

Perancangan *Object Detection System* untuk *Collision Avoidance System* Kapal dengan Pemanfaatan Kamera dan *Light Detection and Ranging* (LiDAR)

Anthony Suryajaya; Totok Yulianto; Yuda Apri Hermawan

Departemen Teknik Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember (ITS)

e-mail:anthony_suryajaya@outlook.com, yuda.apri@na.its.ac.id

Abstrak— Pada era revolusi industri 4.0 ini, perkembangan teknologi menjadi semakin pesat, khususnya pada bagian komputasi Artificial Intelligence (AI) dan Machine Learning (ML) dan pada saat penulisan laporan ini sudah banyak yang mengaplikasikan teknologi ini di industri transportasi baik itu untuk mobil, pesawat dan bahkan kapal. Dalam dunia pelayaran, kejadian tubrukan antar kapal itu sering terjadi karena kelalaian manusia, sehingga diperlukannya sistem bantuan mengemudi untuk meminimalisir kejadian ini sebelum terjadi. Oleh karena itu, penulis mengaplikasikan teknologi object detection, LiDAR, dan microprocessor untuk sistem pencegahan tubrukan yang sederhana. Untuk membuat sistem ini akan dikembangkan program yang bisa memadukan object detection dan pengukuran jarak melalui LiDAR yang akan diuji dengan menggunakan model kapal yang sudah ada dan untuk parameter jarak aman penghindaran akan menggunakan metode blocking area yang dikembangkan oleh (Kijima & Furukawa, 2003). Dengan hasil, setelah pengujian 60 kali dengan variasi kecepatan dengan froude numbe (F_n) dari 0.36 hingga 0.72 sistem berhasil 100% menghindari tubrukan dengan sudut rudder paling optimal 10 derajat untuk skenario head-on collision dan setelah membandingkan 60 data terhadap sudut heading kapal maka didapat rentan sudut 8.9-14.6 derajat untuk heading kapalnya.

Kata Kunci— *Artificial Intelligence (AI)*, *Machine Learning (ML)*, *Object Detection (OD)*, *microprocessor*, *LiDAR*, *Kamera*

I. PENDAHULUAN

Industri Pelayaran merupakan salah satu pekerjaan yang paling beresiko di dunia, karena apabila terjadi kecelakaan berupa tubrukan bisa berdampak kerugian material, ekologis serta bisa menyebabkan kecacatan fisik dan bahkan kehilangan nyawa. Oleh karena itu, sebuah program navigasi autonomous perlu diciptakan sehingga apabila faktor manusia gagal, maka kapal bisa mengetahui dan bertindak untuk menyelamatkan pelaut dan barang muatannya dari kecelakaan.

Pada dunia pelayaran, laut Indonesia merupakan salah satu laut yang memiliki lalu lintas kapal tertinggi di dunia, dimana sekitar 40% kargo dunia lewat laut Indonesia (Anwar, 2020), tentu pada dari padatnya lalu lintas laut ini menimbulkan potensi kecelakaan berupa tubrukan antar kapal, terutama kapal niaga yang berlayar di pelabuhan Indonesia. Mengingat juga bahwa apabila terjadi tubrukan kapal bisa mengancam tidak hanya distribusi barang namun juga bisa mengancam nyawa pelaut dan ekosistem maritim Indonesia.

Beberapa tahun terakhir ini, banyak penelitian untuk

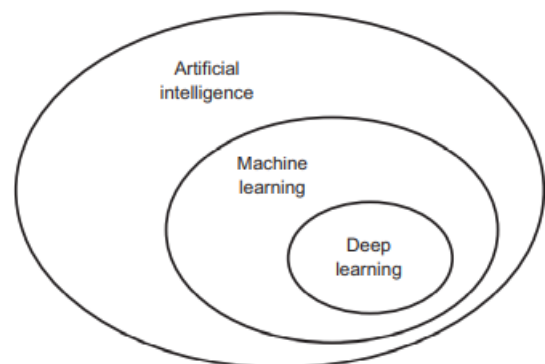
pengembangan Maritime Autonomous Surface Ship (MASS) dimana tujuan MASS ini sendiri adalah untuk mengurangi human error saat pelayaran sehingga banyak prototipe MASS yang sudah dibuat sebelumnya [1] [2]. Tujuan dari studi-studi yang sudah dilakukan sebelumnya adalah untuk menghilangkan manusia dari operasi pengendalian, namun hal ini belum bisa di capai.

Ada beberapa cara yang bisa digunakan untuk menghindari kecelakaan ini [3] [4], seperti meningkatkan otomasi pada navigasi kapal dengan pengembangan sistem autonomous kapal [5]. Dari perkembangan teknologi yang semakin pesat ini terutama di bidang Artificial Intelligence (AI) menghasilkan metode Machine Learning yang memiliki potensi pemanfaatan besar aplikasinya di bidang autonomous vehicle [6]. Namun rata-rata pengembangan autonomous vehicle masih terlalu fokus dengan mobil, sehingga pengembangan untuk MASS masih kurang dan hanya menghasilkan simulasi [7].

Oleh karena itu, dalam tugas akhir ini akan merancang sebuah konsep sistem kontrol otomatis pada kapal yang berdasarkan Machine Learning Object Detection, dimana sistem yang dirancang akan menggunakan konsep Machine Learning dan Object Detection dari sensor kamera supaya kapal bisa membuat pilihan untuk menghindari tubrukan yang meminimalisir kerugian pelaut, ekonomi dan ekosistem maritime Indonesia.

