DESAIN DAN IMPLEMENTASI SISTEM NAVIGASI ROBOBOAT AUTONOMOUS BERBASIS PENGOLAHAN CITRA HSV FILTER

DESIGN AND IMPLEMENTATION NAVIGATION SYSTEM OF ROBOBOAT AUTONOMOUS USING HSV FILTER IMAGE PROCESSING

Anggitrizaka Tatag Anastya
Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom
anggitrizaka@telkomuniversity.ac.id

Abstrak

Roboboat Autonomous merupakan salah satu perkembangan teknologi robot otonom yang dapat diimplementasikan di atas permukaan air atau bisa disebut robot kapal tanpa awak.Sistem navigasi merupakan salah satu faktor terpenting dalam pembuatan roboboat autonomous, yaitu untuk mengatur arah gerak kapal agar melewati jalur sampai ke arah tujuan.
Pada tugas akhir ini penulis merancang dan mengimplementasikan roboboat autonomous dengan raspberry pi sebagai sistem kontrol dan navigasi. Library OpenCV pada raspberry pi digunakan untuk *Image Processing* dalam mendeteksi jalur kapal menggunakan metode HSV Filter . Metode ini akan melakukan pendeteksian jalur kapal berdasarkan warna dasar, kekuatan warna dan tingkat kecerahan warna terhadap jalur yang akan dideteksi, sehingga hasil pendeteksian jalur akan lebih akurat.
Setelah melakukan pengujian, sistem navigasi membutuhkan sekitar 95% dari *CPU Usage*, namun hanya membutuhkan *Memory Usage* sebesar 50MB dengan nilai *Time Process* sebagai delay adalah 0.1-0.3 detik.Hasil pengujian sistem navigasi memiliki nilai akurasi sebesar 92.5%, sehingga dapat disimpulkan penggunaan HSV Filter pada image processing cukup representatif dan responsif dalam mengatur navigasi secara otomatis pada kapal tanpa awak.

Kata Kunci : raspberry pi, HSV Filter, *image processing*, navigasi roboboat

Abstract
Autonomous Roboboat is one of the technological development of autonomous robots that can be implemented on the surface of the water or can be called a robot ship without a crew. The navigation system is one of the most important factors in making roboboat autonomous, that is to set the direction of motion of the ship in order to pass the path to the destination.
In this thesis the author to design and implement autonomous roboboat with raspberry pi as a control and navigation system. OpenCV library on raspberry pi is used for Image Processing in detecting shipping lines using the HSV Filter. This method will make the detection of shipping lines based on the basic color, color strength and brightness of color for the line that will be detected, so that the results would be more accurate path detection. After testing, the navigation system requires about 95% of CPU Usage, Memory Usage yet requires only 50MB with Time Process value as the delay is 0.1-0.3 seconds. Results of testing the navigation system has an accuracy value of 92.5%, so it can be concluded the use of HSV filter on image processing is quite representative and responsive to automatically regulate navigation on the ship without a crew.

Keywords: raspberry pi, HSV filter, image processing, navigation roboboat

1. Pendahuluan

Perkembangan teknologi sekarang ini banyak berfokus untuk memudahkan pekerjaan manusia di segala bidang.Salah satunya adalah teknologi robotika, dimana selain untuk membantu pekerjaan manusia yang tidak terjangkau oleh manusia juga dapat membantu pekerjaan yang beresiko tinggi.Dengan robot pekejaan manusia sudah dapat digantikan. Robot melakukan tugasnya dengan cara dikontrol secara langsung oleh manusia, maupun secara otomatis yang diprogram terlebih dahulu (kecerdasan buatan).

