

DETEKSI IKAN DENGAN MENGGUNAKAN ALGORITMA ADABOOST-SVM

Singgih Putra Pratama ¹⁾, Basuki Rahmat ²⁾, Fetty Tri Anggraeny ³⁾

E-mail: ¹⁾singgihpuptrap.if@gmail.com, ²⁾basukirahmat.if@upnjatim.ac.id,

³⁾fettyanggraeny.if@upnjatim.ac.id

^{1,2,3)}Program Studi Informatika, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

Abstrak

Indonesia merupakan negara yang kaya akan sumber daya alam baik hayati maupun non-hayati. Salah satu sumber daya alam hayati yang sangat banyak jumlahnya di Indonesia adalah laut. Untuk mempermudah mengidentifikasi ikan, dapat memanfaatkan sebuah teknologi yang dapat membantu manusia untuk dapat mengenali ikan dengan menggunakan visi komputer dan pendekatan pemrosesan gambar untuk deteksi ikan menggunakan algoritma *Histogram of Oriented Gradients* (HOG) dan AdaBoost-SVM. Hasil penelitian menunjukkan bahwa metode HOG dan AdaBoost-SVM dapat menghasilkan tingkat akurasi rata-rata sebesar 85%.

Kata kunci: *Deteksi Ikan, Histogram of Oriented Gradients, Adaboost-SVM*

PENDAHULUAN

Indonesia merupakan negara yang kaya akan sumber daya alam baik hayati maupun non-hayati. Salah satu sumber daya alam hayati yang sangat banyak jumlahnya di Indonesia adalah laut, yang menjadikan Indonesia dijuluki negara Maritim karena wilayah perairannya yang lebih luas daripada wilayah daratan. Salah satu sumber daya alam laut yang banyak dimanfaatkan oleh masyarakat Indonesia adalah ikan. Berbagai jenis ikan hidup dan berkembang biak tersebar pada beberapa wilayah perairan yang ada di Indonesia.

Pada penelitian yang dilakukan oleh E. Prianto dan N. K. Suryati [1] terdapat 54 jenis ikan yang terdapat pada Sungai Musi. Dari banyaknya jenisnya ikan tersebut tidak semua orang dapat mengenali jenis-jenis ikan melalui dari segi fisik yang tampak secara visual [2].

Untuk mempermudah mengidentifikasi ikan, dapat memanfaatkan sebuah teknologi yang dapat membantu manusia untuk dapat mengenali ikan. Salah satu teknologi yang berperan penting salah satunya yaitu teknologi visi komputer. Visi komputer merupakan serangkaian teknologi yang memungkinkan perangkat lunak untuk menangkap, menganalisis, dan memproses gambar.

Visi komputer dan pendekatan pemrosesan gambar untuk deteksi ikan bawah air mendapatkan perhatian yang khusus oleh para ilmuwan kelautan [3] dengan mengestimasi keberadaan ikan dari video dan gambar dapat mendukung ahli biologi kelautan untuk memahami lingkungan bawah laut yang alami, mempromosikan pelestariannya dan mempelajari perilaku interaksi antara hewan laut yang menjadi bagian darinya [4].

Untuk melakukan deteksi ikan dibutuhkan sebuah proses pengenalan ikan. Pengenalan ikan merupakan proses mengidentifikasi ikan berdasarkan gambaran bentuk pola tubuh ikan beserta ciri-cirinya [5]. Dari permasalahan tersebut untuk mengatasinya dengan cara pendekatan pengolahan citra yang merupakan salah satu bidang ilmu kecerdasan buatan khususnya visi komputer.

Dalam visi komputer dibutuhkan sebuah proses ekstraksi fitur untuk mengenali bentuk pola tubuh ikan beserta ciri-cirinya. Salah satu ekstraksi fitur yang dapat digunakan visi komputer adalah *Histogram of Oriented Gradients* (HOG). Pada

penelitian yang dilakukan oleh N. Dalal dan B. Triggs [6], HOG merupakan ekstraksi ciri yang dapat mendeteksi objek dengan baik dengan menggunakan SVM sebagai klasifikasinya, dengan harapan dapat meningkatkan tingkat akurasi dalam mendeteksi ikan dapat menggunakan teknik boosting salah satunya adalah algoritma Adaboost.

