

APLIKASI ANDROID-RASPBERRY PI PADA KAPAL TANPA AWAK UNTUK PENCARIAN KORBAN KECELAKAAN LAUT

Noorman Rinanto, Agus Khumaidi

Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal,
Politeknik Perkapalan Negeri Surabaya
JL. Teknik Kimia Kampus PPNS-ITS , Keputih, Sukolilo, Surabaya
Telp : (031) 5947186, Fax : (031) 5942887
Email: noorman.rinanto@gmail.com

Abstract

Finding of marine accident victims carried by sea waves to the outside of incidence point is a major issue in the rescue and evacuation process. In addition, a high safety risk of rescuers while doing its job in bad condition and weather. Therefore, it's necessary to solve it. This research aims to develop an innovation of android information technology on Autonomous Unmanned Surface Vehicle as victim finder using the camera and two ways communication among the ship and ground station (GS). The ship distributed randomly on suspected area while the rescuer will monitor and control it remotely. It equipped by two mode of control operation. Remote mode is used to drive it to the searching point. Autonomous mode is object detection mode based on images color threshold algorithm on Raspberry Pi device. It will send the GPS coordinate data and image capture to the GS after it found an object.

Abstrak

Menemukan posisi korban kecelakaan laut yang terbawa oleh gelombang air laut sampai di luar radius titik kejadian merupakan permasalahan utama dalam proses penyelamatan dan evakuasi. Selain itu, masalah resiko keselamatan yang tinggi dari tim penyelamat saat melakukan tugasnya pada kondisi dan cuaca buruk. Untuk itu dibutuhkan solusi yang dapat mengatasi permasalahan tersebut. Penelitian ini mengusulkan inovasi aplikasi berbasis android pada kapal tanpa awak (Autonomous Unmanned Surface Vehicle) sebagai pencari korban menggunakan kamera dan sistem komunikasi dua arah antara kapal dan ground station. Kapal tersebut disebarkan secara acak di daerah yang dicurigai sementara petugas memonitor dan mengendalikannya secara jarak jauh. Kapal dilengkapi dengan dua mode operasi. Mode remote digunakan untuk menggerakkan kapal menuju titik pencarian. Mode autonomous merupakan mode pendeteksian obyek berbasis pengolahan citra menggunakan algoritma color threshold pada perangkat Raspberry pi. Setelah menemukan obyek, kapal mengirimkan data ke ground station berupa posisi koordinat GPS obyek dan gambar hasil capture obyek.

Kata kunci: *android, raspberry pi, autonomous unmanned surface vehicle*

1. PENDAHULUAN

Tolak ukur keberhasilan proses evakuasi dan penyelamatan korban kecelakaan di laut seperti tenggelamnya kapal dan jatuhnya pesawat di perairan atau laut sangat bergantung dari kesuksesan tim penyelamat dalam menemukan lokasi korban. Korban yang tenggelam biasanya ditemukan tidak jauh dari titik lokasi kejadian kecelakaan. Namun, pada korban yang terapung di laut akan sulit menemukan lokasinya dikarenakan faktor alam seperti terseret gelombang dan angin laut. Hal tersebut menyebabkan area pencarian korban kecelakaan laut menjadi lebih luas seperti yang terjadi pada proses pencarian korban-korban pada kecelakaan jatuhnya pesawat di Indonesia tepatnya di selat

Karimata pada akhir tahun 2014 (Jawa Pos, 2015). Selain itu faktor cuaca yang buruk menyebabkan resiko yang tinggi terhadap keselamatan tim SAR pada saat melakukan proses pencarian korban.

Dari permasalahan tersebut, penelitian ini mengusulkan sebuah inovasi berbasis teknologi informasi dan robotika yang bertujuan membuat *Autonomous Unmanned Surface Vehicle (AUSV)* atau kapal tanpa awak yang berfungsi sebagai alat bantu pencarian korban yang terapung di laut. Kapal tersebut mempunyai 2 (dua) mode operasi yaitu mode otomatis dan mode manual. Untuk mode otomatis yang beroperasi saat kapal melakukan proses pendeteksian obyek atau korban dan mode manual merupakan prose pengendalian manual secara jarak jauh saat obyek