II. TINJAUAN PUSTAKA

A. Machine Learning



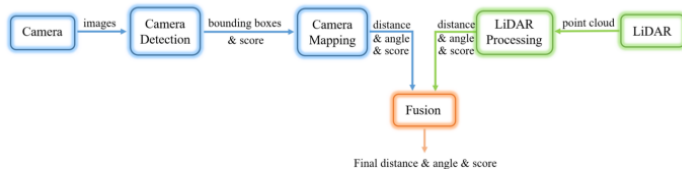
sGambar 1 Perbedaan AI, ML, dan DL

Seperti yang bisa dilihat pada sGambar 1, *Artificial Intelligence* (AI) adalah program dimana komputer mampu melakukan tugas kognitif yang biasanya di lakukan oleh manusia, *Machine Learning* (ML) adalah jenis dari AI dimana program ini mampu “belajar” dari sebuah proses yang sebelumnya tidak ada peraturan yang telah di siapkan

sebelumnya. *Deep Learning* (DL) adalah bagian dari ML, dimana model yang diciptakan ini lebih kompleks atau berlapis agar bisa mencari pola pada data yang lebih kompleks.

B. LiDAR and Camera Detection Fusion in Real- Time Industri Multi Sensor Collision Avoidance System

Penelitian yang menunjukkan implementasi penggunaan sistem *Fusion Multi-sensor* pada lingkungan industrial, dimana dari multi-sensor itu bisa dibuat peraturan tambahan untuk tiap sensor dan sensor-sensor itu saling melengkapi apa yang tidak bisa dilakukan sensor yang lain. Cara kerja sistem ini bisa dilihat pada Gambar 2.

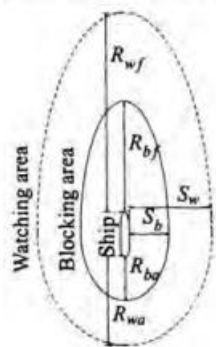


Gambar 2 System Fusion Sensor

Dalam sistem ini, tiap sensor itu memiliki tujuan masing-masing. Dimana sensor kamera berguna untuk menerima data berupa gambar 2 Dimensi untuk *Object Detection* dan LiDAR berguna untuk menerima data komplementer pengukuran jarak antara alat dengan objek tersebut. Gunanya dari fusi ini adalah untuk memprogram alat untuk melakukan sesuatu jika dihadapi scenario yang menggunakan salah satu data spesifik dari salah satu sensor itu, contoh memprogram alat tidak mendekat orang, menajuhi area yang di tandai warna biru atau berhenti sebelum bertubrukan dengan tembok.

C. Automatic Collision Avoidance System dengan Konsep Blocking Area

Sebuah penelitian yang memberikan rumus empiris untuk estimasi jarak keamanan kapal untuk menghindari tubrukan dengan kapal lain, area ini bisa divisualisasikan pada Gambar 3.



Gambar 3 Blocking Area

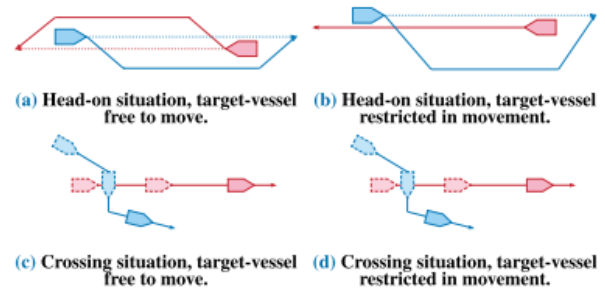
Daerah yang didapat itu berupa garis imajiner elips yang melingkari badan kapal, dimana garis itu menandakan 3 daerah, yakni:

1. R_{bf} : Radius longitudinal *Blocking Area* Fore
2. R_{ba} : Radius longitudinal *Blocking Area* Aft
3. S_b : Radius transversal *Blocking Area*

Ketiga daerah itu di dapat melalui persamaan (1).

$$\begin{cases} R_{bf} = L + (1 + s)T_{90}U \\ R_{ba} = L + T_{90}U \\ S_b = B + (1 + t)D_T \end{cases} \quad (1)$$

D. COLREG-Compliant Collision Avoidance for Unmanned Surface Vehicle using Deep Reinforcement Learning



Gambar 4 Definisi Skenario Tabrakan pada COLREGs

Penelitian yang dilakukan menghasilkan sebuah rekomendasi sistem untuk *Collision Avoidance* yang sesuai dengan hukum COLREGs yang berlaku seperti yang bisa dilihat pada Gambar 4, khusus nya COLREGs part B section 2 Rules:

1. *Rule 14 (Head-on Situation)* : Ketika terdapat dua kapal bermesin yang akan bertemu pada arah yang berlawanan sehingga beresiko untuk saling bertabrakan maka kedua kapal harus merubah arah ke arah *starboard* sehingga saling melewati dari sisi *portside*.
2. *Rule 15 (Crossing Situation)* : Ketika kedua kapal bermesin itu saling menyebrang sehingga menimbulkan erisiko untuk tubrukan maka kapal yang melihat kapal lain di sisi *starboard*nya harus menjauh dan, jika perlu, menghindari situasi crossing secara keseluruhan.
3. *Rule 18 (Responsibilities between Vessels)* : Setiap kapal memiliki tanggung jawab sendiri saat berlayar, dimana:

- a) Kapal bermesin yang sedang berlayar harus menghindari dari kapal lain yang terbatas oleh kemampuannya untuk bermanuver.