Dalam beberapa tahun belakang ini robot yang sedang populer adalah Roboboat Autonomous, yaitu robot yang pada konsepnya dapat dimanfaatkan di atas permukaan air atau disebut robot kapal tanpa awak.Selain itu robot kapal tanpa awak ini bekerja secara autonomous atau secara otomatis yang diprogram terlebih dahulu dengan konsep kecerdasan buatan dengan batasan objek dan tempat kerja.Banyaknya komponen sistem navigasi pendukung seperti beberapa jenis sensor kapal yang membutuhkan uang yang tidak sedikit membuat sering terjadinya masalah dalam

perancangan baik batasan fisik, waktu maupun materi.Hal ini mengakibatkan implementasi Roboboat Autonomous hanya dilakukan oleh kalangan tertentu salah satunya karena batasan tersebut.

Sistem navigasi merupakan bagian yang sangat penting dalam pembuatan robot autonomous, berfungsi sebagai pemandu robot dalam berpindah dari satu tempat ke tempat lain secara otomatis. pada tugas akhir ini penulis merancang desain dan implementasi Roboboat Autonomous menggunakan Raspberry Pi berbasis *image processing*. Sistem navigasi Roboboat Autonomous menggunakan bahasa pemrograman python versi 2.7.2 karena python merupakan bahasa pemrograman yang bisa disebut *Dynamic Typing*, yang mana seorang *programmer* tidak perlu mendeklarasikan tipe dari variabel yang akan digunakan. Selain itu python merupakan bahasa pemrograman yang memiliki dokumentasi yang lengkap, baik dari laman resmi maupun dari beberapa komunitas. Python juga tidak memiliki banyak resource tetapi reliable dalam bekerja sehingga dapat berjalan dengan lancar pada spesifikasi hardware yang minimal[7]. Dan terakhir python merupakan bahasa pemrograman yang multiplatform yaitu dapat digunakan di platform manapun seperti Linux, MacOS, Windows, dan lain lain. Sehingga penggunaan bahasa pemrograman Python sangatlah tepat dalam pembuatan sistem navigasi Roboboat Autonomous

Sistem navigasi pada Roboboat Autonomous bertujuan sebagai pengendali arah manuver ketika kapal sedang berjalan.Sistem navigasi dirancang sedemikian rupa sehingga menghasilkan sistem kerja yang efektif dan efisien. Yang dimaksud efektif adalah ketika arah gerak manuver kapal dapat mengikuti jalur kapal dengan baik dan dengan kecepatan yang telah dikalibrasikan sebelumnya sehingga sibakan air yang mengenai lambung kapal tidak menggoyangkan kapal sehingga tidak mengganggu pendeteksian kamera dalam melihat jalur warna kapal. Efisienitas sistem navigasi kapal adalah terkait penggunaan komponen kapal yang tidak terlalu banyak dan sesuai kebutuhan[7]. Sistem Navigasi pada Roboboat Autonomous ini menggunakan sistem deteksi warna yang akan menentukan arah gerak dan kecepatan kapal. Kamera kapal akan menjadi pendeteksi warna dan diproses oleh Raspberry pi untuk eksekusi selanjutnya. Pada tugas akhir kali ini jalur kapal sudah ditentukan oleh penulis sesuai standar Kontes Kapal Cepat Tak Berawak Nasional (KKCTBN).Implementasi Roboboat Autonomous pada dasarnya digunakan sebagai kapal patroli baik laut maupun danau.

2. Dasar Teori dan Perancangan

2.1 Theresholding

Theresholding merupakan proses pengembangan untuk menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih [1].Secara umum proses pengambangan citra *grayscale* untuk menghasilkan citra biner adalah sebagai berikut.

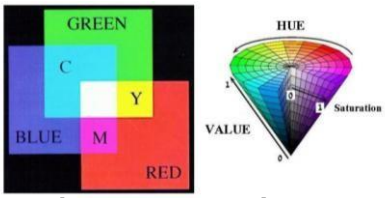
$$G(x,y) = \begin{cases} 1 & f(x,y) \geq T \\ 0 & f(x,y) < T \end{cases} \quad \text{.....(1)}$$

Dimana :
G(x,y) = citra biner
f(x,y) = citra *grayscale*
T = nilai ambang

Nilai T memegang peranan yang sangat penting dalam proses pengambangan. Kualitas hasil citra biner sangat tergantung pada nilai T yang digunakan.

