

Sistem Klasifikasi Emosi Kucing Menggunakan SVM dan Strategi Augmentasi Data

Rizky Ananda Putra¹, Alif Rama Nugroho²

^{1,2}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: ¹*rizkyannandsan@gmail.com, ²aliframanugroho@gmail.com

Abstrak – Komunikasi yang baik dengan hewan peliharaan sering kali terganggu karena manusia kurang memahami suara-suara yang dikeluarkan kucing. Penelitian ini bertujuan membuat sistem yang bisa mengklasifikasikan perasaan kucing secara akurat, agar bisa lebih memahami apa yang mereka rasakan. Masalah utama yang dihadapi adalah ketidakseimbangan jumlah data dalam kategori perasaan yang jarang muncul. Untuk mengatasinya, penelitian ini menggunakan algoritma SVM dengan pendekatan augmentasi data berbasis pemrosesan di memori. Metode ini bisa mengubah dan menyeimbangkan data suara dengan efisien, tanpa membebani penyimpanan fisik. Hasil evaluasi menunjukkan model ini mampu mencapai akurasi sebesar 92,94%. Lebih khusus lagi, kategori perasaan yang melalui proses augmentasi berhasil mencapai skor F1 sebesar 1.00, yang membuktikan bahwa metode ini sangat efektif dalam menangani data yang tidak seimbang. Selain itu, sistem juga terbukti tahan terhadap gangguan suara dari lingkungan sekitar. Kesimpulan penelitian ini menyatakan bahwa penggabungan teknik augmentasi sinyal dengan algoritma SVM mampu menghasilkan sistem yang akurat, efisien, dan responsif terhadap berbagai jenis masukan.

Kata Kunci — Kucing, Emosi, Klasifikasi, SVM, Augmentasi.

1. PENDAHULUAN

Kucing domestik atau *Felis catus* [1], adalah salah satu hewan peliharaan yang paling terkenal di seluruh dunia [2]. Kepopularitasannya tidak hanya karena bentuknya yang lucu, tetapi juga karena kemampuan mereka berinteraksi dengan lingkungan sekitar, terutama manusia [3]. Seiring dengan meningkatnya perhatian masyarakat terhadap manfaat kesehatan mental yang diperoleh dari memiliki hewan peliharaan, jumlah kucing yang dipelihara terus bertambah setiap tahunnya [4]. Dalam hal komunikasi, kucing dewasa biasanya menggunakan suara *meong* untuk berinteraksi dengan pemiliknya, sehingga suara ini menjadi alat utama dalam membangun hubungan antara manusia dan kucing [5]. Karena itu, agar bisa menjaga kesejahteraan hewan sekaligus memperkuat hubungan yang baik antara pemilik dengan hewan peliharaannya, diperlukan pemahaman yang dalam mengenai berbagai jenis suara yang mereka hasilkan [6].

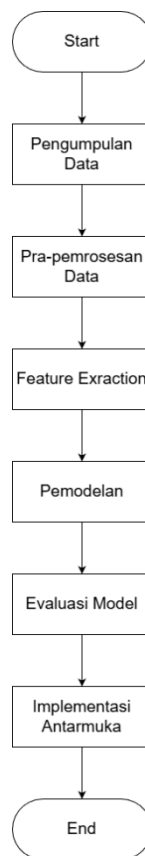
Namun, upaya memahami isyarat dari kucing seringkali terhambat karena perbedaan pemahaman antara kucing dan manusia. Banyak pemilik, terutama yang baru memelihara kucing, sering kali tidak bisa memahami maksud yang ingin disampaikan oleh kucing mereka [7]. Situasi seperti ini bisa menyebabkan respons yang salah. Contohnya, pemilik mungkin memberi makan kucing padahal sebenarnya hanya ingin menyisirnya, atau mengabaikan kucing yang merasa kesepian karena menganggapnya hanya meminta perhatian biasa. Kesalahpahaman seperti ini bisa membuat pemilik bingung dan membuat kucing merasa stres [8].

Untuk mengatasi kesalahpahaman dalam komunikasi, pendekatan berbasis teknologi dengan pengenalan pola suara dianggap sebagai solusi yang sangat menjanjikan [9]. Penggunaan *Machine Learning* (ML), khususnya algoritma *Support Vector Machine* (SVM), dianggap efektif dalam mengklasifikasikan suara yang memiliki karakteristik kompleks [10]. Kemampuan SVM terbukti baik dalam penelitian forensik audio oleh Liu dkk. [11], di mana kombinasi SVM dengan ekstraksi fitur *Mel-frequency Cepstral Coefficients* (MFCC) mampu mencapai tingkat akurasi hingga 99% dalam mendeteksi manipulasi suara stereo. Keunggulan SVM juga terbukti melalui penelitian perbandingan, yang menunjukkan bahwa algoritma ini mampu mengungguli *Random Forest* dengan akurasi 100% pada data tertentu yang memiliki jumlah sampel latih yang memadai [12]. Selain itu, kemampuan SVM dalam memproses sinyal suara untuk mengenali karakteristik emosional yang memiliki kesamaan pola dengan intonasi kebutuhan hewan juga telah dibuktikan dalam penelitian lain [13]. Berdasarkan hasil penelitian tersebut, SVM dianggap sebagai metode yang paling cocok digunakan dalam sistem klasifikasi suara kucing karena mampu menjaga keseimbangan antara tingkat akurasi yang tinggi dan efisiensi dalam penggunaan komputasi.

