

PENERAPAN METODE SUPPORT VECTOR MACHINE PADA APLIKASI DETEKSI TINGKAT KESEGARAN IKAN

Maria Carolina¹, Sandi Tendean², Krisyesika³

¹²Informatika³Bisnis Digital, Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak
e-mail: ¹20421389_maria_c@widyadharm.ac.id, ²sandi_t@widyadharm.ac.id, ³krisyesika@widyadharm.ac.id

Abstract

Indonesia is the second largest producer of marine fish in the world based on data from the Food and Agriculture Organization 2022. Fish is a food that contains excellent protein, fat, vitamins, and minerals. According to data from the Ministry of Marine Affairs and Fisheries, the national fish consumption rate continues to increase, reaching 57.27 kg/capita in 2022. Therefore, it is important to know which fish are fresh and suitable for consumption. In this research, fish freshness detection will be carried out using the Support Vector Machine (SVM) method. The detection process will be carried out using three classes, namely fresh, not fresh, and rotten. SVM is a linear classifier, but it is developed to work on nonlinear data with the concept of kernels in a high-dimensional workspace. The results of testing the SVM model show that the accuracy of the model with the Radial Basic Function (RBF) kernel produces 99 percent accuracy with parameters $C = 10$ and $\gamma = 0.5$. Application testing using two smartphones with four tests resulted in 92 percent accuracy.

Keywords: Support Vector Machine, Fish Freshness Detection, Web, Machine Learning

Abstrak

Indonesia merupakan negara dengan peringkat kedua penghasil ikan laut terbesar di dunia berdasarkan data dari Food and Agriculture Organization 2022. Ikan merupakan bahan pangan yang mengandung protein, lemak, vitamin, dan mineral yang sangat baik. Menurut data Kementerian Kelautan dan Perikanan, Angka Konsumsi Ikan nasional terus meningkat, mencapai 57,27 kg/kapita pada tahun 2022. Oleh karena itu, penting untuk mengetahui ikan yang segar dan layak dikonsumsi. Pada penelitian ini akan dilakukan deteksi kesegaran ikan menggunakan metode Support Vector Machine. Proses deteksi akan dilakukan pada tiga kelas, yaitu segar, tidak segar, dan busuk. Support Vector Machine adalah linear classifier, namun dikembangkan agar dapat bekerja pada data nonlinear dengan konsep kernel pada ruang kerja yang berdimensi tinggi. Hasil pengujian terhadap model Support Vector Machine menunjukkan bahwa akurasi model dengan kernel Radial Basic Function menghasilkan akurasi 99 persen dengan parameter $C = 10$ dan $\gamma = 0,5$. Pengujian aplikasi menggunakan dua smartphone dengan empat kali pengujian menghasilkan akurasi 92 persen.

Kata kunci: Support Vector Machine, Deteksi Kesegaran Ikan, Web, Machine Learning

1. PENDAHULUAN

Indonesia merupakan negara kepulauan berbentuk Republik, terletak di kawasan Asia Tenggara. Indonesia memiliki kurang lebih 17.000 buah pulau dengan luas perairan 3.257.483 km² dan luas daratan 1.922.570 km². Berdasarkan data dari *Food and Agriculture Organization 2022*, Indonesia merupakan negara dengan peringkat kedua penghasil ikan laut terbesar di dunia. Pada data ditahun 2020, Indonesia menghasilkan 6,43 juta ton ikan laut. Ikan merupakan sumber protein, lemak, vitamin, dan mineral yang sangat baik.

Menurut Kementerian Kelautan dan Perikanan, Angka Konsumsi Ikan (AKI) nasional terus meningkat, mencapai 57,27 kg/kapita pada tahun 2022. Angka ini naik dari tahun 2021 yaitu sebesar 55,16 kg/kapita. Ikan mengandung omega 3 yang baik untuk perkembangan otak, serta kalsium, vitamin D, dan fosfor yang baik untuk tulang. Namun, ikan merupakan bahan pangan yang mudah rusak karena kandungan air yang tinggi, sehingga penting bagi masyarakat untuk mengetahui ciri-ciri ikan segar dan layak untuk dikonsumsi.

Pada kehidupan sehari-hari, masih banyak masyarakat yang belum dapat membedakan ikan segar dan ikan tidak segar. Biasanya, konsumen dapat melihat kondisi ikan berdasarkan ciri fisiknya seperti insang, mata, kulit, tekstur daging dan lendir ikan. Namun, tidak semua orang mengetahui dan memahami ciri tersebut. Selain itu, cara manual ini memberikan hasil yang berbeda-beda pada setiap orang. Untuk mempermudah mengidentifikasi tingkat kesegaran ikan, dapat memanfaatkan sebuah teknologi. Salah satu teknologi yang berperan penting adalah *computer vision*.

Computer vision merupakan salah satu cabang ilmu pengetahuan *artificial intelligence* yang mempelajari bagaimana penglihatan manusia dapat diterapkan pada komputer. *Artificial intelligence* (AI) merupakan salah satu teknologi yang mampu meniru perilaku manusia dan kemudian ditanamkan kedalam komputer maupun program. AI memiliki cabang ilmu yang membantu menyempurnakan penerapan dari AI ini sendiri, salah satunya adalah *machine learning*. *Machine Learning* adalah ilmu yang mempelajari tentang algoritma komputer yang dapat mengenali pola-pola di dalam data.

Dalam *machine learning* terdapat metode yang dapat digunakan sesuai dengan cara belajar dari komputer untuk menjalankan tugasnya. Metode yang dapat digunakan salah satunya adalah metode *Support Vector Machine* (SVM). SVM adalah jenis metode klasifikasi *supervised*, yaitu metode dengan model yang mampu untuk memprediksi berdasarkan kasus yang sebelumnya sudah diketahui. Metode SVM melibatkan dua jenis versi yaitu versi linier dan non linier. Versi linier, *hyperplanes* atau himpunan *hyperplanes* digunakan untuk pemisahan kelas. Pada versi non linier, kelas bukan partisi sehingga tidak ada garis lurus yang memisahkan kelas. Dengan bantuan vektor dan margin, *support vector machine* dapat menemukan *hyperplane*.