Artinya, kapal yang berukuran lebih kecil harus memberi jalan kepada kapal yang berukuran lebih besar, karena kemampuan *maneuvering* kapal yang berukuran lebih kecil akan lebih tinggi daripada kapal yang berukuran besar. *Machine Learning dan Ekstraksi Fitur*

III. METODOLOGI

A. Analisis Permasalahan

Merupakan Langkah awal dari pengerjaan Tugas Akhir, dimana permasalahan di *breakdown* menjadi sub masalah sehingga bisa dimengerti dan juga bisa di pecahkan dengan Langkah-langkah yang sesuai

B. Studi Literatur

Setelah intisari dari permasalahan telah ditentukan maka selanjutnya adalah untuk mencari dasar teori dan juga penelitian terdahulu yang pernah dilakukan dan relevan dalam pemecahan masalah.

C. Pembuatan Prototipe

Tahap ini dilakukan setelah mendapat gambaran penyelesaian masalah, dimana perlu dibuat prototipe untuk di uji coba sehingga membuktikan apakah hipotesa awal benar atau tidak. Dalam Tugas Akhir ini, prototipe yang dimaksud adalah prototipe Elektronik yang menggambarkan sistem yang akan dibuat, untuk model kapal yang di uji coba merupakan model kapal yang dibuat pihak lain agar merepresentasikan sistem bekerja pada kapal apapun ukurannya, pada Tugas Akhir ini, spesifikasi model kapal yang digunakan bisa dilihat pada .

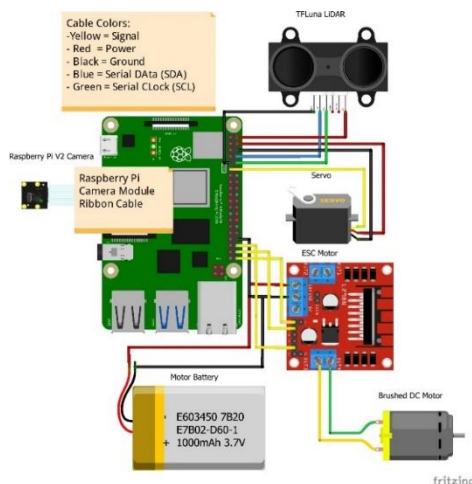
D. Pembuatan Program

Untuk membuat program ini, perlu dipecah menjadi mini program untuk mengurangi kompleksitas pengujian *software* agar bisa diketahui kesalahan sistem dimana, mini program itu terdiri dari:

1. Program *Object Detection*/Kamera: untuk memasitkan *object detection* bisa dilakukan dengan sensor kamera.
2. Program LiDAR: Untuk Menguji kinerja Sensor LiDAR.
3. Program Aktuator: Untuk menguji actuator yang sudah terpasang pada kapal dan juga untuk mencari cara untuk mengendalikannya
4. Program *Wrapper*: Program yang menggabungkan 3 permasalahan kecil itu menjadi satu program besar yang kompleks

E. Perakitan Sistem Elektronik

Dalam tahap ini, sistem sensor dan actuator digabung dengan *microprocessor* sesuai dengan diagram sketsa pada Gambar 6. Dimana *Raspberry Pi* berperan sebagai CPU dan juga *central command unit* untuk actuator, sensor berupa kamera yang berhubungan dengan *ribbon cable* dan juga LiDAR terhubung dengan kabel GPIO. Untuk sistem actuator, servo bisa secara langsung dihubungkan ke *raspberry pi* dengan kabel GPIO karena tidak membutuhkan arus listrik yang besar, hal ini tidak berlaku untuk mesin kapal. Dimana mesin yang digunakan adalah *brushed DC motor* yang memerlukan sumber daya dengan arus yang jauh lebih besar dari kebutuhan *raspberry pi*, sehingga diperlukan komponen tambahan untuk menghubungkan *raspberry pi* dengan motor dan juga mengendalikan arus motor, apabila sumber daya motor langsung dihubungkan ke *raspberry pi* maka *raspberry pi* akan konslet dan terbakar.



Gambar 6 Sketsa Sistem CAS

F. Penggabungan Software dan Hardware

Tahap ini merupakan tahap yang crucial untuk memulai percobaan, mencoba menjalankan sistem sensor dan actuator pada *hardware* yang telah dipasang agar bisa menemukan kesalahan pada *software* sebelum pengujian dimulai

G. Trial Darat (System Function Test)

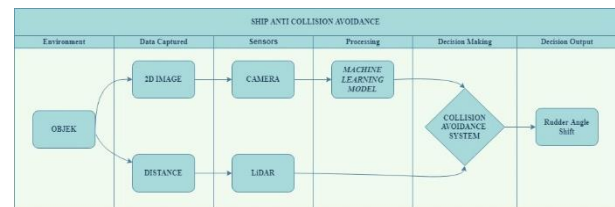
Tahap ini dilakukan untuk mendapatkan performa pengukuran dan pembacaan dari sistem sensor yang telah dipasang pada kapal, agar mengetahui akurasi dan bisa dilakukan kalibrasi jika diperlukan

H. Trial Air

Merupakan tahap terakhir Tugas Akhir ini, dimana dilakukannya percobaan pada air yang tenang untuk mengetahui performa sistem pada simulasi di dunia nyata sesuai yang di definisikan pada rules COLREGs. Dari hasil percobaan akan ditarik kesimpulan juga evaluasi dari seluruh sistem yang telah dibuat.