Berdasarkan kemampuan komputasi yang ada, penelitian ini bertujuan untuk menerapkan algoritma SVM dalam membuat sistem klasifikasi suara kucing yang lebih lengkap dan mampu menangani gangguan. Penelitian ini tidak hanya fokus pada klasifikasi suara kucing dalam kondisi biasa, tetapi juga mencakup berbagai variasi suara spesifik, serta termasuk kategori suara yang tidak relevan seperti *noise*, untuk mengurangi kesalahan dalam pengklasifikasian di lingkungan nyata. Hasil dari model klasifikasi ini kemudian diintegrasikan ke dalam antarmuka berbasis *website* menggunakan kerangka Streamlit, sehingga pengguna dapat menganalisis suara secara langsung dan *real-time*. Dengan pendekatan ini, diharapkan tercipta alat bantu yang praktis dan akurat bagi pemilik hewan peliharaan untuk memahami kebutuhan mereka, serta menguji kinerja algoritma SVM pada *dataset* yang lebih rumit.

2. METODE PENELITIAN

Penelitian tentang pembuatan sistem klasifikasi suara kucing dengan menggunakan algoritma SVM ini dilakukan melalui beberapa tahapan yang disusun secara rapi, seperti terlihat pada Gambar 1. Kerangka kerja ini dibuat dengan terstruktur agar setiap prosesnya bisa di percaya, mulai dari pengambilan data audio hingga pengecekan kemampuan model, sehingga bisa menghasilkan sistem klasifikasi yang tepat, kuat, dan bisa dijadikan dasar ilmiah.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

Data penelitian ini dikumpulkan dari tiga sumber berbeda. Hal ini dilakukan untuk memastikan keberagaman variasi suara dan ketahanan model terhadap *noise*. "*Cat Meow Classification*" adalah sumber data pertama dari repositori publik *Kaggle* milik Larxel [14]. Terdiri dari tiga kelas yaitu *brushing* (manja), *waiting for food* (meminta makan), dan *isolation in an unfamiliar environment* (terisolasi), *dataset* ini berfungsi sebagai data primer yang mewakili kondisi domestik umum.

Selanjutnya, untuk *dataset* kedua berasal dari platform *Kaggle* milik Yagya Raj Pandeya dengan judul "*Cat Sound Classification Dataset*" [15]. Sepuluh label kelas baru ditambahkan dari *dataset* ini. Meliputi kelas *angry* (marah), *defence* (pertahanan), *fighting* (bertengkar), *happy* (bahagia), *hunting mind* (mencari pikiran), *mating* (tidur), *mother call* (panggilan ibu), *paining* (sakit), *resting* (rehat), dan *warning* (peringatan).

Terakhir, ditambahkan satu kelas lagi yang berisi suara yang bukan suara kucing. Suara ini didapatkan dari rekaman mandiri dan pemilihan konten dari *YouTube*. Suara lingkungan yang bukan suara kucing, seperti suara klakson, anjing menggonggong, alat musik, dan suara latar belakang lainnya, termasuk dalam kelas ini. Tujuan menambah kelas ini adalah agar sistem memiliki data untuk dibandingkan, sehingga sistem bisa membedakan antara suara kucing dan suara yang bukan kucing. Pada Tabel 1 terdapat ringkasan mengenai distribusi sumber data dan label kelas yang digunakan dalam penelitian ini.

Tabel 1. Ringkasan Distribusi Label <i>Dataset</i>	
Sumber Data	Label/Kelas
Kaggle (Larxel)	Brushing, Food, Isolation
Kaggle (Yagya Raj Pandeya)	Angry, Defense, Fighting, Happy, HuntingMind, Mating, MotherCall, Paining, Resting, Warning
Mandiri	Non-Cat

2.2 Pra-pemrosesan Data

Pada tahap pra-pemrosesan data dalam penelitian ini, dilakukan penyesuaian format sinyal audio agar lebih standar dan meningkatkan kualitas *input* sebelum proses ekstraksi fitur dimulai. Karena data audio berasal dari berbagai sumber dengan karakteristik perekaman yang berbeda, langkah pertama yang dilakukan secara merata pada seluruh *dataset* adalah penyesuaian laju sampel (*resampling*) menjadi 22.050 Hz. Setelah itu, diterapkan teknik pengurangan derau dengan metode *silence trimming*, yaitu memotong bagian awal dan akhir rekaman yang diam secara otomatis, berdasarkan ambang batas energi sebesar 20 dB. Serangkaian perbaikan sinyal ini ditutup dengan proses normalisasi amplitudo. Proses ini mengubah sinyal gelombang suara diskrit $x[n]$ menjadi rentang nilai antara -1 hingga 1, yang secara matematis dijelaskan dalam Persamaan 1 berikut:

$$x_{norm}[n] = \frac{x[n]}{\max(|x[n]|)} \dots \dots \dots (1)$$

$x_{norm}[n]$ adalah sinyal yang sudah dikoreksi agar nilai-nilainya tidak terlalu besar, dan $\max(|x[n]|)$ adalah nilai tertinggi dari besaran suara tersebut. Langkah ini dilakukan agar tidak ada kesalahan perhitungan karena perbedaan kekuatan suara yang dicatat oleh berbagai perangkat perekam.

Selanjutnya, untuk mengatasi masalah *dataset* yang tidak seimbang, penelitian ini menggunakan teknik augmentasi data. Algoritma sistem dirancang agar mampu membedakan jalur pemrosesan berdasarkan sumber data. Pada *dataset* pertama, data diproses melalui jalur standar tanpa penambahan variasi. Namun, pada *dataset* kedua dan kelas negatif (*non-cat*), diterapkan jalur augmentasi berbasis pemrosesan memori, di mana setiap file audio fisik diolah secara algoritmis menjadi sepuluh variasi data sintetik untuk menyeimbangkan jumlah kelas minoritas.

