e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI: http://dx.doi.org/10.30872/jim.v15i2.4648

Deteksi Ikan Dengan Menggunakan Algoritma Histogram of Oriented Gradients

Fetty Tri Anggraeny [1)], Basuki Rahmat [2)], Singgih Putra Pratama [3)]
Program Studi Informatika, Fakultas Ilmu Komputer,
Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia
E-Mail : fettyanggraeny.if@upnjatim.ac.id [1)]; basukirahmat.if@upnjatim.ac.id [2)]; singgihputrap.if@gmail.com [3)]

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 mengidentifikasikan ikan, dapat memanfaatkan sebuah teknologi yang dapat membantu manusia untuk dapat mengenali ikan dengan menggunakan visi komputer dan pendekatan pemrosesan gambar untuk deteksi ikan dan bukan 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 84.8%.

Kata Kunci – Deteksi Ikan, Histogram of Oriented Gradients, Adaboost-SVM

1. 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 (Prianto & Suryanti, 2010) 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 (Kumaseh, Latumakulita, & Nainggolan, 2013). Untuk mempermudah mengidentifikasikan 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 (Salman, Maqbool, Khan, Jalal, & Shafait, 2019) 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 (Li, Shang, Qin, & Chen, 2016)

Untuk melakukan deteksi ikan dibutuhkan sebuah proses pengenalan ikan. Pengenalan ikan merupakan proses mengidentifikasikan ikan berdasarkan gambaran bentuk pola tubuh ikan beserta ciri-cirinya (Santoso, Setiyono, & Isnanto, 2011). 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 (Dalal & Triggs, 2005), 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. Selain itu pada penelitian yang dilakukan oleh (Purbasari, Intan Y Anggraeny & Harianto, 2018), dengan menggunakan HSV sebagai ruang warnanya dan SVM dalam melakukan klasifikasi menghasilkan tingkat akurasi hingga 100%.

Pada penelitian sebelumnya yang dilakukan oleh (Fu et al., 2019), dengan menggunakan HOG ekstraksi ciri dan melakukan perbandingan SVM dan Adaboost menghasilkan tingkat akurasi rata-rata sebesar 90%. Oleh karena itu penelitian ini, akan di bangun sebuah program klasifikasi ikan dan bukan ikan melalui citra, dimana dalam proses ekstraksi ciri menggunakan Histogram of Oriented Gradients (HOG) dan klasifikasi Adaboost dengan Support Vector Machine (Adaboost-SVM).

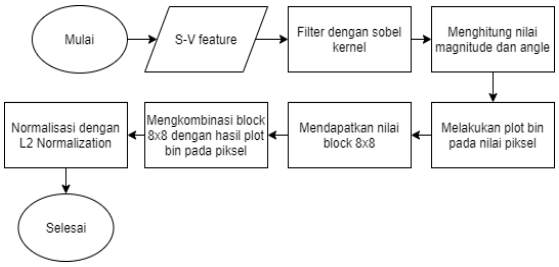
2. TINJAUAN PUSAKA

A. HOG (Histogram of Oriented Gradients)

History of Oriented Gradients (HOG) merupakan sebuah metode yang digunakan pada *image processing* bertujuan untuk deteksi obyek. Teknik ini menghitung nilai gradien dalam daerah tertentu suatu citra. Tiap citra mempunyai karakteristik yang di tunjukkan oleh distribusi gradien. Karakteristik ini diperoleh dengan membagi citra ke dalam daerah kecil yang disebut sel. Tiap sel disusun sebuah histogram dari sebuah gradien dan dari

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI: http://dx.doi.org/10.30872/jim.v15i2.4648

kombinasi gradien tersebut dijadikan sebagai deskripstor yang mewakili sebuah obyek (Permata & Eddy, 2012). Untuk mendapatkan nilai gradien horizontal dan vertikal dapat menggunakan teknik *sobel filter*. Algoritma HOG dapat di lihat pada gambar 1.



