

Klasifikasi Perilaku Kucing Menggunakan Arsitektur Hybrid YOLOv8 dan CNN Berbasis Web

Setyaji Haryo Dewanto*, Tri Listyorini, Endang Supriyati
Program Studi Teknik Informatika, Fakultas Teknik,
Universitas Muria Kudus, Indonesia

Diterima: 25 Desember, 2025 | Revisi: 30 Januari 2026 | Diterbitkan: 27 Februari 2026

DOI: <https://doi.org/10.33005/scan.v21i1.5586>

ABSTRAK

Kesejahteraan hewan peliharaan, khususnya kucing, sangat bergantung pada kemampuan pemilik dalam memahami bahasa tubuh hewan. Namun, interpretasi manual seringkali tidak akurat. Penelitian ini mengembangkan sistem cerdas untuk klasifikasi perilaku kucing memanfaatkan Deep Learning dengan arsitektur hybrid. Sistem menggabungkan YOLOv8 untuk lokalisasi objek dan Convolutional Neural Network (CNN) MobileNetV2 untuk klasifikasi perilaku. Berbeda dari studi terdahulu yang mayoritas berfokus pada penyakit kulit, riset ini menargetkan delapan kelas perilaku spesifik: sleeping, sitting, playing, eating/drinking, grooming, stretching, lying down, dan walking/standing. Metodologi mencakup pengumpulan data multi-source, pra-pemrosesan Laplacian Variance untuk filtrasi blur, serta pelatihan model. Hasil eksperimen menunjukkan YOLOv8 unggul dalam deteksi objek dengan mAP@50 mencapai 96,0%. Pada tahap klasifikasi, CNN mencatatkan akurasi 54,21%, di mana kelas Sleeping mendominasi performa terbaik. Sistem ini telah diimplementasikan ke dalam platform web berbasis Flask, menyediakan alat bantu pemantauan kondisi kucing secara real-time bagi pemilik.

Kata Kunci: Deep Learning, YOLOv8, CNN, MobileNetV2, Klasifikasi Perilaku Kucing

Web-Based Cat Behavior Classification Using Hybrid YOLOv8 and CNN Architecture

ABSTRACT

Pet welfare, particularly for cats, relies heavily on the owner's ability to understand animal body language. However, manual interpretation is often inaccurate. This study develops an intelligent system for cat behavior classification utilizing Deep Learning with a hybrid architecture. The system combines YOLOv8 for object localization and a MobileNetV2-based Convolutional Neural Network (CNN) for behavior classification. Unlike previous studies focusing on skin diseases, this research targets eight specific behavior categories: sleeping, sitting, playing, eating/drinking, grooming, stretching, lying down, and walking/standing. The methodology includes multi-source data collection, Laplacian Variance preprocessing for blur filtering, and model training. Experimental results demonstrate YOLOv8's superiority in object detection with mAP@50 reaching 96.0%. In the classification stage, the CNN recorded an accuracy of 54.21%, with the Sleeping class showing the best performance. This system has been implemented as a Flask-based web platform, providing a real-time cat condition monitoring tool for owners.

Keywords: Deep Learning, YOLOv8, CNN, MobileNetV2, Cat Behavior Classification

How to Cite:

Dewanto, S. H., Listyorini, T., & Supriyati, E. (2026). Klasifikasi Perilaku Kucing Menggunakan Arsitektur Hybrid YOLOv8 dan CNN Berbasis Web. *Scan : Jurnal Teknologi Informasi dan Komunikasi*, 21(1), 1-10.

*Corresponding Author:

Email : setyajiharyodewanto@gmail.com

Alamat : Jl. Lingkar Utara UMK, Gondangmanis,
Bae, Kudus, Jawa Tengah 59327



This article is published under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