2.2 HSV Filter

HSV (*Hue Saturation Value*) dalam model warna RGB (*Red Green Blue*) tidak hanya mewakili warna tetapi juga intensitas cahaya. Model warna HSV transformasi non-linear dari ruang lingkup warna RGB berorientasi pada pengguna dan didasarkan pada pengertian tint, shade dan tone. Ini memiliki nilai-nilai independen untuk Hue, Saturation, dan Value, masing-masing sesuai untuk panjang gelombang, eksitasi, dan kecerahan [3].



Gambar 1 Sistem Koordinat HSV

Pada gambar 1 menunjukkan sistem koordinat HSV sebagai hexacone. Dasar hexacone berwarnahitam dengan HSV = (0, 0, 0). Kebanyakan warna gambar dicatat sebagai (R, G, B). Warna didefinisikan oleh (R, G, B) dimana RGB dinormalisasi dari 0.0 sampai 1.0, dan setara dengan warna (H, S, V) yang ditentukan oleh suatu set formula.

2.3 OpenCV

OpenCV (*Open Computer Vision*) merupakan semacam *library* yang digunakan dalam membuat aplikasi yang berkaitan dengan *computer vision*. OpenCV juga menyediakan banyak fungsionalitas, yang tentunya dapat mempersingkat waktu dalam pembuatan aplikasi yang lumayan rumit [4].

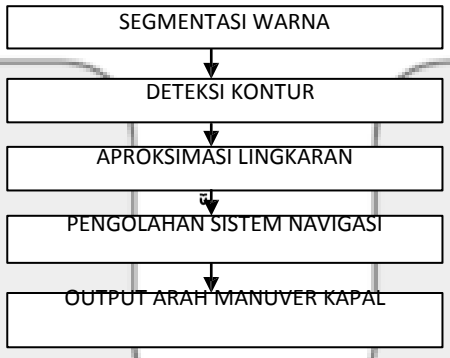
Pada perancangan sistem menjelaskan mengenai alur dari proses yang dikerjakan pada tugas akhir ini. Penjelasan yang ada meliputi alur perancangan desain kapal, hal-hal yang terkait untuk sistem navigasi terkait deskripsi sistem navigasi beserta perancangannya, pengambilan citra, *object detection*, dan *object tracking* sehingga kapal dapat berjalan dengan efektik dan efisien seperti yang diharapkan.

2.4 Perancangan Navigasi Pengambilan Citra

Pengambilan citra atau gambar dilakukan secara kontinu dimana citra yang diambil (*frame*) akan diolah untuk dapat menentukan instruksi yang yang harus dilakukan. Proses deteksi warna dilakukan dengan menggunakan transformasi gambar dari basis RGB (Red, Green, Blue) ke basis HSV (Hue, Saturation, Value)[5]. Transformasi ini dilakukan karena pada basis HSV proses filter warna dapat lebih mudah dilakukan. Setelah transformasi dilakukan selanjutnya merupakan proses *tresholding* dimana proses ini dapat dikatakan sebagai proses filter warna dimana warna yang ingin dideteksi dengan nilai HSV tertentu akan terlihat sebagai gambar putih dan sisanya berwarna hitam[5]. Proses deteksi warna ini dilakukan untuk warna hijau.

2.5 Perancangan Navigasi Object Detection

Dalam proposal tugas akhir ini, warna latar belakang ditentukan berwarna hitam sedang warna objek terdeteksi ditentukan berwarna putih (lebih terang).Algoritma ini digunakan untuk menemukan bouy di dalam gambar dan bisa diaplikasi di sequence image.Konsep dari algoritma ini adalah menampilkan image sebagai matrik pixel dan nilai garis yang sudah pasti [3]. Secara garis besar skenario pengujian sistem dapat digambarkan oleh blok diagram dibawah ini:



Gambar 2Blog diagram sistem navigasi image processing

2.6 Perancangan Sistem

Pada bagian ini dijelaskan perancangan dalam aplikasi yang meliputi (1) gambaran umum sistem,(2) diagram alir sistem, dan (3) perancangan antarmuka.