Gambar 1. Algoritma *Histogram of Oriented Gradients*

Untuk menghitung nilai gradien magnitude dan gradien arah menggunakan persamaan 1 dan 2 berikut.

$$g = \sqrt{g_x^2 + g_y^2} \quad \text{.....(1)}$$

$$\theta = \arctan\frac{g_y}{g_x} \quad \text{.....(2)}$$

Dimana :
g_x : nilai gradien horizontal
g_y : nilai gradien vertikal
g : arah gradien
θ : besarnya gradien

Langkah selanjutnya adalah pembuatan bin untuk setiap selnya, pembuatan bin dapat dilakukan dengan menggunakan persamaan 3 seperti berikut ini.

$$bin = \frac{9*angle}{2*3.14} \quad \text{.....(3)}$$

Dimana :
angle : variabel hasil arah dari proses tahap sebelumnya

Bentuk sel mempengaruhi pixel yang di ambil, di mana bentuk sel tersebut ditentukan oleh tipe geometri blok. Setelah pembuatan blok selesai selanjutnya adalah melakukan normalisasi terhadap nilai ekstraksi ciri dari HOG dengan menggunakan *L2 Normalization* seperti pada persamaan 4.

$$hist = \frac{hist}{\|hist\|^2} \quad \text{.....(4)}$$

Dimana *hist* merupakan nilai dari histogram yang telah dibuat sebelumnya. Cara kerja L2 norm yakni jika terdapat data vektor warna dengan nilai [128, 64, 32], panjang vektor tersebut adalah $\sqrt{128^2 + 64^2 + 32^2} = 146.64$ kemudian pada setiap elemen dibagi 146.64 maka menghasilkan normalisasi vektor [0.87, 0.43, 0.22] apabila nilai vektor yang pertama dikali 2 jadi [256, 128, 64] dan melalui proses normalisasi tersebut akan menghasilkan nilai yang sama yakni [0.87, 0.43, 0.22].

B. Otsu Thresholding

Metode Otsu merupakan metode yang digunakan dalam segementasi 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 (Syafi'i, Wahyuningrum, & Muntasa, 2016). Sehingga citra dapat di pisahkan antara *foreground* dan *background*. Metode Otsu di perkenalkan oleh (Otsu, 1979).

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 5 yang dapat di artikan sebagai distribusi dari sebuah data. Semakin tinggi nilai dari sebuah *variance* semakin tinggi data yang tersebar.

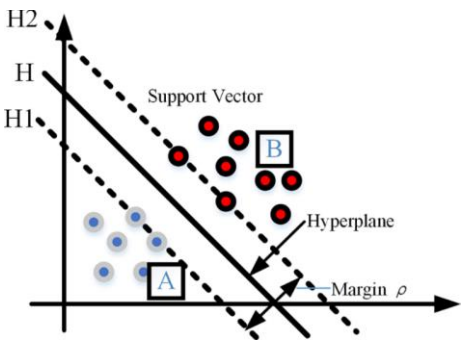
$$\sigma^2 = \frac{\sum_{i=0}^{N}(S_i - \mu)^2}{N} \quad \text{.....(5)}$$

Dimana :
S_i : nilai piksel
μ : mean
N : total piksel

C. SVM (*Support Vector Machine*)

Support Vector Machine merupakan linear *classifier* dan metode yang digunakan dalam pada tahap klasifikasi. Cara kerja SVM berdasarkan pada konsep mencari *hyperlane* terbaik yang berfugnsi sebagai pemisah dua *class*. Konsep dasar dari SVM adalah memaksimalkan *margin* yang merupakan jarak pemisah antar 2 *class*.

Kemudian SVM dikembangkan agar dapat bekerja pada masalah *non*-linear dengan menggunakan konsep kernel *trick* pada ruang kerja berdimensi tinggi (Irawan, Purnomo, & Alamsyah, 2015). Pada umumnya untuk memecahkan sebuah masalah dengan kasus linear dapat menggunakan *hard margin* SVM.



Gambar 2. Contoh cara kerja SVM secara linier (Deng, Cao, Rai, & Gao, 2018)

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI: http://dx.doi.org/10.30872/jim.v15i2.4648

Pada gambar 2 merupakan contoh bagaimana SVM cara bekerja secara linear. Titik yang berada garis putus-putus merupakan *support vector* yang dapat membantu dalam mencari nilai dari sebuah persyaratan persamaan 6 dan terlihat bagaimana nilai w dan b dapat menentukan sebuah *hyperplane.*

$$\min_{w,b} \frac{1}{2} \|w^2\| \text{.....(6)}$$

Dengan syarat:

$$y_i(wx_i + b) \geq 1, \quad i = 1, \ldots m$$

Dimana :

w : nilai dari bobot vektor
b : nilai dari bias
y_i : nilai label yakni 1 dan -1
x_i : nilai data dari ekstraksi fitur

Dikarenakan pemecahan masalah dengan menggunakan *hard margin* SVM tidak dapat memecahkan data yang tidak linear atau data dengan dimensi yang tidak terbatas. Maka menggunakan pemecahan masalah dengan *dual* di mana dengan menggunakan *kernel trick* akan sangat mudah untuk mengklasifikasikan data yang tidak linear. Persamaan *dual* akan lebih mudah dengan menggunakan *Lagrangian duality* (Bottou & Lin, 2006).

