SISTEM DETEKSI RINTANGAN PADA KAPAL OTONOM BERBASIS ALGORITMA FASTER R-CNN MENGGUNAKAN CITRA VISUAL

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang

DISUSUN OLEH :

NATASHA FARAS NABILA

NIM 32602100099

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025

FINAL PROJECT

OBSTACLE DETECTION SYSTEM FOR AUTONOMOUS SHIP BASED ON THE FASTER R-CNN ALGORITHM USING VISUAL IMAGES

Proposed to complete the requirement to obtain a bachelor's degree (S1) at
Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University

Arranged By:

NATASHA FARAS NABILA
NIM 32602100099

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2025

LEMBAR PENGESAHAN

TUGAS AKHIR

SISTEM DETEKSI RINTANGAN PADA KAPAL OTONOM BERBASIS ALGORITMA FASTER R-CNN MENGGUNAKAN CITRA VISUAL

NATASHA FARAS NABILA
NIM 32602100099

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 22 Mei 2025

TIM PENGUJI UJIAN SARJANA :

Badieah, S.T., M.Kom ..... 28-5-2025 .....
NIK. 210615044
(Ketua Penguji)

Ghufron, S.T., M.Kom ..... 28-5-2025 .....
NIK. 210622056
(Anggota Penguji)

Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D ..... 28-5-2025 .....
NIK. 210600017
(Pembimbing)

Semarang, 22 Mei 2025
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, S.T., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Natasha Faras Nabila

NIM : 32602100099

Judul Tugas Akhir : Sistem Deteksi Rintangan Pada Kapal Otonom Berbasis Algoritma Faster R-CNN Menggunakan Citra Visual

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 22 Mei 2025

Yang Menyatakan,

Natasha Faras Nabila

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: Natasha Faras Nabila
NIM	: 32602100099
Program Studi	: Teknik Informatika
Fakultas	: Teknologi industri
Alamat Asal	: Tegal, Jawa Tengah

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **"Sistem Deteksi Rintangan Pada Kapal Otonom Berbasis Algoritma Faster R-CNN Menggunakan Citra Visual"** Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 22 Mei 2025

Yang menyatakan,

Natasha Faras Nabila

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas rahmat dan karunia Allah SWT, penulis dapat menyelesaikan Tugas Akhir dengan judul "Sistem Deteksi Rintangan Pada Kapal Otonom Berbasis Algoritma Faster R-CNN Menggunakan Citra Visual" sebagai syarat memperoleh gelar Sarjana (S1) pada Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya dukungan dari berbagai pihak, materi maupun teknis, oleh karena itu penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. H. Gunarto, S.H., M.H., selaku Rektor Universitas Islam Sultan Agung Semarang.
2. Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T., IPU., ASEAN.Eng., selaku Dekan Fakultas Teknologi Industri.
3. Bapak Moch. Taufik, S.T., M.I.T., selaku Ketua Program Studi Teknik Informatika.
4. Bapak Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D., selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan motivasi selama penyusunan tugas akhir ini.
5. Kedua orang tua, mbah ibu, kakak dan adek, yang telah memberikan doa dan dukungan kepada penulis sehingga bisa menyelesaikan tugas akhir ini.
6. Dan pada semua pihak yang tidak bisa saya sebutkan satu persatu.

Dengan kerendahan hati, penulis menyadari masih banyak kekurangan dalam laporan ini dan mengharapkan saran serta kritik yang membangun untuk perbaikan di masa mendatang.

Semarang, 22 Mei 2025

Natasha Faras Nabila

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Navigasi kapal otonom memerlukan sistem deteksi rintangan yang andal untuk menghindari tabrakan dan menjaga keselamatan operasional selama pelayaran. Salah satu pendekatan yang digunakan dalam pengembangan sistem ini adalah penerapan model *deep learning* Faster R-CNN, yang dikenal memiliki performa tinggi dalam mendeteksi objek pada citra visual. Dataset yang digunakan berasal dari *Obstacle Detection in a Marine Environment*, yang berisi kumpulan gambar maritim dengan anotasi berbagai jenis rintangan yang umum dijumpai di laut. Proses anotasi dilakukan menggunakan *Roboflow*, sebuah platform yang mempermudah proses pelabelan objek, dan hasil anotasi diformat menggunakan standar *COCO*. *Dataset* dibagi menjadi tiga bagian utama, yaitu data *pelatihan*, *validasi*, dan *pengujian* untuk memastikan model mampu melakukan generalisasi dengan baik. Model Faster R-CNN dilatih menggunakan *dataset* tersebut dan dievaluasi berdasarkan metrik *akurasi*, *precision*, *recall*, dan *F1-score*. Hasil evaluasi menunjukkan performa yang sangat baik, dengan *akurasi* mencapai 96,00%, *precision* sebesar 99,20%, *recall* 96,00%, dan *F1-score* 97,41%. Capaian ini menunjukkan bahwa sistem deteksi yang dikembangkan mampu mengidentifikasi rintangan secara efektif, sehingga dapat mendukung peningkatan keselamatan navigasi maritim kapal otonom dalam berbagai kondisi lingkungan maritim.

Kata Kunci: Deteksi Rintangan, Kapal Otonom, Faster R-CNN, *Deep Learning*, Navigasi Maritim.

ABSTRACT

Autonomous ship navigation requires a reliable obstacle detection system to prevent collisions and ensure operational safety during voyages. One approach used in the development of such a system is the application of the deep learning model Faster R-CNN, which is known for its high performance in object detection within visual imagery. The dataset used originates from Obstacle Detection in a Marine Environment, which contains a collection of maritime images annotated with various types of obstacles commonly encountered at sea. The annotation process was conducted using Roboflow, a platform that simplifies the labeling of objects, and the annotations were formatted according to the COCO standard. The dataset was divided into three main parts: training, validation, and testing data, to ensure the model's ability to generalize effectively. The Faster R-CNN model was trained using this dataset and evaluated using the metrics of accuracy, precision, recall, and F1-score. The evaluation results indicate excellent performance, with an accuracy of 96.00%, precision of 99.20%, recall of 96.00%, and an F1-score of 97.41%. These findings demonstrate that the developed detection system is capable of effectively identifying obstacles, thereby supporting enhanced maritime navigation safety for autonomous vessels under various environmental conditions.

Keywords: *Obstacle Detection, Autonomous Ships, Faster R-CNN, Deep Learning, Maritime Navigation.*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam industri maritim *modern*, penggunaan kapal otonom menjadi inovasi penting yang bertujuan untuk meningkatkan efisiensi dan keselamatan operasional di laut. Kapal otonom, yang dirancang untuk beroperasi secara mandiri tanpa intervensi manusia, memiliki potensi besar untuk berbagai aplikasi, seperti pengawasan perairan, patroli keamanan, hingga eksplorasi wilayah terpencil. Salah satu aspek kunci dalam operasional kapal otonom adalah kemampuan mendeteksi rintangan secara akurat. Deteksi rintangan menjadi langkah pertama yang sangat penting untuk memastikan keselamatan pelayaran di lingkungan perairan yang dinamis dan penuh tantangan (Rustina dkk., 2024).

Sebagai negara kepulauan, Indonesia memiliki wilayah perairan yang luas dengan lebih dari 17.000 pulau, menjadikannya jalur pelayaran yang sangat padat (Rifiana dan Adhi, 2022). Selain itu, di wilayah perairan khususnya di laut sendiri banyak sekali menghadapi berbagai rintangan seperti pelampung, kapal portabel, kapal kargo, perahu layar, pendayung, burung camar, dan dermaga. Keberadaan rintangan ini dapat membahayakan keselamatan kapal, khususnya bagi kapal otonom yang harus mengandalkan sistem deteksi berbasis teknologi untuk mengenali dan mengidentifikasi objek-objek tersebut (Milic, 2022). Untuk itu, diperlukan sistem deteksi yang mampu bekerja secara andal dalam berbagai kondisi, seperti perubahan pencahayaan, gangguan dari gelombang, atau variasi cuaca, sehingga rintangan di perairan dapat terdeteksi dengan baik. Sistem ini harus mampu mengenali dan mengidentifikasi rintangan secara akurat berdasarkan data visual, sehingga kapal dapat mengambil tindakan penghindaran secara *real-time* (Della dan Rachmannullah, 2024).

Fokus pengembangan ini adalah pembuatan model sistem deteksi rintangan berbasis algoritma Faster R-CNN (*Faster Region-Based Convolutional Neural Network*), tanpa melibatkan tahap pembuatan jalur penghindaran. Faster R-CNN dipilih karena terbukti efektif dalam mendeteksi objek pada citra visual dengan menghasilkan *bounding box* untuk setiap objek atau jenis rintangan yang terdeteksi (Charli dkk., 2020). Dengan kemampuan tersebut, model ini diharapkan dapat menjadi dasar pengembangan lebih lanjut untuk mendukung sistem deteksi jenis rintangan pada kapal otonom, khususnya di wilayah perairan Indonesia yang kompleks.

1.2 Perumusan Masalah

1. Bagaimana merancang sistem deteksi objek rintangan menggunakan algoritma Faster R-CNN untuk mengenali berbagai jenis objek seperti kapal lain, pelampung, atau rintangan visual lainnya, guna mendukung navigasi kapal otonom di lingkungan laut yang dinamis.
2. Bagaimana Faster R-CNN dapat mendeteksi berbagai rintangan di perairan dengan akurasi tinggi.

1.3 Pembatasan Masalah

1. Sistem yang dikembangkan hanya difokuskan pada deteksi jenis rintangan tanpa mencakup pembahasan terkait arah gerak kapal setelah rintangan teridentifikasi.
2. Fokus deteksi dibatasi pada objek-objek yang berada di permukaan laut, sementara objek seperti fauna laut dan rintangan di bawah permukaan air tidak termasuk dalam ruang lingkup.
3. Pembahasan tidak mencakup kapal yang digunakan sebagai kapal otonom, melainkan difokuskan pada algoritma untuk mendeteksi berbagai jenis rintangan berbasis citra visual.

1.4 Tujuan

Merancang dan mengembangkan model deteksi jenis rintangan menggunakan algoritma Faster R-CNN yang mampu mengidentifikasi objek-objek yang ada di perairan secara akurat untuk diaplikasikan pada kapal otonom.

1.5 Manfaat

Model deteksi jenis rintangan berbasis Faster R-CNN yang dihasilkan diharapkan dapat mendukung peningkatan sistem navigasi kapal otonom dengan memberikan informasi rintangan secara *real-time* sehingga meningkatkan keselamatan dan efisiensi operasional kapal di wilayah perairan.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang akan di pakai penulis dalam pembuatan laporan tugas akhir ini sebagai berikut :

BAB I : PENDAHULUAN

Pada bab pendahuluan menyajikan permasalahan yang dibahas, yaitu latar belakang, rumusan masalah, batasan masalah, manfaat, tujuan, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini disajikan informasi mengenai tinjauan pustaka dari kajian sebelumnya dan dasar teori yang digunakan.

BAB III : METODE PENELITIAN

Pada bab ini disajikan tentang metologi penelitian yang digunakan untuk membuat sistem deteksi jenis rintangan pada kapal otonom menggunakan algoritma Faster R-CNN yang akan dibuat.

BAB IV : HASIL DAN ANALISA

Pada bab ini akan disajikan tentang hasil dan analisa dari sistem deteksi jenis rintangan pada kapal otonom yang telah dibuat.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini disajikan tentang kesimpulan dan saran dari seluruh rangkaian penelitian yang dilakukan.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Kapal otonom atau *Maritime Autonomous Surface Ships* (MASS) merupakan evolusi dalam teknologi maritim yang memungkinkan kapal beroperasi dengan minim atau tanpa intervensi manusia. Berdasarkan definisi dari *International Maritime Organization* (IMO), kapal otonom adalah kapal yang dapat beroperasi secara independen dari interaksi manusia pada berbagai tingkat. Teknologi ini mengintegrasikan kecerdasan buatan (*Artificial Intelligence*), sensor canggih, serta sistem kontrol otomatis untuk menggantikan berbagai fungsi yang sebelumnya dijalankan oleh awak kapal (Della dan Rachmannullah, 2024). Salah satu sistem yang telah dikembangkan untuk mendukung navigasi kapal tanpa awak berbasis pengolahan citra digital menggunakan pendekatan gabungan antara deteksi tepi, transformasi *Hough*, dan metode deteksi *saliensi*. Proses dimulai dengan penentuan garis *horizon* menggunakan deteksi tepi dan transformasi *Hough*, kemudian area di bawah *horizon* dianalisis untuk mengidentifikasi objek melalui metode *saliensi* (Pratama, Boby Mugi dan Mayantasari, 2022).

Penggunaan algoritma *Faster R-CNN* dengan arsitektur *VGG16* telah berhasil diaplikasikan untuk mendeteksi objek berupa lokomotif dan gerbong pada gambar dengan berbagai kondisi pencahayaan dan jarak. Hasil yang diperoleh menunjukkan rata-rata akurasi mencapai 86,40% untuk lokomotif dan 97,23% untuk gerbong. Evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* masing-masing mencapai nilai 88%, 93%, dan 93% (Pardede dan Hardiansyah, 2022). Hasil tersebut menunjukkan bahwa metode ini mampu bekerja secara optimal pada berbagai kondisi lingkungan.

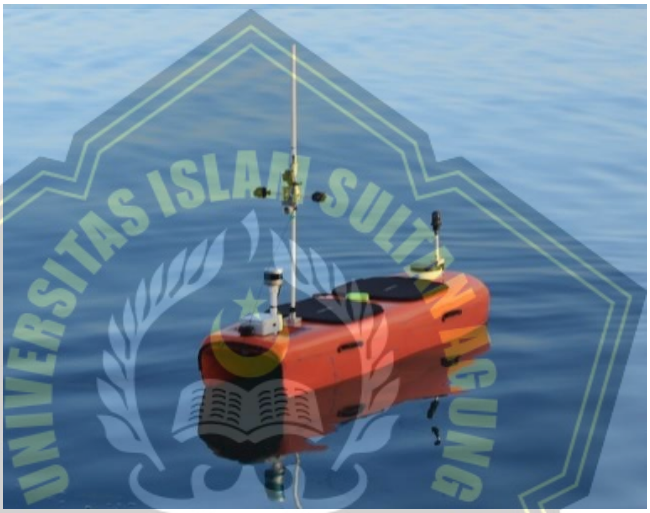
Penerapan algoritma yang sama juga telah digunakan untuk mengenali berbagai jenis burung *lovebird* melalui *bounding box* dan klasifikasi visual, dengan tingkat akurasi bervariasi antara 78% hingga 99% tergantung pada variasi warna, bentuk objek, serta kualitas input citra (Charli dkk., 2020). Hal ini membuktikan kemampuan generalisasi *Faster R-CNN* terhadap objek dengan karakteristik yang beragam. Dalam konteks keamanan biometrik, penggunaan *Faster R-CNN* dengan arsitektur *ResNet50* mampu membedakan wajah asli dan *spoofing* dari video. Model tersebut mencatat rata-rata akurasi 97,07% setelah 21 *epoch* pelatihan, dengan akurasi tertinggi mencapai 97,63%. Meski begitu, performa model menurun drastis hingga 56,48% dalam kondisi pencahayaan gelap. Selain itu, dominasi data wajah pria dalam proses pelatihan menyebabkan wajah wanita *non-spoofing* sering tidak terklasifikasi dengan tepat, sehingga keberagaman data perlu ditingkatkan untuk mendukung akurasi deteksi dalam kondisi nyata (Sunario Megawan dan Wulan Sri Lestari, 2020).