Pada penelitian sebelumnya yang dilakukan oleh [7] dengan menggunakan HOG ekstraksi ciri dan melakukan perbandingan SVM dan Adaboost menghasilkan tingkat akurasi rata-rata sebesar 90%. [8] melakukan penelitian untuk mengklasifikasi *Chronic Kidney Disease* (CKD) dengan menggunakan SVM dan PSO sebagai seleksi fitur. Hasilnya tingkat akurasi dapat ditingkatkan dengan menggunakan Adaboost dari 36.20% hingga 99.50%.

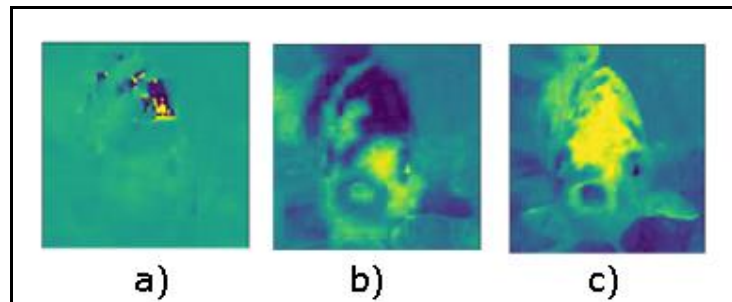
Oleh karena itu penelitian ini, akan di bangun sebuah program deteksi ikan melalui citra, dimana dalam proses ekstraksi ciri menggunakan *Histogram of Oriented Gradients* (HOG) dan Adaboost dengan *Support Vector Machine* (Adaboost-SVM).

METODOLOGI

HSV (*Hue Saturation Value*)

HSV merupakan salah satu dari sekian jenis ruang warna yang dapat digunakan dalam pengolahan citra digital, memiliki 3 komponen nilai warna yang terdiri dari *Hue*, *Saturation*, dan *Value*. Pada dasarnya citra mempunyai nilai RGB sebagai acuan warna [9], oleh karena itu dibutuhkan konversi model RGB ke HSV.

Warna yang dihasilkan dari sebuah *Hue* pada dataset penelitian ini menunjukkan tidak dominan terhadap bentuk ikan, contoh nilai warna HSV pada salah satu sampel dataset ikan seperti pada gambar 1.



Gambar 1. Hasil a) *Hue* b) *Saturation* c) *value*

Sehingga nilai yang digunakan adalah S dan V di mana akan digabungkan sehingga menghasilkan *S-V feature*. Untuk mengubah warna citra dari RGB menjadi HSV dapat menggunakan persamaan 1.

$$\begin{aligned}
 V &= \max\left(\frac{R}{255}, \frac{G}{255}, \frac{B}{255}\right) \\
 S &= \begin{cases} 0 & V = 0 \\ 1 - \frac{\min(\frac{R}{255}, \frac{G}{255}, \frac{B}{255})}{V} & V \neq 0 \end{cases} \\
 H &= \begin{cases} 0^\circ & V = \min(\frac{R}{255}, \frac{G}{255}, \frac{B}{255}) \\ 60^\circ \times \left(\frac{G-B}{255V - \min(R, G, B)} \bmod 6\right) & V = \frac{R}{255} \\ 60^\circ \times \left(\frac{B-R}{255V - \min(R, G, B)} + 2\right) & V = \frac{G}{255} \\ 60^\circ \times \left(\frac{R-G}{255V - \min(R, G, B)} + 4\right) & V = \frac{B}{255} \end{cases} \quad (1)
 \end{aligned}$$

Otsu Thresholding

Metode Otsu merupakan metode yang digunakan dalam segmentasi citra digital. Metode otsu menggunakan nilai ambang secara otomatis, yakni dengan mengubah citra digital warna abu-abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital [10]. Sehingga citra dapat di pisahkan antara *foreground* dan *background*. Metode Otsu di perkenalkan oleh N. Otsu [11].