telah ditemukan atau saat pengiriman kapal ke posisi area tertentu yang dipecurigai terdapat korban kecelakaan laut. Kapal dapat secara otomatis mendeteksi korban dan mengirimkan data informasi berupa posisi koordinat *Global Positioning System (GPS)* dan gambar obyek ke *Ground Station (GS)*. Sistem pengenalan korban menggunakan pengolahan citra berbasis teknologi kamera menggunakan algoritma *ColorThreshold* (Otsu, N., 1979). Pengenalan obyek dilakukan pada perangkat *Raspberry Pi controller*. Obyek yang dideteksi berupa warna pakaian atau pelampung korban. Setelah obyek ditemukan kapal akan secara otomatis mengirimkan data informasi ke *GS* yang berupa perangkat berbasis Android. Sistem komunikasi pada penelitian ini menggunakan komunikasi *WiFi* sebagai perangkat komunikasi data informasi dan *Radio Frequency (RF)* sebagai komunikasi pengendalian jarak jauh.

Diharapkan hasil dari penelitian ini dapat membantu memudahkan proses pencarian korban kecelakaan di laut dan sekaligus dapat mengurangi resiko yang dihadapi oleh tim SAR pada saat melakukan pencarian korban pada cuaca yang buruk.

2. METODE

Algoritma pendeteksi obyek menggunakan metode *Color Threshold* yang pengolahan data gambarnya dilakukan pada perangkat mini PC *Raspberry Pi*. Sistem pengendalian kapal menggunakan metode *PID Controller* pada perangkat *Arduino*.

2.1 Color Threshold

Color Threshold merupakan metode pengolahan citra yang paling sederhana, dimana nilai ambang (*threshold*) digunakan untuk menghasilkan sebuah citra yang bernilai biner (1/0) (Otsu, N., 1979). Proses *threshold* masing-masing *pixel* pada sebuah citra ditandai sebagai *pixel* milik objek. Ada empat jenis *color threshold* secara morfologi yaitu:

1. *Threshold above* Jika koordinat *pixel* yang di ukur nilainya lebih besar dari nilai *threshold*.
2. *Threshold below* Jika koordinat *pixel* yang di ukur nilainya lebih kecil dari nilai *threshold*.
3. *Threshold inside* Jika nilai *pixel* suatu objek berada diantara dua nilai warna ambang.
4. *Threshold outside* jika nilai *pixel* obyek diluar kedua nilai warna ambang.

Pada penelitian ini jenis range *color threshold* yang dipakai adalah *Threshold inside*. Penentuan nilai *threshold* dilakukan dengan kalibrasi kamera

dengan mengambil beberapa contoh gambar pada saat di lapangan. Proses penentuan nilai *threshold (T)* yaitu pada umumnya warna *pixel* suatu objek yang dideteksi ditandai dengan nilai "1" sedangkan *pixel* latar belakang gambar (*background*) ditandai dengan nilai "0" atau dijelaskan dalam persamaan sebagai berikut:

$$g(x, y) = \begin{cases} 1, & \text{iff}(x, y) \geq T \\ 0, & \text{iff}(x, y) < T \end{cases} \dots \dots \dots (1)$$

Dimana *g* adalah titik piksel, *x* adalah titik koornidat sumbu x pada piksel, *y* adalah titik koordinat y piksel, *T* adalah nilai ambang (*threshold*) (Otsu, N., 1979; Taufiq, Y., 2011).

Citra yang digunakan pada penelitian ini menggunakan citra *HSV* yang dikenal lebih baik dalam segi pengenalan terhadap warna dengan intensitas pencahayaan yang kurang atau berlebih daripada citra *RGB* (Adrian, F., Alan, R., 1998). Adapun proses konversi nilai *RGB* ke *HSV* pada citra 256 bit menggunakan teori travis dapat ditunjukkan pada persamaan dibawah ini:

- a. Konversi kedalaman warna RGB 256 bit

$$R' = \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255} \dots \dots \dots (2)$$

Dimana *R* adalah warna piksel *Red*, *G* adalah warna piksel *Green*, *B* adalah warna piksel *Blue*, *R'* adalah perbandingan nilai warna *Red* dengan nilai warna maksimum, *G'* adalah perbandingan nilai warna piksel *Green* dengan warna maksimum, *B'* adalah perbandingan nilai warna piksel *Blue* dengan nilai warna maksimum.

- b. Parameter range warna RGB

$$\begin{aligned} Cmax &= \max(R', G', B') \\ Cmin &= \min(R', G', B') \\ \Delta &= Cmax - Cmin \dots \dots \dots (3) \end{aligned}$$

Dimana *Cmax* adalah konstanta nilai warna batas atas, *Cmin* adalah konstanta nilai warna batas bawah, Δ adalah selisih range warna batas atas dan bawah.

