

Sistem Pendeteksian Rintangan untuk Kapal Tak Berawak dengan Kombinasi Deteksi Tepi, Transformasi Hough dan Deteksi Saliensi

Boby Mugi Pratama ^{#1}, Menasita Mayantasari ^{#2}

^{#1} Program Studi Informatika, Institut Teknologi Kalimantan

^{#2} Program Studi Fisika, Institut Teknologi Kalimantan

Jl. Soekarno-Hatta Km. 15, Karang Joang, Balikpapan, Indonesia

¹ bmpratama@lecturer.itk.ac.id

² menasita@lecturer.itk.ac.id

Accepted on 11-05-2022

Abstract

Kapal tanpa awak yang digunakan untuk berpatroli di perairan pesisir membutuhkan sistem deteksi dan penghindar rintangan yang andal. Hal ini dikarenakan banyak terdapat objek-objek yang dapat menghalangi lajunya kapal di permukaan air. Tantangan dalam pengembangan sistem pendeteksian rintangan berbasis citra bagi kapal tak berawak ukuran kecil adalah performa komputasi yang terbatas. Penelitian dilakukan untuk membuat sebuah sistem pendeteksian rintangan bagi kapal tanpa awak yang akurat dengan proses komputasi yang minimum agar dapat diimplementasikan ke dalam sistem tertanam. Sistem yang dibuat mengkombinasikan deteksi tepi, transformasi Hough dan deteksi saliensi untuk mendeteksi adanya rintangan di atas permukaan air. Dari hasil pengujian didapatkan bahwa performa sistem dapat mendekati kemampuan sistem lain yang lebih kompleks namun belum dapat mengunggulinya. Sistem ini juga diimplementasikan pada sistem tertanam dan menghasilkan kecepatan yang memungkinkan untuk dilakukan pengimplementasian secara *real-time*.

Kata kunci: citra digital, deteksi rintangan, deteksi saliensi, deteksi tepi, kapal tanpa awak, transformasi Hough

I. PENDAHULUAN

Kapal tanpa awak biasanya dapat digunakan untuk menyisir suatu lokasi yang telah ditentukan. Kapal ini dapat membantu Tentara Nasional Indonesia (TNI) dalam menjalankan operasi patrol di laut Indonesia yang sangat luas [1]. Selama ini patroli TNI di laut mengalami banyak kendala berupa terbatasnya kapal patroli dan jumlah personel jika dibandingkan dengan luasnya wilayah perairan Indonesia [2]. Selain itu, kapal tanpa awak diperkirakan akan memiliki peran yang penting di masyarakat pada masa yang akan datang [3]. Kapal tanpa awak yang digunakan untuk berpatroli di perairan pesisir membutuhkan sistem deteksi dan penghindar rintangan yang andal. Hal ini dikarenakan banyak terdapat objek-objek yang dapat menghalangi lajunya kapal

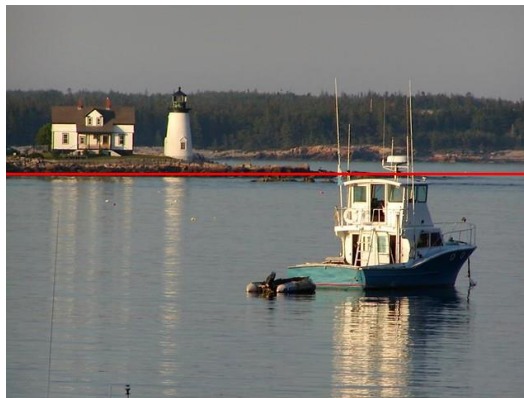
di permukaan air. Objek-objek tersebut antara lain adalah kapal lain, benda terapung dan garis pantai. Objek-objek ini dapat membahayakan kapal tanpa awak apabila tertabrak saat kapal sedang menjalankan misinya. Oleh karena itu, sistem harus dikembangkan untuk dapat merespon dinamika lingkungan yang berubah-ubah tersebut.