Cara kerja dari metode Otsu yakni dengan mengubah data citra menjadi *histogram*. Sebagai contoh citra *greyscale* di mana nilai pixel dari 0 hingga 255. Kemudian apabila mengambil *threshold* dengan nilai $T=2$, citra akan terbagi menjadi 2 kelas.

Kelas 1 (nilai *pixel* < 2) dan kelas 2 (nilai *pixel* > 2) di mana kedua kelas tersebut mewakili *background* dan *foreground* dari sebuah citra yang telah di masukkan. Tetapi kelas 2 dapat menjadi *background* apabila *foreground* lebih gelap dari pada *background*. Untuk menghitung sebuah *variance*, menggunakan persamaan 2 yang dapat di artikan sebagai distribusi dari sebuah data. Semakin tinggi nilai dari sebuah *variance* semakin tinggi data yang tersebar.

$$\sigma^2 = \frac{\sum_{i=0}^N (S_i - \mu)^2}{N} \quad (2)$$

Dari persamaan 2, S_i adalah nilai piksel, μ merupakan mean dan N adalah total piksel dari sebuah gambar.

AdaBoost-SVM

Algoritma AdaBoost merupakan algoritma yang dapat melakukan *boosting* pada klasifikasi yang lemah dan dapat melakukan penyesuaian terhadap nilai *error* [12]. Algoritma klasifikasi yang menggunakan metode *boosting* diharapkan performa dari algoritma tersebut dapat meningkat dari pada tanpa menggunakan metode *boosting* [13].

Dengan menggunakan SVM sebagai base learnernya, maka perlu penyesuaian dalam mengimplementasikannya. SVM yang digunakan adalah SVM dengan kernel RBF (*Radial Basis Function*) seperti pada persamaan 3, terdapat sebuah parameter gamma atau sigma yang dapat digunakan untuk melemahkan klasifikasi tersebut.

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\omega^2}\right) \quad (3)$$

Pada persamaan 3, $\|x - x'\|^2$ merupakan *square Euclidean distance* antara 2 *feature vectors* yakni x dan x' . Sedangkan ω merupakan parameter bebas.

Dikarenakan adanya dilema dalam melakukan penurunan nilai gamma atau sigma, sehingga menggunakan teknik *diverse* AdaBoost-SVM. Mengacu pada jurnal penelitian membahas mengenai *diversity* yang dilakukan oleh P. Melville dan R. J. Mooney [14] di mana mengukur ketidaksamaan antara komponen klasifikasi yang satu dengan semua komponen klasifikasi lainnya, dengan meningkatkan nilai *diversity* dari sebuah komponen klasifikasi akan mendapatkan hasil rata-rata akurasi yang lebih baik. Seperti pseudocode oleh X. Li, L. Wang dan E. Sung [15] berikut ini:

- a. **Input:** Suatu kumpulan *sample* pelatihan dengan label $\{(X_i, Y_i), \dots, (X_N, Y_N)\}$, inialisasi σ, σ_{ini} ; minimal σ, σ_{min} ; langkah dari σ, σ_{step} ; threshold dari *diversity*, *DIV*.
- b. **Initialize:** Bobot suatu *sample* pelatihan: $W_t^1 = 1/N$, untuk semua $i = 1, \dots, N$.
- c. **Do while** ($\sigma > \sigma_{min}$):
 - 1) Gunakan RBFSVM Algoritma untuk melatih suatu komponen klasifikasi, h_t . Pada *sample* bobot pleatihan.
 - 2) Hitung diversity dari h_t : $D_t = \sum_{i=1}^N d_t(x_i)$
 - 3) Hitung kesalahan pelatihannya pada h_t : $\varepsilon_t = \sum_{i=1}^N W_i^t, Y_i \neq h_t(X_i)$.
 - 4) Jika $\varepsilon_t > 0.5$ atau $d_t < DIV$, kurangi nilai σ dengan σ_{step} dan kembali ke langkah 1.
 - 5) Tetapkan bobot untuk component classifier h_t : $\alpha_t = \frac{1}{2} \ln(\frac{1-\varepsilon_t}{\varepsilon_t})$.
 - 6) Update bobot sample pelatihan $w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t \gamma_i h_t(x_i)\}}{C_t}$, $i = 1, \dots, N$ C_t adalah suatu konstanta normalisasi.
- d. **Output:** $f(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