Berdasarkan permasalahan yang ada, penulis melakukan penelitian dengan judul “Penerapan metode *Support Vector Machine* pada aplikasi deteksi tingkat kesegaran ikan”. Aplikasi ini diharapkan mampu mengidentifikasi tingkat kesegaran ikan melalui citra digital mata ikan dengan menerapkan metode SVM.

2. METODE PENELITIAN

2.1 Teknik Pengumpulan Data

Metode pengumpulan data yang digunakan adalah studi literatur berupa jurnal ilmiah dan buku serta melakukan observasi.

2.2 Teknik Analisis dan Perancangan Sistem

Teknik analisis dan perancangan yang digunakan adalah teknik berorientasi objek menggunakan bahasa pemodelan *Unified Modelling Language* (UML).

2.3 Aplikasi Perancang Sistem

Aplikasi yang digunakan untuk merancang sistem adalah StarUML sebagai aplikasi untuk melakukan pemodelan, Visual Studio Code sebagai kode editor, Python sebagai bahasa pemrograman, dan Streamlit sebagai *framework* tampilan web.

2.4 Landasan Teori

2.4.1 Citra Digital

Citra digital merupakan imitasi benda tiga dimensi yang direpresentasikan dalam bentuk dua dimensi melalui kombinasi garis, titik, bentuk, dan warna. Pada teknologi digital, citra merupakan kombinasi piksel yang membuat warna sehingga membentuk imitasi objek dan membuat tekstur tertentu^[1].

2.4.2 RGB

Salah satu ruang warna yang paling umum digunakan adalah RGB. Diruang ini, ada tiga saluran: merah (R), hijau (G), dan biru (B). Setiap warna dalam ruang ini dibuat menggunakan kombinasi dari tiga saluran ini. Kombinasi warna dibuat berdasarkan campuran warna aditif, yang merupakan cara kerja cahaya. Dimulai dengan permukaan hitam, cahaya diproyeksikan ke atasnya, memperoleh warna yang berbeda^[2].

2.4.3 HSV

HSV merupakan sebuah ruang warna yang dianggap paling dekat dalam mendeskripsikan warna oleh mata manusia daripada RGB. Pada sistem ruang warna HSV, berlaku jika nilai R, G, dan B sama maka warna akan menjadi keabuan; yaitu putih, intensitas putih^[3].

2.4.4 Support Vector Machine

Metode *Support Vector Machine* (SVM) adalah metode yang relatif baru untuk memprediksi kasus regresi atau klasifikasi^[4]. Secara konsep kerja, baik data *linear* maupun *nonlinear*, memiliki konsep kerja SVM yang sama. Namun untuk memperoleh *hyperplane* yang tepat, data *nonlinear* perlu dibantu dengan kernel SVM. Kernel adalah sebuah cara yang digunakan untuk memetakan data *nonlinear* berdimensi rendah dan mengubahnya ke dalam ruang dimensi yang lebih tinggi. Tujuannya untuk memudahkan pengklasifikasian data dengan menemukan *hyperplane* yang dapat memisahkan *dataset* secara *linear* dengan baik. Berikut jenis kernel pada SVM beserta persamaannya^[5]:

$$\text{Kernel Linear:} \quad K(x, x_i) = \text{sum}(x * x_i) \quad (1)$$

$$\text{Kernel Polinomial:} \quad K(x, x_i) = 1 + \text{sum}(x * x_i)^d \quad (2)$$

$$\text{Kernel RBF (Radial Basic Function):} \quad K(X, X_i) = \exp(-\gamma \|X - X_i\|^2) \quad (3)$$

2.4.5 Website

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara, dan video atau gabungan dari semua yang terkoneksi melalui internet, sehingga dapat diakses oleh seluruh ataupun siapapun orang yang bisa terkoneksi jaringan internet^[6].

2.4.6 Unified Modeling Language

Unified Modeling Language (UML) adalah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak dengan menggunakan teknik pemrograman berorientasi objek^[7].

2.4.7 Confusion Matrix

Confusion matrix memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* membandingkan hasil kerja antara model dengan hasil akhir sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya ketahui^[8].

Perhitungan yang digunakan untuk menilai performa sebuah klasifikasi yaitu:

Akurasi adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual. Jika nilai akurasi tinggi, maka sebuah sistem akan semakin bagus dalam melakukan prediksi. Nilai akurasi dapat diperoleh dengan persamaan 4 sebagai berikut:

$$\text{Akurasi} = \frac{\text{Prediksi data benar}}{\text{Total data}} \tag{4}$$

Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban sistem. Nilai presisi dapat diperoleh dengan persamaan 5 sebagai berikut:

$$\text{Presisi} = \frac{TP}{TP+FP} \tag{5}$$

Recall adalah salah satu perhitungan keakuratan prediksi yang digunakan sebagai ukuran tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Nilai *recall* dapat diperoleh dengan persamaan 6 sebagai berikut:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{6}$$

F1-score adalah pengaruh relatif hasil kombinasi nilai presisi dengan nilai *recall*. *F1-score* dapat digunakan untuk mengukur kinerja dari sistem klasifikasi yang merupakan rata-rata harmonis dari presisi dan *recall*. Nilai *f1-score* dapat diperoleh dengan persamaan 7 sebagai berikut^[9].

$$F1\text{-score} = 2 \times \frac{(\text{recall} \times \text{precision})}{(\text{recall} + \text{precision})} \tag{7}$$

3. HASIL DAN PEMBAHASAN

3.1 Gambaran Umum Alur Penelitian

Penelitian ini melalui tiga tahapan yaitu: proses pengumpulan data, analisa kebutuhan, dan pemodelan.