Tantangan dalam pengembangan sistem pendeteksian rintangan berbasis citra bagi kapal tak berawak ukuran kecil adalah performa komputasi yang terbatas. Proses komputasi pada kapal tak berawak jenis ini biasanya ditenagai oleh sistem tertanam yang tidak memiliki kapasitas komputasi yang besar [4], [5]. Namun metode deteksi rintangan berbasis pengolahan citra yang banyak berkembang adalah metode berbasis kecerdasan artifisial [6], [7]. Metode-metode berbasis kecerdasan artifisial tersebut tentunya membutuhkan kapasitas komputasi yang besar. Metode jenis ini tentunya tidak tepat jika diterapkan pada sistem tertanam kapal tak berawak. Maka dari itu penelitian ini dilakukan untuk memberikan alternatif sistem pendeteksian rintangan berbasis pengolahan citra yang mengedepankan pada komputasi yang ringan agar dapat diimplementasikan pada sistem tertanam.

II. TINJAUAN PUSTAKA

Secara umum, pendeteksian rintangan pada kapal tak berawak dilakukan berbasis sensor atau berbasis pengolahan citra. Pendeteksian rintangan berbasis sensor ultrasonik utamanya digunakan untuk mendeteksi rintangan berupa batas perairan dan daratan [8]. Sistem pendeteksian jenis ini kurang mampu mendeteksi rintangan yang berukuran kecil atau rintangan dengan bentuk tertentu karena mengandalkan pantulan dari gelombang ultrasonik yang dipancarkan. Sistem ini sering dibantu dengan sistem pendeteksian rintangan berbasis citra untuk menutupi kekurangan tersebut.

Akurasi yang baik pada pendeteksian objek dalam citra pada umumnya dicapai dengan metode berbasis pengolahan citra dan kecerdasan artifisial [6], [7], [9] yang mana membutuhkan kemampuan komputasi sistem yang besar. Pendeteksian juga dapat dilakukan berbasis LiDAR [10]. Namun, pendeteksian berbasis LiDAR tergolong mahal dan tidak aplikatif pada kapal tak berawak berukuran kecil akibat ukuran sensor yang besar. Sehingga dalam rangka mengatasi keterbatasan performa komputasi sistem tertanam Raspberry Pi 4 yang digunakan pada kapal, maka dirancang sebuah sistem pendeteksian rintangan berbasis pengolahan citra yang mengombinasikan deteksi tepi Canny [11], transformasi Hough [12], dan deteksi saliensi [13].



Gambar 1. Pendeteksian horizon pada citra menggunakan deteksi tepi dan transformasi Hough

Dalam melakukan pendeteksian rintangan, mula-mula citra tangkapan kamera diolah menggunakan deteksi tepi Canny dan transformasi Hough. Kedua metode pengolahan citra ini dilakukan untuk mencari garis horizon pada citra sebagaimana ditunjukkan pada Gambar 1. Rintangan yang akan dideteksi menggunakan deteksi saliensi adalah rintangan yang berada di bawah garis horizon ini sehingga komputasi deteksi saliensi tidak dilakukan pada keseluruhan pixel pada citra.

Dalam proses deteksi tepi Canny, noise pada citra terlebih dahulu dikurangi dengan menggunakan filter Gaussian berukuran 5x5. Proses *filtering* ini ditunjukkan dengan persamaan berikut:

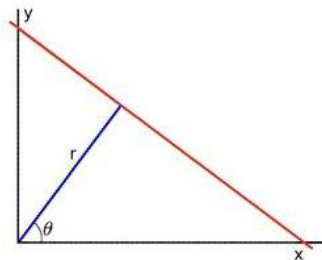
$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \quad (1)$$

dimana B adalah matriks citra hasil filtering, A adalah matriks citra sebelum dilakukan filtering dan * adalah operasi konvolusi. Citra yang sudah dikurangi noise-nya ini kemudian dikonvolusikan dengan kernel Sobel untuk mendapatkan turunan pertama arah horizontal (G_x) dan turunan pertama arah vertikal (G_y). Dari kedua citra ini, dapat dihitung arah (θ) dan gradien tepi (G) dari tiap pixel dengan menggunakan persamaan berikut:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2)$$

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (3)$$

Berdasarkan hasil deteksi tepi tersebut, horizon pada citra dideteksi dengan menggunakan transformasi Hough. Transformasi Hough mengekspresikan garis-garis pada citra hasil deteksi tepi dengan menggunakan sistem koordinat polar sebagaimana persamaan berikut:

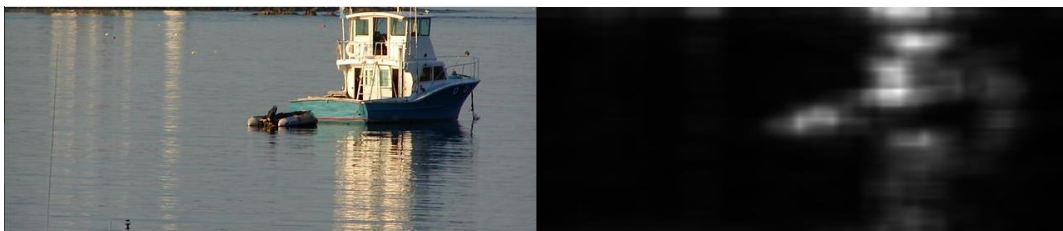


$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right) \quad (4)$$

sehingga:

$$r = x \cos \theta + y \sin \theta \quad (5)$$

Hal tersebut menjadikan apabila terdapat sebuah garis lurus pada citra, tiap pixel pada garis akan memberikan nilai r dan θ yang sama. Jika sepasang nilai r dan θ tertentu sering muncul, maka pada nilai r dan θ tersebut terdapat sebuah garis utama pada citra sebagai garis horizon.



Gambar 2. Deteksi saliensi pada citra menghasilkan peta saliensi

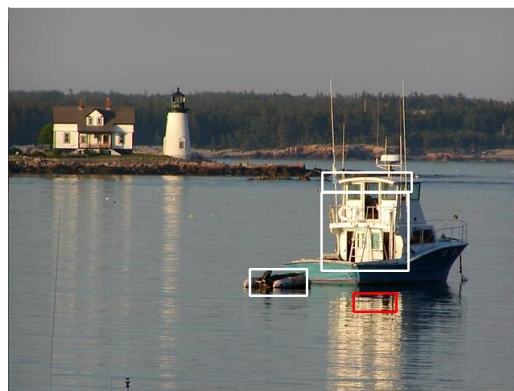
Deteksi saliensi dilakukan dengan mencari jarak Euclidean antara vektor pixel warna $L^*a^*b^*$ pada citra yang diaplikasikan filter Gaussian dan rata-rata vektor $L^*a^*b^*$ dari citra aslinya. Peta saliensi S dari citra I dengan dengan lebar W dan tinggi H dihitung dengan menggunakan persamaan:

$$S(x, y) = ||I_m(x, y) - I_G(x, y)|| \quad (6)$$

dimana $I_m(x, y)$ adalah rata-rata vektor citra yang dihitung dengan persamaan:

$$I_m = \Sigma I / (H \times W) \quad (7)$$

dan $I_G(x, y)$ adalah hasil Gaussian Blur dari citra. Setelah perhitungan saliensi dilakukan, peta saliensi didapatkan dengan cara menormalisasi batas nilai saliensi pada rentang $[0.0, 255.0]$. Contoh citra dan hasil peta saliensinya ditunjukkan sebagaimana pada Gambar 2. Peta saliensi inilah yang kemudian digunakan untuk mendeteksi rintangan dalam citra sebagaimana ditunjukkan pada Gambar 3.



Gambar 3. Rintangan pada citra dapat terdeteksi dari peta saliensi

III. RESEARCH METHOD

Performa dari sistem yang telah dibuat diukur dengan cara pengujian. *Dataset* yang digunakan untuk pengujian sistem berasal dari *dataset* Common Objects in Context (COCO) dari Microsoft [14]. Citra dataset yang terkumpul adalah sebanyak 200 citra. Citra yang dipilih adalah citra yang dianggap representatif dengan kondisi dimana sistem ini akan diterapkan. Beberapa citra yang menjadi *dataset* dalam pengujian sistem ditunjukkan sebagaimana pada Gambar 4.