Penerapan *Faster R-CNN* juga terbukti efektif dalam sistem deteksi okupansi parkir serta penentuan batas *slot* parkir otomatis menggunakan citra dari kamera *CCTV*. Model *ResNet50* yang di-*fine-tune* dengan *dataset PKLot* mampu mencapai akurasi deteksi hingga 99,2% pada data *Barry Street*. Selain itu, *delineasi slot* parkir otomatis yang didukung oleh analisis spasial dan temporal, serta proses *post-processing*, mampu meningkatkan presisi rata-rata dari 37,1% menjadi 80,4% (Acharya dan Khoshelham, 2021). Hasil ini menunjukkan potensi besar untuk diterapkannya sistem parkir cerdas di lingkungan tanpa marka tetap. Secara keseluruhan, algoritma *Faster R-CNN* memiliki potensi signifikan dalam berbagai aplikasi deteksi objek, termasuk dalam sistem deteksi rintangan untuk kapal otonom. Sistem ini dapat bekerja dengan akurasi tinggi dan mampu beradaptasi terhadap kondisi lingkungan kompleks. Proses deteksi dilakukan dengan mengambil citra visual melalui kamera yang dipasang di kapal, kemudian diproses menggunakan metode *deep learning* untuk mengenali dan mengklasifikasikan objek berdasarkan informasi visual (Putro dkk., 2020).

2.2 Dasar Teori

2.2.1 Kapal Otonom

Kapal otonom merupakan salah satu bentuk inovasi dalam dunia maritim yang dirancang untuk dapat beroperasi tanpa awak manusia. Teknologi ini memungkinkan kapal melakukan navigasi, pengambilan keputusan, dan penghindaran rintangan secara otomatis dengan bantuan sensor dan algoritma berbasis kecerdasan buatan. Untuk memberikan gambaran visual mengenai bentuk dan perangkat kapal otonom.



Gambar 2. 1 Kapal Otonom (Milic, 2022)

Pada Gambar 2.1 adalah kapal otonom, kapal yang dilengkapi dengan teknologi otomatisasi untuk dapat beroperasi tanpa campur tangan manusia. Kapal ini menggunakan berbagai sensor seperti kamera, radar, dan lidar untuk memperoleh data tentang lingkungan sekitar dan membuat keputusan *navigasi*. Algoritma yang digunakan untuk kapal otonom melibatkan kecerdasan buatan (AI) dan *machine learning* untuk memungkinkan kapal mengenali objek, memetakan jalur, dan menghindari rintangan secara mandiri (IMO, 2021).

2.2.2 Deteksi Rintangan Pada Kapal Otonom

Deteksi rintangan pada kapal otonom merupakan proses penting untuk mengidentifikasi objek atau hambatan di perairan yang dapat mengganggu jalur pelayaran. Teknologi yang dimanfaatkan dalam sistem ini mencakup sensor visual seperti kamera, lidar, radar, serta teknik pemrosesan citra

digital. Salah satu pendekatan yang banyak digunakan adalah deteksi objek berbasis citra visual menggunakan metode *deep learning*. Algoritma seperti Faster R-CNN terbukti mampu mengenali dan mengklasifikasikan berbagai jenis objek di sekitar kapal, seperti pelampung, kapal portabel, kapal kargo, perahu layar, pendayung, burung camar, dan dermaga (Putro dkk., 2020).

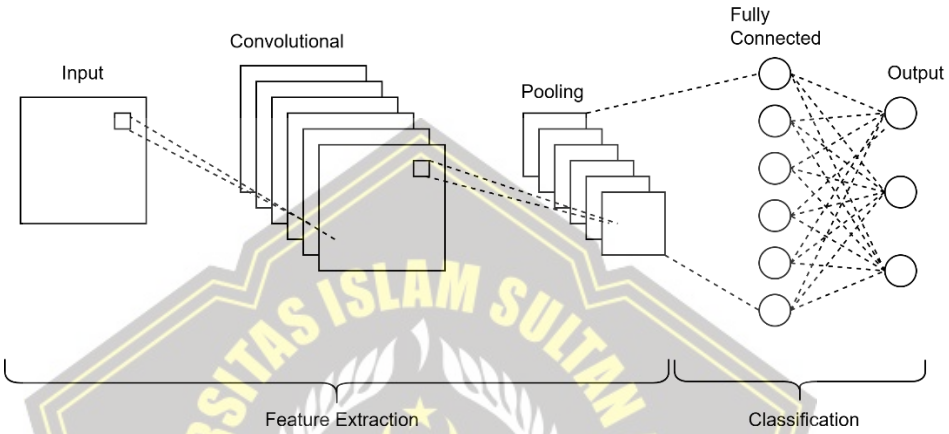
Keberhasilan pendekatan ini telah dibuktikan melalui berbagai studi. Misalnya, (Pratama, Boby Mugi dan Mayantasari, 2022) merancang sistem navigasi kapal tanpa awak dengan menggabungkan metode deteksi tepi, transformasi *Hough,* dan deteksi *saliensi* untuk mengenali objek di permukaan laut. Sistem ini mampu mengidentifikasi rintangan berdasarkan area di bawah garis horizon, sehingga dapat meningkatkan kemampuan navigasi otomatis. Selain itu, (Tersek dkk., 2023) memperkenalkan pendekatan efisien berbasis *deep learning* untuk deteksi rintangan laut yang dapat dijalankan secara *real-time* pada perangkat komputasi tertanam, menjadikannya sangat relevan untuk aplikasi kapal otonom. (Qiao dkk., 2023) juga menyusun ulasan menyeluruh mengenai penerapan *deep learning* pada kendaraan permukaan laut tanpa awak, mencakup berbagai tantangan serta solusi deteksi dan penghindaran rintangan berbasis visual.

2.2.3 *Deep Learning*

Deep learning merupakan salah satu teknologi dalam kecerdasan buatan yang meniru cara kerja otak manusia dalam mengenali pola melalui jaringan saraf tiruan bertingkat. Dalam tugas akhir ini, *deep learning* digunakan untuk membangun sistem deteksi rintangan pada kapal otonom berbasis Faster R-CNN dengan citra visual. Teknologi ini mampu mengenali dan mengklasifikasikan berbagai objek di laut, seperti kapal lain, pelampung, atau dermaga, secara otomatis tanpa aturan eksplisit. Keunggulannya terletak pada kemampuan mengekstraksi fitur langsung dari data gambar, sehingga sangat efektif dalam meningkatkan keselamatan navigasi kapal secara mandiri. Meskipun membutuhkan data yang besar dan perangkat komputasi yang memadai, pendekatan ini dinilai efisien untuk mendeteksi rintangan secara akurat dan *real-time* di lingkungan perairan (Liu dkk., 2020).

2.2.4 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah salah satu model *deep learning* paling populer dan efektif untuk tugas-tugas yang melibatkan pengolahan citra seperti klasifikasi gambar, deteksi objek, dan segmentasi. Proses ini meniru cara kerja otak manusia dalam mengenali pola visual secara bertingkat.

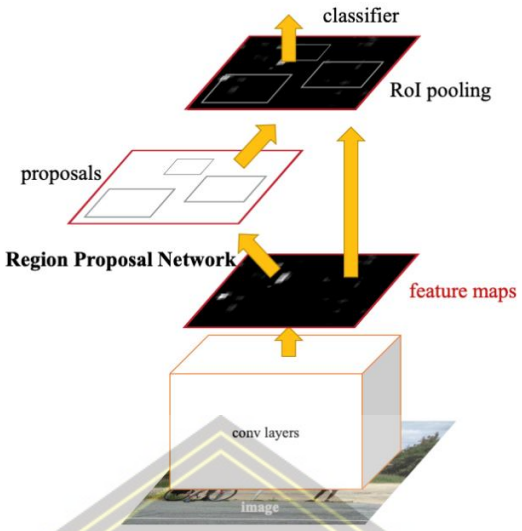


Gambar 2. 2 *Convolutional Neural Network* (CNN)

Pada gambar 2.2 menunjukkan arsitektur dasar dari *Convolutional Neural Network* (CNN) yang terdiri dari dua tahap utama, yaitu *feature extraction* dan *classification*. Pada tahap *feature extraction*, gambar input diproses melalui lapisan konvolusi untuk mengekstraksi fitur-fitur penting, kemudian dilanjutkan dengan *pooling* untuk mereduksi dimensi data. Hasil dari tahap ini diteruskan ke tahap *classification*, di mana fitur yang telah dipadatkan diolah oleh *fully connected layer* untuk menghasilkan *output* akhir berupa kelas dari citra yang diprediksi (Wang dkk., 2022).

2.2.5 Faster R-CNN

Faster R-CNN, yaitu sebuah metode *object detection* yang menggabungkan proses ekstraksi fitur, *region proposal*, dan *object classification* dalam satu kesatuan jaringan. Dengan arsitektur ini, sistem dapat mengenali dan mengklasifikasikan berbagai objek dalam gambar secara efisien dan akurat.

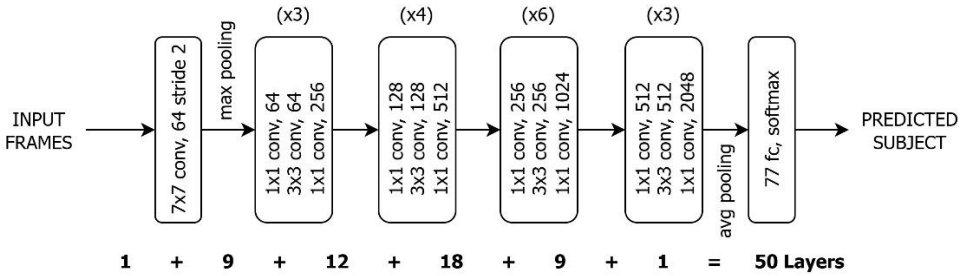


Gambar 2. 3 Arsitektur Faster R-CNN (Gad dan Skelton, 2025)

Pada gambar 2.3 menunjukan bagaimana Faster R-CNN bekerja dalam mendeteksi objek pada citra. Pertama, gambar *input* diproses melalui beberapa *convolutional layers* untuk menghasilkan *feature maps*, yaitu representasi fitur dari gambar tersebut. Selanjutnya, *Region Proposal Network (RPN)* menganalisis *feature maps* dan menghasilkan sejumlah *region proposals* yang berpotensi mengandung objek. *Proposals* ini kemudian dikirim ke *RoI Pooling*, yang menyesuaikan ukuran setiap *proposal* agar bisa diproses secara seragam. Hasil *RoI Pooling* diteruskan ke lapisan *classifier* yang bertugas mengidentifikasi jenis objek dan memperbaiki koordinat *bounding box* (Gad dan Skelton, 2025).

2.2.6 ResNet50

Salah satu komponen penting dalam arsitektur Faster R-CNN adalah bagian ekstraksi fitur. Dalam sistem ini, ResNet50 digunakan sebagai model *backbone* untuk mengekstrak fitur dari gambar input. ResNet50 dipilih karena memiliki struktur jaringan yang dalam (*deep*), namun tetap efisien dalam mengenali pola-pola penting pada gambar. Model ini membantu mengubah gambar mentah menjadi representasi fitur yang bisa dipahami oleh sistem deteksi objek.

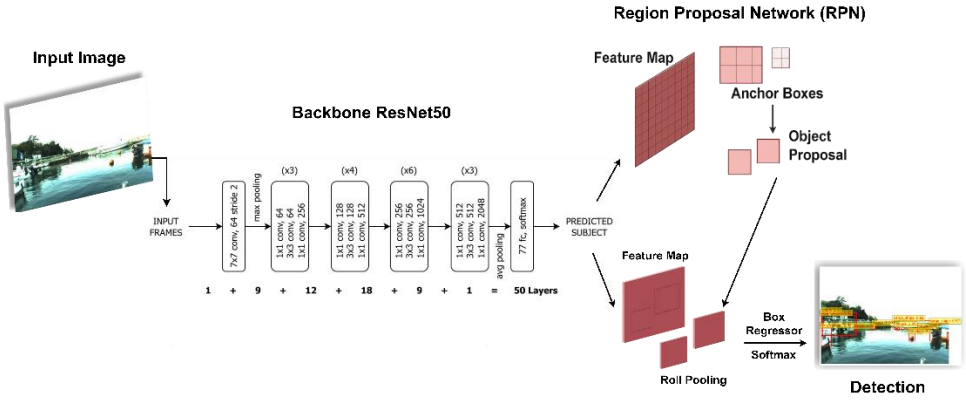


Gambar 2. 4 Arsitektur ResNet50

Pada gambar 2.4 adalah ResNet50 salah satu arsitektur jaringan konvolusional (CNN) yang populer dan digunakan sebagai *backbone* dalam berbagai model *deep learning*, termasuk Faster R-CNN. "ResNet" merupakan singkatan dari *Residual Network*, yang memiliki keunggulan utama berupa *residual blocks* komponen yang memungkinkan aliran informasi dan gradien lebih lancar dalam jaringan yang sangat dalam. ResNet50 terdiri dari 50 lapisan jaringan konvolusi, *pooling*, dan *fully connected* yang membentuk blok-blok utama. Setiap blok memiliki beberapa lapisan konvolusi dengan ukuran kernel yang berbeda, serta *shortcut connection* (tidak digambarkan dalam ilustrasi) yang membantu mengatasi masalah degradasi pada jaringan yang sangat dalam. gambar ini menunjukkan bagaimana input berupa citra akan diproses secara bertahap melalui berbagai blok untuk menghasilkan output prediksi (Renjun dkk., 2022).