3.1.1 Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan data primer yaitu data yang didapat dari sumber pertama. *Dataset* yang digunakan adalah pengambilan citra ikan dengan format jpg/jpeg menggunakan kamera *smartphone* Iphone X. Data yang dikumpulkan adalah data ikan yang berada di Pasar Flamboyan, Kota Pontianak, Kalimantan Barat. Data yang digunakan dalam penelitian ini adalah citra mata dari ikan kembung, ikan nila merah, dan ikan dencis. Pada masing-masing jenis ikan, terdapat tiga kelas yaitu segar, tidak segar, dan busuk. Data yang diperoleh secara keseluruhan dari tiga jenis ikan berjumlah 900 data. Rincian *dataset* ditunjukkan pada Tabel 1.

Tabel 1. Rincian Dataset

Jenis Ikan	Jumlah Keseluruhan Gambar	Jumlah Gambar Per Kelas		
		Segar	Tidak Segar	Busuk
Ikan Nila Merah	300	100	100	100
Ikan Kembung	300	100	100	100
Ikan Dencis	300	100	100	100

3.1.2 Pemodelan

Pemodelan merupakan tahap untuk mempersiapkan model yang akan digunakan dalam *machine learning* sebelum melakukan klasifikasi. Alur pemodelan terdiri dari *input* citra, *preprocessing*, ekstraksi fitur, pelabelan, standarisasi data, *training*, dan klasifikasi SVM.

3.1.2.1 Input Citra

Input citra adalah tahap yang dilakukan untuk meng-*input* data citra mata ikan ke dalam sistem untuk diproses. Citra yang akan di-*input* adalah data citra mata ikan yang sudah dikumpulkan pada tahap pengumpulan data dengan spesifikasi sebagai berikut: format jpg/jpeg, png, heic/heif dan 96 DPI.

3.1.2.2 Preprocessing

Preprocessing dilakukan untuk mempersiapkan citra sebelum melakukan klasifikasi. *Preprocessing* citra dilakukan dengan dua tahap, yaitu *cropping* dan *resize* sebagai berikut:

a. Cropping Citra

Cropping citra dilakukan bertujuan agar mempermudah proses deteksi yang hanya berfokus kepada mata ikan. *Cropping* citra mata ikan dilakukan secara manual dengan ukuran *pixel* yang berbeda-beda, contoh *cropping* pada citra mata ikan ditunjukkan pada Tabel 2.

Tabel 2. Contoh Cropping Pada Citra Mata Ikan

Citra mata ikan segar			
	(a) Ikan Kembang; (b) Ikan Nila Merah; (c) Ikan Dencis		
Citra mata ikan tidak segar			
	(a) Ikan Kembang; (b) Ikan Nila Merah; (c) Ikan Dencis		
Citra mata ikan busuk			
	(a) Ikan Kembang; (b) Ikan Nila Merah; (c) Ikan Dencis		

b. Resize Citra

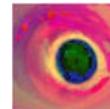
Resize citra adalah tahap mengubah ukuran piksel citra menjadi lebih kecil agar mempercepat komputer dalam melakukan proses pengolahan pada citra mata ikan. Pada tahap *cropping* citra, dijelaskan bahwa proses *cropping* dilakukan secara manual dengan ukuran citra yang berbeda-beda. *Resize* citra bertujuan untuk mengubah ukuran citra menjadi lebih kecil dan seragam. Pada penelitian ini, peneliti mengubah ukuran citra mata ikan menjadi 256 x 256 *pixel*.

3.1.2.3 Ekstraksi Fitur

Ekstraksi fitur dilakukan setelah citra mata ikan melalui tahap *preprocessing*. Fitur yang diekstraksi berupa fitur warna. Ekstraksi warna merupakan suatu tahap dimana komputer membaca data ciri dari citra ikan berdasarkan kombinasi warna. Mata ikan akan mengalami perubahan warna jika dibiarkan seharian pada suhu ruang. Ekstraksi warna menggunakan ruang warna *Red, Green, Blue* (RGB) dan *Hue, Saturation, Value* (HSV). Ekstraksi ruang warna RGB dimulai dari mengambil nilai matriks dari masing-masing saluran warna R, G, dan B, kemudian akan dihitung rata-rata dan standar deviasi.. Ekstraksi ruang warna HSV dimulai dari mengambil nilai matriks dari masing-masing parameter H, S, dan V, kemudian akan dihitung rata-rata dan standar deviasinya. Hasil perhitungan ekstraksi ruang warna RGB dan HSV dirangkum ke dalam Tabel 3 dan Tabel 4. Sampel gambar yang digunakan dalam melakukan perhitungan ruang warna RGB dan HSV ditunjukkan pada Gambar 1 dan Gambar 2.



Gambar 1. Sampel Citra Ikan Segar



Gambar 2. Sampel Citra Ikan Segar HSV

Tabel 3. Rata-rata dan Standar Deviasi RGB

Rata-rata			Standar Deviasi		
R	G	B	R	G	B
179,419769	138,297623	121,415604	69,486936	54,421512	55,036132

Tabel 4. Rata-rata dan Standar Deviasi HSV

Rata-rata			Standar Deviasi		
H	S	V	H	S	V
107,196045	86,394012	183,420059	34,126353	43,139990	66,392869

3.1.2.4 Pelabelan

Pelabelan data dilakukan untuk mengidentifikasi ikan segar, tidak segar, dan busuk berdasarkan fitur warna. Data pelabelan akan dimasukkan ke dalam *dataframe*. Pelabelan dilakukan menggunakan Python.

3.1.2.5 Standarisasi Data

Standarisasi data adalah salah satu tahap yang penting dilakukan pada saat mempersiapkan model *machine learning*, meningkatkan performa model. Standarisasi data dilakukan dengan menggunakan *StandardScaler* dari *library* Scikit-learn Python.