Sehingga algoritma ini dinamakan *Diverse Adaboost-SVM*. Pada algoritma ini *diversity* dihitung berdasarkan jika nilai $h_t(x_i)$ merupakan nilai hasil dari prediksi label yang dilakukan klasifikasi pada sampel x_i dan $f(x_i)$ adalah kombinasi prediksi label dari semua komponen yang ada pada klasifikasi. Nilai *diversity* pada klasifikasi di hitung menggunakan persamaan 4 seperti berikut ini:

$$d_t(x_i) = \begin{cases} 0, & \text{jika } h_t(x_i) = f(x_i) \\ 1, & \text{jika } h_t(x_i) \neq f(x_i) \end{cases} \quad (4)$$

Dimana pada persamaan 4, apabila nilai hasil prediksi klasifikasi SVM tidak sama dengan nilai prediksi sebenarnya akan bernilai 1 dan sebaliknya apabila nilai prediksi tidak sama maka akan bernilai 0.

Pengumpulan Data

Dataset yang digunakan merupakan hasil rekaman sebuah video ikan pada aquarium yang berada di gedung fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur. Video tersebut di ubah menjadi beberapa citra (30 gambar tiap detiknya). Kemudian melakukan proses cropping pada objek ikan dan bukan ikan dengan skala 1:1 yang akan memudahkan proses mengubah ukuran citra pada tahap selanjutnya.

Setelah itu citra akan dibedakan menjadi 2 yakni citra positif (ikan) dan citra negatif (bukan ikan) dengan total 164 citra yang terdiri dari 127 citra positif dan 37 citra negatif. Contoh citra positif dan negatif seperti pada gambar 2 dan gambar 3.



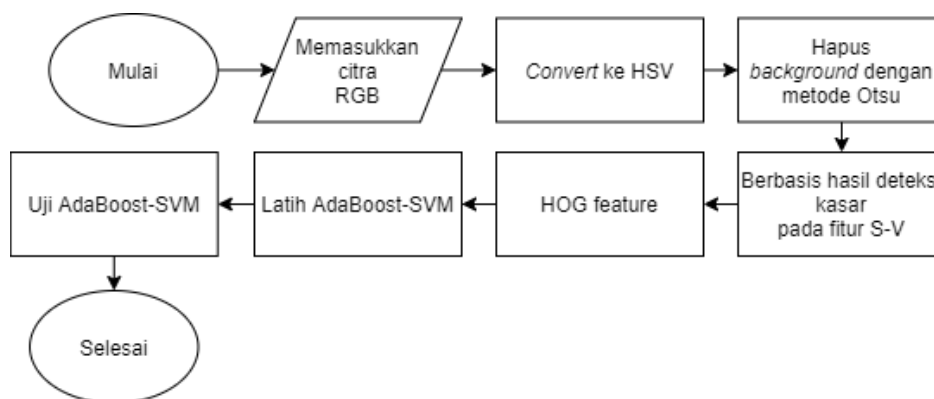
Gambar 2. Citra positif (ikan)



Gambar 3. Citra negatif (bukan ikan)