2.2.7 Faster R-CNN backbone ResNet50

Salah satu komponen utama dalam sistem deteksi objek berbasis *deep learning* adalah arsitektur dari algoritma yang digunakan. Faster R-CNN merupakan salah satu algoritma deteksi objek yang banyak digunakan karena kemampuannya dalam menghasilkan deteksi dengan akurasi tinggi. Gambar berikut menggambarkan arsitektur kerja dari algoritma Faster R-CNN yang menggunakan *backbone* ResNet50 sebagai pengolah informasi visual.



Gambar 2. 5 Arsitektur Faster R-CNN dengan ResNet50

Pada gambar 2.5 merupakan algoritma *Faster R-CNN* (*Region-based Convolutional Neural Network*) salah satu metode deteksi objek berbasis *deep learning* yang mengintegrasikan proses ekstraksi fitur, seleksi wilayah objek, dan klasifikasi dalam satu arsitektur *end-to-end*. Metode ini merupakan pengembangan dari algoritma sebelumnya, yaitu R-CNN dan Fast R-CNN, dengan keunggulan utama pada efisiensi dan kecepatan deteksi. *Faster R-CNN* memperkenalkan komponen baru yang disebut *Region Proposal Network* (*RPN*), yang mampu menghasilkan *region proposals* secara otomatis dan efisien tanpa memerlukan metode eksternal seperti *selective search*.

Dalam implementasinya, algoritma ini menggunakan jaringan konvolusional seperti *ResNet50* sebagai *backbone* untuk mengekstraksi fitur penting dari citra *input*. Seperti ditunjukkan pada gambar 2.5, proses dimulai dari *input image*, yaitu gambar yang dimasukkan ke dalam sistem. Gambar ini diproses melalui jaringan *ResNet50*, yang terdiri dari beberapa *residual blocks*, untuk menghasilkan *feature map* representasi spasial yang mengandung informasi visual dari objek-objek dalam citra. *Feature map* ini kemudian diteruskan ke dua komponen utama. Pertama adalah *Region Proposal Network* (*RPN*), yang menghasilkan sejumlah *anchor boxes* dengan berbagai ukuran dan rasio aspek. RPN mengevaluasi kemungkinan keberadaan objek dalam setiap kotak dan mengeluarkan beberapa *object proposals* yang paling menjanjikan. Kedua, *feature map* beserta *region proposals* dari RPN diproses melalui tahap *RoI (Region of Interest) Pooling*,

yang bertujuan menyeragamkan ukuran proposal agar dapat diproses lebih lanjut secara efisien. Pada tahap akhir, hasil dari *RoI Pooling* dikirim ke dua cabang jaringan: *box regressor* yang bertugas menyempurnakan posisi dan ukuran *bounding box*, serta *softmax classifier* yang mengklasifikasikan objek berdasarkan kelasnya.

Hasil akhir dari proses ini adalah citra dengan sejumlah *bounding box* yang menunjukkan lokasi serta label dari objek-objek yang terdeteksi secara akurat. Dengan pendekatan terintegrasi ini, *Faster R-CNN* terbukti mampu memberikan kinerja deteksi yang tinggi, baik dari segi akurasi maupun kecepatan, sehingga banyak digunakan dalam berbagai aplikasi deteksi objek, termasuk dalam bidang navigasi maritim dan sistem otonom (Charli dkk., 2020).

2.2.8 *Region Proposal Network* (RPN)

Region Proposal Network (RPN) adalah komponen penting dalam arsitektur Faster R-CNN yang berfungsi untuk mengusulkan wilayah-wilayah dalam gambar yang kemungkinan mengandung objek. Seperti yang jelaskan pada gambar 2.4 RPN bekerja dengan cara membaca *feature map* yang dihasilkan oleh *backbone* (seperti ResNet50), lalu menghasilkan sejumlah *anchor boxes* di setiap titik lokasi. Dari *anchor* ini, RPN memprediksi apakah area tersebut mengandung objek atau tidak, serta memperkirakan posisi dan ukuran kotak yang paling sesuai (*object proposals*). RPN memungkinkan sistem untuk hanya memproses wilayah yang relevan saja, sehingga membuat proses deteksi menjadi lebih cepat dan efisien (Shangguan dan Rostami, 2023).

2.2.9 *Feature Map*

Feature map merupakan hasil dari proses ekstraksi fitur dalam jaringan *Convolutional Neural Network* (CNN), yang digunakan untuk merepresentasikan informasi penting dari gambar masukan. Pada setiap lapisan konvolusi, gambar diproses menggunakan filter yang berbeda, di mana masing-masing filter belajar mengenali pola visual spesifik seperti tepi, sudut, atau tekstur. Hasil dari proses ini adalah *feature map*, yaitu representasi

spasial yang menandakan lokasi dan intensitas fitur yang terdeteksi pada citra. Seiring bertambahnya kedalaman jaringan, *feature map* menjadi semakin kompleks dan mengandung fitur tingkat tinggi yang lebih abstrak. Dalam deteksi objek, *feature map* sangat penting karena menjadi dasar bagi tahap-tahap lanjutan seperti *region proposal* dan klasifikasi objek. Arsitektur seperti *Faster R-CNN* memanfaatkan *feature map* dari *backbone network* untuk menghasilkan proposal wilayah dan melakukan deteksi akhir, sehingga kualitas *feature map* sangat memengaruhi akurasi sistem secara keseluruhan (Chen dkk., 2022).

2.2.10 *Bounding Box*

Bounding box adalah kotak persegi panjang yang digunakan untuk menggambarkan lokasi objek yang terdeteksi dalam citra. Dalam Faster R-CNN, *bounding box* ditentukan oleh algoritma untuk menunjukkan posisi dan ukuran objek yang terdeteksi, yang memungkinkan sistem untuk mengenali objek dengan lebih baik dan menghindari tabrakan pada kapal otonom (Malviya, 2023).

2.2.11 Evaluasi Model

Setelah model selesai dilatih, langkah berikutnya adalah mengevaluasi kinerjanya. Evaluasi ini dilakukan untuk memastikan bahwa model mampu melakukan prediksi dengan baik dan sesuai dengan tujuan yang diharapkan. Pada tahap ini, digunakan beberapa metrik evaluasi utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Setiap metrik tersebut memiliki fungsi tersendiri dalam menggambarkan seberapa baik model bekerja dari berbagai aspek.

1. *Accuracy*

   Accuracy merupakan ukuran yang menunjukkan proporsi prediksi yang benar dari seluruh data yang diuji. Metrik ini termasuk yang paling dasar dan sering digunakan untuk menilai kinerja model klasifikasi secara keseluruhan, karena memberikan gambaran umum tentang seberapa akurat model dalam melakukan prediksi.

$$Accuracy = \frac{TP+TN}{\mathrm{TP}+TN+\mathrm{FP}+F\mathrm{N}} \quad (1)$$

Keterangan:

a. TP (*True Positive*): jumlah data positif yang diprediksi benar oleh model
b. TN (*True Negative*): jumlah data negatif yang diprediksi benar oleh model
c. FP (*False Positive*): jumlah data negatif yang diprediksi salah sebagai positif
d. FN (*False Negative)*: jumlah data positif yang diprediksi salah sebagai negatif

Semakin tinggi nilai *accuracy,* maka semakin baik performa model dalam mengklasifikasikan data secara keseluruhan. Namun, metrik ini menjadi kurang representatif apabila terdapat ketidakseimbangan kelas, yaitu ketika jumlah data antar kelas tidak seimbang.

2. *Precision*

Precision digunakan untuk mengukur tingkat ketepatan model dalam mengklasifikasikan data sebagai positif. Metrik ini dihitung sebagai rasio antara jumlah prediksi positif yang benar dengan total seluruh prediksi positif yang dihasilkan oleh model.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. *Recall*

Recall mengukur kemampuan model dalam mendeteksi atau mengenali seluruh data yang benar-benar termasuk dalam kelas positif. Nilai *recall* diperoleh dari perbandingan antara jumlah prediksi positif yang benar dengan total jumlah data yang sebenarnya positif.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4. *F1-Score*

F1-Score adalah metrik yang menggabungkan *Precision* dan *Recall* dalam satu nilai, dengan memperhitungkan kedua metrik tersebut secara seimbang. Metrik ini sangat berguna ketika diperlukan keseimbangan

antara *Precision* dan *Recall*, terutama pada dataset yang memiliki distribusi kelas yang tidak seimbang.

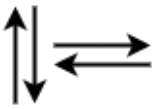
$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

F1-Score akan bernilai tinggi hanya jika *Precision* dan *Recall* sama-sama tinggi, sehingga metrik ini sangat berguna dalam mengukur kinerja model secara keseluruhan, terutama pada sistem yang menghadapi data yang tidak seimbang atau sensitif terhadap kesalahan klasifikasi.

2.2.12 *Flowchart*

Tabel 2. 1 Simbol *Flowchart* dan Fungsinya

No	Simbol	Nama	Fungsi Simbol
1		*Terminal*	Menandai awal atau akhir proses.
2		*Output/Input*	Menunjukkan masukan atau keluaran data.
3		*Process*	Menyatakan langkah pemrosesan data.
4		*Decision*	Mengindikasikan pilihan ya atau tidak.
5		*Predefined Process*	Menunjukkan subproses yang telah ditetapkan.

6		*Flow*	Menghubungkan simbol dan arah alur proses.

Pada tabel 2.1 dijelaskan simbol-simbol dasar dalam *flowchart* beserta fungsi utamanya. Simbol-simbol ini merupakan elemen penting dalam visualisasi alur proses sistem karena masing-masing memiliki peran spesifik dalam merepresentasikan aktivitas seperti awal/akhir proses, *input/output* data, pemrosesan, pengambilan keputusan, subproses, dan arah alur kerja. Penyajian dalam bentuk tabel dan gambar ini bertujuan untuk mempermudah pemahaman struktur logika sistem secara sistematis dan terstandarisasi.

BAB III
METODOLOGI PENELITIAN

3.1 Deskripsi Sistem

Sistem yang akan dibuat merupakan sistem deteksi jenis rintangan pada kapal otonom menggunakan algoritma Faster R-CNN dengan arsitektur ResNet50 sebagai *backbone*. Sistem ini dibangun menggunakan bahasa pemrograman Python dengan *framework* PyTorch sebagai dasar pengolahan data dan pelatihan model *deep learning*. Gambar yang digunakan sebagai *input* diambil langsung dari dataset *Obstacle Detection in a Marine Environment* yang mana dalam dataset tersebut berisi gambar yang diambil langsung dari kamera yang terpasang di kapal otonom, lalu diproses oleh sistem untuk mendeteksi dan mengenali objek atau rintangan yang berada di sekitar kapal. Proses deteksi ini dilakukan melalui pendekatan *object detection* yang merupakan bagian dari metode *deep learning*, di mana sistem akan menghasilkan *bounding box* pada objek yang terdeteksi, lengkap dengan klasifikasinya. Tujuan dari sistem ini adalah untuk membantu kapal otonom mengenali lingkungan sekitarnya secara otomatis, sehingga dapat meningkatkan keamanan dan navigasi dalam operasinya.

3.2 Studi Literatur

Arsitektur yang digunakan adalah Faster R-CNN dengan ResNet50 sebagai backbone, yang dikenal kuat dalam mengenali objek pada citra visual. Tujuannya adalah mendeteksi berbagai jenis rintangan di laut yang dapat mengganggu navigasi kapal otonom. Dataset yang digunakan mencakup tujuh jenis rintangan yang umum dijumpai di perairan, yaitu pelampung, kapal portabel, kapal kargo, perahu layar, pendayung, burung camar, dan dermaga. Dataset tersebut akan melalui proses pelatihan, pengujian, dan validasi menggunakan Roboflow. Dengan sistem ini, diharapkan kapal otonom dapat mengenali rintangan secara akurat dan mengambil keputusan untuk menghindari potensi tabrakan di laut.

3.3 Analisis Kebutuhan Sistem

Sebelum membangun sistem deteksi rintangan pada kapal otonom, diperlukan sejumlah perangkat lunak dan platform pendukung. Setiap komponen sistem memiliki peran penting dalam keseluruhan proses mulai dari pengolahan data, pelatihan model hingga pembuatan antarmuka pengguna. Adapun rincian kebutuhan sistem yang digunakan dapat dilihat pada tabel 3.1.

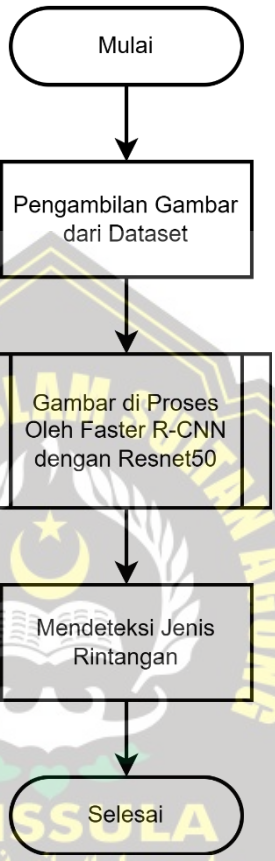
Tabel 3. 1 Kebutuhan Sistem

Kebutuhan Sistem	**Fungsi**
Python	Bahasa Pemprograman
Pytorch	*Framework* Deep Learning
Roboflow	Anotasi gambar, pembagian dataset, dan konversi ke format *COCO*
Google Colab (*Colaboratory*)	Pemanfaatkan T4 GPU untuk proses *training* model secara efisien
Streamlit	Membangun *user interface* berbasis web

Pada tabel 3.1 menjelaskan berbagai kebutuhan sistem yang digunakan dalam pembangunan sistem deteksi rintangan kapal otonom. Python digunakan sebagai bahasa pemrograman utama karena fleksibilitas dan dukungannya terhadap berbagai pustaka machine learning. Pytorch digunakan sebagai *framework* deep learning yang memudahkan implementasi dan pelatihan model. Roboflow dimanfaatkan untuk anotasi gambar, pembagian dataset, dan konversi anotasi ke format COCO yang sesuai dengan kebutuhan model deteksi objek. Google Colab digunakan karena menyediakan GPU T4 yang mempercepat proses pelatihan model secara efisien. Sementara itu, Streamlit digunakan untuk membangun antarmuka pengguna berbasis web sehingga hasil deteksi dapat divisualisasikan secara interaktif.