- c. Konversi nilai H (*hue*)

$$H = \begin{cases} 0, & \Delta = 0 \\ 60 \times \left(\frac{G' - B'}{\Delta} \text{ mod } 6 \right), & Cmax = R' \\ 60 \times \left(\frac{B' - R'}{\Delta} + 2 \right), & Cmax = G' \\ 60 \times \left(\frac{R' - G'}{\Delta} + 4 \right), & Cmax = B' \end{cases} \dots \dots \dots (4)$$

Dimana *H* adalah nilai warna *Hue*.

- d. Konversi nilai nilai S (*Saturation*)

$$S = \begin{cases} 0, & Cmax \\ \frac{\Delta}{Cmax}, & Cmax \end{cases} \dots \dots \dots (5)$$

Dimana S adalah nilai warna *Saturation*.

e. Konversi nilai V (Value)

$$V = Cmax \dots\dots\dots(6)$$

Dimana V adalah nilai warna *Value*.

2.2 PID Controller

PID *Controller* merupakan sebuah sistem pengendali yang terdiri dari tiga buah sistem pengendali yaitu sistem pengendali *proportional (P)*, *Integral (I)*, *Derivative (D)*. Dimana sinyal masukan (*input*) dari sistem berasal dari *feedback* berupa nilai *error (e)* yang merupakan selisih antara nilai *setpoint (sp)* dengan nilai *present value (pv)* sebagaimana ditunjukkan pada persamaan (7). Adapun persamaan fungsi sinyal keluaran (*output*) dari sistem PID pada waktu kontinu seperti dinyatakan pada persamaan (8).

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

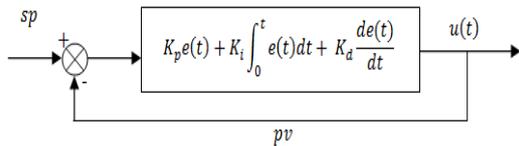
atau

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) (7)$$

Dimana $u(t)$ adalah sinyal keluaran pengendali PID, K_p adalah konstanta proporsional, K_i adalah konstanta integral, K_d adalah konstanta turunan, T_i adalah waktu integral, T_d adalah waktu turunan, sedangkan $e(t)$ adalah sinyal *error* (Katsuhiko, O., 1994).

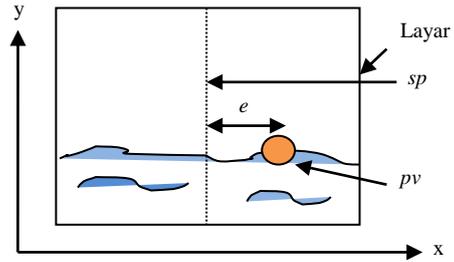
$$e = sp - pv \quad (8)$$

Dimana e adalah nilai *error*, sp adalah nilai *setpoint*, pv adalah nilai *present value* (Katsuhiko, O., 1994). Jika digambarkan dalam bentuk blok diagram sistem pengendali PID tersebut seperti ditunjukkan pada gambar 1.



Gambar 1. Blok diagram sistem pengendali PID.

Pada penelitian ini pengendali PID digunakan untuk mengendalikan gerakan *rudder* kapal dengan *error* yang didapatkan dari hasil komputasi citra kamera. Dimana nilai sp berupa nilai koordinat x piksel layar (*screen*). Sedangkan pv diambil dari koordinat x piksel posisi obyek ditemukan sekarang. seperti yang ditunjukkan pada gambar 2.



Gambar 2. Ilustrasi pengambilan nilai piksel citra digital hasil kamera.

2.3 Sistem kerja perangkat

Sistem kerja perangkat pada penelitian ini seperti ditunjukkan pada gambar 3. Kamera pada kapal akan mengambil citra lingkungan sekitar tempat lokasi terjadinya kecelakaan laut. Saat kapal mendeteksi obyek maka citra dan posisi koordinat GPS yang didapatkan akan di kirim ke *ground station* untuk dianalisa oleh petugas. Kemudian petugas memutuskan untuk melakukan pengendalian secara manual (*Manual Mode*) atau membiarkan kapal menuju ke titik lokasi korban secara otomatis (*Autonomous Mode*).