2.6.1 Gambaran Umum Sistem

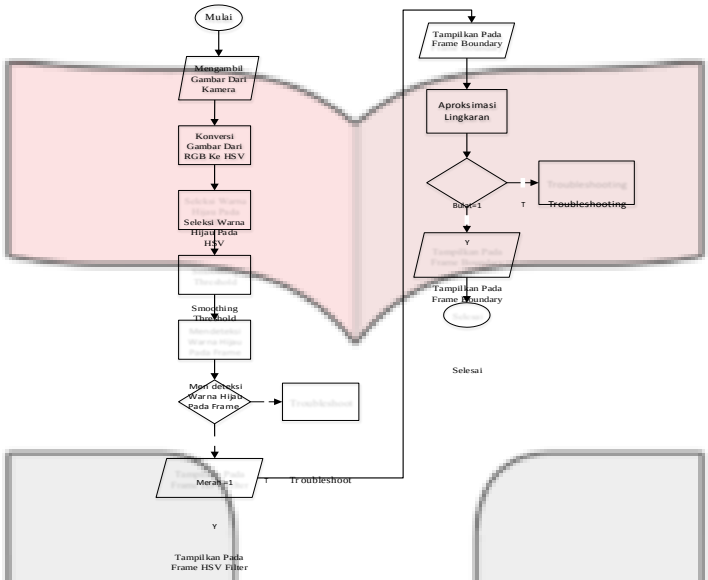
Gambaran umum dari sistem dijelaskan melalui Gambar 3.5:

Gambar 3 Gambaran Umum Sistem

Pada Gambar 3 di atas dapat dilihat gambaran umum sistem dari aplikasi navigasi.Pada gambar ini menjelaskan bahwa Raspberry Pi dapat difungsikan sebagai multitasking, yaitu berfungsi sebagai sistem kendali dan sistem navigasi.Sistem kendali dipengaruhi oleh hasi dari navigasi pendeteksian jalur bola sehingga dapat mengendalikan arah manuver belok ke kiri atau ke kanan maupun kecepatan yang digunakan melambat atau di percepat. Sehingga kapal dapat berjalan sesuai bentuk jalur dari garis *start* sampai *finish.*

2.6.2 Diagram Alir Sistem

Untuk membantu proses penghitungan, ditentukan sebuah *frame boundary* yang melingkupi area lintasan manuver yang akan dilalui objek jalur susunan bola terdeteksiPengambilan citra atau gambar dilakukan secara kontinu dimana citra yang diambil (*frame*) akan diolah untuk dapat menentukan instruksi yang yang harus dilakukan [4]. Proses deteksi warna dilakukan dengan menggunakan transformasi gambar dari basis RGB (Red, Green, Blue) ke basis HSV (Hue, Saturation, Value). Transformasi ini dilakukan karena pada basis HSV proses filter warna dapat lebih mudah dilakukan[5]. Setelah transformasi dilakukan selanjutnya merupakan proses *tresholding* dimana proses ini dapat dikatakan sebagai proses filter warna dimana warna yang ingin dideteksi dengan nilai HSV tertentu akan terlihat sebagai gambar putih dan sisanya berwarna hitam. Proses deteksi warna ini dilakukan untuk warna hijau. Flow chart deteksi pengambilan citra dan deteksi warna akan dijelaskan pada gambardibawah ini:



Gambar 4 Flowchart Deteksi Warna

Pada gambar diatas menjelaskan bahwa jika suatu objek telah terlacak dan dikenali, sesuai warna yang terdeteksi dalam frame boundary maka kapal akan bergerak sesuai penghitungan baik arah gerak maupun kecepatannya. Satu frame boundary akan dibagi menjadi dua: frame kiri dan frame kanan. Dari pembatas frame tengah dengan garis pendeteksi jalur kiri ataupun kanan akan membentuk segitiga. Akan dibandingkan dari dua segitiga tersebut, apabila segitiga kiri lebih kecil luasnya daripada segitiga kanan maka nilai dari kecepatan negatif artinya harus belok ke kiri.Sebaliknya apabila luas segitiga kiri lebih besar dari kanan maka nilai kecepatannya positif artinya harus belok ke kanan. Untuk kecepatan gerak kapal, dilihat dari garis sudut pendeteksi jalur kiri maupun kanan bawah dengan pembatas garis horizontal bagian bawah frame boundary. Semakin kecil sudut yang terbentuk maka kecepatan belok semakin besa Jika sudut bouy pembatas sebelah kiri lebih kecil dari 20 derajat, variabel kecepatannya 100.Kemudian apabila sebelah kanan lebih kecil dari 20 derajat, variabel kecepatannya -100. Jika sudut bouy pembatas sebelah kiri lebih kecil dari 40 derajat, variabel kecepatannya 50 dan apabila sebelah kanan lebih kecil dari 40 derajat, variabel kecepatannya -50 dan seterusnya.

3. Pembahasan

3.1 Implementasi Sistem

Setelah perancangan sistem lengkap maka dilakukan implementasi sesuai dengan perancangan yang telah dilakukan. Implementasi dilakukan menggunakan bahasa pemrograman agar dapat diketahui apakah semua fungsi pada rancangan sebelumnya dapat berjalan dengan baik atau tidak. Kemudian dilakukan pengujian aplikasi untuk mengevaluasi apabila ada kekurangan di dalam program sistem navigasi.

3.1.1 Implementasi Perangkat Lunak

Dalam pelaksanaan implementasi digunakan berbagai macam perangkat lunak sebagai berikut.

1. Python 2.7.2
2. Numpy
3. *Open Computer Vision*
4. *Pi Camera Array Module*

3.1.2 Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam implementasi program ini yaitu:

1. Raspberry Pi tipe B+

2. Raspberry Pi Camera
3. Motor Servo SG90

3.3 PengujianPerformansi

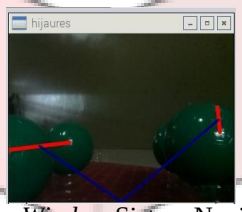
Pengujian performansi adalah pengujian dengan cara mendemokan program navigasi Roboboat Autonomous secara terperinci dengan membedakan sudut deteksi sehingga dapat melihat hasil kecepatan. Selain itu juga melihat warna yang terdeteksi dan membandingkan antara frame boundary kiri maupun kanan. Pengujian performansi memperlihatkan tampilan hasil program navigasi Roboboat Autonomous dalam beberapa skenario penghitungan sudut terhadap objek terdeteksi yang mempengaruhi arah manuver dan kecepatan kapal.

3.3.1 Pengujian Performansi Arah Manuver Kapal

Terdapat tiga skenario pengujian yang akan diuji yaitu ketika kapal harus bermanuver ke kanan, ketika kapal harus bermanuver ke kiri, dan ketika kapal tidak mendeteksi jalur.

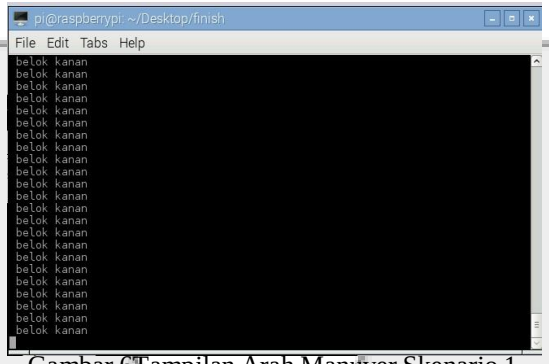
- Skenario 1

Gambar dibawah adalah tampilan ketika segitiga yang terbentuk antara *frame window* bagian kiri lebih besar daripada *frame window* bagian kanan.