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i \text{.....(7)}$$

Dengan syarat:

$$0 \leq \alpha_i \leq C$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Dimana :
α_{ij} : alpha dengan index i atau j

Pada penelitian ini menggunakan kernel *Radial Basis Function* (RBF). Untuk mempermudah perhitungan *Langrangian duality*, dapat menggunakan pemecahana masalah dengan *quadratic programming* dengan persamaan 8.

$$\min_{\alpha} \alpha^T P\, a + q^T \alpha \text{.....(8)}$$

Dengan syarat:
$G\alpha \leq h$
$A\alpha = b$

Dari persamaan *quadratic programming* tersebut akan dapat dilakukan pencarian nilai *support vector* secara mudah. Perhitungan dari sebuah persamaan ini menggunakan library yang bernama CVXOPT.

D. AdaBoost

Algoritma AdaBoost merupakan algoritma yang dapat melakukan *boosting* pada klasifikasi yang lemah dan dapat melakukan penyesuaian terhadap nilai *error* (Freund & Schapire, 1999). Algoritma klasifikasi yang menggunakan metode *boosting* diharapkan performa dari algoritma tersebut dapat meningkat dari pada tanpa menggunakan metode *boosting* (Tri Anggraeny & Yuniar Purbasari, 2019).

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

Pada penelitian yang dilakukan oleh (Pratama, Rahmat, & Anggraeny, 2020) 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 (Melville & Mooney, 2005) 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 (Xuchun Li et al., 2008) berikut ini:

Input: Suatu kumpulan *sample* pelatihan dengan label $\{(X_i, Y_i), \ldots, (X_N, Y_N)\}$, inisialisasi σ, σ_{ini}; minimal σ, σ_{min}; langkah dari σ, σ_{step}; threshold dari *diversity*, *DIV*.

Initialize: Bobot suatu *sample* pelatihan:

$W_t^1 = 1 / N$ .....(9)

untuk semua $i = 1, \ldots, N$.

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 :
   $D_t = \sum_{i=i}^{N} d_t(x_i)$ .....(10)
3. Hitung kesalahan pelatihannya pada h_t :
   $\varepsilon_t = \sum_{t=1}^{N} W_i^t, Y_i \neq h_t(X_i)$ .....(11)
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 :
   $\alpha_t = \frac{1}{2} \ln(\frac{1-\varepsilon_t}{\varepsilon_t})$ .....(12)
6. Update bobot sample pelatihan
   $w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(X_i)\}}{C_t}$ .....(13)
   i = 1,....,N C_t adalah suatu konstanta normalisasi.

Output:

$f(x) = sign(\sum_{t=1}^{T} \alpha_t\, h_t(x))$ .....(14)

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

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI: http://dx.doi.org/10.30872/jim.v15i2.4648

diversity pada klasifikasi di hitung menggunakan persamaan 9 seperti berikut ini:

$$d_t(x_i) = \begin{cases} 0, \ jika\ h_t(x_i) = f(x_i) \\ 1, \ jika\ h_t(x_i) \neq f(x_i) \end{cases} \text{.....(15)}$$

Dimana pada persamaan 9, 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.

3. METODE PENELITIAN

A. 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.

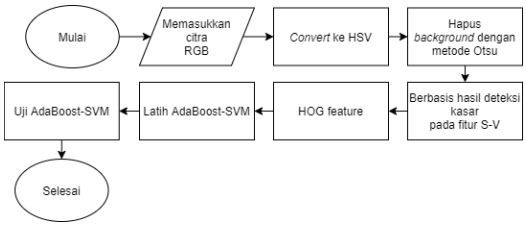
Setelah itu citra akan dibedakan menjadi 2 yakni citra positif (ikan) dan citra negatif (bukan ikan) dengan total 164 citra yang terdiri dari 127 citra positif dan 37 citra negatif. Contoh citra positif dan negatif seperti pada gambar 3 dan gambar 4.