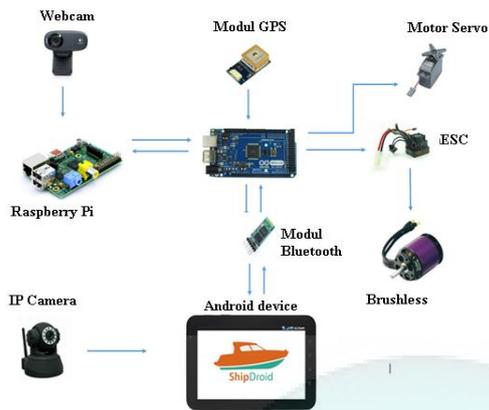


Gambar 3. Sistem kerja perangkat.

2.4 Perancangan

Perancangan kapal tanpa awak pencari korban di laut pada penelitian ini dilakukan melalui 2 (dua) tahap yaitu perancangan *hardware* dan *software*. Perancangan *hardware* terdiri dari sebuah model kapal yang dilengkapi dengan perangkat elektronik seperti Raspberry Pi sebagai perangkat pengolah data citra dari *webcam*, mikrokontroler Arduino Mega ADK yang berfungsi sebagai perangkat *interface I/O* ke modul- modul elektronik lainnya. Modul-modul elektronik itu antara lain seperti modul bluetooth HC-05 sebagai penghubung komunikasi antara perangkat Android ke pengendali kapal, modul Neo GPS sebagai perangkat untuk mengetahui posisi kapal di bumi, IP camera yang dipakai untuk *streaming video* secara langsung melalui komunikasi *wifi* ke perangkat Android. Selain itu juga terdapat motor brushless yang berfungsi sebagai penggerak utama kapal, *Electronic Speed Control (ESC)* sebagai perangkat pengendali motor brushles dan motor servo sebagai perangkat pengatur gerakan rudder kapal. Keseluruhan rancangan sistem

hardware seperti ditunjukkan pada gambar 3.



Gambar 4. Rancangan hardware.

Sedangkan perancangan *software* terdiri atas pemrograman Android menggunakan Android Studio, pemrograman pada mikrokontroler Arduino Mega ADK menggunakan android sketch dan pemrograman pada Raspberry Pi menggunakan pemrograman Phyton IDE.

2.4.1 Model Kapal

Kapal tanpa awak yang digunakan pada penelitian ini adalah berjenis catamaran yang mempunyai 2 (dua) lambung dengan ukuran dimensi utama seperti ditunjukkan pada tabel 1. Model kapal seperti ditunjukkan pada gambar 4.

Tabel 1. Dimensi Utama kapal.

Length Over All (LOA)	1,00 meter
Beam (B)	0,40 meter
Draft (T)	0,02 meter
Draught (H)	0,10 meter
CB (Coeffisien Block)	0,2

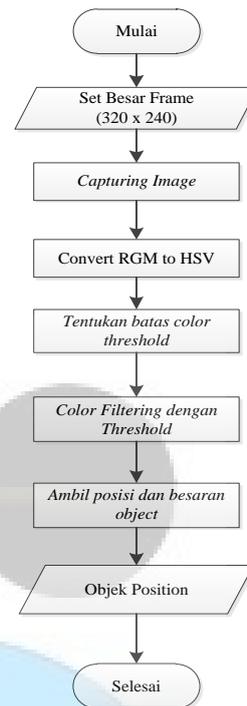


Gambar 5. Model kapal catamaran.

2.4.2 Perancangan software Pendeteksi Korban

Software pendeteksi korban yang ada pada perangkat Raspberry Pi dibuat dengan menggunakan bahasa pemrogram Phyton dan *library* OpenCV sebagai pembantu algoritma pengenalan obyek. OpenCV merupakan sebuah *library* yang dikenal pada pengolahan citra dan video. Adapun sistematika proses pendeteksian

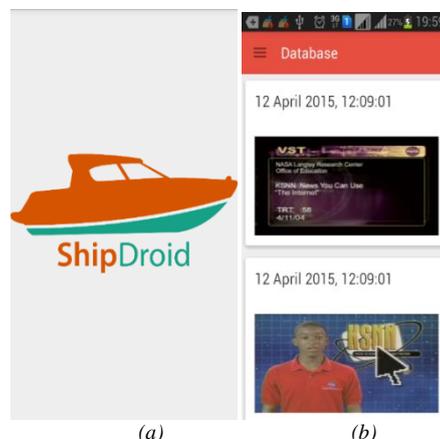
obyek seperti disajikan dalam diagram alur pada gambar 5.