Gambar 5 *Frame Window* Sistem Navigasi Skenario 1

Kemudian, akan dilakukan perhitungan luas dari masing masing segitiga sehingga akan menentukan arah manuver kapal. Apabila luas segitiga pada frame bagian kiri lebih besar daripada frame bagian kanan maka kapal bermanuver ke kanan. Hal ini dapat dilihat dari gambar terminal Raspberry Pi di bawah ini:

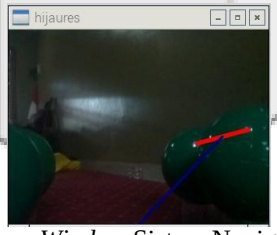


Gambar 6Tampilan Arah Manuver Skenario 1

Selain dari Terminal akan terlihat dari arah gerak motor servo ke arah kanan sesuai dengan besar sudut beloknya kemudian juga tampilan pada terminal Raspberry Pi yang ditampilkan

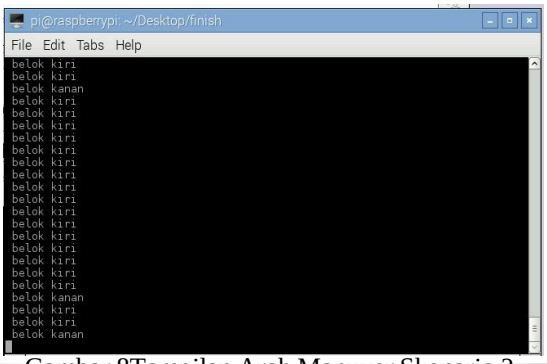
- Skenario 2

Sama seperti pada skenario pertama akan ditampilkan gambar seperti dibawah ini:



Gambar 7 *Frame Window* Sistem Navigasi Skenario 2

Terbentuk dua segitiga antara frame kiri dan frame kanan yang akan menentukan arah manuver kapal. Apabila luas segitiga pada frame bagian kiri lebih kecil daripada frame bagian kanan maka kapal bermanuver ke kiri. Hal ini dapat dilihat dari gambar terminal Raspberry Pi di bawah ini:

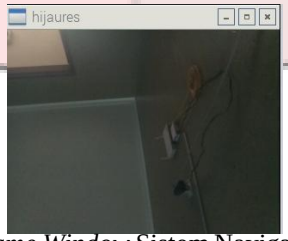


Gambar 8Tampilan Arah Manuver Skenario 2

Selain dari Terminal akan terlihat dari arah gerak motor servo ke arah kiri sesuai dengan besar sudut beloknya kemudian juga ditampilkan pada terminal Raspberry Pi yang ditampilkan.

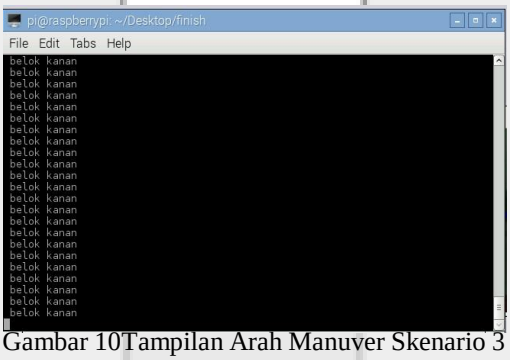
- Skenario 3

Hasil deteksi tanpa mendeteksi objek dapat ditampilkan pada gambar 25 dibawah ini.



Gambar 9*Frame Window* Sistem Navigasi Skenario 3

Pada skenario ketiga kondisi dimana sistem navigasi sama sekali tidak mendeteksi jalur. Sehingga arah manuver akan selalu ke kanan, berputar terus sampai menemukan objek terdeteksi. Hasil tampilan arah manuver dapat ditampilkan pada gambar 25 dibawah ini.