IV. HASIL EKSPERIMEN DAN PEMBAHASAN

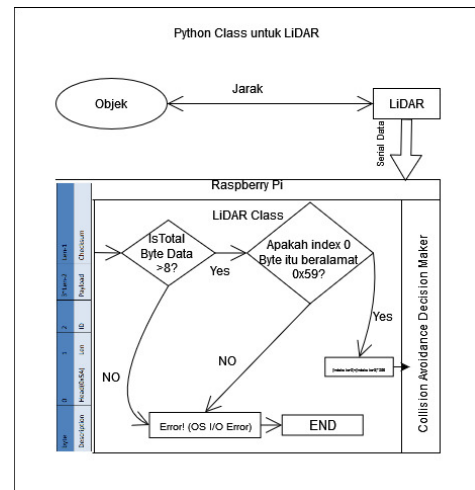
A. Perancangan Sistem Collision Avoidance



Gambar 5 Flowchart Rancangan Sistem CAS

Dari Gambar 5, bisa dilihat *outline* dari rancangan sistem CAS yang akan dibuat. Dimana sistem akan menerima data dari lingkungan/*environment* berupa gambar 2D dan jarak dari *object* dengan dua sensor *independent*, lalu kedua data akan di proses secara *concurrent* sendiri-sendiri. Dimana gambar 2D akan di Dimana gambar 2D akan di masukan pada model *machine learning* untuk deteksi objek terlebih dahulu sebelum ke dalam *decision maker* CAS, tapi jarak akan langsung dijadikan data primer untuk CAS karena jarak disesuaikan dengan variasi kecepatan dan juga perhitungan *blocking area*.

B. Pengujian Code Kamera dan LiDAR



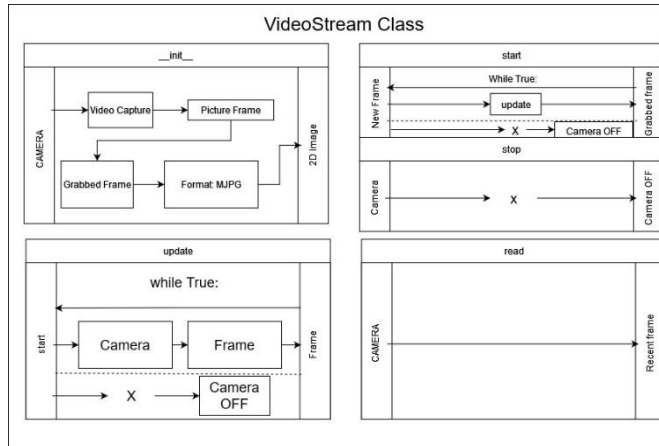
Gambar 7 Python Class untuk LiDAR

Apabila Rancangan CAS dari Gambar 5, maka Langkah selanjutnya adalah untuk membuat program yang memenuhi rancangan tersebut. Dimana program akan dibuat secara individu setelah di *breakdown* tiap masalah dalam rancangan tersebut, menjadi 4 program yakni:

1. Program *object detection*
2. Program LiDAR
3. Program Actuator
4. *Wrapper Code/Sistem CAS*

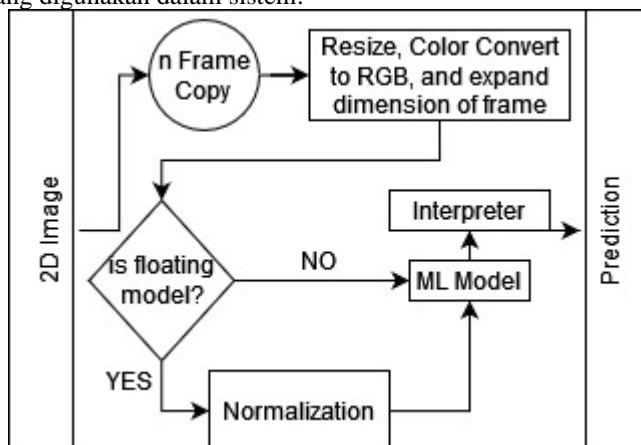
Setelah me

Dimana program yang pertama akan dibuat cara kerja sesuai dengan diagram Gambar 7, yang merupakan *while True loop* yang berarti pengukuran akan terus dilakukan sensor hingga program diberikan perintah untuk berhenti atau terganggu operasinya. Untuk program *Object Detection*, cara kerjanya adalah untuk memecahkan program besar menjadi dua program yang lebih kecil yakni, program untuk aktivasi *livestream* kamera dan juga program untuk *object detection* sendiri. Dimana untuk *livestream* akan mengikuti cara kerja dari Gambar 8.



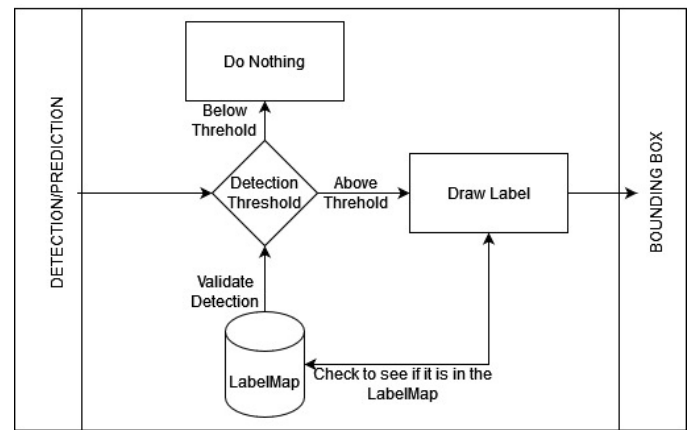
Gambar 8 Flowchart Program Livestream Kamera

Setelah membuat program untuk *livestream*, program *object detection* akan dibuat mengikuti dengan cara kerja dari Gambar 9 untuk menampilkan prediksi dari model *machine learning* yang digunakan dalam sistem.