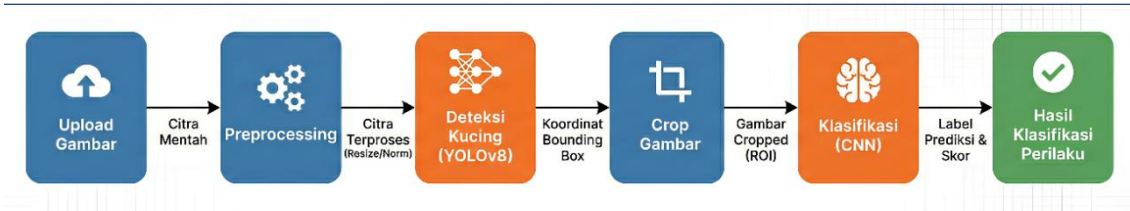
PENDAHULUAN

Kucing merupakan salah satu hewan peliharaan dengan populasi terbesar di masyarakat. Seiring dengan meningkatnya kesadaran akan kesejahteraan hewan (*animal welfare*), pemahaman mengenai kondisi fisik dan psikologis kucing menjadi prioritas bagi pemilik [1]. Indikator kesehatan hewan seringkali tercermin dari perubahan perilaku harian, seperti pola tidur, kebiasaan makan, atau aktivitas *grooming*. Sayangnya, tidak sedikit pemilik yang mengalami kesulitan dalam menafsirkan bahasa tubuh dan sinyal perilaku hewan peliharaan secara tepat. Ketidakmampuan ini berisiko menyebabkan keterlambatan dalam penanganan medis apabila terjadi gangguan kesehatan pada hewan [2].

Akselerasi teknologi *Artificial Intelligence* (AI), spesifiknya pada ranah *Deep Learning*, menawarkan solusi efisien untuk otomatisasi pemantauan. Arsitektur berbasis *Convolutional Neural Network* (CNN) telah terbukti andal tidak hanya dalam pengolahan citra, tetapi juga dalam klasifikasi pola kompleks lainnya seperti pengenalan suara huruf hijaiyah berbasis Android [3] dan analisis akurasi data audio [4]. Fleksibilitas CNN ini menjadi landasan kuat untuk penerapannya dalam analisis visual perilaku hewan. Di bidang lain, bukti efektivitas metode ini juga terlihat pada penelitian medis terkini, di mana Idris et al. [5] berhasil menerapkan metode CNN untuk identifikasi kanker paru-paru pada citra histopatologi dengan akurasi tinggi. Sejalan dengan kemajuan tersebut, teknologi serupa juga telah diterapkan dalam studi pemantauan hewan. Chen et al. [6] mengembangkan sistem pemantauan perilaku kucing berbasis Raspberry Pi menggunakan YOLOv3-Tiny dengan akurasi tinggi, namun terbatas pada enam perilaku dasar dan perangkat keras tertentu. Di sisi lain, Guo et al. [7] memanfaatkan *Deep Learning* untuk memantau perilaku ayam broiler, membuktikan efektivitas CNN dalam analisis etologi hewan.

Selain itu, metode CNN juga telah banyak dikembangkan untuk berbagai kebutuhan klasifikasi kucing, seperti pada klasifikasi spesies anjing, kucing, dan harimau [8] dan sistem rekomendasi pengobatan penyakit hewan Hidayat et al. [9]. Dalam konteks klasifikasi kucing, penelitian Gunawan et al. [10] dan Seftania et al. [11] berfokus pada klasifikasi jenis ras kucing menggunakan CNN dan YOLOv8. Sementara itu, Dalimunthe et al. [12] merancang sistem deteksi aktivitas kucing menggunakan YOLOv5 untuk notifikasi keberadaan hewan. Penelitian terbaru oleh Meilita dan Yustanti [1] menggunakan YOLOv8 untuk mendeteksi penyakit kulit pada kucing. Meskipun metode-metode tersebut efektif, mayoritas penelitian sebelumnya berfokus pada identifikasi ras, deteksi keberadaan objek, atau deteksi penyakit fisik, dan belum secara komprehensif menangani klasifikasi ragam perilaku (*behavioral analysis*) kucing pada *platform* web yang aksesibel.

Berdasarkan analisis celah penelitian (*gap analysis*) tersebut, penelitian ini bertujuan untuk membangun sistem klasifikasi perilaku kucing yang mengintegrasikan model deteksi objek YOLOv8 dan model klasifikasi CNN (MobileNetV2). Sistem ini dirancang untuk mengenali delapan kategori perilaku spesifik: *sleeping*, *sitting*, *playing*, *eating/drinking*, *grooming*, *stretching*, *lying down*, dan *walking/standing*. Kebaruan penelitian ini terletak pada pendekatan *hybrid* yang memisahkan lokalisasi objek dan klasifikasi perilaku untuk meningkatkan akurasi pada *Region of Interest* (ROI), serta implementasinya pada platform web yang mempermudah akses bagi pengguna umum.