Variasi data sintetik pada jalur augmentasi ini dihasilkan melalui beberapa teknik manipulasi sinyal digital. Teknik pertama adalah penambahan *noise Gauss* yang menambahkan gangguan acak pada sinyal asli untuk mensimulasikan suara latar belakang, dengan persamaan model aditif yang terdapat pada Persamaan 2.

$$x_{aug}[n] = x[n] + \alpha \cdot w[n] \dots \dots \dots (2)$$

Di mana $w[n]$ adalah *noise* putih yang mengikuti distribusi normal dan α adalah faktor yang mengubah besarnya *noise*. Selain itu, digunakan teknik *Pitch Shifting* untuk mengubah frekuensi dasar (f_0) tanpa mengubah durasi, teknik *Time Stretching* untuk mempercepat atau memperlambat tempo audio dengan mengubah rasio durasi waktu ($t \rightarrow \beta t$), serta teknik *Time Shifting* untuk menggeser posisi waktu sinyal (*rolling*). Penggunaan kombinasi dari teknik-teknik tersebut bertujuan untuk memperkaya representasi fitur dalam ruang latih, sehingga model memiliki kemampuan generalisasi yang lebih baik terhadap berbagai variasi kondisi akustik.

2.3 Feature Extraction

Setelah pra-pemrosesan, tahap ekstraksi fitur bertujuan mengubah sinyal berupa gelombang waktu menjadi bentuk angka yang singkat namun mampu mewakili sifat khas suara. Penelitian ini memilih fitur *Mel-Frequency Cepstral Coefficients* (MFCC) karena kemampuannya meniru cara pendengaran mamalia non-linear dalam membedakan timbre suara, terutama dengan sensitivitas yang lebih tinggi terhadap frekuensi rendah. Cara mengubah frekuensi (f) menjadi skala *Mel* (M) dijelaskan dalam Persamaan 3.

$$M(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \dots \dots \dots (3)$$

Dari 20 koefisien MFCC yang diambil, kemudian digabungkan dengan fitur *Zero Crossing Rate* (ZCR) sebagai penanda tingkat kekasaran getaran suara harmonik, serta fitur *Chroma* untuk mengetahui nada utama yang terdengar. Selain itu, fitur *Root Mean Square* (RMS) digunakan untuk mengukur tingkat energi atau kekerasan suara, seperti yang dijelaskan dalam Persamaan 4.

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N |x_n|^2} \dots \dots \dots (4)$$

Karena durasi *input* audio berbeda-beda, semua fitur yang dihasilkan (MFCC, *Chroma*, ZCR, RMS) digabungkan dengan menghitung rata-rata (*mean*) dan standar deviasi (*std*). Hasilnya disatukan menjadi satu vektor fitur berdimensi tunggal yang bisa digunakan sebagai *input* untuk algoritma klasifikasi.

2.4 Pemodelan

Tahap pemodelan bertujuan untuk membuat fungsi keputusan yang bisa mengubah vektor fitur suara menjadi 14 kelas label target. Karena SVM sangat efektif dalam menangani banyak dimensi fitur, algoritma ini dipilih. Sebelum pelatihan dimulai, *dataset* fitur dibagi menjadi data latih dan data uji dengan rasio 75:25. Teknik stratifikasi digunakan agar proporsi setiap kelas, dari variasi emosi kucing hingga kelas *noise*, terdistribusi secara seimbang dan mewakili pada kedua bagian data, sehingga hasil evaluasi tidak bias.

Untuk menjaga konsistensi dalam pengolahan data, proses pelatihan sistem dilakukan dalam sebuah alur kerja terintegrasi yang disebut *pipeline*. Sebelum *pipeline* dimulai, data fitur distandarisasi menggunakan *StandardScaler*, alat yang membuat distribusi data memiliki rata-rata nol dan varians satu. Karena SVM adalah algoritma berbasis jarak yang sensitif terhadap skala fitur, tahapan ini sangat penting. Selain itu, penyetelan parameter dilakukan dengan metode *Grid Search* yang dikombinasikan dengan *Stratified K-Fold Cross-Validation* agar model memiliki performa terbaik. Sistem secara otomatis menguji berbagai kombinasi parameter seperti jenis kernel, nilai regularisasi *C*, dan koefisien *gamma*. Setelah itu, sistem memilih konfigurasi terbaik sebagai model akhir yang akan diekspor.

2.5 Evaluasi Model

Untuk menilai seberapa baik model SVM yang telah dilatih dalam mengenali 14 jenis suara (13 emosi kucing dan 1 kategori *noise*), digunakan metode evaluasi berupa *Confusion Matrix*. Seperti yang dijelaskan oleh Pushpa Singh dkk. [16], *Confusion Matrix* adalah alat visualisasi yang berguna untuk mengetahui sejauh mana pengklasifikasi mampu mengenali data dari berbagai kelas.

Dalam kasus klasifikasi multi-kelas, evaluasi dilakukan berdasarkan empat komponen dasar hasil prediksi yang terdapat pada Tabel 2.