Gambar 10Tampilan Arah Manuver Skenario 3

Selain dari Terminal akan terlihat dari arah gerak motor servo ke arah kanan sesuai dengan besar sudut beloknya kemudian juga ditampilkan pada terminal Raspberry Pi yang ditampilkan.

3.3.2 Pengujian Akurasi Arah Manuver Kapal

Pada pengujian ini akan dilakukan penghitungan persentase akurasi ketika kapal harus bermanuver ke kiri maupun kanan berdasarkan sudut deteksi. Penghitungan nilai sudut deteksi dilakukan dengan cara manual menggunakan busur. Hal ini bertujuan untuk menghitung tingkat akurasi arah navigasi secara *real condition*. Nilai presentase akurasi tiap sudut dihitung menggunakan rumus dibawah ini:

$$\text{PERSENTASE AKURASI } x = \frac{\sum x}{n} \text{ X } 100\%$$

Dimana:

x = arah yang akan diukur tingkat akurasinya yaitu kiri atau kanan
n = total sampel yang telah ditentukan yaitu 20

Hasil dari penghitungan persentase akurasi arah navigasi berdasarkan sudut dapat dilihat pada tabel dibawah ini:

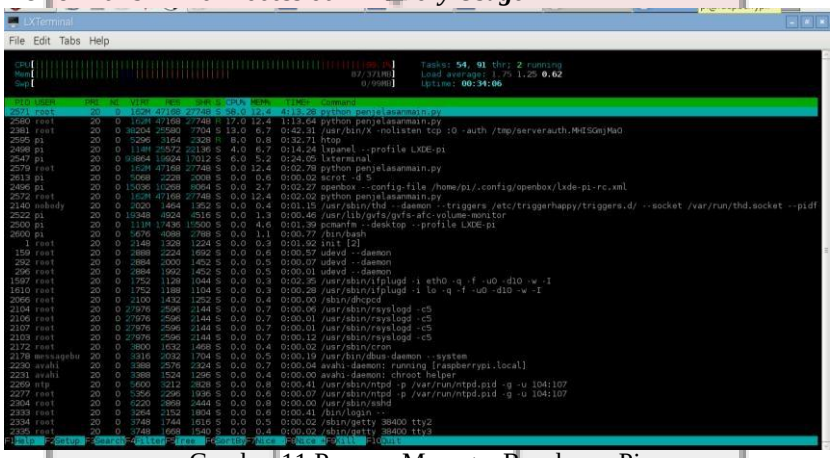
Tabel 4Hasil pengujian akurasi arah navigasi kapal

Besar Sudut	Arah Manuver	Persentase Akurasi Rata-Rata

	Kiri	Kanan	Berdasarkan Sudut
<20 ^{o}	90%	100%	95%
<40 ^{o}	85%	100%	92.5%
<60 ^{o}	80%	100%	90%
		Presentase Akurasi	**92.5%**

Pada tabel diatas menunjukkan bahwa presentase akurasi arah navigasi kapal sebesar 92.5%.Hal ini menunjukkan bahwa program berjalan responsif dalam mengatur arah navigasi secara otomatis pada kapal tanpa awak.