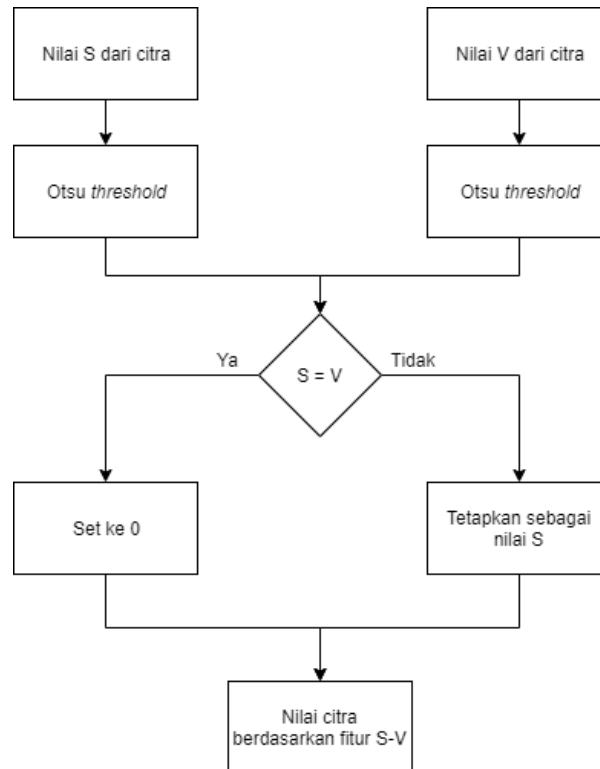
Desain Sistem

Bagian ini merupakan Desain sistem dari deteksi ikan menggunakan *Histogram of Oriented Gradients* (HOG) dan AdaBoost-SVM. Secara keseluruhan, desain sistem ditampilkan pada gambar 4.



Gambar 4. Desain sistem

Pada gambar 4, memasukkan citra RGB merupakan gambar yang di ambil pada dataset yang telah di buat. Kemudian citra diubah dari RGB menjadi HSV, nilai yang digunakan adalah nilai saturation dan value saja. Dikarenakan nilai hue cenderung sama seperti saturation, kemudian masing-masing di hapus *background* menggunakan *Otsu Thresholding*. Nilai saturation dan value akan di seleksi seperti algoritma pada gambar 5.

Gambar 5. **Fitur s-v**

Setelah melewati proses pada gambar 5, maka nilai s-v tersebut di ekstraksi ciri dengan menggunakan HOG. Kemudian citra di bagi menjadi 2 yakni citra untuk latih klasifikasi dan uji coba klasifikasi, dengan langkah melakukan pelatihan mesin klasifikasinya terlebih dahulu kemudian melakukan uji coba terhadap klasifikasi tersebut.

HASIL DAN PEMBAHASAN

Pada penelitian ini uji coba dilakukan dengan menggunakan klasifikasi Adaboost-SVM dengan RBF Kernel. Setiap percobaan memiliki parameter yang berbeda-beda. Parameter yang akan di uji coba adalah Sigma dan pada SVM. Pada akhir pengujian parameter dilakukan percobaan perbandingan dengan beberapa klasifikasi yakni dengan Adaboost murni (*Decision Tree* sebagai *base learner*) dan SVM dengan RBF kernel.

Percobaan perubahan parameter dilakukan bertujuan untuk mengetahui tingkat error dan akurasi terbaiknya. Pada setiap melakukan percobaan citra latih dan citra uji untuk klasifikasi AdaBoost-SVM bernilai tetap.

Percobaan Perubahan Sigma

Percobaan perubahan *sigma* ini dilakukan untuk mengetahui nilai *sigma* yang terbaik dan hasil yang terbaik tersebut akan digunakan pada tahap uji coba perubahan nilai *C*. Percobaan perubahan *sigma* ini diawali dengan nilai 100 hingga 20. Setiap percobaan akan dilakukan pengurangan nilai *sigma* sebesar 10. Hasil keseluruhan dari percobaan perubahan sigma ini dapat dilihat pada tabel 1.