Tabel 2. Tabel Confusion Matrix		
Kelas Data	Positif Prediksi	Negatif Prediksi
Positif Aktual	TP	FN
Negatif Aktual	FP	TN

Berdasarkan komponen di atas, kinerja model diukur menggunakan empat metrik utama yang direkomendasikan oleh Miller dkk. [17] untuk pemodelan klasifikasi, yaitu:

1. Akurasi (*Accuracy*)

Akurasi adalah perbandingan antara prediksi yang benar dengan total data uji. Akurasi menunjukkan keseluruhan kemampuan sistem secara keseluruhan. Rumus untuk menghitung akurasi bisa dilihat pada Persamaan 5:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \dots \dots \dots (5)$$

2. Presisi (*Precision*)

Presisi mengukur seberapa tepat model dalam mengklasifikasikan prediksi positif. Presisi penting untuk mengurangi kesalahan FP, seperti memastikan suara anjing tidak dianggap sebagai suara kucing. Rumus presisi bisa dilihat pada Persamaan 6.:

$$Precision = \frac{TP}{TP+FP} \dots \dots \dots (6)$$

3. Recall (*Sensitivity*)

Recall mengukur kemampuan model dalam menemukan semua informasi positif yang relevan dari data yang sebenarnya positif. Metrik ini penting untuk mencegah kesalahan FN. Rumus recall bisa dilihat pada Persamaan 7:

$$Precision = \frac{TP}{TP+FN} \dots \dots \dots (7)$$

4. *F1-Score*

F1-Score adalah rata-rata harmonik dari presisi dan recall. *F1-Score* memberikan penilaian yang lebih seimbang dibandingkan akurasi, terutama ketika data tidak seimbang antar kelas. Rumus *F1-Score* bisa dilihat pada Persamaan 8:

$$F1 - Score = 2 \frac{Precision \times Recall}{Precision + Recall} \dots \dots \dots (8)$$

2.6 Implementasi Antarmuka

Tahap implementasi antarmuka bertujuan untuk mengubah model komputasi yang rumit menjadi aplikasi yang bisa digunakan oleh pengguna dengan mudah. Antarmuka sistem ini dibuat menggunakan *framework Streamlit* yang berbasis *Python*. Alasan memilih *Streamlit* adalah karena kemampuannya dalam membuat aplikasi data web dengan cepat, serta kemudahan integrasinya dengan berbagai *library* pemrosesan data di *Python* tanpa perlu membuat bagian antarmuka depan secara terpisah.

Arsitektur antarmuka dirancang sesuai dengan alur diagram aktivitas sistem, yang terdiri dari dua modul halaman utama yang dikelola melalui mekanisme *Session State*. Mekanisme ini bertugas untuk menyimpan data variabel agar tetap ada saat pengguna berpindah dari satu halaman ke halaman lain.

2.6.1 Halaman Validasi Pengunjung

Modul ini berperan sebagai pintu masuk pertama. Pengguna harus mengisi informasi identitas seperti Nama Pemilik dan Nama Kucing sebelum diperbolehkan menggunakan fitur klasifikasi. Sistem melakukan pemeriksaan logis untuk memastikan semua kolom data terisi dengan lengkap.

2.6.2 Halaman Klasifikasi dan Visualisasi

Antarmuka ini adalah bagian utama dari sistem, dilengkapi dengan widget pengunggah berkas. *Widget* ini diatur untuk menerima file berformat audio *.wav* dan *.mp3*. Setelah berkas diunggah, antarmuka secara otomatis memulai proses dari kelas *Preprocessing* dan *Classification* di bagian belakang sistem. Hasilnya ditampilkan dalam dua bentuk, yaitu label kelas emosi yang diprediksi beserta tingkat keyakinan dalam bentuk persentase, serta visualisasi grafik batang yang menampilkan distribusi probabilitas terhadap 14 kelas label. Hal ini memberikan informasi yang transparan kepada pengguna mengenai bagaimana model membuat keputusan.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Tahap pengumpulan data sudah selesai dengan berhasil mengumpulkan materi audio dari tiga sumber yang sudah ditentukan sesuai dengan rencana penelitian. Semua file audio yang didapat disimpan dalam folder terpisah agar lebih mudah dikenali. Di *dataset* utama yang berasal dari repositori Laxel, berhasil dikumpulkan contoh suara untuk tiga kelas utama dengan format file *.wav*.

Sementara itu, dari *dataset* tambahan yang berasal dari Yagya Raj Pandeya, diperoleh sepuluh folder berisi kategori emosi, masing-masing folder berisi 10 file audio dalam format *.mp3*. Terakhir, untuk kategori *non-cat*, proses perekaman sendiri dan penambahan manual berhasil mengumpulkan 26 contoh suara lingkungan yang beragam. Dari hasil perhitungan jumlah data fisik tersebut, ditemukan adanya ketidakseimbangan dalam jumlah data (*class imbalance*) yang cukup signifikan antara *dataset* utama, *dataset* tambahan, dan kelas *non-cat*.

3.2 Pra-pemrosesan Data

Modul pra-pemrosesan telah diterapkan pada semua data mentah yang telah ditata sebelumnya. Pada tahap awal standarisasi sinyal, semua file audio, baik berformat *.wav* maupun *.mp3*, berhasil diubah ke laju sampel yang sama dan amplitudo-nya dinormalisasi. Penggunaan algoritma penghapusan bagian hening juga berhasil mengurangi rata-rata durasi file dengan menghilangkan bagian yang tidak penting, sehingga sinyal menjadi lebih padat dan informatif.

Hasil terbesar pada tahap ini datang dari penerapan strategi augmentasi data. Berdasarkan log sistem, pemrosesan pada *dataset* sekunder dan *dataset non-cat* berhasil menghasilkan variasi data sintesis secara langsung di memori. Setiap file fisik pada kelas-kelas ini berhasil diubah menjadi 10 vektor fitur berbeda melalui variasi *noise*, *pitch*, dan *speed*. Sebaliknya, *dataset* primer diproses menggunakan jalur standar tanpa penambahan variasi, sehingga mempertahankan karakteristik asli.