Gambar 1. Arsitektur Sistem *Hybrid YOLOv8* dan CNN untuk Klasifikasi Perilaku Kucing
Sumber: Data Penelitian

METODE PENELITIAN

Kerangka kerja pengembangan sistem dalam studi ini mengadopsi model *Prototype*, yaitu suatu pendekatan iteratif yang memungkinkan evaluasi dan penyempurnaan fungsi sistem dilakukan secara bertahap mulai dari identifikasi kebutuhan, pembangunan purwarupa, hingga evaluasi pengguna [13]. Pendekatan ini dipilih untuk memastikan integrasi antara model deteksi objek dan klasifikasi perilaku berjalan optimal sebelum sistem diimplementasikan sepenuhnya pada platform web.

Implementasi teknis dari purwarupa tersebut digambarkan dalam diagram alur sistem pada Gambar 1 alur kerja sistem dimulai dari tahap pra-pemrosesan citra menggunakan metode *Laplacian Variance* untuk menyaring kualitas citra. Selanjutnya, citra diproses menggunakan algoritma YOLOv8 untuk mendeteksi lokasi objek, yang kemudian dilanjutkan dengan proses *cropping* area relevan. Potongan citra tersebut akhirnya diklasifikasikan oleh model CNN MobileNetV2 untuk menentukan jenis perilaku kucing.

Pengumpulan Data

Untuk menghimpun data yang diperlukan, teknik yang diaplikasikan adalah *multi-source scraping* dari platform penyedia gambar publik, yaitu Unsplash, Pexels, dan Pixabay, untuk mendapatkan variasi visual yang beragam. Penggunaan sumber data publik yang bervariasi ini umum dilakukan dalam penelitian klasifikasi objek untuk memperkaya variasi visual data latih [14]. *Dataset* yang dikumpulkan dikelompokkan menjadi dua kategori utama sesuai kebutuhan arsitektur model *hybrid*. Pertama, dataset untuk deteksi objek yang terdiri dari 1.706 citra. Data ini dianotasi menggunakan platform Roboflow dengan label tunggal "cat" dan dibagi menjadi data latih (*train*), validasi (*val*), dan uji (*test*). Kedua, *dataset* untuk klasifikasi perilaku yang berjumlah total 3.138 citra. Data ini diklasifikasikan ke dalam delapan kelas perilaku spesifik. Pembagian dataset dilakukan secara proporsional untuk mencegah *overfitting* [15].

Arsitektur Sistem dan Pra-pemrosesan

Penelitian ini mengusulkan arsitektur sistem yang bekerja secara sekuensial atau bertingkat (*cascaded*). Tahap awal melibatkan pra-pemrosesan citra menggunakan metode Laplacian Variance untuk mendeteksi tingkat keburaman (*blur*). Rumus variansi Laplacian didefinisikan sebagai:

$$\text{Blur Score} = \sigma^2 = \frac{1}{N} \sum_{i,j} (L(i,j) - \mu)^2 \quad (1)$$

Di mana $L(i,j)$ merupakan nilai intensitas piksel hasil konvolusi citra dengan kernel Laplacian, dan μ adalah nilai rata-rata intensitas piksel dari citra Laplacian tersebut. Citra

yang memenuhi standar kualitas berdasarkan nilai *blur score* kemudian dianalisis memakai arsitektur YOLOv8 (*You Only Look Once v8*) sebagai metode lokalisasi objek, khususnya mendeteksi keberadaan kucing pada citra. Agar model memiliki ketahanan yang optimal terhadap variasi input, teknik augmentasi data berupa rotasi dan *flipping* diterapkan selama proses pelatihan [16]. Hasil dari tahap ini berupa *Bounding Box* (BBBox) yang menandai area objek kucing. Selanjutnya, area citra yang berada di dalam *Bounding Box* tersebut dipotong (*cropping*) dan diubah ukurannya (*resizing*) menjadi berukuran:

$$224 \times 224 \text{ piksel} \quad (2)$$

Ukuran ini disesuaikan dengan kebutuhan input pada tahap pemrosesan berikutnya dalam sistem klasifikasi perilaku.