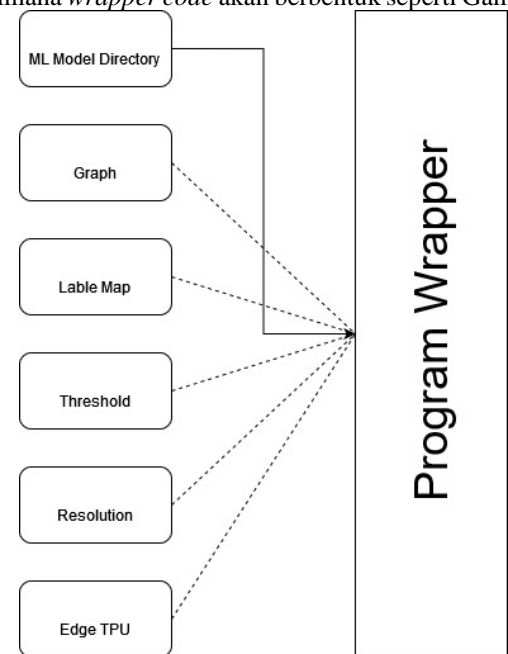
Gambar 9 Flowchart Object Detection Menampilkan Prediksi Model ML

Dan juga Gambar 10 untuk menggambarkan *bounding box* terhadap objek yang telah berhasil di prediksi model *machine learning*.



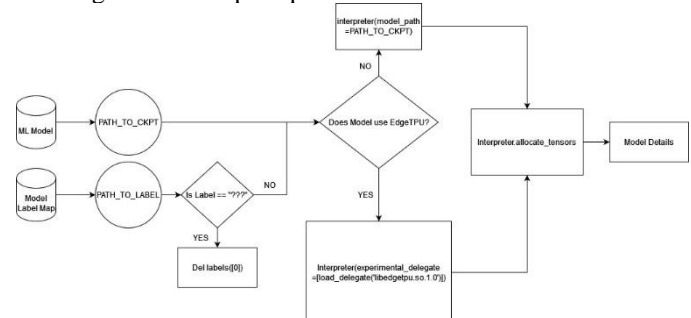
Gambar 10 Flowchart untuk Menggambar Bounding Box dari Prediksi Objek

Setelah program untuk *object detection* dan juga LiDAR telah dibuat, berikutnya adalah untuk membuat program untuk actuator, motor dan servo untuk menemukan kombinasi *ChangeDutyCycle* (CDC) yang tepat dari hasil *trial and error*, maka akan di dapat hasil data tentang actuator seperti pada Gambar 14 dan Gambar 16, lalu menggabungkan ke 3 fungsi itu dengan menggunakan *wrapper code* atau dengan sistem CAS. Dimana *wrapper code* akan berbentuk seperti Gambar 11.



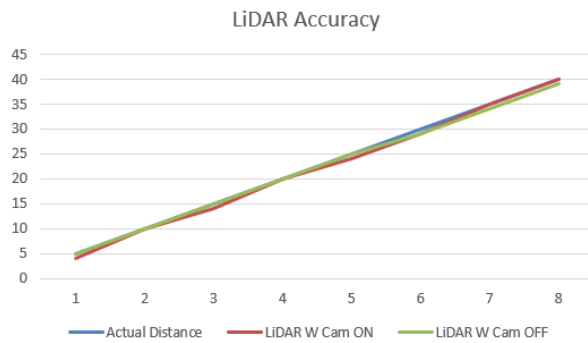
Gambar 11 Flowchart Wrapper Code

Dimana untuk memasukan model *machine learning* sendiri akan mengikuti cara seperti pada Gambar 12.



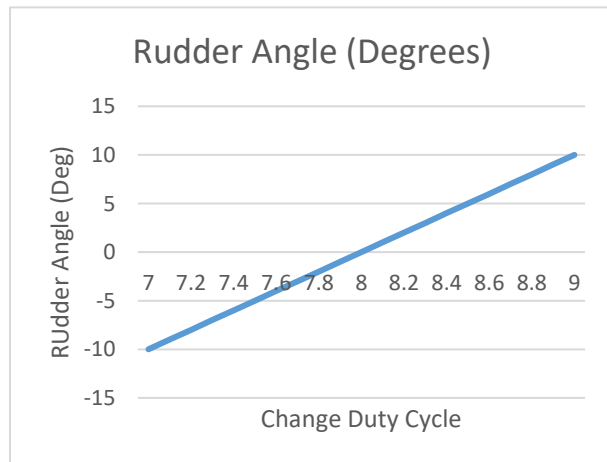
Gambar 12 Cara Import Model Machine Learning

C. Trial Darat



Gambar 13 Hasil Pengukuran LiDAR pada Keadaan Kamera ON dan OFF

Dalam percobaan darat, sistem yang diuji adalah fungsi dari sensor, berupa LiDAR dan *onboard Camera*. Dimana dalam pengujian sensor LiDAR akan diuji dengan mengukur jarak benda dengan jarak interval yang akan di validasikan dengan pengukuran menggunakan meteran yang satuan ukurnya menggunakan standar internasional, dimana hasil ini bisa dilihat pada **Error! Reference source not found.** Dari 20 kali pengukuran dengan interval 5 cm itu diketahui bahwa untuk jarak yang lebih dari 1 m sensor masih bisa digunakan karena pada jarak tersebut pengukuran LiDAR pada jarak tersebut <5% apabila sensor digunakan bersamaan dengan sistem kamera ML.