Tabel 1. **Hasil percobaan perubahan nilai sigma AdaBoost-SVM**

Percobaan ke-	Sigma	C	Akurasi (%)	Precision	Recall
1	100	1	81	0,81	1
2	90	1	81	0,81	1
3	80	1	81	0,81	1
4	70	1	81	0,81	1
5	60	1	81	0,81	1
6	50	1	81	0,81	1
7	40	1	89	0,88	1
8	30	1	89	0,88	1
9	20	1	89	0,88	1

Pada tabel 1, nilai *sigma* 100 hingga 50 mendapatkan nilai akurasi yang stagnan yakni sebesar 81% kemudian nilai 40 hingga 20 mendapatkan akurasi yang lebih baik yakni sebesar 89%. Maka dari itu pada penelitian ini menetapkan nilai *sigma* 40 yang terbaik perolehan *sigmanya*.

Percobaan Perubahan C pada SVM

Parameter yang dapat di ubah selanjutnya adalah parameter *C* pada SVM sebagaimana SVM menjadi base learner dari teknik Adaboost. Sehingga dilakukan uji coba parameter *C* pada SVM. Dari percobaan 3.2 sebelumnya mendapatkan nilai *sigma* terbaik yakni sebesar 40. Oleh karena itu *sigma* 40 akan selalu tetap nilainya dan yang akan dilakukan percobaan adalah nilai *C*-nya.

Karena parameter *C* berfungsi sebagai untuk mengatur berapa banyak menghindari kesalahan klasifikasi pada saat pelatihan. Semakin besar nilai *C* maka akan semakin kecil margin dari hyperplane jika hyperplane dapat berkerja dengan baik maka akan menghasilkan nilai akurasi yang lebih baik. Maka dari itu nilai *C* ini berpengaruh besar, dan setiap uji coba dilakukan penambahan nilai sebesar 0.5 hingga mencapai 5. Secara keseluruhan hasil dari percobaan perubahan *C* dapat di lihat pada tabel 3 berikut ini:

Tabel 2. Hasil percobaan perubahan nilai C pada SVM

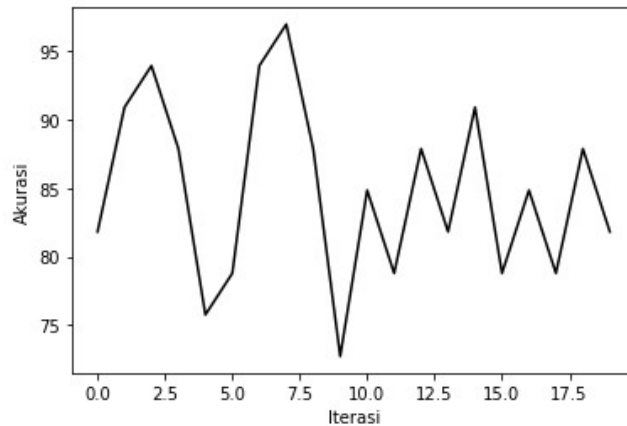
Percobaan ke-	Sigma	C	Akurasi (%)	Precision	Recall
0	40	1	89	0,88	1
1	40	1.5	89	0,88	1
2	40	2	89	0,88	1
3	40	2.5	85	0,875	0,95
4	40	3	85	0,875	0,95
5	40	3.5	85	0,88	0,95
6	40	4	89	0,913	0,9545
7	40	4.5	93	0,92	1
8	40	5	93	0,9166	1

Pada tabel 2, percobaan yang ke 0 merupakan hasil dari percobaan 3.2 sebelumnya, kemudian hasil nilai *C* mendapatkan nilai akurasi yang stagnan pada percobaan 0 hingga 2 dan akhirnya menurun nilai akurasinya ke 85%. Pada nilai *C* sebesar 4.5 mendapatkan peningkatan nilai akurasi yang lebih baik dari pada percobaan sebelumnya yakni 93%.

Hal ini membuktikan jika semakin besar nilai *C* yang digunakan maka optimisasi yang digunakan akan memilih nilai margin yang lebih kecil pada hyperplane jika hal tersebut akan bekerja dengan baik maka akan mendapatkan akurasi yang lebih baik pula. Setelah mendapatkan nilai parameter yang tepat yakni *sigma* 40 dengan *C* sebesar 4.5.