Gambar 4. Contoh citra dataset untuk pengujian

Melalui pengujian menggunakan dataset, hasil pendeteksian rintangan selanjutnya dikategorikan menjadi tiga kategori yaitu: *true positive* (TP), *false positive* (FP) dan *false negative* (N). TP adalah kondisi saat rintangan pada citra terdeteksi oleh sistem. FP adalah kondisi saat tidak terdapat rintangan pada citra, namun sistem mendeteksi adanya rintangan. FN adalah kondisi saat rintangan pada citra tidak terdeteksi oleh sistem. Kondisi TP, FP dan N ditunjukkan sebagaimana pada Gambar 5. Dengan menggunakan ketiga variabel tersebut, performa sistem diukur dengan menggunakan analisis *precision* dan *recall* [15]. *Precision* (P) dan *recall* (R) dihitung menggunakan persamaan berikut:

$$P = \frac{TP}{TP + FP} \quad (8)$$

$$R = \frac{TP}{TP + FN} \quad (9)$$



Gambar 5. Kategori hasil pendeteksian: A adalah FN, B adalah FP dan C adalah TP

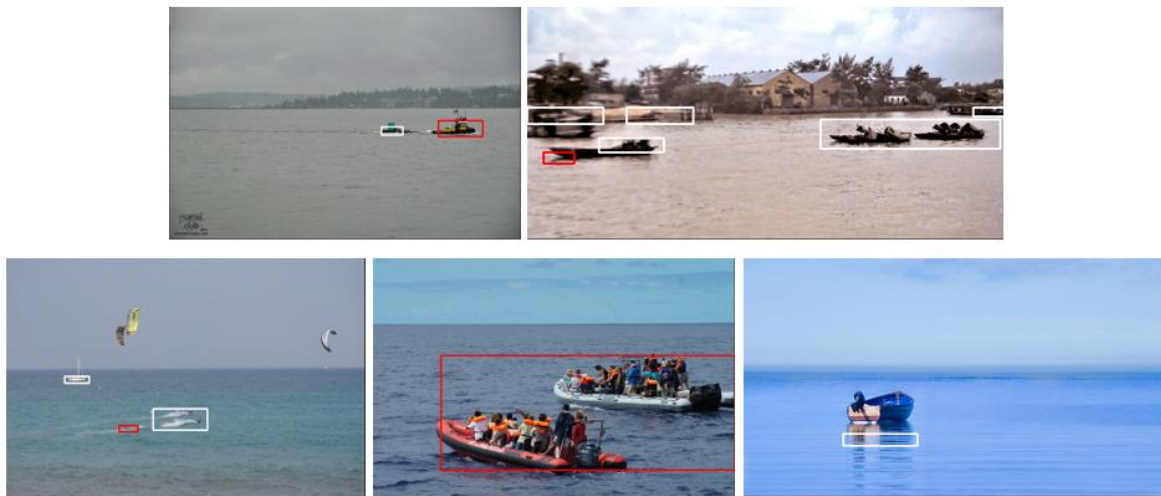
Analisis *precision* dan *recall* ini digunakan karena telah umum digunakan berbagai peneliti dalam penelitian terkait pendeteksian objek. Hal tersebut akan memudahkan dalam membandingkan hasil penelitian ini dengan penelitian yang lainnya.

IV. HASIL DAN PEMBAHASAN

TABEL 1
HASIL PENGUJIAN SISTEM

| TP | FP | N | P | R |
|-----|----|----|------|------|
| 229 | 82 | 41 | 0,74 | 0,85 |

Pengujian sistem pendeteksian menggunakan citra dataset menghasilkan data sebagaimana diperlihatkan pada Tabel 1. Nilai *precision* sistem berada pada angka 0,74. Performa sistem semakin baik apabila apabila nilai *precision* semakin mendekati nilai 1 dan sebaliknya apabila nilai *precision* mendekati 0. Sedangkan nilai *recall* sistem berada pada angka 0,80. Performa sistem semakin baik apabila apabila nilai *recall* semakin mendekati nilai 1 dan sebaliknya apabila nilai *recall* mendekati 0. *Precision* menggambarkan performa sistem dalam hal akurasi dari prediksi yang bersifat positif. Sedangkan *recall* menggambarkan performa sistem dalam hal sensitivitas, yaitu seberapa mudah suatu objek dapat terdeteksi sebagai rintangan oleh sistem. Beberapa citra hasil pengujian sistem diperlihatkan sebagaimana pada Gambar 6.