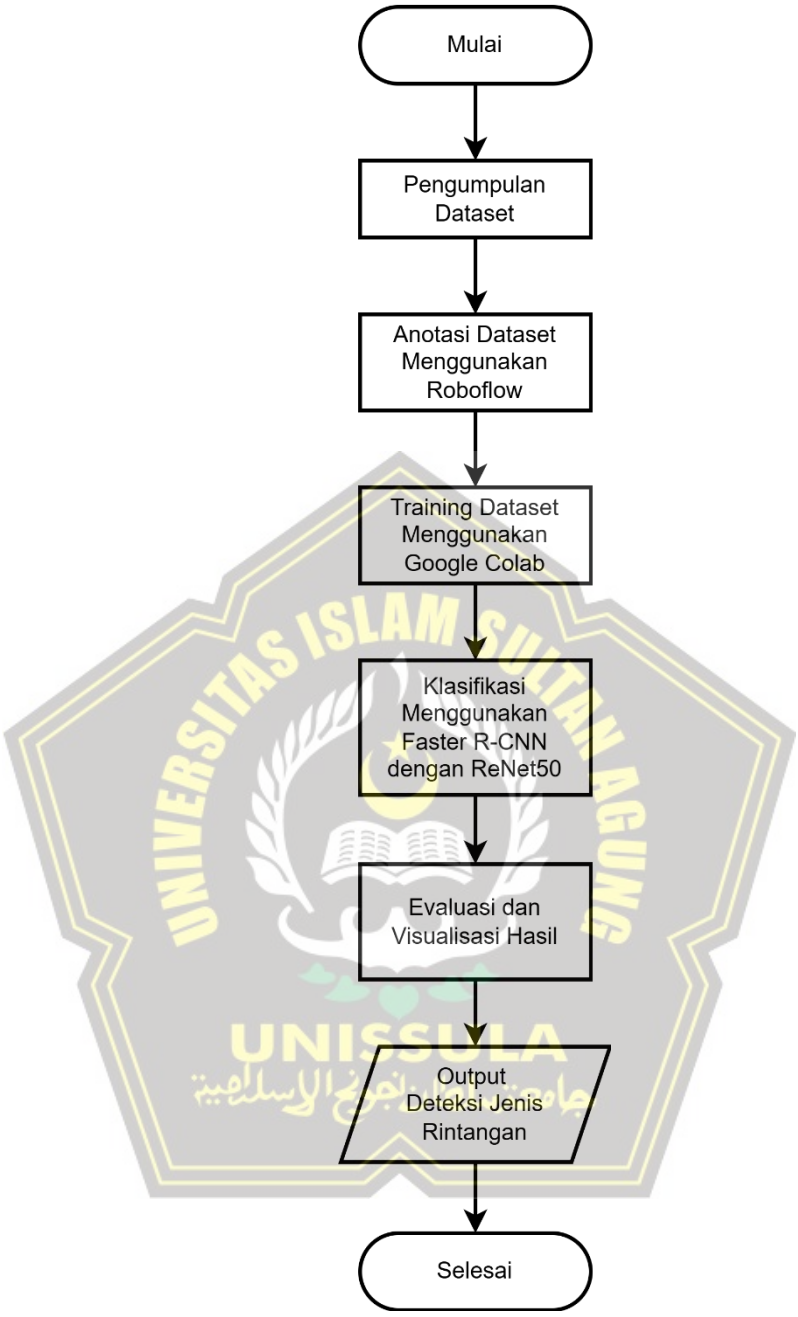
3.4 Rancang Alur Model

Pada tahap ini, dilakukan perancangan alur kerja model sistem yang akan dibuat. Untuk mempresentasikannya, alur tersebut akan digambarkan dalam sebuah *flowchart* sederhana yang ditampilkan pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Alur Kerja Sistem

Pada gambar 3.1 merupakan diagram alur kerja sistem deteksi jenis rintangan pada kapal otonom menggunakan Algoritma Faster R-CNN dengan arsitektur ResNet50. Diagram ini menjelaskan tahapan utama dalam proses pendeteksian rintangan, dimulai dari pengambilan gambar dari dataset yang tersedia. Selanjutnya, gambar-gambar tersebut diproses oleh model Faster R-CNN yang didukung oleh *backbone* ResNet50 untuk mendeteksi dan mengklasifikasikan objek yang teridentifikasi sebagai rintangan di perairan. Setelah proses klasifikasi selesai, sistem akan mengeluarkan hasil berupa jenis rintangan yang terdeteksi, yang kemudian dapat digunakan untuk mendukung navigasi kapal otonom secara lebih aman dan efisien.



Gambar 3. 2 *Flowchart* Perancangan Model Deteksi Jenis Rintangan

Pada gambar 3.2 merupakan diagram dari proses perancangan sistem deteksi jenis rintangan untuk kapal otonom menggunakan algoritma Faster R-CNN dengan *backbone* ResNet50. Proses dimulai dari pengumpulan dan anotasi dataset menggunakan Roboflow, dilanjutkan dengan pelatihan model di Google Colab. Model tersebut digunakan untuk melakukan klasifikasi objek rintangan pada citra visual laut. Hasil klasifikasi kemudian dievaluasi

dan divisualisasikan untuk mengukur performa model sebelum menghasilkan *output* berupa jenis rintangan yang terdeteksi. Diagram ini menggambarkan alur kerja sistem secara sistematis dari awal hingga akhir.

3.5 Tahapan Perancangan Model

3.5.1 Pengumpulan Data

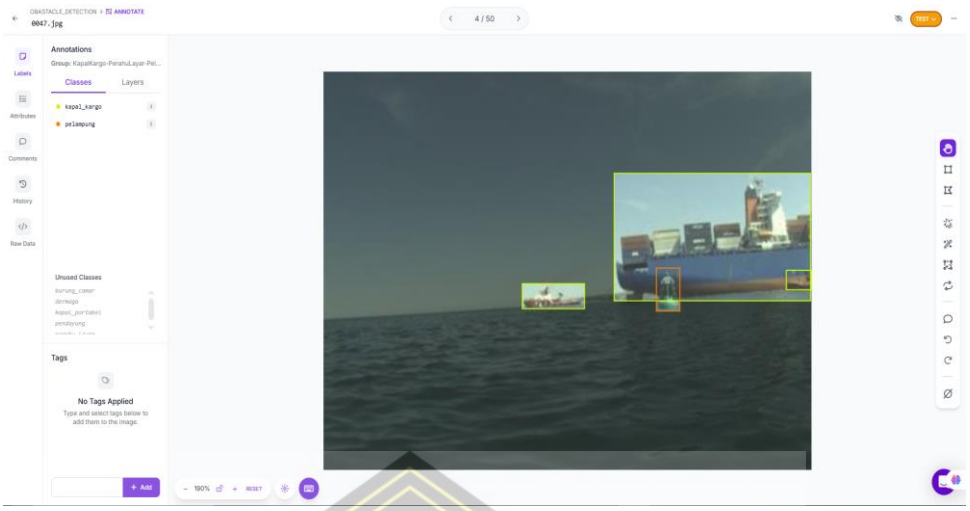
Pengumpulan data menggunakan dataset yang bersumber dari *Obstacle Detection in a Marine Environment*[1], berisi citra visual dari berbagai objek rintangan di perairan. Gambar-gambar dalam dataset tersebut kemudian dianotasi ulang menggunakan platform Roboflow untuk menyesuaikan dengan kebutuhan model deteksi berbasis *deep learning*. Proses anotasi dilakukan dengan menandai letak objek menggunakan *bounding box* serta memberikan label kelas pada masing-masing objek. Dataset hasil anotasi diklasifikasikan ke dalam tujuh kelas rintangan, yaitu: pelampung, kapal portabel, kapal kargo, perahu layar, pendayung, burung camar, dan dermaga. Hasil akhir anotasi diekspor dalam format COCO (.json) agar kompatibel dengan arsitektur model Faster R-CNN berbasis ResNet50 yang digunakan. Format COCO ini mendukung struktur data yang terorganisir dengan baik, mencakup pembagian data ke dalam folder *training*, *validation*, dan *testing* untuk proses pelatihan serta evaluasi model secara optimal. Dataset hasil anotasi ini juga dapat diakses melalui platform Roboflow[2].

3.5.2 Anotasi Dataset

Anotasi objek pada citra laut yang berisi beberapa rintangan di laut. Setiap objek yang terdeteksi diberikan *bounding box* dengan label yang sesuai berdasarkan kelasnya. Proses anotasi dilakukan secara manual menggunakan platform Roboflow, yang mempermudah dalam menandai dan mengelompokkan objek pada citra. Anotasi dilakukan oleh satu orang anotator untuk menjaga konsistensi pelabelan antar gambar. Setiap objek rintangan yang terdapat pada citra laut diberi *bounding box* dan label kelas berdasarkan jenis objek tersebut.

---

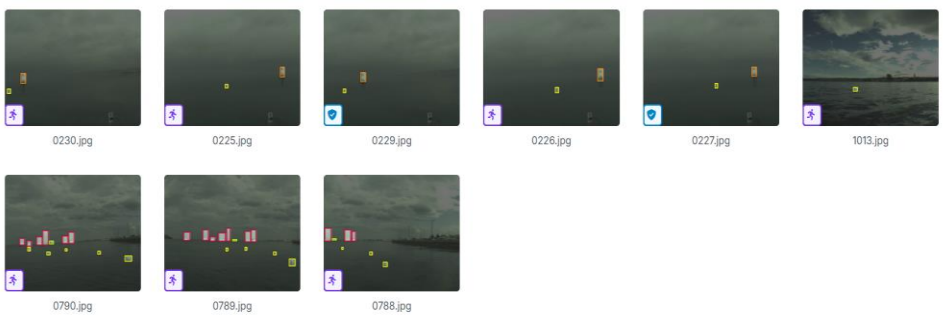
[1] https://box.vicos.si/borja/viamaro/index.html
[2] https://universe.Roboflow.com/shafarras/obastacle_detection



Gambar 3. 3 Anotasi Dataset Berdasarkan Kelas Objek Rintangan

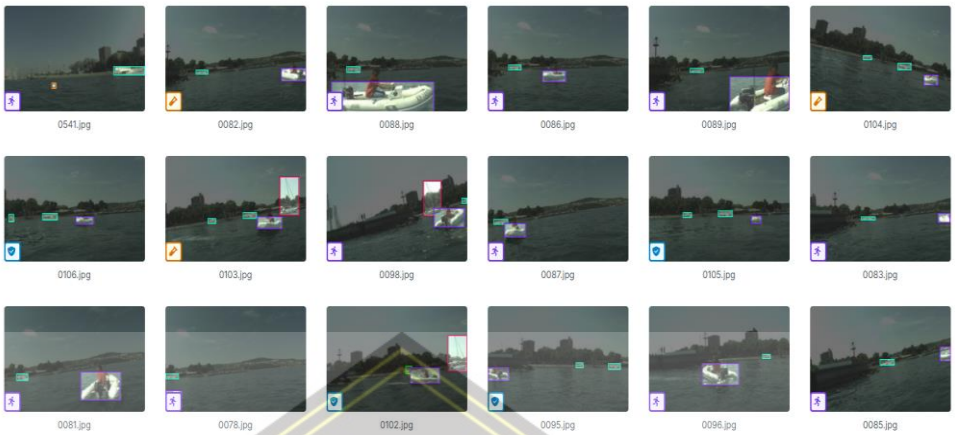
Pada gambar 3.3 adalah hasil anotasi citra yang digunakan dalam proses pelatihan model deteksi objek untuk sistem deteksi rintangan kapal otonom. Dalam citra tersebut, terdapat dua jenis objek yang telah diberi label atau anotasi menggunakan *bounding box*, yaitu kapal kargo dan pelampung. Objek kapal kargo tampak berada di sisi kanan gambar dengan ukuran yang besar dan telah diberi kotak anotasi berwarna kuning untuk menunjukkan keberadaannya secara akurat. Sementara itu, pelampung juga terdeteksi di area tengah gambar dan di dekat kapal kargo, keduanya berada pada permukaan laut dan juga diberi *bounding box*. Anotasi ini dilakukan untuk memberikan informasi visual kepada model mengenai posisi dan kelas objek yang ada dalam citra.

Citra visual yang berupa objek burung camar yang sudah di anotasi sesuai dengan kelasnya.



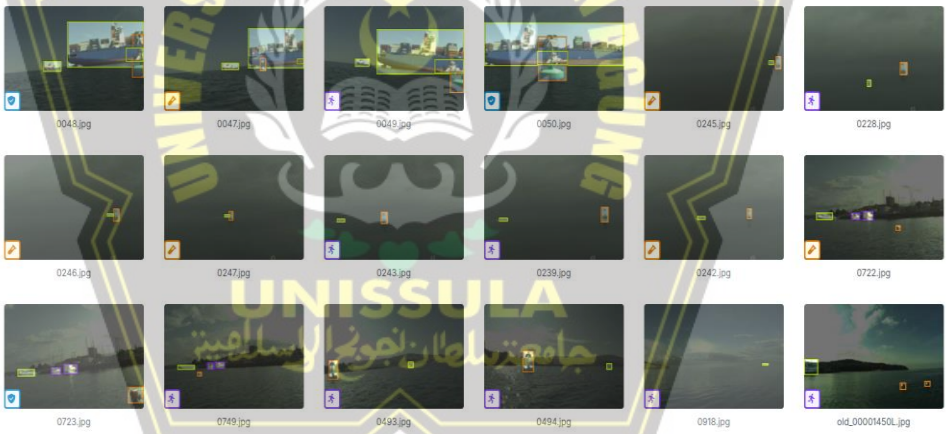
Gambar 3. 4 Dataset Burung Camar

Citra visual yang berupa objek dermaga yang sudah di anotasi sesuai dengan kelasnya.



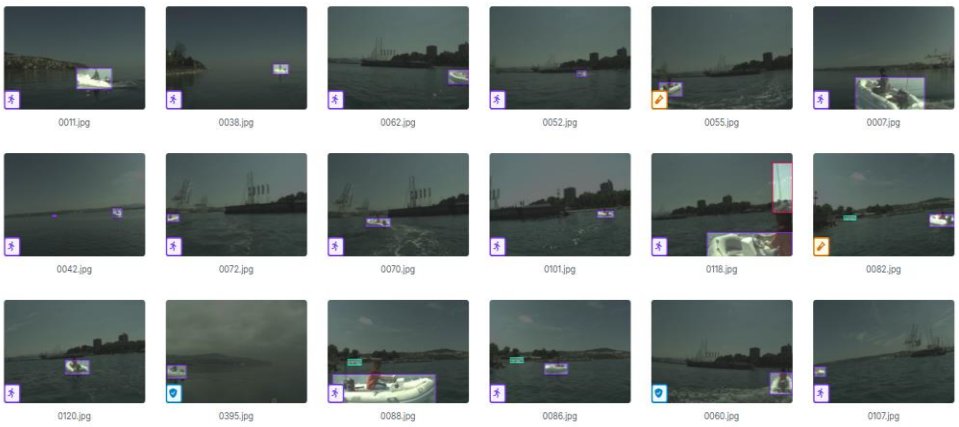
Gambar 3. 5 Dataset Dermaga

Citra visual yang berupa objek kapal kargo yang sudah di anotasi sesuai dengan kelasnya.