Gambar 3. Citra positif (ikan)

Gambar 4. Citra negatif (bukan ikan)

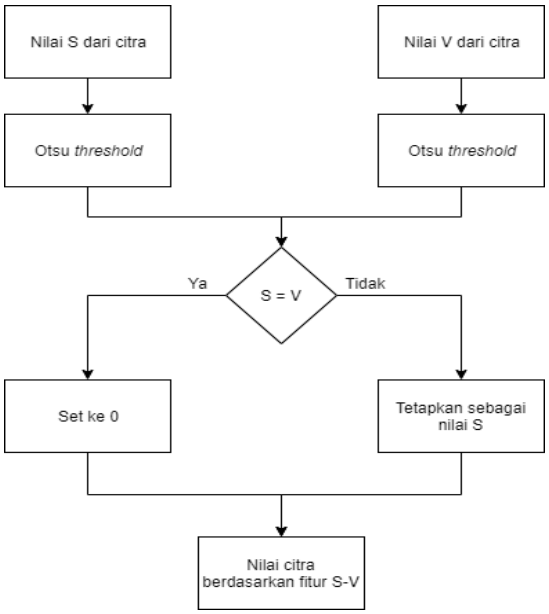
B. Desain Sistem

Bagian ini merupakan Desain sistem dari deteksi ikan dan bukan ikan menggunakan *Histogram of Oriented Gradients* (HOG) dan AdaBoost-SVM. Secara keseluruhan, desain sistem ditampilkan pada gambar 5.



Gambar 5. Desain sistem

Pada gambar 5, 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 6.



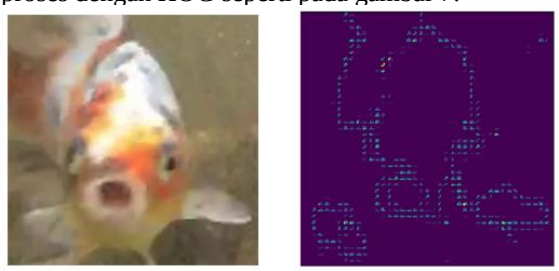
Gambar 6. Fitur s-v

Setelah melewati proses pada gambar 6, 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.

4. 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 C pada SVM. 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.

Dataset dibagi menjadi 2 bagian yakni data latih dan data uji, yang di mana data latih sebanyak 131 dan data uji sebanyak 33. Hasil citra yang telah melakukan proses ektraksi fitur HOG dengan konfigurasi nilai bin 9 dengan ukuran sel sebesar 8x8 menghasilkan sebanyak 1024 nilai fitur, tampilan citra yang telah di proses dengan HOG seperti pada gambar 7.



Gambar 7. Hasil citra dengan ekstaksi fitur HOG

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI: http://dx.doi.org/10.30872/jim.v15i2.4648

A. 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 di awali dengan nilai 100 hingga 20. Setiap percobaan akan dilakukan pengurangan nilai *sigma* sebesar 10. Hasil keseluruhan dari percobaan perubahan sigma ini dapat di lihat 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 *sigma*nya.

B. 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 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 2.

Tabel 2. Hasil percobaan perubahan nilai *C* pada SVM

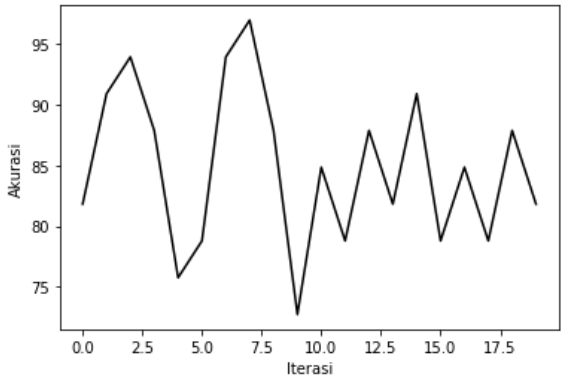
Percobaan ke-	Sigma	C	Akurasi (%)	Precision	Recall
0	40	1	89	0,88	1
1	40	1,5	89	0,88	1
2	40	2	89	0,88	1
3	40	2,5	85	0,875	0,95
4	40	3	85	0,875	0,95
5	40	3,5	85	0,88	0,95
6	40	4	89	0,913	0,95
7	40	4,5	93	0,92	1
8	40	5	93	0,9166	1

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.

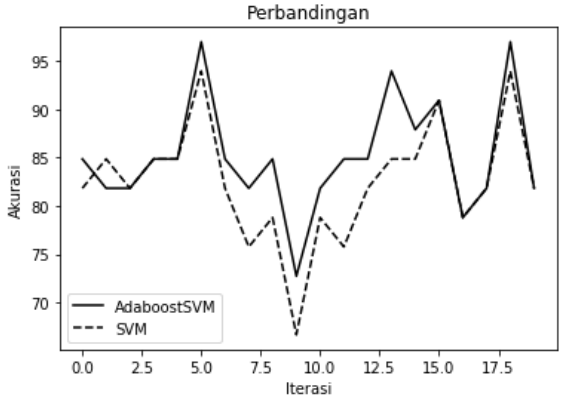
C. Analisa Tingkat Akurasi Klasifikasi

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 gambar 8.