Perubahan jumlah data ini langsung memengaruhi keseimbangan distribusi kelas. Perubahan jumlah data dari kondisi mentah menjadi data latih yang siap digunakan disajikan pada Tabel 3.

Tabel 3. Perbandingan Jumlah Data Sebelum dan Sesudah Pra-pemrosesan

Sumber Data	Label/Kelas	Format Awal	Jumlah Awal	Setelah Augmentasi
Kaggle (Larxel)	Brushing	.wav	127	127
	Food	.wav	92	92
	Isolation	.wav	221	221
Kaggle (Yagya Raj Pandeya)	Angry	.mp3	10	100
	Defense	.mp3	10	100
	Fighting	.mp3	10	100
	Happy	.mp3	10	100
	HuntingMind	.mp3	10	100
	Mating	.mp3	10	100
	MoherCall	.mp3	10	100
	Paining	.mp3	10	100
	Resting	.mp3	10	100
	Warning	.mp3	10	100
Mandiri	Non-Cat	.mp3	26	260
TOTAL			566	1700

Seluruh data hasil pemrosesan dan augmentasi tersebut kemudian diekstraksi menjadi matriks fitur numerik dan disimpan ke dalam file *features.csv*.

3.3 Feature Extraction

Proses mengubah sinyal audio menjadi bentuk angka telah berhasil dilakukan pada semua data latih yang sudah diproses sebelumnya. Hasil dari tahapan ini adalah terbentuknya data yang terstruktur dalam format *.csv*. Dari pengecekan terhadap file *features.csv* yang dihasilkan, setiap sampel audio kini diwakili oleh vektor fitur yang memiliki dimensi 54. Dimensi tersebut berasal dari kombinasi empat jenis fitur statistik yang dihitung selama seluruh durasi audio. Komposisi lengkap dari vektor fitur yang berhasil diekstraksi disajikan pada Tabel 4.

Tabel 4. Struktur Vektor Fitur Hasil Ekstraksi

Jenis Fitur	Stastistik	Jumlah Dimensi (Kolom)	Keterangan
MFCC	Rerata (<i>Mean</i>)	20	Representasi warna suara utama
	Standar Deviasi (<i>Std</i>)	20	Variasi warna suara
Chroma	Rerata (<i>Mean</i>)	12	Representasi kelas nada (pitch)
ZCR	Rerata (<i>Mean</i>)	1	Indikator kekasaran sinyal
RMS	Rerata (<i>Mean</i>)	1	Indikator energi/kekerasan suara
TOTAL		54	Input Model SVM

Selain 54 kolom fitur numerik (dilabeli sebagai *f0* hingga *f53*), struktur data akhir juga mencakup kolom identitas berupa nama berkas dan label target numerik yang mewakili 14 kelas.

3.4 Pemodelan

Pelatihan model dilakukan dengan menggunakan algoritma SVM pada *dataset* fitur yang telah dibagi secara proporsional. Proses penghitungan dilaksanakan di lingkungan perangkat keras yang dilengkapi dengan CPU AMD Ryzen 5 dan RAM sebesar 24 GB, yang terbukti mampu mengolah matriks fitur berdimensi $N \times 54$ secara efisien.

Berdasarkan skenario penyesuaian hyperparameter dengan metode *GridSearchCV*, sistem telah mengevaluasi berbagai kombinasi nilai parameter untuk menemukan konfigurasi yang memberikan hasil generalisasi terbaik. Dari serangkaian eksperimen validasi silang berbasis *Stratified 5-Fold*, model menunjukkan konvergensi yang optimal. Hasil pencarian menunjukkan bahwa kombinasi parameter terbaik adalah dengan menggunakan fungsi kernel *RBF*, dengan nilai regularisasi *C* sebesar 5 dan koefisien *gamma scale*. Detail tentang ruang pencarian dan parameter yang dipilih dapat dilihat pada Tabel 5.

Tabel 5. Hasil Optimasi *Hyperparameter* SVM

Hyparameter	Search Space	Parameter Terpilih
Kernel	linear, rbf, poly, sigmoid	rbf
C	0.001, 0.01, 0.1, 1, 5, 10	5
Gamma	scale, auto, 1e-3, 1e-2, 1e-1, 1, 10	scale

Model dengan konfigurasi parameter terbaik kemudian dilatih kembali menggunakan seluruh data latih untuk memaksimalkan pemahaman pola. Model akhir tersebut selanjutnya disimpan dalam format biner bernama *best_svm.pkl* dengan ukuran file yang *compact*, sehingga siap digunakan dalam antarmuka sistem tanpa menguras memori saat proses prediksi berlangsung.