3.3.3 Pengujian Performansi *Time Process* dan *Memory Usage*



Gambar 11 Process Manager Raspberry Pi

Pada implementasi program dapat berjalan dengan lancar dengan delay 0.1 detik ketika program mulai dijalankan dan ketika program ditutup. Dapat dilihat pada gambar diatas Time Process merupakan nilai dari delay berjalannya program yaitu rata rata 0.1-0.3 detik. Ketika program berjalan, program memiliki nilai framerate sebesar 32. Maksudnya dalam satu detik kamera akan mengambil 32 gambar sehingga gambar terlihat bergerak dan di setiap gambar warna akan dideteksi dengan resolusi tiap gambar 640x480 yang akan disimpan kedalam memori. Yang berarti dibutuhkan *delay* minimal 1 detik untuk memproses citra untuk melakukan pendeteksian. *Delay* dipengaruhi oleh faktor penggunaan hardware penunjang yang mempengaruhi penggunaan tegangan Raspberry Pi seperti *servo motor*, sehingga dibutuhkan tegangan tambahan. Selain itu delay juga dipengaruhi oleh *CPU Usage* seperti menjalankan aplikasi lain bersamaan. Pada gambar diatas menunjukan tabel operasi baik *CPU Usage* maupun *Memory Usage.*Program sistem navigasi ini membutuhkan sekitar 98% dari *CPU Usage.*Dengan menjalankan Program sistem navigasi melalui terminal , Sistem Navigasi berjalan dengan dengan nilai *delay* minimum 1 detik.Penyimpanan proses perekaman akan dihapus oleh sistem ketika program dihentikan.

4. Kesimpulan

Hasil implementasi dan pengujian alpha dan beta program sistem navigasi Roboboat Autonomous yaitu program ini dapat berjalan dengan baik sesuai dengan fungsi yang diharapkan. Oleh karena itu, kesimpulan penelitian ini adalah bahwa program ini:

1. Sistem Navigasi dapat mendeteksi warna menggunakan filter HSV dan menentukan arah gerak servo sesuai posisi jalur yang ditentukan
2. Raspberry pi dapat melakukan kerja secara *multitasking* sebagai prosesor yang mengatur sistem navigasi maupun sistem kendali gerak servo.
3. Pengujian navigasi sistem memiliki nilai akurasi sebesar 92.5%, sehingga dapat disimpulkan program berjalan responsif dalam mengatur arah navigasi secara otomatis pada kapal tanpa awak.
4. Nilai *Time Process* ketika program sedang berjalan adalah dengan *delay* sekitar 0.1-0.3 Detik namun program menggunakan source CPU Usage yang sangat besar yaitu sebesar 95% dan program berjalan dengan lancar dengan rata rata penggunaan Memory Usage sebesar 50 MB.

5. Daftar Pustaka:

[1] Pritish Sachdeva, Schrutik Katchii, "*Paper Review on Raspberry Pi B+*," Fakultas Elektro dan Komunikasi, Institut Teknologi Xavier, Mumbai, India.

[2] Whithame D. Reeve, 2013. LWA TV on Raspberry Pi

[3] Hermawan Prabowo, Soegianto Soelistiono," *Multiple Object Tracking Using Open Computer Vision*," Fakultas Sains dan Teknologi, Universitas Airlangga, Surabaya.

[4] M. Irwan Bustami, Agus Siswanto, Afrizal N.T., M. Fajri R., Chindra Saputra, "Mendeteksi Bentuk Warna Bola Pada Robot Humanoid Soccer Dengan Menggunakan Raspberry Pi," STIKOM Dinamika Bangsa, Jambi.

[5] Hermawati, Fajar Astuti (2013). Pengolahan Citra Digital Konsep dan Teori. Yogyakarta: Penerbit CV Andi Offset

[6] Riwinoto,S.T., M.Kom., "Penggunaan Algoritma Hough Tranforms Deteksi BentukLingkaran pada Ruang 2D," Teknik Informatika, Politeknik Batam, Batam.

[7] Tri Bagus Susilo, Adi Prima Yudhistira, Ogi Tridho Maroghi, Ahlan Zulfakhri, Fatih Khamdani, Mohamad Ansori, Andi Pratama,"AUVSI Autonomous Surface Vehicle Journal Paper," Universitas Diponegoro, Semarang.

[8] K.S. Shilpashree, Lokesha H., Hadimani Shivkumar,"*Implementation Of Image Processing In Raspberry Pi"*,Kalpataru Institute Of Technology, India