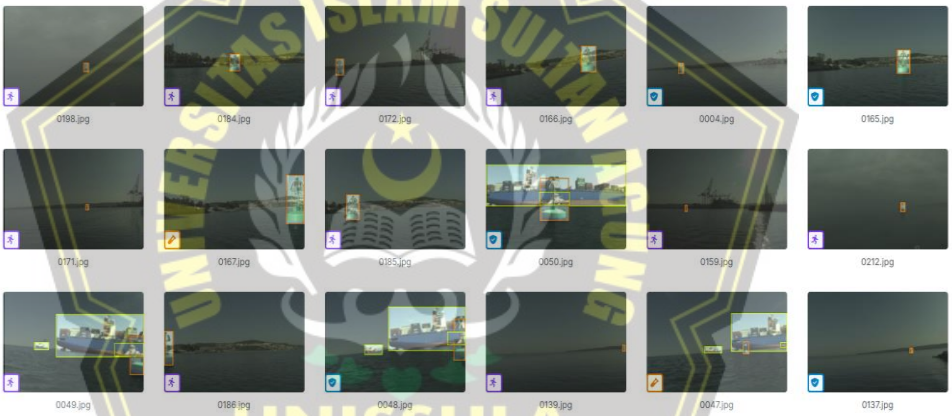
Gambar 3. 6 Dataset Kapal Kargo

Citra visual yang berupa objek kapal portabel yang sudah di anotasi sesuai dengan kelasnya.



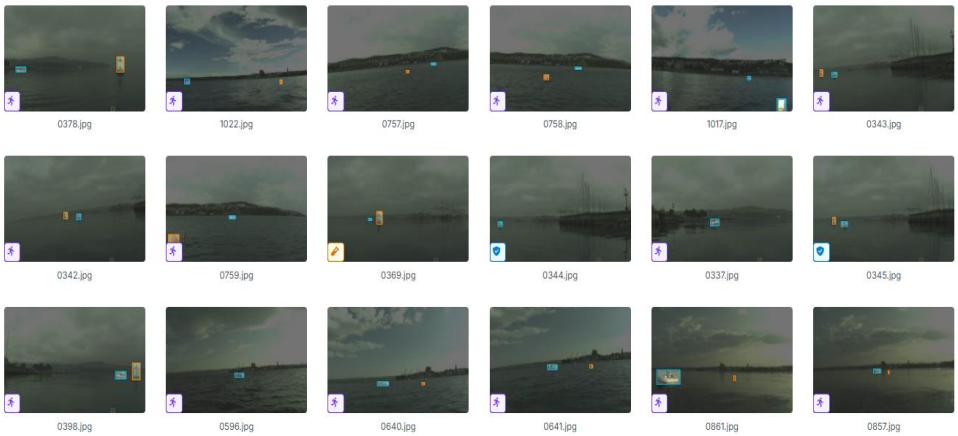
Gambar 3. 7 Dataset Kapal Portabel

Citra visual yang berupa objek pelampung yang sudah di anotasi sesuai dengan kelasnya.



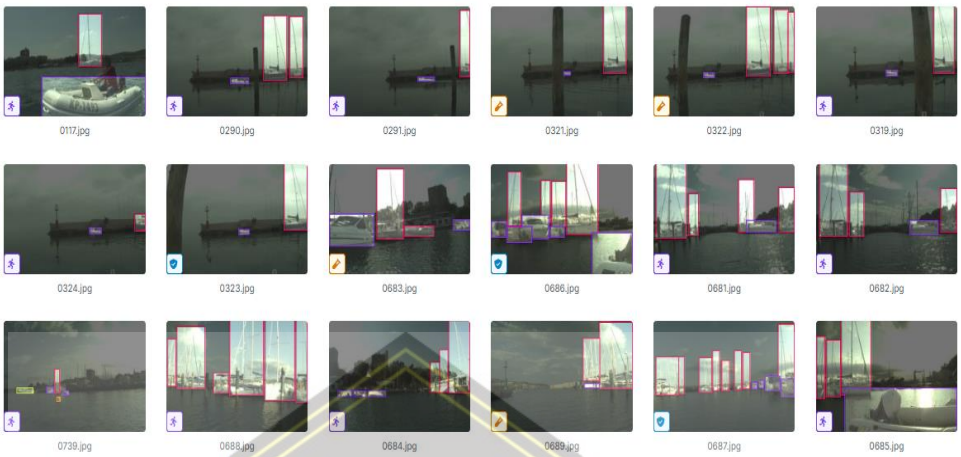
Gambar 3. 8 Dataset Pelampung

Citra visual yang berupa objek pendayung yang sudah di anotasi sesuai dengan kelasnya.



Gambar 3. 9 Dataset Pendayung

Citra visual yang berupa objek perahu layar yang sudah di anotasi sesuai dengan kelasnya.



Gambar 3. 10 Dataset Perahu Layar

Pada gambar 3.4 hingga gambar 3.10 merupakan dataset yang telah dianotasi dan diklasifikasikan ke dalam tujuh kelas objek rintangan berdasarkan jenisnya. Setiap gambar menampilkan hasil anotasi dari citra visual yang berasal dari dataset, di mana objek-objek telah diberi *bounding box* serta label kelas sesuai dengan kategorinya. Anotasi dilakukan menggunakan *platform* Roboflow untuk memastikan setiap objek dalam gambar teridentifikasi dengan benar dan dapat digunakan dalam pelatihan model deteksi rintangan berbasis Faster R-CNN dengan arsitektur ResNet50. Proses ini bertujuan untuk mengoptimalkan kemampuan model dalam mengenali berbagai jenis rintangan yang mungkin muncul di perairan saat diterapkan pada sistem kapal otonom.

3.5.3 *Preprocessing*

Preprocessing dilakukan menggunakan platform Roboflow untuk menyiapkan data sebelum tahap pelatihan model. Langkah-langkah preprocessing yang diterapkan meliputi *Auto-orient*, *Resize*, dan Augmentasi. Ketiga langkah ini dijalankan secara sistematis agar data yang digunakan dalam pelatihan model berada dalam kondisi optimal.

1) *Auto-Orient*

Penyesuaian orientasi gambar secara otomatis (*auto-orient*). Fitur ini memastikan bahwa seluruh gambar berada dalam posisi orientasi yang benar, sesuai dengan metadata gambar aslinya. Hal ini penting untuk mencegah ketidaksesuaian arah objek yang dapat memengaruhi akurasi deteksi model.

2) *Resize*

Dalam dataset awal memiliki dimensi yang bervariasi, salah satunya berukuran 512x584 *piksel*. Agar model dapat memproses data secara konsisten, seluruh gambar diubah ukurannya menjadi 640x640 *piksel*.

3) Augmentasi Data (*Saturation Adjustment*)

Saturation augmentation yaitu untuk meningkatkan kemampuan generalisasi model terhadap berbagai kondisi pencahayaan, dilakukan augmentasi gambar dengan menyesuaikan tingkat saturasi warna.

Tujuan dari augmentasi ini adalah untuk meningkatkan kemampuan generalisasi model dengan memberikan berbagai kondisi visual yang mungkin muncul di dunia nyata, seperti pencahayaan yang redup atau terlalu terang. Dengan demikian, model menjadi lebih robust dan akurat ketika menghadapi gambar dengan kondisi yang bervariasi selama proses deteksi objek.

3.5.4 *Data Splitting*

Dataset split adalah proses penting dalam tahap pengolahan data sebelum dilakukan pelatihan model. Proses ini bertujuan untuk membagi dataset menjadi tiga bagian utama, yaitu *train set*, *validation set*, dan *test set*. Masing-masing bagian memiliki fungsi tersendiri dalam memastikan bahwa model yang dibangun mampu belajar dengan baik, divalidasi secara objektif, dan diuji performanya terhadap data baru.

Tabel 3. 2 Pembagian Dataset *Training*, *Validation*, dan *Testing*

Dataset *Split*	**Jumlah Gambar**	**Presentase**
train set	598 gambar	82%
valid set	85 gambar	12%
test set	43 gambar	6%

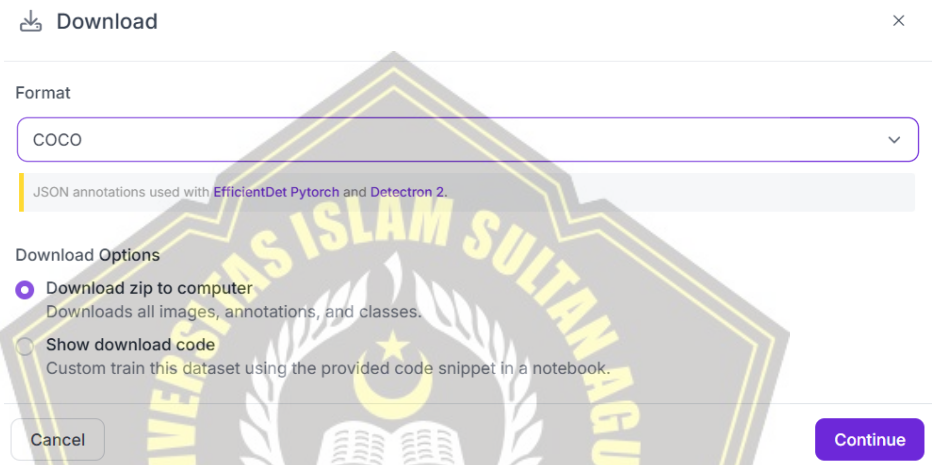
Pada Tabel 3.1 menunjukkan pembagian dataset menjadi tiga bagian, yaitu *train set* (82%), *valid set* (12%), dan *test set* (6%) dari total 726 gambar. Pembagian ini bertujuan untuk melatih, memvalidasi, dan menguji performa model secara seimbang.

Train set digunakan sebagai data utama dalam proses pelatihan model. Melalui data ini, model belajar mengenali pola atau karakteristik dari objek yang dianalisis. Karena pelatihan memerlukan banyak contoh, maka proporsi *train set* biasanya merupakan yang terbesar. Setelah itu, *validation set* digunakan untuk mengevaluasi performa model selama pelatihan. Evaluasi ini tidak memengaruhi proses pembelajaran model secara langsung, melainkan berfungsi untuk memantau apakah model mulai mengalami overfitting atau tidak. Terakhir, *test set* digunakan setelah pelatihan selesai. Data ini benar-benar baru bagi model dan digunakan untuk mengukur kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

Pembagian dataset seperti ini dilakukan agar proses pelatihan berjalan optimal, evaluasi performa menjadi objektif, dan hasil akhir dari model dapat dipercaya. Dengan proporsi yang seimbang dan sesuai kebutuhan, pembagian dataset akan sangat membantu dalam menghasilkan model yang akurat dan andal.

3.5.5 *Data Formatting*

Setelah proses anotasi selesai adalah mengunduh dataset dalam format yang sesuai dengan kebutuhan model yang akan digunakan. Pemilihan format ini penting karena setiap algoritma atau *framework* memiliki standar struktur data masing-masing. Dalam hal ini, format COCO dipilih karena kompatibel dengan berbagai model deteksi objek populer seperti EfficientDet dan Detectron2.

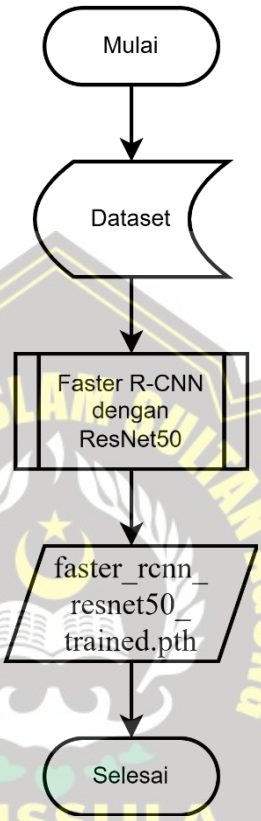


Gambar 3. 11 *Export Data Formatting*

Pada gambar 3.14 menunjukkan proses *data formatting* dengan mengekspor data ke dalam format anotasi COCO sebelum dataset diunduh, yang merupakan langkah penting setelah pembagian data. Format COCO dipilih karena kompatibel dengan *framework* PyTorch dan Detectron2. Opsi unduhan yang dipilih adalah "*Download zip to computer*", yang akan mengekspor dan mengunduh semua gambar, anotasi dalam file JSON, serta daftar kelas dalam satu paket. Langkah ini bertujuan untuk mempersiapkan dataset dalam struktur yang sesuai agar dapat langsung digunakan dalam pelatihan model deteksi objek.

3.5.6 *Training* Menggunakan *Google Colab*

Tahap Selanjutnya yaitu melakukan *training* dataset dengan menggunakan google colab. Untuk melatih dataset yang akan digunakan untuk mendeteksi jenis rintangan pada kapal otonom.



Gambar 3. 12 *Flowchart Training* Dataset

Berdasarkan gambar 3.15 yang menampilkan *flowchart training* dataset, proses pelatihan dimulai dari tahap pemuatan dataset yang telah diformat sebelumnya dalam format COCO. Dataset ini dimuat menggunakan *custom class* COCODataset yang memproses gambar dan anotasi untuk digunakan dalam pelatihan model. Selanjutnya, proses berlanjut ke tahap pemrosesan dataset menggunakan *DataLoader*, kemudian dilakukan pelatihan menggunakan model Faster R-CNN dengan *backbone* ResNet50 yang diimplementasikan dalam *framework* PyTorch. Proses pelatihan ini dijalankan di Google Colab, yang memungkinkan pemanfaatan T4 GPU untuk mempercepat komputasi dan meningkatkan efisiensi pelatihan.

Dalam pelatihan ini, terdapat beberapa parameter penting yang perlu diatur untuk memastikan model belajar secara efektif. Berikut adalah tabel dari penjelasan parameter pelatihan model Faster R-CNN yang Anda berikan:

Tabel 3. 3 Parameter pelatihan model Faster R-CNN.