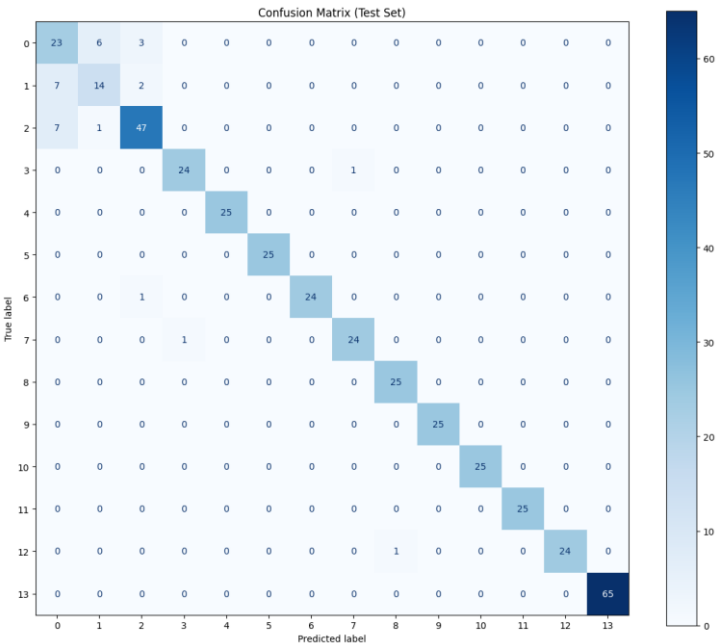
3.5 Evaluasi Model

Evaluasi kemampuan model dilakukan pada data uji yang mencakup 25% dari seluruh *dataset*, yaitu data yang tidak pernah dilihat oleh model selama masa pelatihan. Dengan menggunakan metrik evaluasi standar, model SVM yang dibuat berhasil mencapai akurasi global sebesar 92,94%. Angka ini menunjukkan bahwa sistem memiliki kemampuan generalisasi yang sangat baik dalam mengenali 14 kategori suara berbeda.

Analisis lebih lanjut mengenai kinerja model untuk setiap kelas disajikan dalam Tabel 6. Tabel tersebut merangkum nilai presisi, recall, dan *F1-Score* yang diperoleh dari laporan klasifikasi.

Tabel 6. Laporan Klasifikasi

Kelas	Precision	Recall	<i>F1-Score</i>
0	0,62	0,72	0,67
1	0,67	0,61	0,64
2	0,89	0,85	0,87
3	0,96	0,96	0,96
4	1,00	1,00	1,00
5	1,00	1,00	1,00
6	1,00	0,98	0,98
7	0,96	0,96	0,96
8	0,96	0,98	0,98
9	1,00	1,00	1,00
10	1,00	1,00	1,00
11	1,00	1,00	1,00
12	1,00	0,96	0,98
13	1,00	1,00	1,00



Gambar 2. Confusion Matrix Display

Visualisasi *Confusion Matrix* pada Gambar 2 menampilkan distribusi hasil prediksi model secara lebih rinci. Matriks tersebut menunjukkan pola diagonal yang mendominasi, yang menunjukkan bahwa sebagian besar data uji berhasil diprediksi dengan benar sesuai dengan label aslinya. Dengan melihat intensitas warna dan sebaran angka pada matriks, dapat ditarik dua poin utama:

1. Kotak berwarna biru gelap di sudut kanan bawah (Label 13) menunjukkan hasil yang sangat memuaskan. Model berhasil mengklasifikasikan seluruh 65 sampel suara *non-cat* secara benar tanpa ada kesalahan (100% *True Positive*). Selain itu, tidak ada suara kucing yang salah dikategorikan sebagai *non-cat*. Hal ini membuktikan bahwa sistem memiliki kemampuan untuk menangani gangguan suara lingkungan dengan sangat baik.
2. Kelas-kelas dari *dataset* sekunder yang telah diolah dengan metode augmentasi (Label 3 hingga 12) menunjukkan tingkat akurasi yang sangat konsisten. Kira-kira semua kelas memiliki angka prediksi benar antara 24 hingga 25 sampel, dengan kesalahan sangat sedikit, hanya 0 hingga 1 sampel yang salah per kelas. Ini menunjukkan bahwa strategi augmentasi *in-memory* berhasil meningkatkan variasi data, sehingga model bisa membedakan emosi spesifik seperti *Angry*, *Happy*, hingga *Warning* secara akurat.

3.6 Implementasi *User Interface* (UI)

3.6.1 Halaman Validasi Pengunjung

Pada saat aplikasi dijalankan pertama kali, pengguna akan dibawa ke halaman Daftar Hadir yang bertindak sebagai pintu masuk awal untuk memvalidasi sistem. Di halaman tersebut, pengguna wajib mengisi formulir identitas yang terdiri dari dua kolom, yaitu Nama Pemilik dan Nama Kucing.

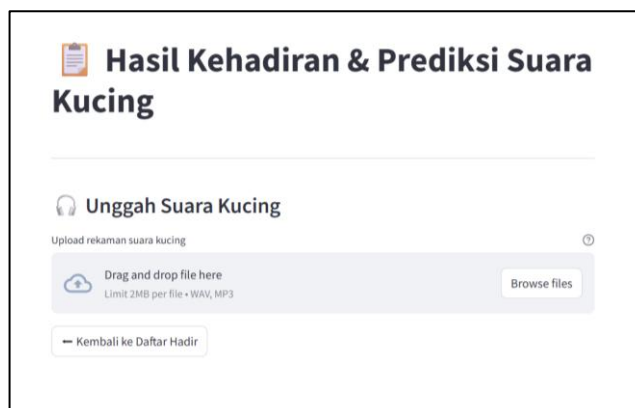


Gambar 3. Halaman Validasi Pengunjung

Visualisasi antarmuka halaman ini ditampilkan pada Gambar 3 di atas. Sistem menggunakan mekanisme validasi *input* untuk memastikan data tetap valid, di mana tombol akses Masuk secara default tidak aktif dan hanya akan dinyalakan setelah semua kolom data terisi. Mekanisme ini bertujuan untuk mencegah masuknya nilai kosong (*null values*) ke dalam sesi penyimpanan (*session state*) sistem.