Gambar 6. Flowchart proses pengenalan objek.

2.4.3 Perancangan Software Android

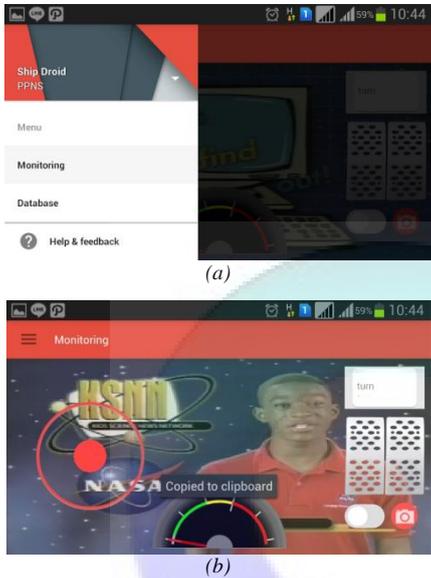
Dalam penelitian ini *interface* pengendali yang digunakan pada *ground sector* adalah aplikasi berbasis android yang dibuat dengan menggunakan software android studio *IDE*. Android studio dikembangkan dengan bahasa pemrograman *Java* oleh *Google corp*. Adapun smartphone android yang digunakan memiliki spesifikasi minimum mempunyai OS (*Operating System*) Android Ice Cream Sandwich *Interface* terdiri dari beberapa tampilan yaitu *splash screen*, dan list data base seperti ditunjukkan pada gambar 6.



Gambar 7. Tampilan splash screen (a) dan (b) list database.

Sedangkan rancangan drawer, dan data

monitoring seperti ditunjukkan pada gambar 7. Kemudian detail database seperti ditunjukkan pada gambar 8. Setiap form (xml) mempunyai fungsi tersendiri dalam sistem, berikut ini adalah penjelasan detail dari masing-masing form.



Gambar 8. Tampilan drawer dan monitoring form



Gambar 9. Rancangan detail view database

3. PENGUJIAN DAN ANALISA

Proses pengujian sistem pada penelitian ini dilakukan dengan beberapa tahapan yaitu pengujian mode kapal, streaming video, dan pengambilan data GPS.

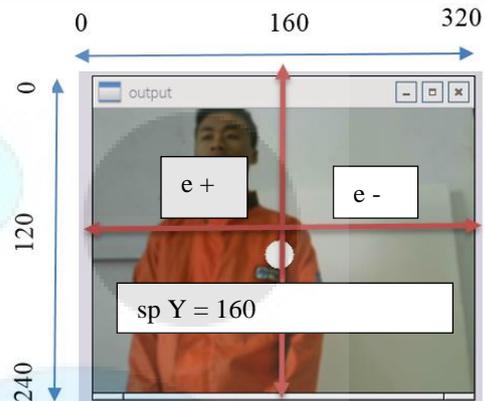
3.1 Pengujian Kapal

Pengujian mode operasi pada kapal ini terdiri dari 2 (dua) mode yaitu mode *autonomous* (otomatis), dan mode *remote* (manual).

3.1.1 Mode Otomatis

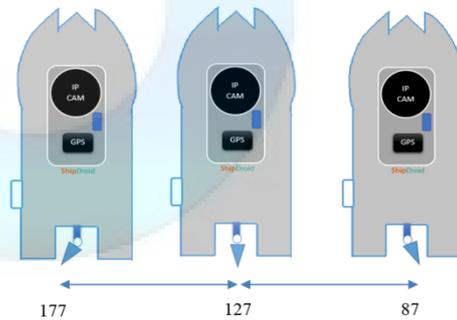
Pengujian ini bertujuan untuk menguji respon rudder saat kapal beroperasi dengan mode otomatis dimana rudder dikendalikan dengan metode *PID controller*. Input data untuk sistem kontrol *PID*

adalah data posisi objek dalam pixel x, y dan output berupa pulsa PWM untuk mengendalikan servo rudder. Nilai error didapat dengan mengurangkan nilai *sp* saat benda berada diposisi tengah dengan posisi benda saat beroperasi, pengujian kontrol *PID* dengan mengambil data respon sistem terhadap error. Nilai *Kp*, *Ki*, *Kd* didapat dengan melakukan metode *tuning and trial*. Mikrokontroler akan menampilkan data sudut rudder secara periodik. Input data dari kamera seperti ditunjukkan pada gambar 9.