No.	Parameter	Penjelasan
1	*Learning Rate* (lr=0.0001)	Mengontrol besar perubahan bobot saat update; nilai terlalu besar membuat pelatihan tidak stabil, terlalu kecil membuatnya lambat.
2	*Optimizer* (Adam)	Menggunakan Adam yang adaptif terhadap data *sparse* dan cocok untuk deteksi objek; alternatif umum lainnya adalah SGD dengan momentum.
3	Jumlah *Epoch* (*num_epochs*=50)	Menentukan berapa kali seluruh dataset digunakan dalam pelatihan; terlalu sedikit menyebabkan *underfitting*, terlalu banyak bisa *overfitting*.
4	*Batch Size* (*batch_size*=2)	Jumlah data yang diproses dalam satu iterasi; disesuaikan dengan kapasitas GPU (RAM 8 GB cocok untuk *batch* kecil).
5	Transformasi Citra	Gunakan `ToTensor()` untuk mengubah gambar ke tensor; dapat ditambah augmentasi untuk memperkaya variasi data.
6	Jumlah Kelas (*num_classes*)	Harus sesuai dengan jumlah kelas dalam dataset ditambah 1 kelas *background*.
7	*Non-Maximum Suppression* (*iou_threshold*=0.5)	Digunakan untuk menyaring deteksi ganda; menentukan ambang batas dua kotak prediksi dianggap tumpang tindih atau tidak.
8	model.parameters()	Kumpulan parameter model yang dioptimalkan oleh *optimizer*; penting untuk proses *backpropagation*.
9	*Mixed Precision* (*GradScaler* + *autocast*)	Digunakan untuk mempercepat pelatihan dan mengurangi penggunaan memori GPU.

Setelah pelatihan selesai, model disimpan ke dalam file faster_rcnn_resnet50_trained.pth. Proses diakhiri dengan evaluasi model dan visualisasi hasil dalam bentuk *confusion matrix* serta metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*, untuk memastikan kemampuan model dalam mendeteksi rintangan pada kapal otonom di kondisi nyata.

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil dan Analisis

4.1.1 Pengumpulan dan Pengolahan Data

1. Pengumpulan Data

Data yang digunakan berasal dari dataset *Obstacle Detection in a Marine Environment*, yang dirancang untuk mendukung pengembangan sistem deteksi rintangan di lingkungan laut, khususnya bagi kapal otonom. Dataset ini diperoleh dari lebih dari lima puluh jam rekaman video yang mendokumentasikan kondisi nyata di perairan. Dari rekaman tersebut, gambar-gambar dipilih secara manual dengan mempertimbangkan keragaman kondisi visual untuk memastikan data yang digunakan bersifat representatif. Proses seleksi ini menghasilkan sebanyak 1.325 gambar sebagai kumpulan data awal.

Selanjutnya, dilakukan proses anotasi ulang terhadap 726 gambar terpilih menggunakan *platform* Roboflow. Anotasi dilakukan secara manual dengan menandai objek-objek yang dianggap sebagai rintangan dalam navigasi laut. Setiap objek diberikan penandaan menggunakan *bounding box* serta label kelas sesuai kategorinya. Dataset hasil anotasi diklasifikasikan ke dalam tujuh kelas rintangan, yaitu:

1) Kapal portabel
2) Kapal kargo
3) Perahu layar
4) Pendayung
5) Burung camar
6) Dermaga

Hasil anotasi kemudian diekspor dalam format COCO (.json) yang sesuai untuk digunakan pada model deteksi objek berbasis deep learning seperti Faster R-CNN.

2. Pengolahan Data

Setelah proses anotasi selesai, langkah berikutnya adalah pengolahan data, yaitu menyiapkan dataset agar siap digunakan dalam pelatihan dan evaluasi model.

Data yang telah dianotasi dan diekspor dalam format COCO kemudian dibagi ke dalam tiga subset utama:

1) Data pelatihan (*training set*): 598 gambar
2) Data pengujian (*testing set*): 85 gambar
3) Data validasi (*validation set*): 43 gambar

Pembagian ini dilakukan secara proporsional untuk memastikan bahwa model memiliki cukup data untuk belajar, divalidasi selama proses pelatihan, serta diuji terhadap data yang belum pernah dilihat sebelumnya. Pembagian ini juga bertujuan untuk mengukur kemampuan model dalam melakukan generalisasi terhadap data baru di lingkungan laut yang kompleks. Pengolahan data ini merupakan tahap penting dalam *pipeline* sistem deteksi objek, karena memastikan bahwa data berada dalam format dan struktur yang sesuai dengan kebutuhan model, serta mendukung evaluasi performa model secara objektif.

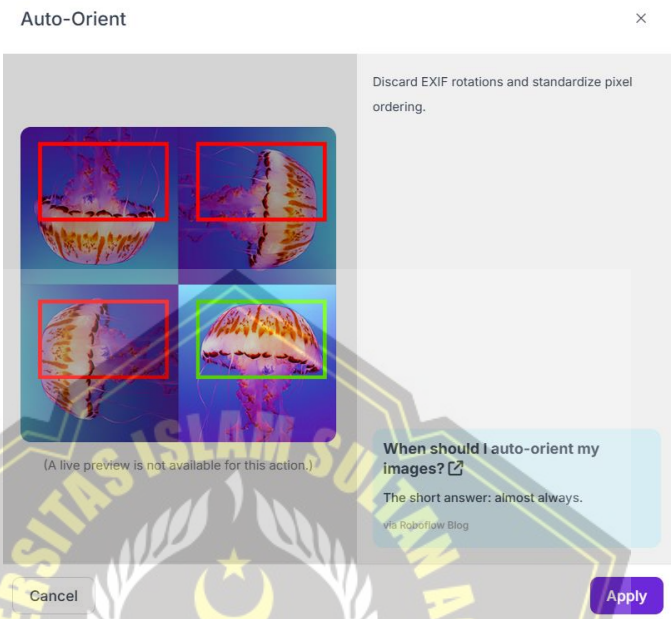
4.1.2 *Preprocessing Data*

Sebelum data digunakan dalam proses pelatihan model deteksi objek, dilakukan tahap *preprocessing* untuk memastikan bahwa data memiliki format dan kualitas visual yang konsisten. Proses ini dilakukan menggunakan platform Roboflow, yang menyediakan berbagai fitur untuk penyesuaian dan peningkatan kualitas gambar. Terdapat tiga langkah utama yang dilakukan dalam tahap *preprocessing*, yaitu:

1. *Auto-orient*

Auto-Orient merupakan salah satu fitur penting dalam tahap praproses data citra. Fitur ini berfungsi untuk menghilangkan informasi rotasi dari metadata EXIF serta menstandarkan urutan piksel pada setiap gambar. Dengan menerapkan auto-orientasi, semua citra yang digunakan dalam proses pelabelan dan pelatihan model akan memiliki orientasi

yang seragam. Hal ini bertujuan untuk menghindari kesalahan dalam deteksi objek akibat perbedaan sudut pandang atau rotasi gambar yang tidak terdeteksi secara eksplisit oleh model.



Gambar 4. 1 *Auto-Orient*

Pada gambar 4.1 menunjukkan proses *auto-orient* pada citra menggunakan platform Roboflow. Kotak merah memperlihatkan citra-citra dengan orientasi yang berbeda-beda akibat metadata EXIF saat pengambilan gambar. Sementara itu, kotak hijau menunjukkan hasil setelah dilakukan proses *auto-orient*, di mana citra telah diselaraskan agar tampil secara konsisten dan sesuai dengan posisi objek yang sebenarnya.

Dataset citra diperoleh dari kamera yang dipasang langsung pada kapal otonom. Karena kapal beroperasi di lingkungan laut yang dinamis, posisi kamera sering mengalami perubahan orientasi akibat gerakan kapal, ombak, angin, atau kondisi cuaca lainnya. Kondisi ini menyebabkan citra yang dihasilkan memiliki orientasi yang tidak seragam, seperti miring, terbalik, atau menyamping.

Untuk mengatasi hal tersebut, dilakukan proses *auto-orient* pada seluruh citra sebagai bagian dari tahap praproses (*preprocessing*). Fitur

auto-orient secara otomatis membaca metadata EXIF dari gambar dan menyesuaikan arah gambar agar tampil dengan orientasi yang benar. Proses ini juga menyusun ulang piksel agar konsisten untuk seluruh dataset. Langkah ini sangat penting untuk memastikan setiap gambar memiliki orientasi yang seragam, sehingga model tidak kebingungan dalam mengenali objek yang sama dari arah berbeda. Dengan demikian, proses anotasi dan pelatihan model menjadi lebih akurat dan efisien.

2. *Resize*

Resize digunakan untuk mengubah ukuran gambar agar sesuai dengan input model dan mempercepat proses pelatihan. Ukuran yang umum digunakan adalah 640×640 piksel.

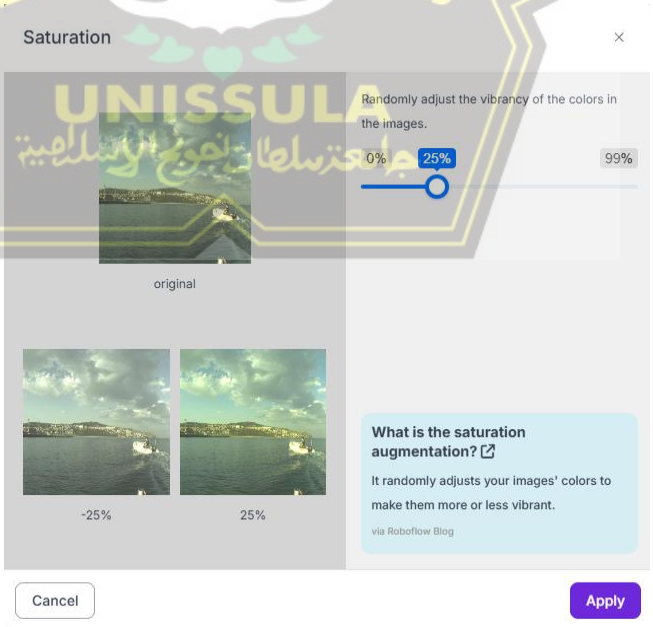
Gambar 4. 2 *Resize*

Pada gambar 4.2 menunjukkan tampilan fitur *Resize* di platform Roboflow, yang digunakan untuk mengubah ukuran semua gambar dalam dataset menjadi 640 x 640 piksel. Ukuran ini dipilih karena merupakan dimensi standar yang banyak digunakan dalam model deteksi objek, serta memberikan keseimbangan antara detail visual dan efisiensi komputasi. Penyesuaian ini dilakukan tanpa mengubah isi atau label objek dalam gambar, sehingga informasi penting tetap terjaga untuk

pelatihan model. Pada proses praproses data, salah satu langkah penting yang dilakukan adalah penyesuaian ukuran (*resize*) gambar. Selanjutnya, fitur *Resize* pada platform Roboflow digunakan untuk menyeragamkan ukuran seluruh citra dalam dataset menjadi 640 x 640 piksel. Penyeragaman dimensi ini sangat penting karena model *deep learning*, seperti Faster R-CNN atau YOLO, umumnya membutuhkan format *input* citra yang tetap dan konsisten agar dapat memproses data secara optimal. Jika ukuran gambar berbeda-beda, maka proses pelatihan bisa menjadi tidak efisien, dan model akan kesulitan mengenali pola atau objek dengan konsisten. Dengan ukuran gambar yang seragam, proses ekstraksi fitur dan pembelajaran model menjadi lebih stabil dan cepat, serta dapat meningkatkan akurasi deteksi pada data validasi maupun data nyata di lapangan.

3. Augmentasi

   Augmentasi saturasi digunakan untuk menyesuaikan tingkat kecerahan warna dalam gambar secara acak, guna meningkatkan variasi data pelatihan.



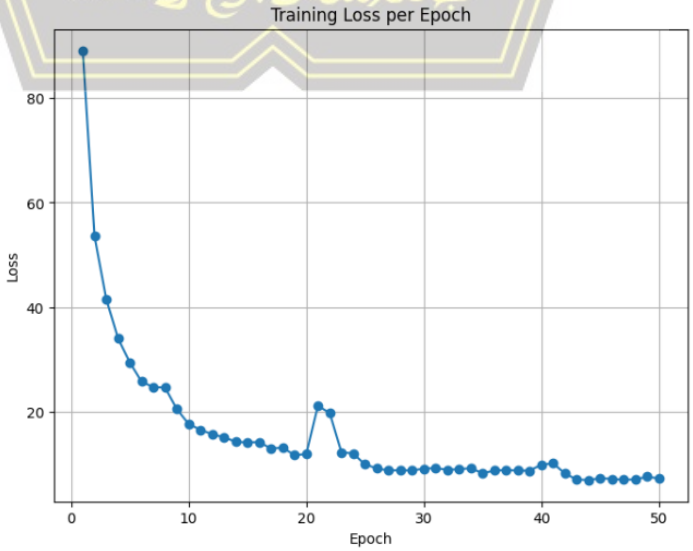
Gambar 4. 3 *Saturation Augmentation*

Pada gambar 4.3 menunjukkan salah satu metode augmentasi data yang digunakan yaitu *saturation augmentation* atau augmentasi saturasi. Teknik ini digunakan untuk mengubah tingkat kecerahan warna pada citra, sehingga menghasilkan variasi data visual tanpa mengubah struktur atau label objek di dalam citra. Dalam *platform* Roboflow, proses ini dapat dilakukan secara otomatis dengan mengatur persentase perubahan saturasi, seperti pada gambar ini yang memperlihatkan perubahan saturasi sebesar -25% (mengurangi kecerahan warna) dan +25% (meningkatkan kecerahan warna).

Seluruh tahapan *preprocessing* ini bertujuan untuk meningkatkan kualitas data masukan dan memastikan bahwa model dapat belajar dari dataset yang seragam, informatif, dan relevan dengan kondisi nyata di lingkungan laut.

4.1.3 *Optimizer* dan Parameter *Training*

Pada tahap pelatihan model, digunakan optimizer Adam dengan learning rate sebesar 0.0001 untuk mengoptimalkan parameter-parameter model. Selain itu, penerapan teknik mixed precision training dilakukan dengan memanfaatkan GradScaler dari CUDA, yang bertujuan untuk meningkatkan efisiensi dan kecepatan pelatihan tanpa mengorbankan akurasi. Proses pelatihan dilakukan selama 50 epoch, dan setiap epoch dicatat nilai loss-nya dalam sebuah daftar untuk keperluan evaluasi dan analisis performa model.



Gambar 4. 4 *Training Loss* per *Epoch*

Pada gambar 4.4 memperlihatkan perkembangan nilai *training loss* model Faster R-CNN dengan *backbone* ResNet50 selama 50 *epoch*.