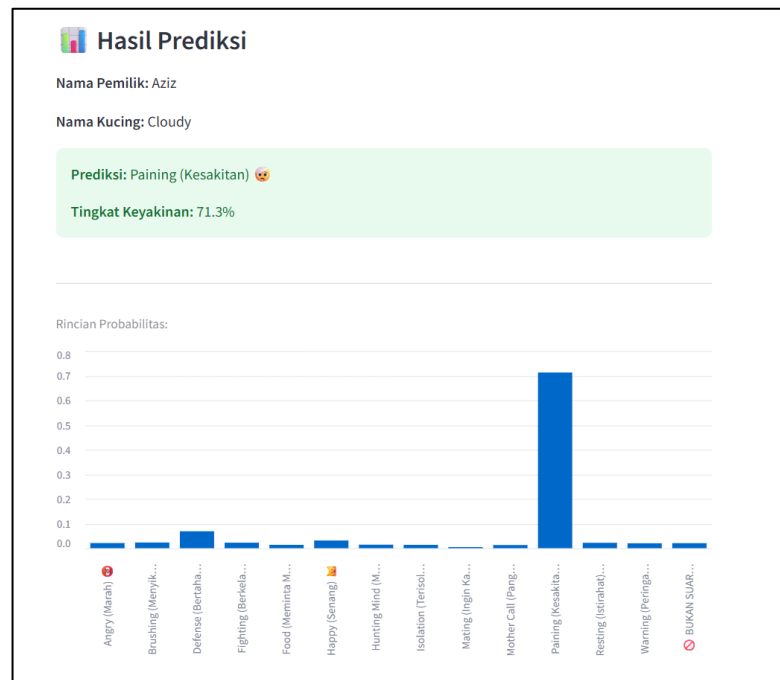
3.6.2 Halaman Klasifikasi dan Visualisasi

Setelah proses validasi identitas berhasil dilakukan, pengguna akan diarahkan ke halaman utama. Antarmuka ini dilengkapi dengan fitur *file uploader* yang mendukung format audio standar yaitu *.wav* dan *.mp3*. Sebagai alat verifikasi data, sistem menyediakan pemutar audio yang memungkinkan pengguna untuk memeriksa kualitas rekaman yang diunggah sebelum komputasi dimulai. Tampilan halaman klasifikasi dapat dilihat pada gambar 4 di bawah.



Gambar 4. Halaman Unggah File Audio untuk Klasifikasi Suara Kucing

Setelah berkas audio diterima, sistem secara otomatis menjalankan proses pra-pemrosesan dan prediksi model di *backend*. Hasil klasifikasi kemudian ditampilkan dalam dua komponen informasi utama, seperti yang ditunjukkan pada Gambar 5 di bawah, sebagai berikut:



Gambar 5. Halaman Hasil Klasifikasi dan Visualisasi

4. SIMPULAN

Berdasarkan analisis mendalam terhadap hasil implementasi dan pengujian sistem klasifikasi emosi suara kucing yang telah dibangun, dapat ditarik beberapa kesimpulan utama, yaitu:

1. Penelitian ini berhasil mengembangkan sistem cerdas berbasis SVM yang terintegrasi dengan antarmuka web *Streamlit*. Sistem tersebut terbukti mampu memproses *input* audio dan mengklasifikasikannya ke dalam 14 kategori (13 emosi dan 1 *noise*) secara interaktif. Keberhasilan ini menunjukkan bahwa model komputasi yang rumit bisa dikemas menjadi aplikasi praktis yang mudah digunakan.
2. Penggunaan teknik augmentasi *signal-domain augmentation* secara *in-memory* telah terbukti menjadi solusi penting dalam mengatasi masalah kelangkaan data. Metode ini berhasil menyeimbangkan distribusi data latih menjadi 1.700 sampel dan meningkatkan performa model pada kelas minoritas secara signifikan. Hal ini dibuktikan oleh capaian *F1-Score* sempurna sebesar 1,00 pada kelas-kelas yang diaugmentasi, seperti *Defense*, *Fighting*, dan *non-cat*.
3. Kelebihan utama dari model yang dihasilkan adalah kemampuannya dalam mengatasi gangguan suara lingkungan. Sistem menunjukkan kemampuan diskriminatif yang tinggi dalam membedakan suara target (kucing) dengan suara gangguan (*noise*) seperti gonggongan anjing atau instrumen musik, yang ditunjukkan dengan presisi 100% pada kelas *non-cat*. Akurasi global sebesar 92,94% membuktikan bahwa model ini dapat diandalkan untuk digunakan secara umum.
4. Kelemahan sistem terlihat pada kemampuannya dalam melakukan generalisasi untuk kelas-kelas dari *dataset* primer yang tidak melalui proses augmentasi. Terdapat perbedaan penampilan yang jelas, di mana kelas *Brushing* dan *Food* memiliki tingkat kesalahan klasifikasi yang lebih tinggi (*F1-Score* < 0.70) dibandingkan kelas lainnya. Hal ini menunjukkan bahwa ketergantungan pada data fisik yang terbatas tanpa variasi sintetik dapat menurunkan sensitivitas model terhadap pola akustik yang mirip.

5. SARAN

Sebagai tindak lanjut dari temuan mengenai ketimpangan performa yang masih terjadi pada *dataset* primer, penelitian berikutnya sangat dianjurkan untuk fokus pada peningkatan kualitas *dataset* melalui pengumpulan data riil secara besar-besaran, terutama pada kategori *Brushing* dan *Food*. Penambahan data secara alami ini dinilai sangat penting untuk memberikan variasi fitur suara yang lebih natural dibandingkan hanya bergantung pada manipulasi data sintetik. Dengan demikian, diharapkan dapat meningkatkan kemampuan model dalam membedakan kelas-kelas yang memiliki pola suara yang mirip. Selain itu, dianjurkan untuk melanjutkan

pengembangan dengan mengeksplorasi dan membandingkan pendekatan *Deep Learning* seperti arsitektur *Convolutional Neural Networks* (CNN) atau *Recurrent Neural Networks* (RNN). Penerapan metode ini bertujuan untuk menguji apakah mekanisme ekstraksi fitur otomatis berbasis spektrogram mampu memberikan hasil generalisasi yang lebih stabil dan akurasi yang lebih merata di seluruh kelas dibandingkan metode ekstraksi fitur manual seperti MFCC dengan SVM yang digunakan pada penelitian ini.