Pengembangan Model Klasifikasi

Tahap akhir adalah klasifikasi perilaku mengimplementasikan model *Convolutional Neural Network* (CNN) yang berbasis pada arsitektur MobileNetV2. Penggunaan MobileNetV2 dengan teknik *Transfer Learning* dan *Depthwise Separable Convolution* dipilih karena efisiensi komputasinya yang tinggi [17]. Biaya komputasi pada arsitektur ini dihitung dengan rumus:

$$Cost = D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (3)$$

Di mana D_K merupakan ukuran kernel, di mana variabel M merepresentasikan total saluran masukan (*input channel*), sedangkan N menunjukkan saluran luaran (*output channel*). Guna menangani ketidakseimbangan data (*imbalanced data*), diterapkan teknik *Class Weighting* pada fungsi *loss* selama pelatihan [18].

Evaluasi Model

Skenario pengujian dilakukan menggunakan lingkungan komputasi *cloud* Google Colab dengan dukungan akselerasi perangkat keras GPU Tesla T4 [19]. Kinerja model deteksi YOLOv8 dievaluasi menggunakan metrik standar meliputi parameter *Precision*, *Recall*, serta nilai *Mean Average Precision* (mAP) yang diukur menggunakan *threshold Intersection over Union* (IoU) sebesar 0.5. Sementara itu, model klasifikasi CNN dievaluasi berdasarkan akurasi keseluruhan (*Overall Accuracy*) dan *F1-Score* per kelas. Analisis kesalahan prediksi dilakukan menggunakan *Confusion Matrix* [20].

HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengukur kinerja masing-masing model secara terpisah maupun secara terintegrasi.

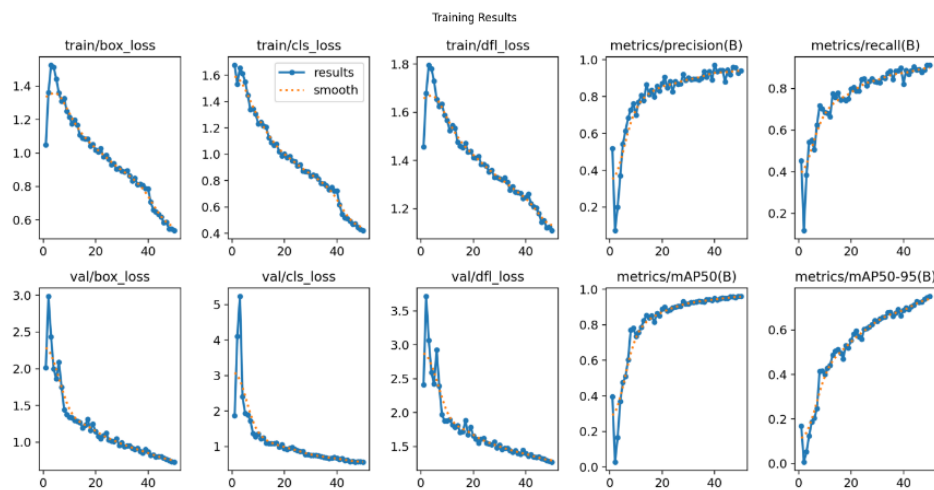
Hasil Pelatihan Model Deteksi (YOLOv8)

Pelatihan model deteksi dilakukan selama 50 *epoch*. Hasil evaluasi pada data validasi menunjukkan performa yang sangat tinggi, seperti yang tertera secara rinci pada Tabel 1.

Tabel 1
Hasil Kinerja Model Deteksi Objek YOLOv8 pada Data Validasi

Metrik	Nilai	Keterangan
Precision (P)	0.941	Tingkat ketepatan model dalam mendeteksi positif (sedikit <i>False Positive</i>).
Recall (R)	0.911	Sensitivitas model atau kemampuannya menemukan seluruh objek kucing dalam citra.
mAP@50	0.960	Nilai presisi rata-rata yang dihitung saat ambang batas <i>Intersection over Union</i> (IoU) mencapai 0.5.
mAP@50-95	0.752	Rata-rata presisi pada berbagai tingkat IoU (0.5 hingga 0.95), menunjukkan akurasi posisi kotak deteksi.
<i>Inference Time</i>	10.0 ms	Rata-rata waktu yang dibutuhkan untuk memproses satu gambar (kecepatan deteksi).