Tabel 4. 1 Hasil *Training 50 Epoch*

Epoch	Total Loss
5	29.4
10	17.7
15	14.1
20	11.9
25	9.9
30	9.2
35	8.3
40	9.9
45	7.3
50	7.3

Pada tabel 4.1 menunjukkan hasil pelatihan model Faster R-CNN dengan backbone ResNet-50, di mana total *training loss* mengalami penurunan signifikan dari *epoch* 5 hingga *epoch* 15, yaitu dari 29,4 menjadi 14.1 Setelah itu, penurunan *loss* berlangsung lebih lambat dan mulai stabil di bawah nilai 10 mulai dari *epoch* 25. Pada *epoch* 50, total *loss* mencapai nilai 7,3 yang mengindikasikan bahwa model mulai mendekati konvergensi. Dengan penurunan *loss* yang konsisten, pelatihan selama 50 *epoch* sudah cukup baik, namun evaluasi lebih lanjut menggunakan data pengujian diperlukan untuk memastikan performa model di kondisi nyata.

4.2 Hasil Evaluasi Model

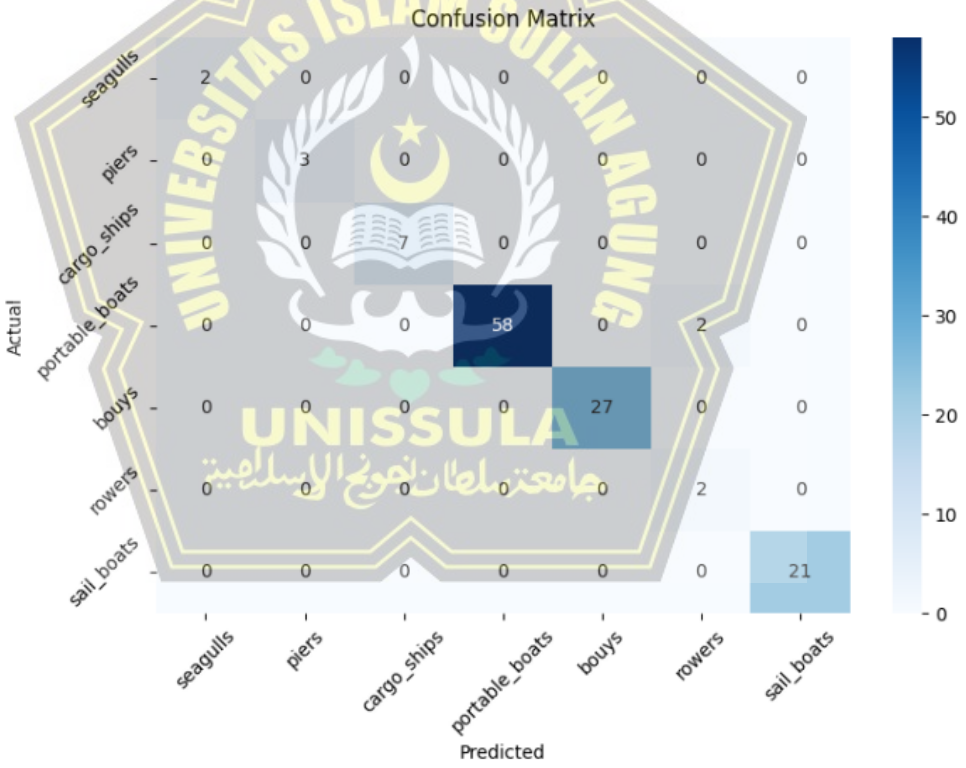
Tabel 4. 2 Metrik Evaluasi Model

Metrix Evaluasi	Skor(%)
Accuracy	96.00 %
Precision	99.20 %
Recall	96.00 %
F1-Score	97.41 %

Tabel 4.2 menunjukkan metrik evaluasi kinerja model Faster R-CNN (ResNet-50) pada data uji, dengan *accuracy* sebesar 96,00%, yang menunjukkan persentase prediksi benar secara keseluruhan. *Precision* 99,20% mengindikasikan bahwa sebagian besar prediksi rintangan model akurat, sementara *recall* 96,00% menunjukkan model berhasil mendeteksi sebagian besar rintangan yang ada. *F1-score* 97,41% mencerminkan keseimbangan yang baik antara *precision* dan *recall*, menandakan performa deteksi yang baik dengan sedikit kesalahan dalam identifikasi rintangan.

4.2.1 *Confusion Matrix*

Berikut adalah *confusion matrix* yang menunjukkan kinerja model dalam mengklasifikasikan berbagai jenis objek selama proses pengujian.



Gambar 4. 5 *Confusion Matrix*

Pada gambar 4.5 menunjukkan *confusion matrix* kinerja model Faster R-CNN dalam mendeteksi tujuh kelas objek pada dataset maritim, yaitu burung camar (*seagulls*), dermaga (*piers*), kapal kargo (*cargo_ships*), kapal portabel (*portable_boats*), pelampung (*buoys*), pendayung (*rowers*), dan perahu layar

(*sail_boats*). Model menunjukkan performa sangat baik pada beberapa kelas, seperti:

1. Kapal Portabel (*portable_boats*): 58 prediksi benar.
2. Pelampung (*buoys*): 27 prediksi benar.
3. Perahu Layar (*sail_boats*): 21 prediksi benar.

Model juga mampu mengenali kelas kapal kargo (*cargo_ships*) 7 benar prediksi benar, dermaga (*piers*) 3 prediksi benar, burung camar (*seagulls*) 2 prediksi benar, dan pendayung (*rowers*) 2 prediksi benar, meskipun dengan jumlah data yang terbatas. Terdapat sedikit kesalahan, seperti:

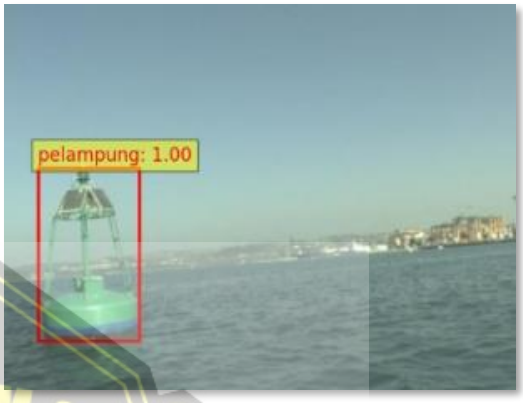
1. 2 pelampung (*buoys*) salah diklasifikasikan sebagai kapal portabel (*portable_boats*).
2. 2 pendayung (*rowers*) salah diklasifikasikan sebagai pelampung (*buoys*).

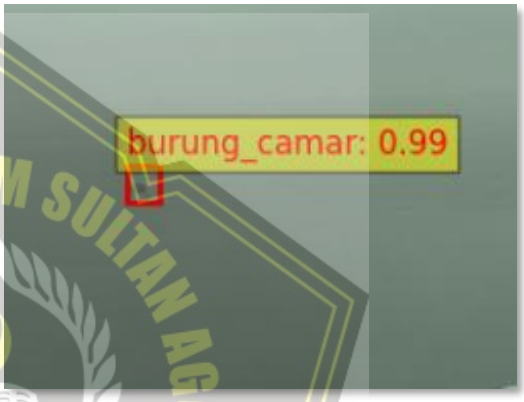
Kesalahan ini kemungkinan disebabkan oleh kemiripan visual antar objek atau jumlah data pelatihan yang tidak seimbang pada beberapa kelas. Secara keseluruhan, model menunjukkan performa yang cukup andal dalam mengenali objek-objek maritim, namun evaluasi lebih lanjut dan penyeimbangan data masih diperlukan untuk meningkatkan akurasi deteksi.

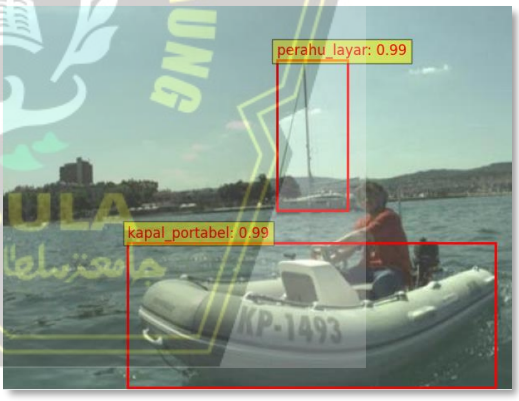
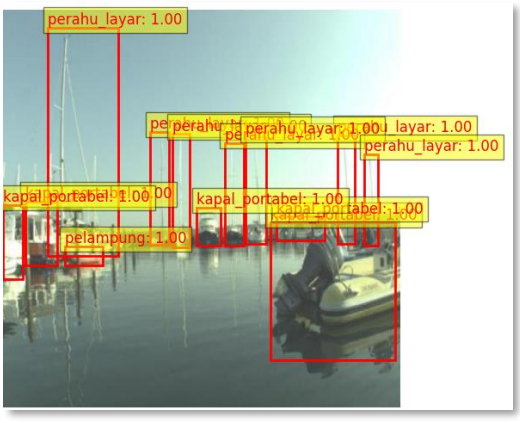
4.2.2 Hasil Deteksi Rintangan

Hasil Deteksi Rintangan pada Kapal Otonom saat menjalankan hasil deteksi gambar baru Menggunakan Model Faster R-CNN. Tabel 4.3 menunjukkan hasil deteksi rintangan yang berhasil dilakukan oleh sistem deteksi berbasis algoritma Faster R-CNN pada citra visual laut.

Tabel 4. 3 Hasil Deteksi Jenis Rintangan pada Kapal Otonom

Jenis Rintangan	Hasil	Objek Terdeteksi
Pelampung	100%	
Perahu Layar & Kapal Portabel	100%	
Kapal Kargo	87%	

Pendayung	99%	
Burung Camar	99%	
Dermaga	97%	

Kapal Kargo & Pelampung	100%	
Perahu Layar & Burung Camar	90%-100%	
Perahu Layar & Kapal Portabel	99%	
Perahu layar, Kapal Portabel, Pelampung	100%	

Tabel 4.3 merupakan hasil deteksi rintangan di lingkungan maritim menggunakan model Faster R-CNN dengan *backbone* ResNet50. Model ini berhasil mengidentifikasi berbagai jenis rintangan, seperti pelampung, burung camar, perahu layar, kapal kargo, pendayung, kapal portabel, dan dermaga, yang semuanya berpotensi memengaruhi jalur navigasi kapal otonom. Setiap objek yang terdeteksi ditandai dengan *bounding box* merah dan diberi label sesuai jenisnya, beserta skor keyakinan yang sesuai. Hasil ini, seperti yang ditunjukkan pada Tabel 4.3, menunjukkan kemampuan model untuk mengenali objek dengan tingkat akurasi yang tinggi, meskipun ada tantangan yang ditimbulkan oleh variasi pencahayaan dan kondisi lingkungan.

4.3 *Deployment*

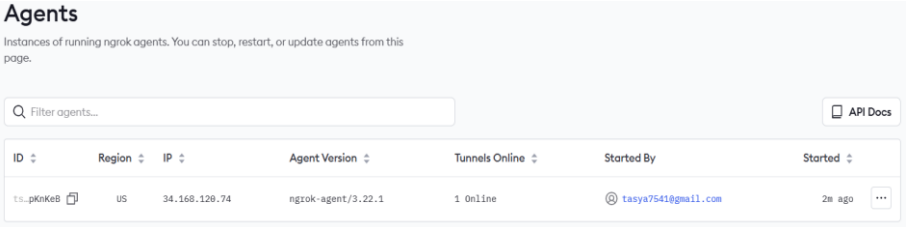
Deployment merupakan proses menjalankan model yang telah dilatih ke dalam sebuah aplikasi agar dapat digunakan secara langsung. Tujuan dari tahap ini adalah untuk memungkinkan model melakukan prediksi secara *real-time* dan dapat diakses oleh pengguna. Dalam implementasinya, model Faster R-CNN yang telah dilatih untuk mendeteksi objek maritim diterapkan ke dalam bentuk aplikasi agar dapat dimanfaatkan secara praktis dalam mendukung sistem navigasi kapal otonom.

4.3.1 *Streamlit*

Streamlit merupakan *framework* berbasis Python yang digunakan untuk membangun antarmuka pengguna (*user interface*) berbasis web secara interaktif. Melalui *Streamlit*, hasil deteksi rintangan pada citra laut dapat divisualisasikan dalam bentuk gambar yang telah diberi *bounding box* dan label kelas sesuai prediksi model. Penggunaan Streamlit memungkinkan pengujian sistem secara langsung melalui *browser* tanpa memerlukan keahlian pengembangan web. Untuk mempermudah akses aplikasi dari jaringan luar, digunakan ngrok sebagai solusi *tunneling* yang menghubungkan aplikasi lokal ke internet.

1. *Ngrok Agents*

Agents pada *dashboard* Ngrok yang berfungsi untuk memantau dan mengelola instansi agen Ngrok yang sedang berjalan.

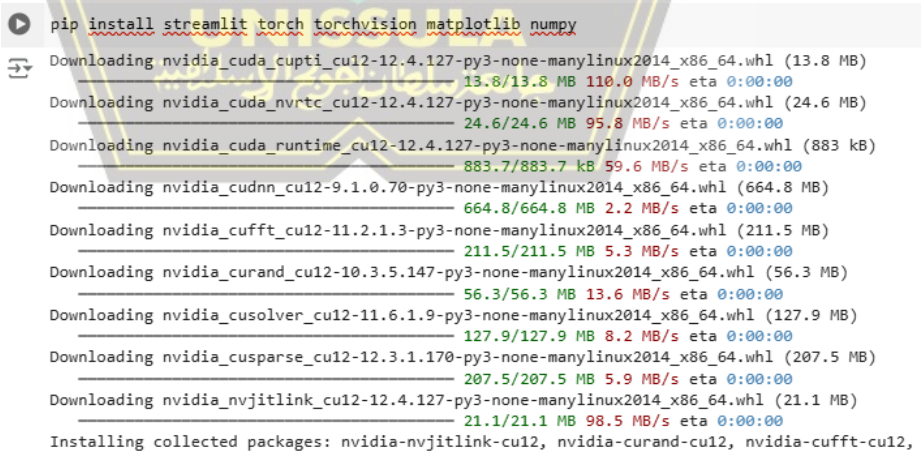


Gambar 4. 6 *Ngrok Agents*

Pada gambar 4.6 menampilkan alamat IP publik 34.168.120.74. Agen ini menggunakan versi ngrok-agent/3.22.1 dan saat ini memiliki satu tunnel yang aktif atau *online*. Agen tersebut dijalankan oleh pengguna dengan email. Informasi ini berguna bagi pengguna untuk mengetahui status koneksi, lokasi server, serta aktivitas agen secara *real-time* melalui *dashboard* Ngrok.