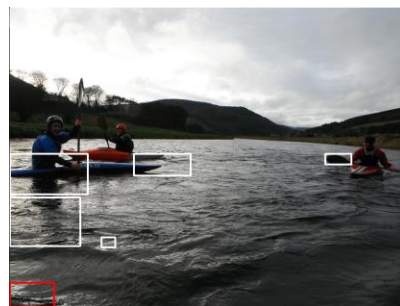
Gambar 6. Beberapa citra hasil pengujian sistem

Kesalahan pendeteksian yang dialami disebabkan oleh berbagai hal. Kesalahan pendeteksian pada kategori N umumnya disebabkan oleh kesalahan dalam proses deteksi horizon menggunakan transformasi Hough. Pada Gambar 7 diperlihatkan bahwa garis horizon yang terdeteksi (garis merah) menggunakan transformasi Hough bukanlah garis horizon yang sesungguhnya. Hal tersebut menyebabkan objek rintangan yang berada di atasnya menjadi tidak terdeteksi. Untuk menangani hal ini, diperlukan metode deteksi horizon yang lebih akurat untuk citra permukaan air.



Gambar 7. Kesalahan deteksi horizon menyebabkan kesalahan pendeteksian

Kesalahan pendeteksian pada kategori FP umumnya disebabkan oleh refleksi dari permukaan air. Kesalahan pendeteksian ini dicontohkan sebagaimana pada Gambar 8 dimana refleksi pada permukaan air banyak terdeteksi sebagai objek rintangan. Untuk menangani hal ini, metode deteksi saliensi yang digunakan untuk mendeteksi objek dapat dimodifikasi atau dikombinasikan dengan metode lain.



Gambar 8. Kesalahan deteksi akibat refleksi pada permukaan air

Perbandingan performa sistem dalam penelitian ini dengan sistem lain pada penelitian sebelumnya ditunjukkan sebagaimana pada Tabel 2. Sistem pendeteksian dalam penelitian ini belum mampu untuk menghasilkan performa yang lebih baik terutama jika dihadapkan dengan sistem yang berbasis pada *deep learning*. Hal ini dikarenakan sistem yang telah dibuat mengutamakan kebutuhan komputasi yang rendah agar dapat diterapkan pada sistem tertanam. Pada pengujian menggunakan Raspberry Pi 4, sistem ini mampu memproses citra dalam waktu rata-rata sebesar 0,47 s menggunakan citra beresolusi 640x480 pixel. Sementara itu, sistem lain dalam penelitian sebelumnya belum diterapkan pada sistem tertanam. Hal tersebut dikarenakan sistem lain tersebut membutuhkan kapasitas komputasi perangkat yang besar, terutama jika menggunakan *convolutional neural network* yang mana harus menggunakan *Graphic Processing Unit* (GPU) untuk mendapatkan kecepatan komputasi yang memadai.

TABEL II
PERBANDINGAN PERFORMA SISTEM

| Model | P | R | Metode | Platform |
|----------------|----------|----------|--|-----------------|
| Model 1 [6] | 0,89 | 0,91 | <i>Convolutional neural network</i> | PC |
| Model 2 [16] | 0,92 | 0,80 | <i>Semantic segmentation</i> | PC |
| Penelitian ini | 0,74 | 0,85 | Deteksi tepi, transformasi Hough dan deteksi saliensi | Raspberry Pi 4 |

V. KESIMPULAN

Melalui penelitian ini, dibuat sebuah sistem pendeteksian rintangan untuk kapal tanpa awak berbasis pengolahan citra digital. Sistem yang telah dibuat mengkombinasikan deteksi tepi, transformasi Hough dan deteksi saliensi untuk mendeteksi objek pada citra. Deteksi tepi dan transformasi Hough digunakan untuk mendeteksi horizon pada citra. Bagian citra di bawah horizon selanjutnya diproses menggunakan deteksi saliensi. Pengujian sistem belum memperlihatkan performa pendeteksian yang lebih baik jika dibandingkan dengan penelitian sebelumnya. Namun sistem dalam penelitian ini berpotensi untuk dikembangkan menjadi sistem pendeteksian secara *real-time* karena beban komputasi yang ringan dan dapat diimplementasikan dalam sistem tertanam Raspberry Pi 4.