Gambar 14 Hasil Uji Sudut *Rudder*

Selain penujian sensor, perlu juga dilakukan pengujian sistem actuator sebelum pengujian di air untuk mencari konfigurasi elektronik yang tepat di dalam program, terutama untuk konfigurasi *ChangeDutyCycle* (CDC)nya *servo* untuk mengendalikan *rudder* kapal. Untuk menguji *servo*, maka diperlukan busur untuk mengukur perubahan sudut dan juga penanda yang dipasang pada poros *rudder* kapal untuk mengamati perubahan sudut *rudder*. Dari pengujian tersebut, diketahui bahwa pergerakan sudut *rudder* kapal terbatas dengan jarak sudut 10°, dimana untuk nilai CDC 7 akan menghasilkan perubahan sudut *rudder* 80° (*Port*), untuk menghasilkan sudut netral berupa 90° maka diperlukan nilai CDC 8, dan untuk sudut 110° (*Starboard*). Seperti yang bisa dilihat pada Gambar 14.

D. Trial Air

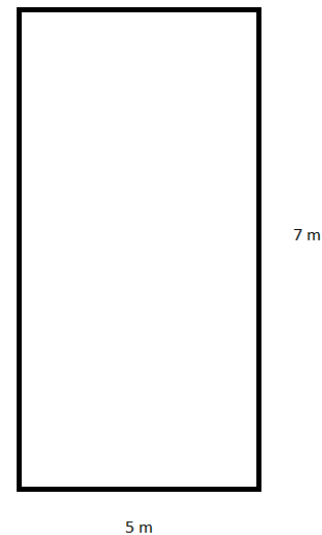
Setelah skenario ditentukan, dalam Tugas Akhir ini scenario yang dipilih adalah *Head-On Collision*, maka selanjutnya adalah pengambilan data pada air untuk menguji kinerja sistem. Data yang akan diambil dari pengujian air adalah

1. Performa model kapal
2. Kinerja sistem

Dimana untuk mengambil data performa kapal, akan diuji pada kolam yang diukur terlebih dahulu untuk bisa mendapat data kecepatan model. Dimana pengambilan data akan diuji dengan waktu kapal untuk menempuh panjang kolam dari ujung ke ujung kolam. Dengan hasil pengambilan data itu bisa di jadikan data tabular, seperti Tabel 1.

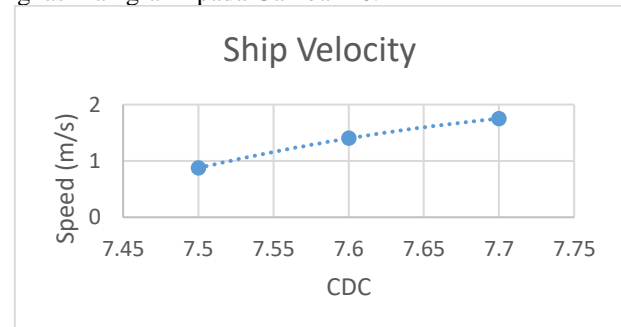
Tabel 1 Data Kecepatan Kapal

Engine Function Test				Panjang Kolam = 7 m
No.	Cdc	Speed (m/s)	Speed (Knots)	Time (s)
1	7.5	0.875	1.70086	8
2	7.6	1.4	2.721376	5
3	7.7	1.75	3.40172	4



Gambar 15 Visualisasi Dimensi Kolam Uji

Apabila data pada Tabel 1 di visualisasikan, maka akan menghasilkan grafik pada Gambar 16.



Gambar 16 Visualisasi Data Kecepatan Kapal

Dari pengambilan data untuk *servo* dan motor maka sistem bisa di konfigurasi agar menyesuaikan dengan performa kapal. Jadi perhitungan *blocking area* akan terlihat seperti Tabel 2. Sehingga bisa diketahui bahwa untuk variasi yang lebih cepat, maka *blocking area* yang dibutuhkan juga bertambah cukup signifikan.

Tabel 2 Perhitungan *Blocking Area* untuk Model

Cdc	Speed (m/s)	Rbf (m)	Rba (m)	S (m)
7.5	0.875	2.17	1.12	0.93
7.6	1.4	2.30	1.17	1.00
7.7	1.75	2.37	1.19	1.04
7.8	1.66666667	2.35	1.18	1.03

Apabila data dibandingkan dengan kapal dengan ukuran yang di skala 10:1 dengan model maka perhitungan akan menjadi seperti Tabel 3.

Tabel 3 Perhitungan *Blocking Area* pada Kapal Pembanding

speed (knot)	Speed (m/s)	Rbf (m)	Rba (m)	S (m)	Froude number kapal	Fn Model
17.01011	8.75	293.28	137.76	138.24	0.36066	0.36066
27.21617	14	313.15	144.38	150.95	0.577056	0.577056
34.02022	17.5	323.21	147.74	157.50	0.72132	0.72132
32.40021	16.66667	320.98	146.99	156.04	0.686971	0.686971

Dimana, setelah konfigurasi sistem telah dilakukan, maka Langkah selanjutnya adalah untuk mengambil data model dan juga prototipe sistem di kolam uji coba dengan variasi kecepatan dan juga jarak parameter *blocking area* yang divariasikan menjadi dua, parameter jarak penuh dan 75% dari jarak parameter *blocking area*.