3.1.2.6 Training

Tahap pelatihan data melibatkan pembagian *dataset* menjadi data latih dan data uji dengan ukuran data uji sebesar 30 persen dari total dataset dan *random state* 42 untuk mendapatkan akurasi terbaik. Dalam penelitian ini, setiap jenis ikan memiliki 300 data, sehingga total dataset mencapai 900 data setelah menggabungkan ketiga *dataframe*.

3.1.2.7 Klasifikasi SVM

Tahap klasifikasi dilakukan setelah ekstraksi fitur, di mana citra mata ikan dikonversi ke dalam *dataframe* dan melalui tahap *training* data. Model SVM akan dilatih menggunakan kernel *Radial Basic Function*

(RBF) dan parameter *default*. Peneliti akan menjelaskan proses klasifikasi SVM dengan perhitungan manual, menggunakan metode klasifikasi *multiclass one-vs-rest*. Tahap klasifikasi meliputi *input* citra, menghitung kernel RBF, mengoptimalkan fungsi dualitas *lagrange multiplier*, dan menemukan fungsi pemisah (*hyperplane*).

a. Input Data

Input data dilakukan dengan menggunakan data satu sampel dari ketiga kelas. Sampel data masing-masing kelas ditunjukkan pada Tabel 5.

Tabel 5. Sampel Data Masing-Masing Kelas

Fitur	Kelas Satu	Kelas Dua	Kelas Tiga
Mean r	0,856480057	0,705054428	-0,062572606
Mean g	0,932068681	0,862761605	0,177239317
Mean b	0,813554086	0,319898339	0,110325486
Stddev r	1,051275349	1,293444879	0,253080278
Stddev g	0,710928872	1,119696832	0,291232494
Stddev b	1,282952505	0,966952965	0,449677255
Mean h	-0,006400187	-0,818692014	-0,648408822
Mean s	-0,809351006	-0,02959037	-0,35559483
Mean v	0,771885825	0,664131417	-0,16491335
Stddev h	0,760387484	0,443541018	0,701563077
Stddev s	0,585167211	0,180058309	1,289772859
Stddev v	1,445049993	1,607634229	0,674485112

b. Menghitung Kernel RBF

Berdasarkan data pada Tabel 5, maka didapatkan data untuk masing-masing kelas sebagai berikut.

- 1) Kelas Satu = (0.856, 0.932, 0.814, 1.051, 0.711, 1.283, -0.006, -0.809, 0.772, 0.760, 0.585, 1.445)
- 2) Kelas Dua = (0.705, 0.863, 0.320, 1.293, 1.120, 0.967, -0.819, -0.030, 0.664, 0.444, 0.180, 1.608)
- 3) Kelas Tiga = (-0.063, 0.177, 0.110, 0.253, 0.291, 0.450, -0.648, -0.356, -0.165, 0.702, 1.290, 0.674)

Setelah didapatkan data untuk masing-masing kelas, akan dilakukan perhitungan jarak $\|X - X_i\|$ pada setiap kelas dan hasilnya ditunjukkan pada Tabel 6.

Tabel 6. Hasil Perhitungan $\|X - X_i\|$

$\ X_1 - X_2\ $	$\ X_1 - X_3\ $	$\ X_2 - X_3\ $	$\ X_1 - X_1\ $	$\ X_2 - X_2\ $	$\ X_3 - X_3\ $
1.472	2.451	2.476	0	0	0

Setelah menghitung jarak $\|X - X_i\|$, akan dilakukan perhitungan kernel RBF dengan $\gamma = 0,1$ disetiap pasangan K , menggunakan rumus Persamaan 3. Hasil perhitungan ditunjukkan pada matriks K berikut:

$$K = \begin{bmatrix} 1 & 0,8051 & 0,5456 \\ 0,8051 & 1 & 0,5417 \\ 0,5456 & 0,5417 & 1 \end{bmatrix}$$

Setelah menghitung matriks K , maka dilakukan perhitungan terhadap y . Nilai y merupakan nilai dari label atau nilai dari kelas yang sudah diberikan. *Dataset* yang digunakan pada penelitian ini merupakan *dataset multiclass*, sehingga menggunakan SVM *one-vs-rest*, yaitu satu kelas *vs* seluruh kelas. Kelas yang ditentukan terdapat pada pada Tabel 7.

Tabel 7. Nilai Label Pada Y

y_1	y_2	y_3
-1	1	1

Perhitungan dilakukan terhadap $y_i y_j$ sebanyak data. Karena data memiliki tiga kelas dan matriks kernel K yang terbentuk menjadi matriks 3×3 , maka perhitungan y juga berbentuk matriks 3×3 . Setelah menghitung $y_i y_j$, maka terbentuklah matriks y sebagai berikut.

$$y_i y_j = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

c. Mengoptimalkan Fungsi Dualitas Lagrange Multiplier

Setelah mendapatkan matriks kernel RBF, tahap selanjutnya adalah mengoptimalkan fungsi dualitas *lagrange multiplier*. Dualitas *lagrange multiplier* berguna memastikan nilai atau harga dari suatu fungsi yang dibatasi oleh suatu keadaan untuk mencari minimum relatif ataupun maksimum. Matriks K merupakan hasil perhitungan dari $K(X, X_i) = \exp(-\gamma \|X - X_i\|^2)$. Matriks tersebut akan berkorelasi dengan $\alpha_i \alpha_j$ dalam Persamaan 8. Persamaan 8 menggunakan kernel K sebagai pengganti *dot-product* $x_i x_j$. Maksimalkan:

$$Ld = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{8}$$

$$Ld \max = (\alpha_1 + \alpha_2 + \alpha_3) - \frac{1}{2} (((1)(-1)(-1)\alpha_1 \alpha_1) + ((0.8051)(-1)(1)\alpha_1 \alpha_2) + ((0.5456)(-1)(1)\alpha_1 \alpha_3) + ((0.8051)(1)(-1)\alpha_2 \alpha_1) + ((1)(1)(1)\alpha_2 \alpha_2) + ((0.5417)(1)(1)\alpha_2 \alpha_3) + ((0.5456)(1)(-1)\alpha_3 \alpha_1) + ((0.5417)(1)(1)\alpha_3 \alpha_2) + ((1)(1)(1)\alpha_3 \alpha_3)$$