Gambar 10. Ilustrasi posisi objek dan error pada frame 340 x 240.

Sedangkan untuk data pulsa servo ditunjukkan pada gambar 10.

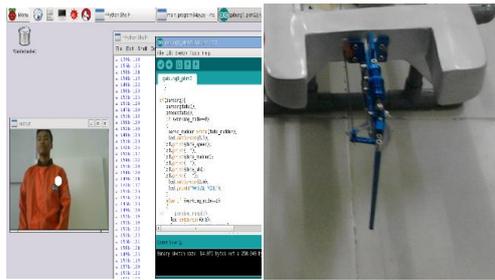


Gambar 11. Data nilai sudut servo

Dalam pengujian ini parameter yang digunakan yaitu $K_p = 2$, $K_i = 0$, dan $K_d = 1$ dengan set point rudder adalah 127. Dan untuk mendapatkan nilai rudder sekarang (P_v) dengan menggunakan persamaan:

$$\text{Sudut rudder} = 127 - \text{PID}$$

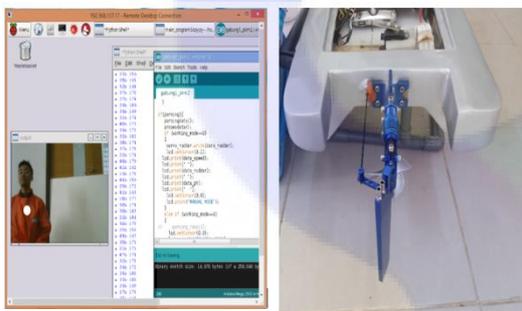
Beberapa hasil pengujian seperti ditunjukkan pada gambar 11,12,dan 13.



Gambar 12. Ketika objek berada di posisi tengah



Gambar 13. Ketika objek di samping kanan frame



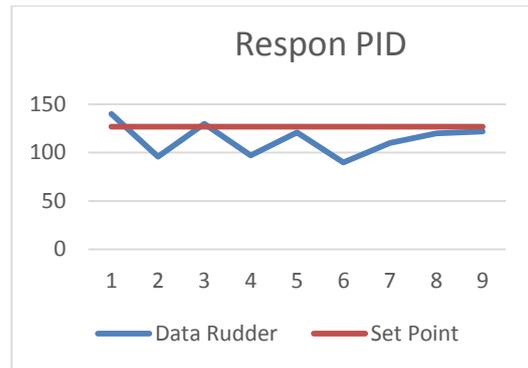
Gambar 14. Ketika objek di samping kiri frame

Dari hasil percobaan diatas didapatkan data input output dengan kontroler PID seperti pada tabel 2.

Tabel 2. Hubungan Pixel dengan posisi rudder mode otomatis

No	Data Rudder	Set Point
1	140	127
2	96	127
3	130	127
4	97	127
5	121	127
6	90	127
7	110	127
9	120	127
10	122	127

Jika data di atas disajikan menjadi grafik maka respon dapat dilihat pada gambar 14.



Gambar 15. Grafik respon PID

3.1.2 Mode Manual

Pada mode manual ini kapal secara penuh dikendalikan melalui interface android yang dibuat. Adapun data yang dikendalikan berupa pergerakan rudder, kecepatan putaran motor, dan sudut servo. Interface akan mengirim data sesuai dengan protokol. Sedangkan pada sisi arduino harus mampu melakukan parsing data sehingga dapat membedakan data untuk kecepatan motor, sudut rudder, sudut servo, serta mode gerakan kapal otomatis atau manual. Pada arduino data data parsing yang diterima tidak dapat langsung digunakan tetapi melalui mapping karena range data antara keduanya berbeda. Adapun perbedaan data antara keduanya dapat dilihat pada tabel 3 dibawah ini.