Percobaan Klasifikasi AdaBoost-SVM

Pada tahap ini akan dilakukan uji coba klasifikasi AdaBoost-SVM dengan konfigurasi nilai sigma 40 dan C sebesar 4.5, percobaan dilakukan sebanyak 20 kali sehingga menghasilkan data seperti pada gambar



Gambar 6. Hasil klasifikasi AdaBoost-SVM

Akurasi terbaik pada algoritma HOG dan Adaboost-SVM saat pengujian dilakukan mencapai 96.9% sedangkan untuk rata-rata hasil akurasi mencapai 84.8%.

KESIMPULAN

Dari hasil perancangan hingga pengujian sistem disimpulkan, dengan menggunakan nilai sigma yang besar pada RBFSVM menjadikan klasifikasi tersebut lemah, dan jika menggunakan nilai sigma yang lebih kecil pada RBFSVM menjadikan klasifikasi tersebut kuat. Selain itu AdaBoost yang digunakan untuk *boosting* dengan SVM menggunakan jenis *diverse* Adaboost.

Serta penerapan *boosting* pada SVM menggunakan Adaboost dapat berjalan dengan baik. Hal tersebut dapat dibuktikan dari hasil dari percobaan sebanyak 20 kali dengan menghasilkan tingkat akurasi rata-rata sebesar 85%.

DAFTAR RUJUKAN

- [1] E. Prianto dan N. K. Suryati, "Komposisi Jenis dan Potensi Sumber Daya Ikan di Muara Sungai Musi," *J. Lit. Perikan. Ind. Vol.16 No.1 Maret 2010*, pp. 1-8, 2010.
- [2] M. R. Kumaseh, L. Latumakulita dan N. Nainggolan, "Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding," 2013.
- [3] A. Salman, S. Maqbool, A. H. Khan, A. Jalal dan F. Shafait, "Real-time fish detection in complex backgrounds using probabilistic background modelling," 2019.
- [4] X. Li, M. Shang, H. Qin dan L. Chen, *Fast accurate fish detection and recognition of underwater images with Fast R-CNN*, 2015.
- [5] S. J. Santoso, B. Setiyono dan R. R. Isnanto, "Pengenalan Jenis-Jenis Ikan Menggunakan Metode Analisis Komponen Utama," 2006.
- [6] N. Dalal dan B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005.
- [7] L. Fu, J. Duan, X. Zou, G. Lin, S. Song, B. Ji dan Z. Yang, "Banana detection based on color and texture features in the natural environment," *ScienceDirect*, 2019.
- [8] A. F. Indriani dan M. A. Muslim, "SVM Optimization Based on PSO and AdaBoost to Increasing Accuracy of CKD Diagnosis," 2019.

- [9] B. Y. B. Putranto, W. Hapsari dan K. Wijana, "SEGMENTASI WARNA CITRA DENGAN DETEKSI WARNA HSV UNTUK MENDETEKSI OBJEK," 2011.
- [10] S. I. Syafi'i, R. T. Wahyuningrum dan A. Muntasa, "SEGMENTASI OBYEK PADA CITRA DIGITAL MENGGUNAKAN OTSU THRESHOLDING," *Jurnal Informatika*, 2015.
- [11] N. Otsu, "A Threshold Selection Method from Grayscale Histogram," 1979.
- [12] Y. Freund dan R. E. Schapire, "A Short Introduction to Boosting," 1999.
- [13] F. T. Anggraini dan I. Y. Purbasari, JARINGAN SARAF TIRUAN DAN MODIFIKASINYA MENGGUNAKAN SUPERVISED LEARNING, Sidoarjo: Indomedia Pustaka, 2019.
- [14] P. Melville dan R. J. Mooney, "Creating Diversity In Ensembles Using Artificial Data," 2004.
- [15] X. Li, L. Wang dan E. Sung, "AdaBoost with SVM-based component classifier," 2008.