Sumber: Data Penelitian

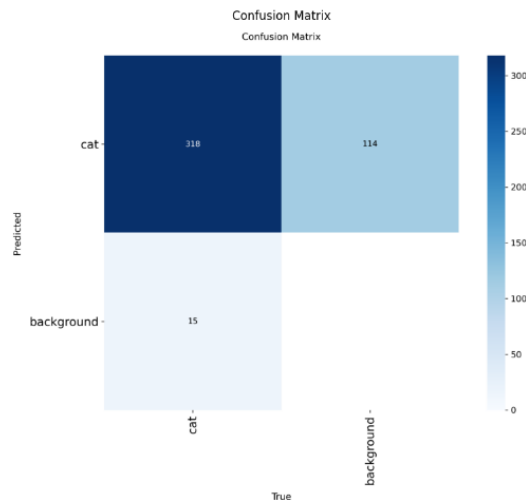


Gambar 2. Training Loss & Metrics

Sumber: Data Penelitian

Tingginya nilai mAP@50 (96%) menunjukkan bahwa YOLOv8 sangat efektif dalam membedakan objek kucing dari latar belakang. Kurva pelatihan pada Gambar 2 menunjukkan penurunan *loss* yang stabil, mengindikasikan model tidak mengalami *overfitting*.

Analisis lebih lanjut menggunakan *Confusion Matrix* (Gambar 3) memperlihatkan bahwa model berhasil mendeteksi 318 *instance* kucing dengan benar dan hanya melewati 15 *instance* (*False Negative*).



Gambar 3. Confusion Matrix

Sumber: Data Penelitian

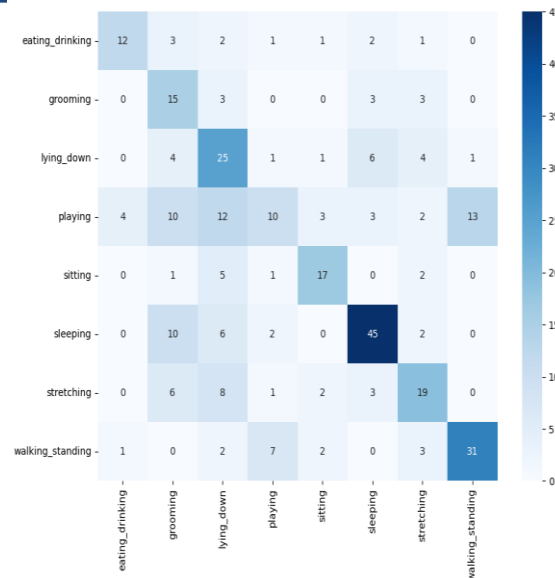
Tabel 2
Klasifikasi CNN per Kelas

Kelas	Precision	Recall	F1-Score
<i>Sleeping</i>	0.73	0.69	0.71
<i>Walking/Standing</i>	0.69	0.67	0.68
<i>Sitting</i>	0.65	0.65	0.65
<i>Eating/Drinking</i>	0.71	0.55	0.62
<i>Stretching</i>	0.53	0.49	0.51
<i>Lying Down</i>	0.40	0.60	0.48
<i>Grooming</i>	0.31	0.62	0.41
<i>Playing</i>	0.43	0.18	0.25
Rata-rata (<i>Macro Avg</i>)	0.55	0.56	0.54

Sumber: Data Penelitian

Hasil Klasifikasi Perilaku (CNN)

Model CNN dievaluasi pada data uji terpisah. Akurasi keseluruhan (*Overall Accuracy*) yang dicapai adalah 54,21%. Meskipun lebih rendah dibandingkan deteksi objek, hasil ini signifikan mengingat kompleksitas klasifikasi 8 kelas perilaku yang memiliki kemiripan visual tinggi. Hasil ini sejalan dengan karakteristik arsitektur MobileNetV2 yang mengutamakan efisiensi resource dibandingkan kedalaman jaringan yang sangat kompleks, sehingga cocok untuk implementasi pada aplikasi penerjemah atau sistem real-time lainnya [21], [22]. Rincian performa untuk setiap kelas disajikan pada Tabel 2. Berdasarkan Tabel 2, kelas *Sleeping* memiliki performa terbaik (F1-Score 0.71), sedangkan kelas *Playing* memiliki performa terendah (0.25). Analisis kesalahan dapat dilihat pada *Confusion Matrix* di Gambar 4.



Gambar 4. Confusion Matrix
Sumber: Data Penelitian

Kesalahan klasifikasi terbesar terjadi antara kelas *Playing*, *Lying Down*, dan *Grooming*. Hal ini disebabkan oleh ambiguitas pose, kucing yang sedang bermain seringkali berada dalam posisi berbaring atau berguling yang mirip dengan *Lying Down*.