Syarat 1: $\alpha_1 - \alpha_2 - \alpha_3 = 0$, Syarat 2: $\alpha_1, \alpha_2, \alpha_3 \geq 0$

Untuk *dataset* dengan jumlah besar, masalah dualitas optimasi pada Persamaan 8 dapat diselesaikan dengan metode numerik seperti *Quadratic Programming*. Dengan bantuan perangkat lunak, maka didapatkan nilai alpha sebagai berikut.

$$\alpha_1 = 1, \alpha_2 = 0,77, \alpha_3 = 0,22, b = 0,7486$$

d. Menemukan Fungsi Pemisah (Hyperplane)

Hasil perhitungan α dan b menunjukkan bahwa data *training* adalah *support vector*. Hal ini karena nilai $\alpha > 0$. Nilai b didapatkan dari bantuan proses perhitungan menggunakan perangkat lunak. Setelah menemukan nilai α dan b , maka model SVM siap digunakan untuk menggunakan fungsi pemisah (*hyperplane*) persamaan 9.

$$f(x) = \sum_{i=0}^N \alpha_i y_i K(x_i, x) \cdot x + b \tag{9}$$

Dimana N merupakan jumlah *support vector* dan K merupakan fungsi kernel.

3.1.2.8 Hyperparameter Tuning

Tahap ini adalah proses penyesuaian *hyperparameter* algoritma *machine learning* untuk meningkatkan kinerja model. Setelah melalui proses *training* SVM menggunakan parameter *default*, maka model akan kembali dilatih menggunakan parameter lain untuk membuat perbandingan antara performa masing-masing parameter. Kernel RBF akan di-*set* menggunakan nilai $C = [1, 10, 100, 1000]$ dan $\gamma = [0.01, 0.1, 0.2, 0.5]$. Penentuan nilai pada C dan γ adalah kombinasi antara nilai *default* dengan nilai yang ditentukan secara *random* untuk mencari parameter dengan akurasi terbaik. *Grid search* akan melatih SVM dengan kombinasi dari masing-masing nilai parameter pada kernel. Setelah model dilatih dan dievaluasi menggunakan *k-fold cross-validation*, maka didapatkan model dengan skor validasi terbaik, yaitu model kernel RBF dengan *mean score* tertinggi yaitu 0,98; nilai parameter $C = 10$; dan $\gamma = 0,5$.

3.2 Evaluasi

Evaluasi berguna untuk melihat akurasi model yang telah dilatih dan sudah melalui tahap *hyperparameter tuning*. Jika pada tahap sebelumnya peneliti hanya menggunakan satu sampel data, maka pada tahap evaluasi peneliti akan menggunakan seluruh *data testing*. Evaluasi yang digunakan adalah *confusion matrix*. Perhitungan *accuracy* dilakukan menggunakan Persamaan 4.

$$Accuracy = \frac{\text{Prediksi data benar}}{\text{Total data}} = \frac{265}{270} = 0,98 = 98 \%$$

Setelah melakukan perhitungan *accuracy* terhadap keseluruhan *data testing*, maka dilakukan perhitungan terhadap *precision*, *recall*, dan *f1-score* pada setiap kelas menggunakan Persamaan 5, 6, dan 7. Hasil perhitungan ditunjukkan pada Tabel 8.

Tabel 8. Hasil Perhitungan Precision, Recall, dan F1-Score

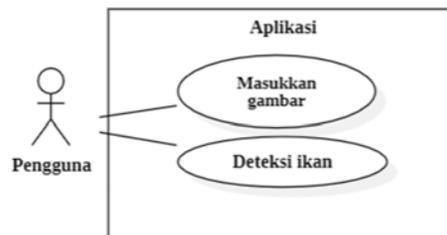
Kelas Satu (Segar)			Kelas Dua (Tidak Segar)			Kelas Tiga (Busuk)		
Precision (%)	Recall (%)	F1-Score (%)	Precision (%)	Recall (%)	F1-Score (%)	Precision (%)	Recall (%)	F1-Score (%)
100	97	98	94	100	97	100	98	99

3.3 Perancangan Unified Modeling Language (UML)

Dalam perancangan aplikasi deteksi tingkat kesegaran ikan, pemodelan yang digunakan adalah Unified Modeling Language (UML) untuk memberikan gambaran mengenai aplikasi yang akan dibangun menggunakan beberapa diagram seperti *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

3.3.1 Use Case Diagram

Pada Gambar 3 ditunjukkan diagram *use case* aplikasi deteksi kesegaran ikan. *Diagram use case* merupakan diagram yang menggambarkan secara keseluruhan antara suatu sistem dengan pengguna. Pada gambar diagram *use case* tersebut, pengguna dapat melakukan dua aktivitas yaitu masukkan gambar dan deteksi ikan.