Tabel 3. Data mapping arduino dan interface

No		Data Interface	Data Arduino
1	Analog Rudder	-13 Sampai 136	177 Sampai 87
2	Kecepatan	0 Sampai 100	57 Sampai 77
3	Data Servo	30 Sampai 160	150 sampai 60
4	Mode Kerja	1 Dan 0	1 Dan 0

Berikut adalah program arduino untuk mapping data diatas dengan memanfaatkan program map pada arduino, adapun aturan dalam proses mapping sebagai berikut:

$$Hasil = \text{map}(\text{min1}, \text{max1}, \text{min2}, \text{max2});$$

```
data_speed=map(dt[0].toInt(),0,100,55,77);
data_rudder=map(dt[1].toInt(),10,110,-177,-85);
data_rudder=data_rudder*-1;
data_servo=map(dt[2].toInt(),30,160,-150,-60);
data_servo=data_servo*-1;
working_mode=dt[3].toInt();
```

Berikut adalah gambar hasil kendali menggunakan remote.



Gambar 16. Kapal ketika kondisi normal



Gambar 19. Uji coba kapal dengan kecepatan 55%.



Gambar 17. Uji coba kapal dengan tombol analog diarahkan ke kanan dan kecepatan 0%

Dari percobaan ini didapatkan hasil mapping data seperti pada tabel 4.

Tabel 4. Hasil mapping

Keterangan:

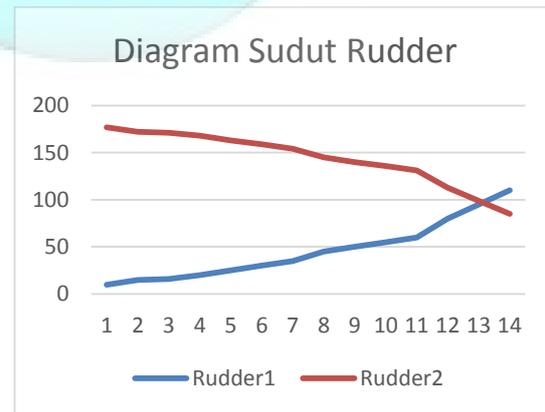
- K= Kecepatan K'=Pulsa Brushless
- R= Sudut rudder R'= Pulsa Rudder
- S= Sudut servo S'= Pulsa Servo
- M=Mode gerak M=Mode gerak

Dari tabel 4 didapatkan grafik perubahan data seperti pada gambar 19, 20, dan 21.

No	Data Interface				Data Arduino			
	K	R	S	M	K'	R'	S'	M
1	0	10	30	1	55	177	150	1
2	1	15	32	0	55	172	149	0
3	5	16	40	1	56	171	143	1
4	10	20	45	1	57	168	140	1
5	13	25	55	1	58	163	133	1
6	15	30	65	1	58	159	126	1
7	20	35	70	0	59	154	122	0
8	30	45	71	1	62	145	122	1
9	35	50	75	0	63	140	119	0
10	45	55	90	0	65	136	108	0
11	50	60	98	0	66	131	103	0
12	75	80	100	0	72	113	102	0
13	80	95	145	1	73	99	70	1
14	100	110	160	1	77	85	60	1



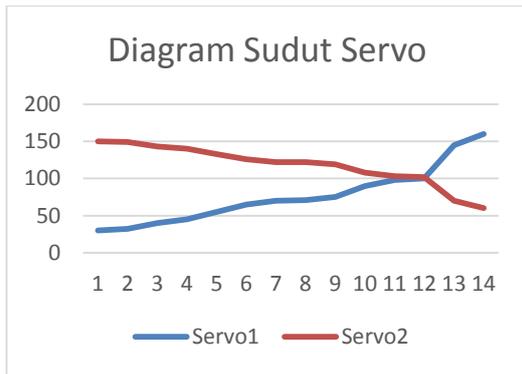
Gambar 20. Grafik kecepatan



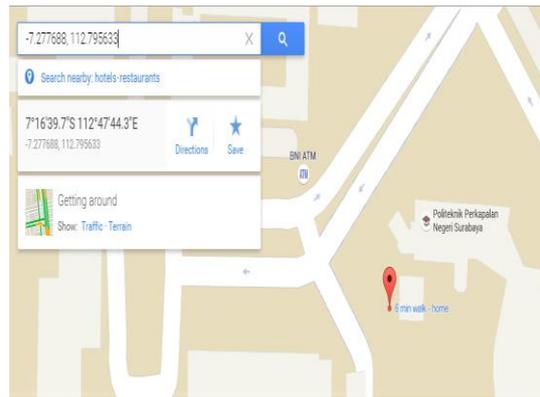
Gambar 21. Grafik data sudut rudder



Gambar 18. Uji coba kapal dengan tombol analog diarahkan ke kiri dan kecepatan 0%



Gambar 22 Grafik data sudut servo



Gambar 23. Hasil capture google map



Gambar 23. Trial kapal di kolam uji coba

3.2 Pengujian GPS

Pengujian GPS digunakan untuk mengambil data *latitude* dan *longitude* dari sensor Neo GPS Starter Kit dan membandingkan data yang dibaca dengan data lokasi sesungguhnya. Perbandingan menggunakan fasilitas google maps. Pengambilan data dilakukan pada 10 titik koordinat yang berbeda dan hasilnya seperti ditunjukkan pada tabel 5.