DAFTAR PUSTAKA

- [1] B. Nussberger dkk., “A common statement on anthropogenic hybridization of the European wildcat (*Felis silvestris*),” *Front Ecol Evol*, vol. Volume 11-2023, 2023, doi: 10.3389/fevo.2023.1156387.
- [2] S. Dwi Permana Putra, F. Fikri, dan O. Tivani Kiki, “Identifikasi Jenis Kucing Berdasarkan Pola Citra Menggunakan Convolutional Neural Network,” *Seminar Nasional Teknologi & Sains*, vol. 4, no. 1, hlm. 765–770, 2025.
- [3] Ú. Morales-Rodríguez, J. Carbajal-Hernández, B. Luna-Benoso, G. De Jesús Rodríguez-Jordán, dan J. C. M. Perales, “Recognition of domestic *Felis Catus* vocalisations using a computational associative model,” *Journal of Animal Behaviour and Biometeorology*, hlm., 2024, doi: 10.31893/jabb.2024020.
- [4] L. R. Kogan, J. Currin-McCulloch, C. Bussolari, dan W. Packman, “Cat owners’ disenfranchised guilt and its predictive value on owners’ depression and anxiety,” *Human-Animal Interactions*, vol. 2023, Jan 2023, doi: 10.1079/HAI.2023.0044;ISSUE:ISSUE:DOI.
- [5] S. Schötz, J. van de Weijer, dan R. Eklund, “Context effects on duration, fundamental frequency, and intonation in human-directed domestic cat meows,” *Appl Anim Behav Sci*, vol. 270, hlm. 106146, 2024, doi: <https://doi.org/10.1016/j.applanim.2023.106146>.
- [6] C. de Mouzon, R. Di-Stasi, dan G. Leboucher, “Human perception of cats’ communicative cues: human-cat communication goes multimodal,” *Appl Anim Behav Sci*, vol. 270, hlm. 106137, 2024, doi: <https://doi.org/10.1016/j.applanim.2023.106137>.
- [7] D. de S. Machado, R. R. Vicentini, L. da S. Gonçalves, S. Luchesi, E. Otta, dan A. C. Sant’Anna, “Unraveling Feline Myths: A Review About Misperceptions and Beliefs Surrounding Domestic Cat Behavior,” dalam *Pets*, MDPI, 2025, hlm. 32.
- [8] E. Prato-Previde dkk., “What’s in a Meow? A Study on Human Classification and Interpretation of Domestic Cat Vocalizations,” *Animals (Basel)*, vol. 10, no. 12, hlm. 1–18, Des 2020, doi: 10.3390/ANI10122390.
- [9] P. Wei, F. He, L. Li, dan J. Li, “Research on sound classification based on SVM,” *Neural Comput Appl*, vol. 32, no. 6, hlm. 1593–1607, 2020, doi: 10.1007/s00521-019-04182-0.
- [10] S. Kavitha dan J. Manikandan, “Improved methodology of svm to classify acoustic signal by spectral centroid,” *Journal of Trends in Computer Science and Smart Technology*, vol. 3, no. 4, hlm. 294–304, 2022.
- [11] T. Liu, D. Yan, R. Wang, N. Yan, dan G. Chen, “Identification of Fake Stereo Audio Using SVM and CNN,” *Information*, vol. 12, no. 7, hlm. 263, Jun 2021, doi: 10.3390/info12070263.
- [12] Md. R. Ansari, S. A. Tumpa, J. A. F. Raya, dan M. N. Murshed, “Comparison between Support Vector Machine and Random Forest for Audio Classification,” dalam *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, IEEE, Sep 2021, hlm. 1–4. doi: 10.1109/ICECIT54077.2021.9641152.
- [13] R. A. A., M. Nasrun, dan C. Setianingsih, “Human Emotion Detection with Speech Recognition Using Mel-frequency Cepstral Coefficient and Support Vector Machine,” dalam *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, IEEE, Apr 2021, hlm. 1–6. doi: 10.1109/AIMS52415.2021.9466077.
- [14] “Cat Meow Classification.” Diakses: 7 Desember 2025. [Daring]. Tersedia pada: <https://www.kaggle.com/datasets/andrewmvd/cat-meow-classification>
- [15] “Cat Sound Classification Dataset.” Diakses: 7 Desember 2025. [Daring]. Tersedia pada: <https://www.kaggle.com/datasets/yagtapandeya/cat-sound-classification-dataset>
- [16] P. Singh, N. Singh, K. K. Singh, dan A. Singh, “Chapter 5 - Diagnosing of disease using machine learning,” dalam *Machine Learning and the Internet of Medical Things in Healthcare*, K. K. Singh, M. Elhoseny, A. Singh, dan A. A. Elngar, Ed., Academic Press, 2021, hlm. 89–111. doi: <https://doi.org/10.1016/B978-0-12-821229-5.00003-3>.
- [17] C. Miller, T. Portlock, D. M. Nyaga, dan J. M. O’Sullivan, “A review of model evaluation metrics for machine learning in genetics and genomics,” *Frontiers in Bioinformatics*, vol. 4, hlm. 1457619, Sep 2024, doi: 10.3389/FBINF.2024.1457619/FULL.