2. *Installation library streamlit*

Gambar berikut menunjukkan proses instalasi beberapa pustaka Python yang dibutuhkan untuk menjalankan sistem deteksi rintangan pada kapal otonom.



Gambar 4. 7 Instalasi Paket *Streamlit* dan Ngrok

Pada gambar 4.7 menunjukkan proses instalasi paket *Streamlit* dan Ngrok menggunakan perintah pip install. Instalasi ini diperlukan agar

aplikasi web berbasis *Streamlit* bisa dijalankan dan diakses secara *online* melalui Ngrok.

3. *Link streamlit*

Gambar berikut menampilkan hasil *output* dari proses menjalankan aplikasi *Streamlit* yang telah diintegrasikan dengan ngrok.

Gambar 4. 8 *Output* Konfirmasi *Streamlit* dan Alamat Akses Ngrok

Pada gambar 4.8 merupakan alamat sementara dari aplikasi berbasis *Streamlit* yang dijalankan secara lokal dan dibagikan ke publik menggunakan layanan Ngrok. *Streamlit* digunakan sebagai *platform* untuk membangun aplikasi web interaktif berbasis Python, sedangkan Ngrok berfungsi sebagai alat tunneling yang memungkinkan server lokal diakses secara *online* melalui jaringan internet. Dengan menggunakan Ngrok, aplikasi yang berjalan di komputer lokal dapat diakses oleh pihak lain tanpa perlu melakukan *deploy* ke *server hosting*.

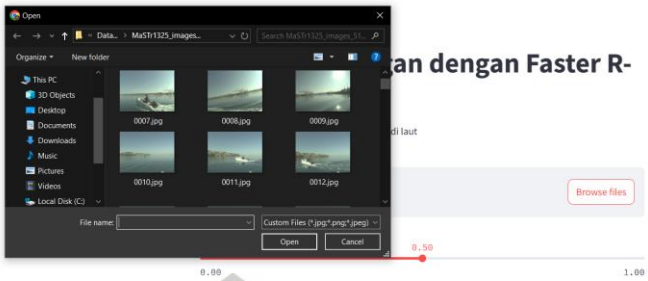
4. *Dashboard streamlit*

Dashboard dari sistem deteksi rintangan di laut dengan Faster R-CNN.

Gambar 4. 9 Tampilan *Dasboard Streamlit*

Pada gambar 4.9 pengguna dapat mengunggah gambar melalui fitur *drag and drop* atau tombol *"Browse files"* dan mengatur nilai *threshold* deteksi menggunakan *slider* untuk mengontrol sensitivitas deteksi objek.

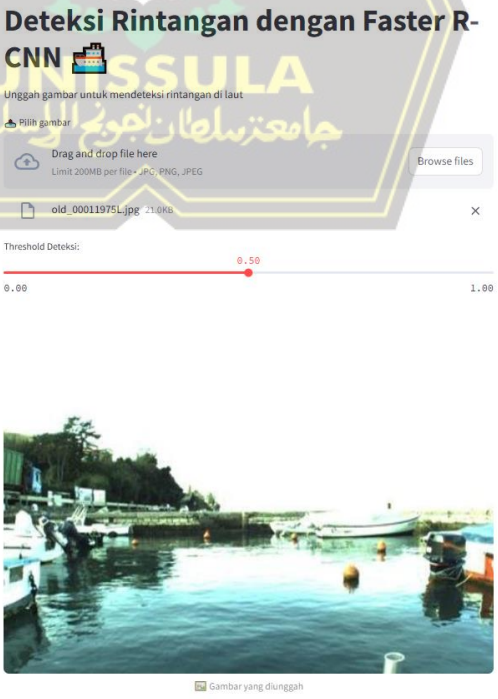
Gambar di bawah ini adalah tampilan saat pengguna mengunggah gambar ke dalam sistem deteksi rintangan di laut dengan menggunakan algoritma Faster R-CNN.



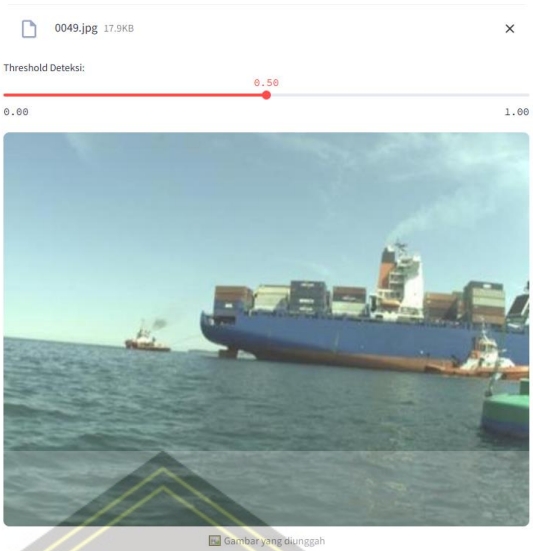
Gambar 4. 10 Tampilan Mengunggah Gambar pada Sistem

Pada gambar 4.10 menunjukkan tampilan saat jendela file *explorer* muncul untuk memilih file gambar dengan format JPG, PNG, atau JPEG dari penyimpanan lokal.

Tampilan antarmuka yang dibangun menggunakan *Streamlit* untuk sistem deteksi rintangan di laut berbasis algoritma Faster R-CNN.gambar 4.11 dan 4.12 adalah tampilan saat berhasil menggunggah gambar yang akan di deteksi.



Gambar 4. 11 Tampilan Berhasil Mengunggah Gambar 1



Gambar 4. 12 Tampilan Berhasil Mengunggah Gambar 2

Pada gambar 4.11 dan gambar 4.12 yang berhasil diunggah terdapat *slider* untuk mengatur *threshold* deteksi yang digunakan untuk memfilter hasil deteksi.

Pada gambar 4.13 menampilkan sistem berhasil mendeteksi gambar 1 objek deteksi yang ada di permukaan laut.

Gambar 4. 13 Tampilan Sistem Berhasil Mendeteksi Gambar 1

Pada gambar 4.14 menampilkan sistem berhasil mendeteksi gambar 2 objek deteksi yang ada di permukaan laut.

Gambar 4. 14 Tampilan Sistem Berhasil Mendeteksi Gambar 2

Pada gambar 4.13 dan gambar 4.14 sistem menampilkan pratinjau gambar dengan *bounding box* berwarna merah yang menandai objek-objek yang terdeteksi seperti kapal portabel dan pelampung. Setiap *bounding box* dilengkapi label nama objek dan skor probabilitas yang menunjukkan tingkat keyakinan model terhadap deteksi tersebut. Gambar ini memperlihatkan hasil deteksi sistem yang berhasil menampilkan objek secara akurat sesuai *threshold* yang telah diatur sebelumnya, menandakan bahwa proses deteksi berjalan dengan baik.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penerapan algoritma Faster R-CNN dengan arsitektur ResNet-50 telah berhasil membangun sistem yang mampu mendeteksi objek-objek rintangan di permukaan perairan, seperti kapal lain, pelampung, dermaga, dan berbagai hambatan visual lainnya. Sistem ini dirancang untuk mendukung pergerakan kapal otonom secara cerdas di lingkungan laut yang kompleks dan dinamis.

Kinerja model menunjukkan hasil yang sangat baik dengan tingkat akurasi mencapai 96,00%, precision 99,20%, recall 96,00%, dan F1-score 97,41%. Nilai-nilai ini mencerminkan ketepatan dan konsistensi dalam mengenali serta mengklasifikasikan berbagai jenis rintangan di laut. Selain itu, grafik penurunan *training loss* yang stabil mengindikasikan proses pembelajaran berlangsung optimal tanpa kecenderungan overfitting.

Pencapaian ini memperlihatkan bahwa Faster R-CNN memiliki kapabilitas tinggi dalam melakukan deteksi objek dengan akurasi yang andal. Meski demikian, pengembangan lanjutan tetap diperlukan, terutama dalam menghadapi kondisi cuaca ekstrem serta integrasi dengan sistem penghindaran rintangan secara real-time untuk meningkatkan efektivitas dan daya tahan sistem navigasi kapal otonom.

.

5.2 Saran

Berdasarkan keterbatasan yang ditemui selama proses pengembangan, berikut beberapa saran untuk peningkatan sistem di masa depan:

1. Pengembangan Jalur Rekomendasi Penghindaran Otomatis

   Sistem yang dibangun saat ini hanya berfokus pada deteksi rintangan tanpa mengatur arah pergerakan kapal setelah rintangan terdeteksi. Oleh karena itu, perlu dikembangkan integrasi sistem navigasi yang mampu merespons hasil deteksi dengan menentukan jalur penghindaran secara

otomatis, sehingga kapal dapat bergerak secara mandiri dengan aman dan efisien.

2. Penambahan Variasi Jenis Rintangan

   Objek yang digunakan dalam dataset masih terbatas pada rintangan permukaan seperti kapal dan pelampung. Untuk meningkatkan kemampuan deteksi, disarankan agar jenis rintangan diperluas, mencakup objek lain seperti sampah plastik, batang kayu, atau elemen terapung lainnya yang juga berpotensi mengganggu navigasi kapal otonom.

DAFTAR PUSTAKA

Acharya, D., dan Khoshelham, K. (2021). *Parking Occupancy Detection and Slot Delineation Using Deep Learning: A Tutorial Visual parking space detection using deep features View project Visual-inertial odometry for indoor positioning View project* (Nomor July). https://doi.org/10.34727/2021/isbn.978-3-85448-045-7

Charli, F., Syaputra, H., Akbar, M., Sauda, S., dan Panjaitan, F. (2020). Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird. *Journal of Information Technology Ampera*, *1*(3), 185–197. https://doi.org/10.51519/journalita.volume1.isssue3.year2020.page185-197

Chen, T., Zhang, Z., Cheng, Y., Awadallah, A., dan Wang, Z. (2022). The Principle of Diversity: Training Stronger Vision Transformers Calls for Reducing All Levels of Redundancy. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12010–12020. https://doi.org/10.1109/CVPR52688.2022.01171

Della, R. H., dan Rachmannullah, M. A. F. (2024). KAPAL OTONOM: KONSEP, TEKNOLOGI, DAN PROSPEK TRANSPORTASI MARITIM. In *Eureka Media Aksara*. https://repository.penerbiteureka.com/media/publications/584106-kapal-otonom-konsep-teknologi-dan-prospe-9676c22e.pdf

Gad, A. F., dan Skelton, J. (2025). *Faster R-CNN Explained for Object Detection Tasks*. DigitalOcean. https://www.digitalocean.com/community/tutorials/faster-r-cnn-explained-object-detection

IMO. (2021). Outcome of the Regulatory Scoping Exercise for the Use of Maritime Autonomous Surface Ships (Mass). *MSC.1/Circ.1638*, *44*(0), 1–105.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., dan Pietikäinen, M. (2020). Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, *128*(2), 261–318.

https://doi.org/10.1007/s11263-019-01247-4

Malviya, N. (2023). *Object Detection — Anchor Box VS Bounding Box*. Medium. https://medium.com/@nikitamalviya/object-detection-anchor-box-vs-bounding-box-bf1261f98f12

Milic, B. (2022). *Obstacle Detection in a Marine Environment*. Visual Cognitive Systems Laboratory, University of Ljubljana, Faculty of Computer and Information Science. https://box.vicos.si/borja/viamaro/index.html

Pardede, J., dan Hardiansyah, H. (2022). Deteksi Objek Kereta Api menggunakan Metode Faster R-CNN dengan Arsitektur VGG 16. *MIND Journal*, *7*(1), 21–36. https://doi.org/10.26760/mindjournal.v7i1.21-36

Pratama, Boby Mugi dan Mayantasari, M. (2022). *View of Sistem Pendeteksian Rintangan untuk Kapal Tak Berawak dengan Kombinasi Deteksi Tepi, Transformasi Hough dan Deteksi Saliensi.pdf*.

Putro, E. C., Awangga, R. M., dan Andarsyah, R. (2020). *Tutorial Object Detection People With Faster region-Based Convolutional Neural Network(Faster R-CNN)* (R. M. Awangga (ed.)). Kreatif, 2020. https://books.google.co.id/books?id=wawGEAAAQBAJ&lpg=PA1&pg=PA1#v=onepage&q&f=false

Qiao, Y., Yin, J., Wang, W., Duarte, F., Yang, J., dan Ratti, C. (2023). Survey of Deep Learning for Autonomous Surface Vehicles in Marine Environments. *IEEE Transactions on Intelligent Transportation Systems*, *24*(4), 3678–3701. https://doi.org/10.1109/TITS.2023.3235911

Renjun, X., Junliang, Y., Yi, W., dan Mengcheng, S. (2022). Fault Detection Method Based on Improved Faster R-CNN: Take ResNet-50 as an Example. *Geofluids*, *2022*. https://doi.org/10.1155/2022/7812410

Rifiana, S. F., dan Adhi, S. I. (2022). Mengapa Indonesia disebut Negara Maritim? *kompas.com*. https://www.kompas.com/global/read/2022/09/06/212900570/mengapa-indonesia-disebut-negara-maritim-?page=all

Rustina, E., Al-Habib, J., Aulian, D., Ayu, D., dan Sinaga, J. S. (2024). *Keselamatan Perairan Dan Pengelolaan Lalu Lintas Maritim* (S. D. Astuti

(ed.); Pertama). https://books.google.co.id/books?id=x_P3EAAAQBAJ&lpg=PA1&ots=PRG2b9mr-F&dq=Keselamatan maritim merupakan aspek penting dalam pengelolaan sumber daya laut%2C terutama di negara kepulauan seperti Indonesia. Kecelakaan di laut dapat disebabkan oleh berbagai

Shangguan, Z., dan Rostami, M. (2023). Improved Region Proposal Network for Enhanced Few-Shot Object Detection. *Neural Networks*, *180*. https://doi.org/https://doi.org/10.1016/j.neunet.2024.106699

Sunario Megawan, dan Wulan Sri Lestari. (2020). Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, *9*(3), 261–267. https://doi.org/10.22146/.v9i3.231

Tersek, M., Zust, L., dan Kristan, M. (2023). *eWaSR -- an embedded-compute-ready maritime obstacle detection network*. 1–18. http://arxiv.org/abs/2304.11249

Wang, K. J., Jiang, H. F., dan Lee, Y. X. (2022). A multiple-stage defect detection model by convolutional neural networks. *Computers & Industrial Engineering*. https://doi.org/https://doi.org/10.1016/j.cie.2022.107166