Gambar 3. Use Case Diagram

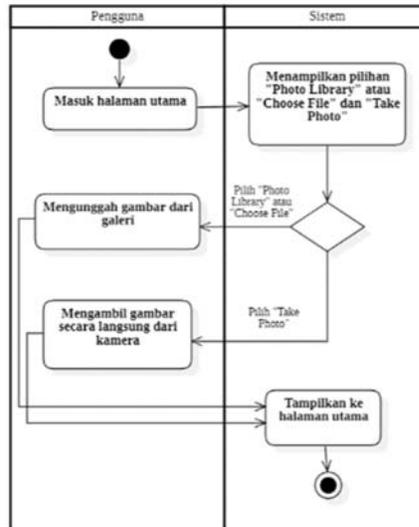
Tabel 9. Penjelasan Use Case

No	Use Case	Keterangan
1	Masukkan gambar	Proses <i>input</i> gambar ke dalam aplikasi
2	Deteksi ikan	Proses deteksi ikan menggunakan klasifikasi SVM

3.3.2 Activity Diagram

3.3.2.1 Activity Diagram Unggah Gambar

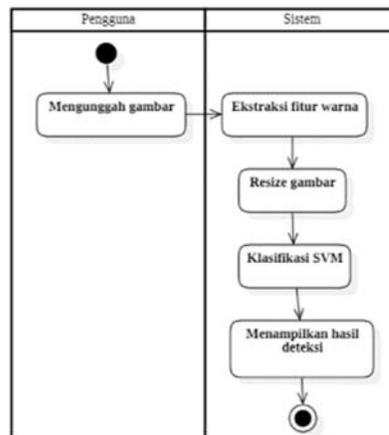
Pada Gambar 4 ditunjukkan *activity diagram* atau diagram aktivitas masukkan gambar. Pada *activity diagram* tersebut, pengguna akan masuk ke halaman utama, kemudian pilihan *photo library* atau *choose file* dan *take photo* akan ditampilkan. Jika pengguna memilih opsi *photo library* atau *choose file*, maka sistem akan menampilkan gambar yang berada di galeri, kemudian pengguna memilih gambar yang ingin diunggah dan sistem akan menampilkan gambar ke halaman utama. Jika pengguna memilih opsi *take photo*, maka sistem akan membuka kamera, kemudian pengguna mengambil gambar menggunakan kamera dan sistem akan menampilkan gambar ke halaman utama.



Gambar 4. Activity Diagram Masukkan Gambar

3.3.2.2 Activity Diagram Deteksi Ikan

Pada Gambar 5 ditunjukkan *activity diagram* atau diagram aktivitas deteksi ikan. Pada *activity diagram* tersebut, pengguna akan mengunggah gambar, kemudian sistem akan melakukan ekstraksi fitur warna terhadap gambar yang telah diunggah. Ekstraksi fitur warna yang digunakan adalah ekstraksi ruang warna RGB dan HSV. Setelah melakukan ekstraksi fitur warna, sistem melakukan *resize* terhadap gambar agar mempermudah proses deteksi. Setelah melakukan *resize* gambar, sistem akan melakukan klasifikasi SVM dan menampilkan hasil deteksi ke halaman utama.

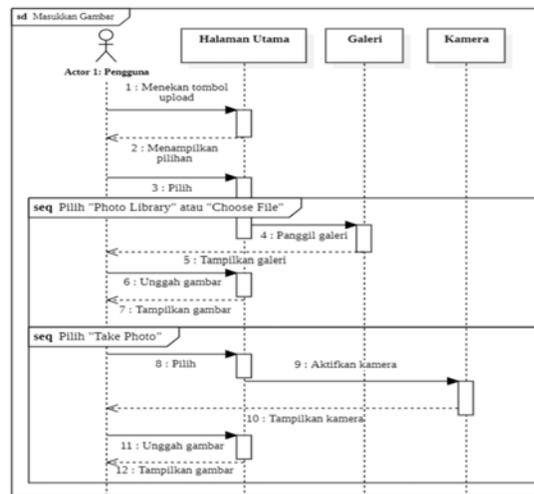


Gambar 5. Activity Diagram Deteksi Ikan

3.3.3 Sequence Diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi antar objek secara rinci pada aplikasi. Pada aplikasi ini, terdapat dua *sequence diagram*.

3.3.3.1 Sequence Diagram Unggah Gambar

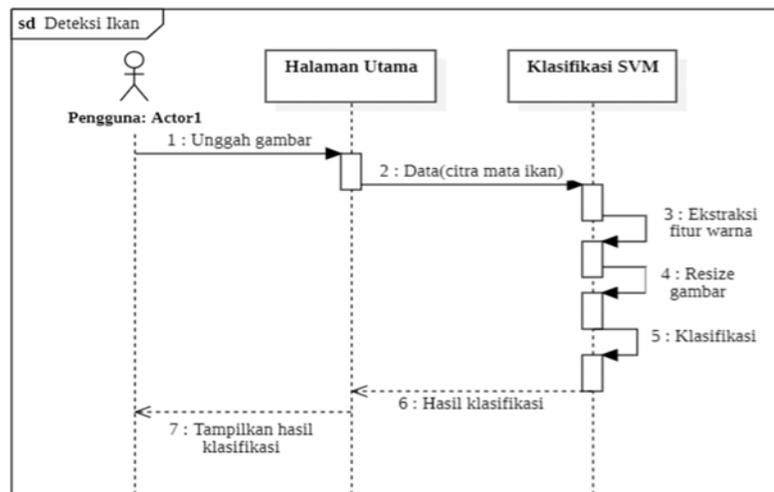


Gambar 6. Sequence Diagram Masukkan Gambar

Pada Gambar 6 ditunjukkan *sequence diagram* atau diagram sekuen masukkan gambar. Pada *sequence diagram* tersebut, terdapat satu aktor dan tiga objek yaitu halaman utama, galeri, dan kamera. Setelah pengguna masuk ke halaman utama, pilihan *photo library* atau *choose file* dan *take photo* akan ditampilkan. Jika pengguna memilih opsi *photo library* atau *choose file*, maka sistem akan menampilkan gambar yang berada di galeri, kemudian pengguna memilih gambar yang ingin diunggah, dan sistem akan menampilkan gambar ke halaman utama. Jika pengguna memilih opsi *take photo*, maka sistem akan membuka kamera, kemudian pengguna mengambil gambar menggunakan kamera dan sistem akan menampilkan gambar ke halaman utama.