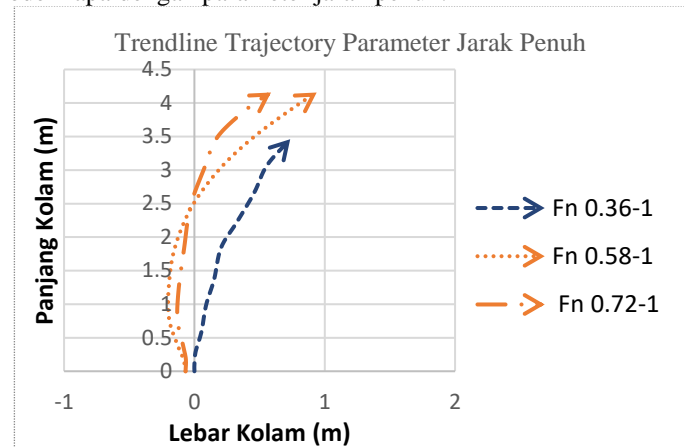
Dari 60 kali pengujian, diketahui bahwa sistem telah berhasil menghindari tabrakan sebanyak 60 kali. Maka sistem bisa disimpulkan mempunyai *success rate* 100% dan untuk tiap variasinya akan menghasilkan bentuk lintasan/*trajectory* yang berbeda-beda.

Tabel 4 Sampel Koordinat yang Didapat dari *Blender*

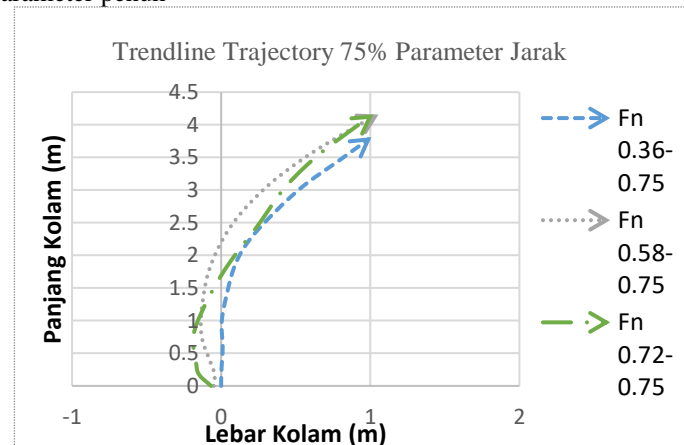
	1		2		3		4		5	
f	x	y	x	y	x	y	x	y	x	y
10	0	0	0	0	0	0	0	0	0	0
20	0	0.01	0	0.01	0	0.01	0	0.01	0	0.01
30	0	0.1	0	0.1	0	0.1	0	0.1	0	0.1
40	0	0.29	0	0.29	0	0.29	0	0.29	0.07	0.29
50	0.12	0.59	0	0.59	0.11	0.59	0.05	0.59	0.168995	0.59
60	0.1	0.98	0	0.98	0.17	0.98	0	0.98	0.190239	0.98
70	0.15	1.37	0	1.37	0.32	1.37	0.09	1.37	0.330239	1.37
80	0	1.84	0	1.84	0.42	1.84	0.13	1.84	0.486763	1.84
90	0.1	2.22	0.14	2.22	0.71	2.22	0.19	2.22	0.658228	2.22
100	0.22	2.66	0.26	2.66	0.84	2.66	0.33	2.66	0.84343	2.66

Untuk pengambilan data *trajectory*, model kapal akan di rekam dengan *handphone* dalam format video, lalu setelah perekaman video tersebut akan dirubah menjadi model 3D oleh *software fspy* dan kemudian dari model acuan itu akan dicari satu per satu koordinat pergerakan kapal dalam *software* yang

berbeda Bernama *blender*, seperti yang di tunjukan pada Tabel 4. Hasil proses *blende* akan dimasukkan dalam excel, dimana akan di periksa, olah dan visualisaikan seperti pada Gambar 17 untuk *trajectory*, yang menunjukkan rata-rata bentuk lintasan model kapaldengan parameter jarak penuh.

Gambar 17 Visualisasi *Trajectory* Rata-Rata Model Parameter Jarak Penuh

Dan juga Gambar 18, yang menunjukkan bentuk lintasan rata-rata untuk model yang menggunakan 75% dari parameter jarak yang diperhitungkan dengan *blocking area*. Dari kedua grafik itu maka bisa disimpulkan bahwa kecepatan semakin tinggi maka perlu *blocking area* yang lebih banyak dan jarak apabila menggunakan dari 100% *blocking area* akan menghasilkan sudut *heading* yang lebih besar daripada penggunaan jarak parameter penuh

Gambar 18 Visualisasi *Trajectory* Rata-Rata Model 75% Parameter Jarak

Selain *trajectory*, data air yang perlu adalah sudut *heading* kapal. dengan metode yang digunakan untuk mencari data sudut *heading* ini miirp dengan metode pencarian data *trajectory*, namun yang di olah dalam sudut adalah perubahan lateral secara sumbu horizontal kapal sehingga, data sudut *heading* akan dijadikan data tabular seperti Tabel 5.