Gambar 8. Hasil klasifikasi ikan dan bukan ikan dengan 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%. sedangkan untuk perbandingan dengan tanpa menggunakan Adaboost menghasilkan hasil seperti pada gambar 9.



Gambar 9. Perbandingan Adaboost-SVM dengan RBFSVM

Perbandingan tersebut dapat terlihat apabila tanpa menggunakan teknik *boosting* pada klasifikasi SVM menghasilkan nilai akurasi yang kurang memuaskan. Sehingga teknik Adaboost dapat membantu klasifikasi untuk meningkatkan hasil akurasinya.

e-ISSN 2597-4963 dan p-ISSN 1858-4853 DOI: http://dx.doi.org/10.30872/jim.v15i2.4648

5. KESIMPULAN

Berdasarkan 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. 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 84.8%.

DAFTAR PUSTAKA

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. https://doi.org/10.1109/CVPR.2005.177

Deng, Z., Cao, M., Rai, L., & Gao, W. (2018). A two-stage classification method for borehole-wall images with support vector machine. *PLoS ONE*, *13*(6), 1–19. https://doi.org/10.1371/journal.pone.0199749

Freund, Y., & Schapire, R. E. (1999). A short introduction to bosting. *Journal of Japanese Society for Artificial Intelligence*, *14*(5), 771–780. Retrieved from https://cseweb.ucsd.edu/~yfreund/papers/IntroToBoosting.pdf

Fu, L., Duan, J., Zou, X., Lin, G., Song, S., Ji, B., & Yang, Z. (2019). Banana detection based on color and texture features in the natural environment. *Computers and Electronics in Agriculture*, *167*(July), 105057. https://doi.org/10.1016/j.compag.2019.105057

Irawan, F., Purnomo, A., & Alamsyah, D. (2015). Deteksi Mobil pada Citra Digital Menggunakan C-HOG dan Support Vector Machine. *GI MDP & MDP BUSINESS Journal*, (x), 1–12.

Kumaseh, M. R., Latumakulita, L., & Nainggolan, N. (2013). Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding. *Jurnal Ilmiah Sains*, *13*(1), 74. https://doi.org/10.35799/jis.13.1.2013.2057

Li, X., Shang, M., Qin, H., & Chen, L. (2016). Fast accurate fish detection and recognition of underwater images with Fast R-CNN. *OCEANS 2015 - MTS/IEEE Washington*, 1–5. https://doi.org/10.23919/oceans.2015.7404464

Melville, P., & Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. *Information Fusion*, *6*(1), 99–111. https://doi.org/10.1016/j.inffus.2004.04.001

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Permata, C., & Eddy, I. K. (2012). *Deteksi Mobil Menggunakan Histogram of Oriented Gradient*. Institut Teknologi Sepuluh Nopember. Retrieved from http://repository.its.ac.id/3084/

Pratama, S. P., Rahmat, B., & Anggraeny, F. T. (2020). Deteksi Ikan Dengan Menggunakan Algoritma Adaboost-SVM. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, *1*(2), 458–466. Retrieved from http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/92/64

Prianto, E., & Suryanti, N. K. (2010). Komposisi jenis dan potensi sumber daya ikan di muara Sungai Musi. *Jurnal Penelitian Dan Perikanan Indonesia*, *16*(1), 1–8. Retrieved from file:///C:/Users/User/Downloads/3393-8613-1-SM.pdf

Purbasari, Intan Y Anggraeny, F. T., & Harianto, N. (2018). Classification of broiler chicken eggs using support vector machine (svm) and feature selection algorithm. *ICITB Proceedings*.

Salman, A., Maqbool, S., Khan, A. H., Jalal, A., & Shafait, F. (2019). Real-time fish detection in complex backgrounds using probabilistic background modelling. *Ecological Informatics*, *51*(February), 44–51. https://doi.org/10.1016/j.ecoinf.2019.02.011

Santoso, S. J., Setiyono, B., & Isnanto, R. R. (2011). *Pengenalan jenis-jenis ikan menggunakan metode analisis komponen utama*. Universitas Diponegoro. Retrieved from http://eprints.undip.ac.id/25746/

Syafi'i, S. I., Wahyuningrum, R. T., & Muntasa, A. (2016). Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding. *Jurnal Informatika*, *13*(1), 1–8. https://doi.org/10.9744/informatika.13.1.1-8

Tri Anggraeny, F., & Yuniar Purbasari, I. (2019). *Jaringan Syaraf Tiruan dan Modifikasinya Menggunakan Supervised Learning*. Surabaya: Indomedia Pustaka.