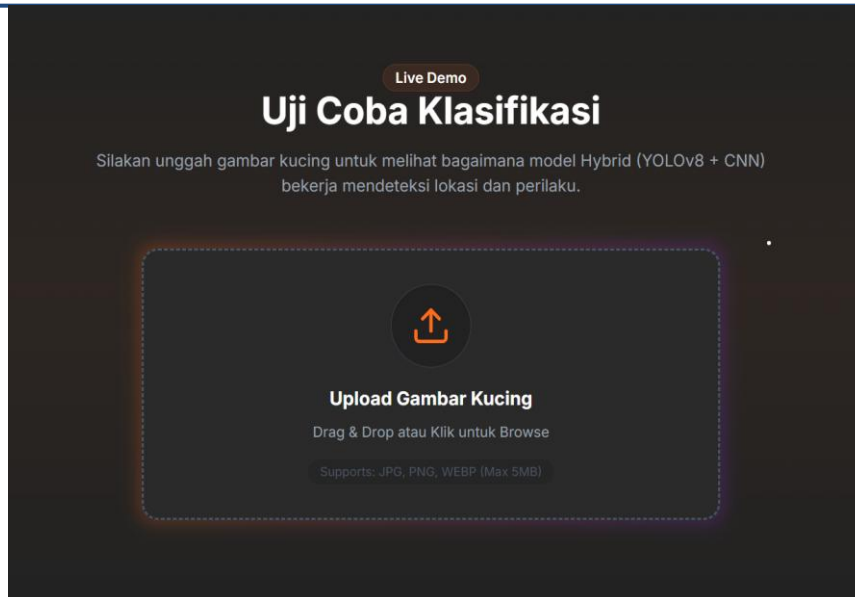
Implementasi dan Analisis Sistem Terintegrasi

Sistem klasifikasi perilaku kucing dibangun dengan mengintegrasikan model YOLOv8 dan CNN MobileNetV2 ke dalam satu *pipeline* pemrosesan sekuensial pada platform web. Mekanisme integrasi ini mengikuti fungsi matematis Kode 4., di mana hasil deteksi objek menjadi input bagi proses klasifikasi.

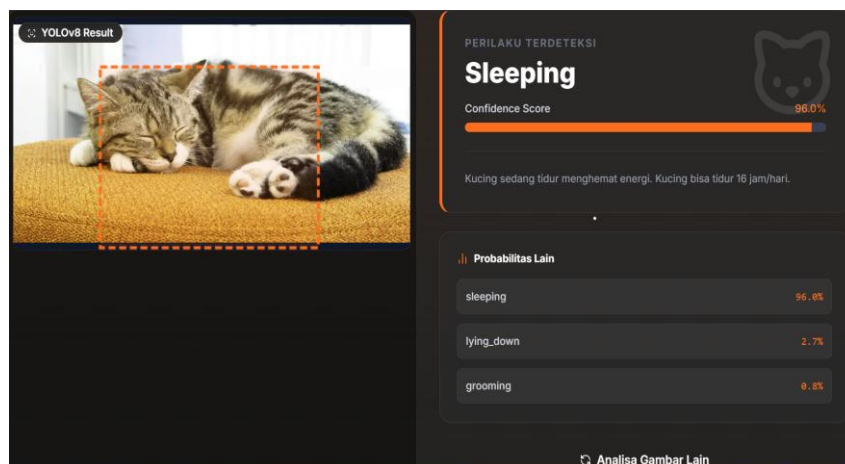
$$y = F_{CNN} \left(\text{Crop}(F_{YOLO}(I)) \right) \quad (4)$$

Kinerja sistem hibrida ini sangat bergantung pada keberhasilan tahap pertama (deteksi). Berdasarkan hasil pengujian, model YOLOv8 memiliki nilai *Recall* sebesar 91,1%. Secara teoritis, probabilitas keberhasilan sistem dalam memberikan prediksi perilaku yang benar adalah hasil perkalian antara sensitivitas detektor dan akurasi klasifikasi. Meskipun akurasi gabungan tampak moderat, sistem ini memiliki keunggulan signifikan dalam meminimalisir *False Positive* dari latar belakang (*background*). Jika dibandingkan dengan penggunaan CNN secara langsung pada citra penuh (*full image*) tanpa *cropping* YOLO, model akan kesulitan fokus pada objek kucing, terutama pada latar belakang yang kompleks. Hal ini relevan dengan temuan Ayhallansyah et al. [23] yang menekankan pentingnya ekstraksi fitur yang akurat dalam pemrosesan citra.