Tabel 5. Data *Latitude/ Longitude*

No	L o k a s i Data Latitude/ Longitude	G P S
1	-7.277688, 112.795633	
2	-7.277867, 112.795835	
3	-7.277809, 112.795676	
4	-7.277704, 112.795887	
5	-7.277580, 112.795629	
6	-7.277753, 112.795673	
7	-7.277690, 112.795684	
8	-7.277736, 112.795648	
9	-7.277697, 112.795696	
10	-7.277642, 112.795734	

Dari data hasil pembacaan data GPS yang ditunjukkan pada tabel 5 jika dilakukan pengujian dengan menggunakan fasilitas google map maka didapatkan hasil seperti yang ditunjukkan pada gambar 23.

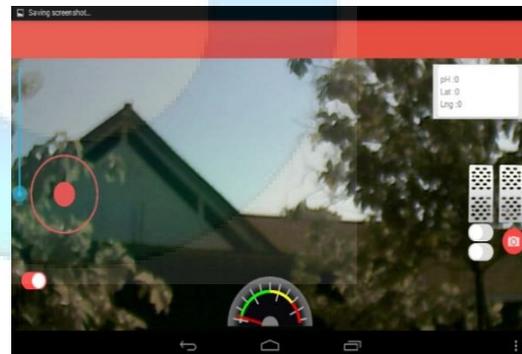
3.3 Pengujian streaming video

Pada pengujian ini jaringan wifi yang digunakan adalah *hotspot tethering wifi smartphone*. Seperti ditunjukkan pada gambar 24.



Gambar 24. Sistem komunikasi IP camera dengan android

Untuk hasil pengambilan gambar dapat dilihat pada gambar 24.



Gambar25. Hasil pengambilan gambar pada kapal.

4. SIMPULAN DAN SARAN

Dari pengujian sistem pada penelitian ini dapat disimpulkan beberapa hal sebagai berikut:

1. Untuk mempermudah cara mendapatkan perbedaan nilai data menggunakan program *mapping*.
2. Sistem dapat bekerja dengan baik sesuai sistem kerja yang diinginkan
3. Jalur komunikasi antara android dan IP camera menggunakan *tethering hotspot* dapat berjalan lancar.
4. Nilai konstanta K_p , K_i , dan K_d yang paling sesuai digunakan adalah $K_p = 2$, $K_i=0$, dan $K_d=1$.

5. DAFTAR RUJUKAN

- Jawa Pos National Network, 2015. *Jasad Korban AirAsia Tersebar di Perairan Sulbar dan Sulsel*. [Online] (Updated 31 Januari 2015) Availableat: <http://m.jpnn.com/news.php?id=284809>.hth.korban air asia[Accessed 15 April 2015]
- Otsu, N., 1979. "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62-66.
- Yusuf Taufiq, 2011. *Membedakan Objek Menggunakan Metode Thresholding dan Fungsi Morfologi*. Depok: Universitas Indonesia.
- Ford Adrian, Roberts Alan, 1998. *Colour Space Conversions*. United Kingdom.
- Ogata, Katsuhiko, 1994. *Teknik Kontrol Automatik Jilid 1*, Diterjemahkan Oleh Ir. Edi Leksono, Erlangga, Jakarta.
- Upton Eben.2012.*Programming the Raspberry Pi*.United Kingdom:Raspberry Pi Foundation.
- Reinius Staffan.2013.*Object recognition using the OpenCV Haar cascade-classifier on the iOS platform*.Uppsala Universitet:Department of Information Technology.
- Viraktamath, Katti Mukund,Khatawkar Aditya,Kulkarni Pavan.2013. *Face Detection and Tracking using OpenCV*.India:Faculty, Department of Electronics and Communication Engineering.
- Wiscott, 2000. *PID without PhD.Embedded system programming*, UK