DAFTAR PUSTAKA

- [1] B. P. Zen, "The Concept of Big Data Analysis for Maritime Information on Indonesian Waters using K-Means Algorithm," *INISTA: Journal of Informatics, Information System, Software Engineering and Applications*, vol. 3, no. 2, Art. no. 2, Jul 2021, doi: 10.20895/inista.v3i2.200.
- [2] D. Hardianto dan W. D. Aryawan, "Pembuatan Konsep Desain Unmanned Surface Vehicle (USV) untuk Monitoring Wilayah Perairan Indonesia," *J. Tek. ITS*, vol. 6, no. 2, hlm. G65–G70, Sep 2017, doi: 10.12962/j23373539.v6i2.23366.
- [3] C. Barrera, I. Padron, F. S. Luis, dan O. Llinas, "Trends and challenges in unmanned surface vehicles (USV): From survey to shipping," *TransNav Int. J. Mar. Navig. Saf. Sea Transp.*, vol. Vol. 15 No. 1, 2021, doi: 10.12716/1001.15.01.13.
- [4] A. Gonzalez-Garcia dan H. Castañeda, "Guidance and Control Based on Adaptive Sliding Mode Strategy for a USV Subject to Uncertainties," *IEEE J. Ocean. Eng.*, vol. 46, no. 4, hlm. 1144–1154, Okt 2021, doi: 10.1109/JOE.2021.3059210.
- [5] D. Sousa, M. Luís, S. Sargento, dan A. Pereira, "An Aquatic Mobile Sensing USV Swarm with a Link Quality-Based Delay Tolerant Network," *Sensors*, vol. 18, no. 10, hlm. 3440, Okt 2018, doi: 10.3390/s18103440.
- [6] L. Ma, W. Xie, H. Huang, dan School of Information Science and Engineering, Harbin Institute of Technology, Weihai 264209, China, "Convolutional neural network based obstacle detection for unmanned surface vehicle," *Math. Biosci. Eng.*, vol. 17, no. 1, hlm. 845–861, 2020, doi: 10.3934/mbe.2020045.
- [7] W. Wang dan X. Luo, "Autonomous Docking of the USV using Deep Reinforcement Learning Combine with Observation Enhanced," dalam 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Agu 2021, hlm. 992–996. doi: 10.1109/AEECA52519.2021.9574371.
- [8] S. Sembiring dan K. Exaudi, "Perancangan Robot Kapal dengan Perilaku Menghindari Rintangan," *KNTIA*, vol. 4, no. 2017, hlm. A117–A124, 2017.
- [9] B. M. Pratama, D. Gunawan, dan R. A. G. Gultom, "Deep learning-based object detection and geographic coordinate estimation system for GeoTiff imagery," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, hlm. 012003, Jul 2020, doi: 10.1088/1742-6596/1577/1/012003.
- [10] J. Villa, J. Aaltonen, dan K. T. Koskinen, "Path-Following With LiDAR-Based Obstacle Avoidance of an Unmanned Surface Vehicle in Harbor Conditions," *IEEEASME Trans. Mechatron.*, vol. 25, no. 4, hlm. 1812–1820, Agu 2020, doi: 10.1109/TMECH.2020.2997970.
- [11] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, hlm. 679–698, Nov 1986, doi: 10.1109/TPAMI.1986.4767851.
- [12] L. A. F. Fernandes dan M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," *Pattern Recognit.*, vol. 41, no. 1, hlm. 299–314, Jan 2008, doi: 10.1016/j.patcog.2007.04.003.
- [13] R. Achanta, S. Hemami, F. Estrada, dan S. Susstrunk, "Frequency-tuned salient region detection," dalam 2009 IEEE conference on computer vision and pattern recognition, 2009, hlm. 1597–1604.
- [14] T.-Y. Lin dkk., "Microsoft COCO: Common Objects in Context," dalam *Computer Vision – ECCV 2014*, Cham, 2014, hlm. 740–755. doi: 10.1007/978-3-319-10602-1_48.
- [15] K. M. Ting, "Precision and Recall," dalam *Encyclopedia of Machine Learning*, C. Sammut dan G. I. Webb, Ed. Boston, MA: Springer US, 2010, hlm. 781–781. doi: 10.1007/978-0-387-30164-8_652.
- [16] M. Kristan, V. Sulić Kenk, S. Kovačič, dan J. Perš, "Fast Image-Based Obstacle Detection From Unmanned Surface Vehicles," *IEEE Trans. Cybern.*, vol. 46, no. 3, hlm. 641–654, Mar 2016, doi: 10.1109/TCYB.2015.2412251.