Sistem diimplementasikan menggunakan kerangka kerja Flask. Pendekatan berbasis web ini memberikan fleksibilitas akses yang lebih luas bagi pengguna dibandingkan aplikasi *native*, sebagaimana diterapkan juga pada sistem deteksi kualitas pangan berbasis CNN [24]. Pada antarmuka pengguna, hasil penggabungan ditampilkan secara visual. Gambar asli yang diunggah pengguna diberi *overlay* berupa kotak pembatas (*bounding box*) dari YOLOv8 untuk menunjukkan lokasi kucing, diikuti dengan label perilaku dan skor kepercayaan (*confidence score*) dari CNN MobileNetV2.



Gambar 5. Halaman Uji Coba Klasifikasi Perilaku Kucing (*Upload Gambar*)
Sumber: Data Penelitian



Gambar 6. Tampilan Hasil Deteksi YOLOv8 dan Klasifikasi Perilaku oleh CNN
Sumber: Data Penelitian

SIMPULAN

Penelitian ini berhasil mengembangkan sistem klasifikasi perilaku kucing berbasis web menggunakan metode *hybrid* YOLOv8 dan CNN MobileNetV2 yang terintegrasi secara efektif. YOLOv8 terbukti sangat andal sebagai *ekstraktor Region of Interest* (ROI) karena mampu mendeteksi objek dengan akurat, ditunjukkan oleh mAP@50 sebesar 96,0% dan *Recall* 91,1%, sehingga proses klasifikasi dapat dilakukan pada area yang relevan. Sementara itu, MobileNetV2 mampu mengklasifikasikan delapan perilaku kucing dengan akurasi 54,21%, namun masih menghadapi tantangan pada perilaku yang bersifat dinamis seperti *playing* karena citra tunggal belum cukup menangkap informasi gerak (temporal). Implementasi pada platform web juga berhasil mengintegrasikan deteksi *blur*, deteksi objek, dan klasifikasi dalam satu alur kerja yang efisien dan mudah digunakan, sehingga berpotensi dimanfaatkan untuk pemantauan perilaku kucing secara praktis dengan biaya lebih rendah. Meski demikian, keterbatasan utama

penelitian ini lebih dipengaruhi oleh pemilihan *input* berbasis gambar dan variasi kondisi *dataset* (pencahayaan, sudut, latar) yang dapat menurunkan ketepatan klasifikasi pada kelas tertentu. Karena itu, penelitian lanjutan disarankan menggunakan data video atau metode berbasis temporal serta memperluas *dataset* yang lebih beragam dan mendekati kondisi nyata agar performa sistem lebih stabil dan dapat diandalkan di berbagai situasi.