Tabel 5 Rata-Rata Sudut *Heading* Kapal

Average Heading (Deg)	
Variasi Fn 0.36-1	12.6
Variasi Fn 0.36-0.75	14.5
Variasi Fn 0.58-1	13.4

Average Heading (Deg)	
Variasi Fn 0.58-0.75	14.4
Variasi Fn 0.72-1	13.9
Variasi Fn 0.72-0.75	14.6

Setelah melakukan pengujian air sebanyak 60 kali, sistem sudah berhasil menghindari tabrakan 60 kali. Bisa disimpulkan bahwa sistem memiliki *success rate* 100% karena mampu menghindari 60 dari 60 kali percobaan tabrakan dengan variasi kecepatan dan juga jarak *blocking area* yang berbeda, dimana dicoba saat parameter jarak penuh dan juga saat dipakai 75% dari perhitungan parameter jarak *blocking area*.

V. KESIMPULAN

Setelah melakukan percobaan sistem maka kesimpulan yang bisa di ambil dari Tugas Akhir ini adalah;

1. Sistem yang dibuat mampu melakukan *object detection* dan pengukuran jarak dengan kamera dan sensor LiDAR.
2. Setelah pengujian di darat untuk memastikan sistem sensor dan aktuator bekerja, bisa disimpulkan bahwa:
 - a. Kamera *object detection* berhasil mendeteksi objek
 - b. Akurasi LiDAR mengukur jarak dengan kondisi program *object detection* aktif memiliki akurasi 95.75% dan kalau *object detection* tidak aktif, akurasi tersebut meningkat hingga 98.91%
 - c. *Servo* mampu dikendalikan dengan merubah nilai *ChangeDutyCycle* (CDC) dengan perubahan sudut sejauh 10°, dimana sudut minimal *rudder* adalah 80° yang berarti kapal berbelok arah *port*, dan maksimal sudut *rudder* adalah 100° yang berarti kapal berbelok arah *starboard*.
3. Setelah 60 kali pengujian di air dengan variasi *froude number* antara 0.36 hingga 0.72, sistem berhasil 60 kali dari 60 kali pengujian. Berarti sistem mempunyai *success rate* 100% untuk skenario *head-on collision* menggunakan metode *blocking area* [8]. Dimana untuk tiap variasi memiliki sudut *heading*:
 - a. Fn 0.36: memiliki rata-rata sudut *heading* 12.6° untuk jarak parameter penuh, dan sudut *heading* rata-rata 14.5° untuk 75% jarak parameter.
 - b. Fn 0.57: memiliki rata-rata sudut *heading* 13.4° untuk jarak parameter penuh, dan sudut *heading* rata-rata 14.4° untuk 75% jarak parameter.
 - c. Fn 0.72: memiliki rata-rata sudut *heading* 8.9° untuk jarak parameter penuh, dan sudut *heading* rata-rata 14.6° untuk 75% jarak parameter.

Namun, pembacaan dari LiDAR masih cenderung salah karena LiDAR merupakan sensor yang sensitif terhadap cahaya sehingga saat pengujian di lapangan terbuka yang disinari cahaya matahari, sensor LiDAR salah membaca dan terjadi *margin of error* yang cukup signifikan, dimana

sensor membaca jarak menjadi lebih dekat dari jarak aslinya.

4. Sistem bisa diaplikasikan di kapal, namun perlu pengembangan untuk meningkatkan kinerja dan juga fungsi otomasi sistem menjadi *advance guidance system* untuk mencapai *Maritime Autonomous Surface Ship* (MASS) yang kompleks.

DAFTAR PUSTAKA

- [1] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annu. Rev. Control*, vol. 41, pp. 71–93, Jan. 2016, doi: 10.1016/J.ARCONTROL.2016.04.018.
- [2] M. Schiaretti, L. Chen, and R. R. Negenborn, "Survey on autonomous surface vessels: Part II - Categorization of 60 prototypes and future applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10572 LNCS, pp. 234–252, 2017, doi: 10.1007/978-3-319-68496-3_16.
- [3] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. H. A. J. M. van Gelder, "Ship collision avoidance methods: State-of-the-art," *Saf. Sci.*, vol. 121, no. April 2019, pp. 451–473, 2020, doi: 10.1016/j.ssci.2019.09.018.
- [4] R. Lagisetty, N. K. Philip, R. Padhi, and M. S. Bhat, "Object detection and obstacle avoidance for mobile robot using stereo camera," *Proc. IEEE Int. Conf. Control Appl.*, no. August, pp. 605–610, 2013, doi: 10.1109/CCA.2013.6662816.
- [5] M. Akdağ, P. Solnør, and T. A. Johansen, "Collaborative collision avoidance for Maritime Autonomous Surface Ships: A review," *Ocean Eng.*, vol. 250, no. February, p. 110920, 2022, doi: 10.1016/j.oceaneng.2022.110920.
- [6] A. N. Catapang and M. Ramos, "Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle," *Proc. - 6th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2016*, pp. 441–445, 2017, doi: 10.1109/ICCSCE.2016.7893614.
- [7] H. Zhou, Z. Ren, M. Marley, and R. Skjetne, "A guidance and maneuvering control system design with anti-collision using stream functions with vortex flows for autonomous marine vessels," pp. 1–15, 2021, [Online]. Available: <http://arxiv.org/abs/2106.02405>.
- [8] K. Kijima and Y. Furukawa, "Automatic collision avoidance system using the concept of blocking area," *IFAC Proc. Vol.*, vol. 36, no. 21, pp. 223–228, 2003, doi: 10.1016/S1474-6670(17)37811-4.