3.3.3.2. Sequence Diagram Deteksi Ikan

Pada *sequence diagram* deteksi ikan, terdapat satu aktor dan dua objek yaitu halaman utama dan klasifikasi SVM. Setelah masuk ke halaman utama, pengguna akan mengunggah gambar. Gambar yang telah diunggah akan diproses oleh klasifikasi SVM. Proses yang dilakukan adalah ekstraksi fitur warna RGB dan HSV, *resize* gambar, klasifikasi, serta menampilkan hasil deteksi ke halaman utama. *Sequence diagram* atau diagram sekuen deteksi ikan ditunjukkan pada Gambar 7.



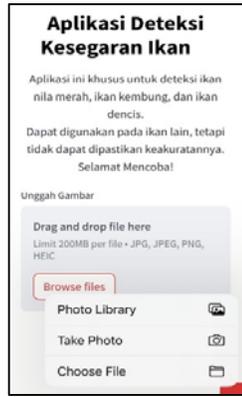
Gambar 7. Sequence Diagram Deteksi Ikan

3.4 Tampilan Antarmuka Aplikasi

Tampilan antarmuka merupakan implementasi dari aplikasi. Tampilan antarmuka dibangun berdasarkan tampilan yang sudah dirancang pada tahap perancangan antarmuka. Berikut adalah tampilan antarmuka pada aplikasi deteksi kesegaran ikan.

3.4.1 Tampilan Halaman Utama

Tampilan halaman utama berisi nama aplikasi, keterangan tentang aplikasi, dan *button browse file* yang berisi pilihan kamera dan unggah gambar. Jika memilih opsi *take photo*, maka pengguna dapat mengambil gambar secara langsung menggunakan kamera. Jika pengguna opsi *photo library* atau *choose file*, maka pengguna dapat mengunggah gambar dari galeri. Tampilan halaman utama ditunjukkan pada Gambar 8.

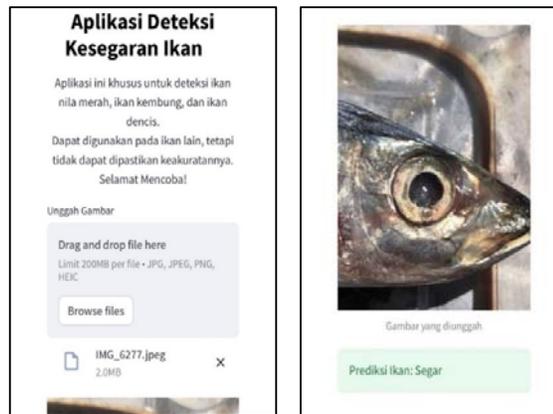


Gambar 8. Tampilan Halaman Utama

3.4.2 Tampilan Hasil Prediksi

Tampilan hasil prediksi terdapat beberapa komponen, yaitu nama aplikasi, keterangan tentang aplikasi, *button browse file* yang berisi pilihan kamera atau unggah gambar, gambar ikan yang berisi hasil dari jepretan kamera ataupun hasil dari unggahan gambar dan keterangan dari hasil prediksi.

Jika pengguna memilih opsi *take photo*, maka pengguna akan dialihkan ke kamera *smartphone*, kemudian pengguna mengambil gambar mata ikan menggunakan kamera. Hasil jepretan kamera serta hasil klasifikasi akan ditampilkan. Jika pengguna memilih opsi *photo library* atau *choose file*, maka pengguna akan dialihkan ke penyimpanan gambar pada *smartphone*. Pengguna memilih gambar yang ingin dideteksi dan hasil unggahan gambar serta hasil klasifikasi akan ditampilkan. Tampilan hasil prediksi ditunjukkan pada Gambar 9.



Gambar 9. Tampilan Hasil Prediksi

3.5 Pengujian

Pengujian merupakan tahap untuk menguji apakah model yang dibangun dapat mendeteksi citra yang belum dilatih. Pada penelitian ini, pengujian dilakukan menggunakan dua *smartphone*, yaitu Iphone X dan OnePlus 5 dengan empat kali pengujian terhadap citra ikan untuk masing-masing kelas. Hasil pengujian ditunjukkan pada Tabel 10.

Tabel 10. Hasil Keseluruhan Pengujian

Perangkat	Kelas	Pengujian	Ikan Nila	Ikan Kembung	Ikan Dencis
Iphone X	Segar (S)	1	S	S	S
		2	S	TS	S
		3	S	S	S
		4	S	S	S
	Tidak Segar (TS)	1	TS	TS	TS
		2	TS	TS	TS
		3	TS	TS	S
		4	S	TS	TS
	Busuk (B)	1	S	TS	S
		2	S	TS	TS
		3	S	S	S

		4	S	S	S
OnePlus 5	Segar (S)	1	S	S	S
		2	S	S	TS
		3	TS	S	S
		4	S	S	S
	Tidak Segar (TS)	1	TS	TS	TS
		2	TS	TS	TS
		3	TS	TS	TS
		4	S	TS	TS
	Busuk (B)	1	S	S	S
		2	TS	S	TS
		3	TS	TS	TS
		4	TS	S	TS
Keterangan : Data yang diberi warna abu-abu adalah data hasil pengujian yang salah.					

Berdasarkan Tabel 10, maka dilakukan perhitungan nilai akurasi pada masing-masing perangkat sebagai berikut.

$$\text{Iphone X} = \text{Akurasi} = \frac{\text{Prediksi data benar}}{\text{Total data}} = \frac{21}{36} = 0,58 = 58\%$$

$$\text{OnePlus 5} = \text{Akurasi} = \frac{\text{Prediksi data benar}}{\text{Total data}} = \frac{21}{36} = 0,58 = 58\%$$

Berdasarkan perhitungan di atas, akurasi yang dihasilkan cukup rendah, hal ini disebabkan oleh data kelas busuk yang secara keseluruhan bernilai salah yaitu dengan akurasi 0 persen. Model yang digunakan sebagai data latih adalah model citra ikan yang hanya berfokus pada mata ikan. Sedangkan aplikasi yang dirancang tidak melakukan deteksi objek yang berfokus pada mata ikan sehingga pada saat pengujian menghasilkan akurasi yang rendah. Oleh karena itu, peneliti akan melakukan pengujian ulang pada data kelas busuk terhadap ketiga jenis ikan menggunakan data uji yang hanya berfokus pada mata ikan. Hasil pengujian ulang pada data dengan kelas busuk menggunakan data uji yang berfokus pada mata ikan dengan perangkat Iphone X dan OnePlus 5 ditunjukkan pada Tabel 11.