DAFTAR PUSTAKA

- [1] B. Meilita and W. Yustanti, "Sistem deteksi penyakit kulit kucing menggunakan algoritma You Only Look Once (YOLO) v8," *Journal of Emerging Information Systems and Business Intelligence*, vol. 5, no. 2, pp. 178–188, 2024.
- [2] I. Y. Pangestu and S. R. Ramadhani, "Perancangan Sistem Deteksi Penyakit Kulit Pada Kucing Menggunakan Deep Learning Berbasis Android," *Teknika*, vol. 12, no. 3, pp. 173–182, 2023.
- [3] F. I. Muqith, E. Supriyati, and T. Listyorini, "Klasifikasi pengucapan huruf Hijaiyah berbasis Android menggunakan CNN dengan fitur mel-spectrogram," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 10, no. 1, pp. 67–78, 2025.
- [4] F. N. Rahman, T. Listyorini, and E. Supriyati, "Analisis Akurasi Cnn Pada Data Olah Suara Manusia Menggunakan Parameter Koefisien Mfcc Dan Max Length," *Jurnal Digit: Digital of Information Technology*, vol. 15, no. 1, pp. 1–9, 2025.
- [5] M. Idris, F. T. Anggraeny, and R. Mumpuni, "Identifikasi Kanker Paru-Paru Pada Gambar Histopatologi Menggunakan Metode Convolutional Neural Network," *Scan: Jurnal Teknologi Informasi dan Komunikasi*, vol. 18, no. 3, pp. 37–44, 2023.
- [6] R.-C. Chen, V. S. Saravanarajan, and H.-T. Hung, "Monitoring the behaviours of pet cat based on YOLO model and raspberry Pi," *International Journal of Applied Science and Engineering*, vol. 18, no. 5, pp. 1–12, 2021.
- [7] Y. Guo, S. E. Aggrey, P. Wang, A. Oladeinde, and L. Chai, "Monitoring behaviors of broiler chickens at different ages with deep learning," *Animals*, vol. 12, no. 23, p. 3390, 2022.
- [8] M. Murdifi and S. Uyun, "Klasifikasi Hewan Anjing, Kucing, dan Harimau Menggunakan Metode Convolutional Neural Network (CNN)," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 10, no. 3, pp. 331–340, Sep. 2025.
- [9] F. Hidayat, M. Z. Sarwani, and R. Hariyanto, "Rekomendasi Pengobatan Pada Penyakit Kucing Menggunakan Metode Decision Tree (Studi Kasus: Klinik Drh. Panti Absari)," *INTEGER: Journal of Information Technology*, vol. 9, no. 2, 2024.
- [10] R. Gunawan, D. M. I. Hanafie, and A. Elanda, "Klasifikasi Jenis Ras Kucing Dengan Gambar Menggunakan Convolutional Neural Network (CNN)," *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 18, no. 4, pp. 1–8, 2024.
- [11] P. D. A. D. Seftania, M. A. D. Widyadara, and D. Swanjaya, "Implementasi Yolov8 Untuk Klasifikasi Jenis Kucing," in *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 2025, pp. 679–686.
- [12] M. B. P. Dalimunthe, E. Putra, and S. Harahap, "Deteksi Aktivitas Kucing di Rumah Menggunakan Kamera Berbasis Pengolahan Citra Digital," *Innovative: Journal Of Social Science Research*, vol. 5, no. 2, pp. 3505–3512, 2025.

- [13] F. Fenando, "Implementasi E-Commerce Berbasis Web pada Toko Denia Donuts Menggunakan Metode Prototype," *JUSIFO (Jurnal Sistem Informasi)*, vol. 6, no. 2, pp. 66–77, 2020.
- [14] M. I. Rahayu, F. Faiqunisa, and N. Nugraha, "Klasifikasi Ras Kucing Menggunakan Metadata Data Set Kaggle Dengan Framework Yolo V5," *Jurnal Penelitian dan Pengembangan Teknologi Informasi dan Komunikasi*, vol. 12, no. 1, pp. 14–18, 2023.
- [15] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," Jan. 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [16] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," *arXiv preprint arXiv:2204.08610*, 2022.
- [17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [18] D. Utomo and T. F. Utomo, *Deep Learning Dasar Teori dan Implementasi dengan Python dan Keras: Diandra Kreatif*. Penalicious, 2025.
- [19] E. Bisong, *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.
- [20] S. Saepudin, N. Sujana, M. M. Mutoffar, and A. A. Haryanto, "Analisis Kinerja YOLOv8 Optimalisasi Roboflow untuk Deteksi Ekspresi Wajah Emosional Dengan Machine Learning," *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 6, no. 2, pp. 115–124, 2024.
- [21] A. Hadhiwibowo, S. R. Asri, and R. A. Dinata, "Penerapan Convolutional Neural Network dengan Arsitektur Mobilenetv2 Pada Aplikasi Penerjemah dan Pembelajaran Bahasa Isyarat," *TIN: Terapan Informatika Nusantara*, vol. 4, no. 8, pp. 518–523, Jan. 2024.
- [22] M. A. Pratama, "Implementasi Arsitektur MobileNetV2 untuk Deteksi Penyakit Antraknosa dan Busuk Buah pada Cabai Rawit," in *Seminar Nasional Teknologi & Sains*, 2025, pp. 501–506.
- [23] S. F. Ayhallansyah, M. A. Ardiansyah, M. A. Fawwazansa, and F. T. Anggraeny, "Deteksi Gambar Palsu Menggunakan Deep Learning," *Scan: Jurnal Teknologi Informasi dan Komunikasi*, vol. 19, no. 3, pp. 156–162, 2025.
- [24] F. Paramudita and M. I. Zulfa, "Aplikasi Android pendeteksi kualitas beras berbasis machine learning menggunakan metode Convolutional Neural Network," *Jurnal Pendidikan Dan Teknologi Indonesia*, vol. 3, no. 7, pp. 297–305, 2023.