Tabel 11. Hasil Pengujian Ulang Dengan Data Uji Busuk Baru

Perangkat	Kelas	Pengujian	Ikan Nila	Ikan Kembang	Ikan Dencis
Iphone X	Segar (S)	1	S	S	S
		2	S	TS	S
		3	S	S	S
		4	S	S	S
	Tidak Segar (TS)	1	TS	TS	TS
		2	TS	TS	TS
		3	TS	TS	S
		4	S	TS	TS
	Busuk (B)	1	B	B	B
		2	B	B	B
		3	B	B	B
		4	B	B	B
OnePlus 5	Segar (S)	1	S	S	S
		2	S	S	TS
		3	TS	S	S
		4	S	S	S
	Tidak Segar (TS)	1	TS	TS	TS
		2	TS	TS	TS
		3	TS	TS	TS
		4	S	TS	TS
	Busuk (B)	1	B	B	B
		2	B	B	B
		3	B	B	B
		4	B	B	B
Keterangan : Data yang diberi warna abu-abu adalah data hasil pengujian yang salah.					

Berdasarkan Tabel 11, maka dilakukan perhitungan ulang nilai akurasi pada masing-masing perangkat sebagai berikut.

$$\text{Iphone X} = \text{Akurasi} = \frac{\text{Prediksi data benar}}{\text{Total data}} = \frac{33}{36} = 0,92 = 92\%$$

$$\text{OnePlus 5} = \text{Akurasi} = \frac{\text{Prediksi data benar}}{\text{Total data}} = \frac{33}{36} = 0,92 = 92\%$$

Setelah melakukan pengujian ulang, akurasi yang dihasilkan menjadi lebih baik. Akurasi yang dihasilkan dari pengujian dua perangkat adalah 92 persen.

4. KESIMPULAN

Berdasarkan hasil pengujian aplikasi deteksi kesegaran ikan, peneliti menyimpulkan beberapa hal sebagai berikut:

- Aplikasi dapat dijalankan dengan cara *input* gambar, kemudian sistem akan memproses pengklasifikasian dan sistem akan memberikan hasil prediksi ke halaman utama. Pengklasifikasian menggunakan metode *Support Vector Machine* (SVM). Akurasi model SVM untuk pengklasifikasian mencapai 99 persen menggunakan kernel *Radial Basic Function* (RBF) dengan parameter $C = 10$ dan $\gamma = 0.5$.
- Pengujian model dilakukan sebanyak dua kali, sebab pengujian pertama menghasilkan akurasi yang kurang memuaskan yaitu 58 persen. Akurasi yang rendah tersebut diakibatkan oleh data kelas busuk yang secara keseluruhan bernilai salah serta *dataset* yang kurang bervariasi. Oleh sebab itu dilakukan pengujian kedua dengan data uji kelas busuk yang difokuskan pada mata ikan memperoleh akurasi sebesar 92 persen.

5. SARAN

Adapun saran untuk pengembangan aplikasi ini adalah:

- Menambahkan parameter bentuk dan tekstur untuk melakukan deteksi.
- Menambahkan *dataset* yang lebih banyak dan bervariasi.
- Aplikasi dapat melakukan deteksi objek sehingga hanya mengenali objek ikan saja.
- Aplikasi dapat melakukan deteksi mata ikan sehingga pada saat dijalankan, aplikasi langsung dapat melakukan *cropping* otomatis pada area mata ikan.

DAFTAR PUSTAKA

- [1] Kirana, Kartika Candra. (2021). *Pengolahan Citra Digital Teori Dan Penerapan Pengolahan Citra Digital Pada Deteksi Wajah*. Ahlimedia Press. Malang.
- [2] Hidayatullah, Muh. (2020). *Digital Imaging Menggunakan Adobe Photoshop CS6*. Yayasan Barcode. Makassar.
- [3] Zonyfar, Candra. (2020). *Pengolahan Citra Digital: Sebuah Pengantar*. Desanta Muliavisitama. Serang.
- [4] Khowarizmi, Al. (2021). *Pengantar Teknologi Informasi (Dalam Perkembangan Data Science)*. Umsu Press. Medan.
- [5] Jollyta, Deny, Prihandoko, Alayuma Hajjah, Elin Haerani, dan Muhammad Siddik. (2023). *Algoritma Klasifikasi Untuk Pemula Solusi Python Dan RapidMiner*. Deepublish. Sleman.
- [6] Alam, Guntur, Alwendo Wahyu Aranski, Sarah Astiti, Riko Andrian Putra, dan Sandhy Fernandez. (2023). *Pemrograman Web 2 Studi Kasus Layout Dan CRUD*. PT. Sonpedia Publishing Indonesia. Jambi.
- [7] Sukamto, Rosa Arini dan M. Shalahuddin. (2019). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika Bandung. Bandung.
- [8] Asri, Yessy, Dwina Kuswardani, Widya Nita Suliyanti, dan Chrystyna Monica Tambunan. (2019). *Algoritma C4.5 Klasifikasi Titik Dan Jenis Gangguan Pada Jaringan Distribusi Penyulang*. Uwais Inspirasi Indonesia. Sidoarjo.
- [9] Susandri dan Suranti Ratri. (2022). *Machine Learning: Review Omnibus Law UU Cipta Kerja*. CV Jakad Media Publishing. Surabaya.