https://jurnal.pustakagalerimandiri.co.id/index.php/pustakadata
DOI: https://doi.org/10.55382/jurnalpustakadata.v5i2.1449

Jurnal Pustaka Data
Pusat Akses Kajian Database, Analisa Teknologi, dan Arsitektur Komputer
Vol. 5 No. 2 (2025) 391 – 397
E ISSN: 2809-7483

Deteksi Kesehatan Kucing Menggunakan Arsitektur VGG-16 Berbasis Website

Muhammad Rizki Setyawan[1], Fajar Rahardika Bahari Putra[2*], Rendra Soekarta[3], Yolanda Iriana Manurung[4]
Teknik Informatika, Universitas Muhammadiyah Sorong, Sorong, Indonesia
[1]Rizki@um-sorong.ac.id, [2*]fajar_rbp@um-sorong.ac.id, [3]Rendrasoekarta@email.com, [4]yolandaIrmanurung@gmail.com

Abstract

Cat owners often face difficulties in recognizing the early signs of their pets' health conditions. Limited knowledge and restricted access to veterinarians often lead to delays in disease treatment. Therefore, a system is needed that can quickly and accurately identify the health status of cats. As a solution to this problem, a web-based cat health classification system was developed using a Convolutional Neural Network (CNN) with the VGG-16 architecture. The dataset was collected through web scraping and then processed using preprocessing techniques. The CNN model was trained and evaluated using a confusion matrix and classification report. To ensure that all features function properly, system testing was conducted using the Blackbox Testing method. The training results showed that the developed CNN model achieved an accuracy of 97%, with high precision, recall, and F1-score values. The web-based system allows users to upload cat images for analysis without requiring additional applications. The results of this study indicate that the system can increase cat owners' awareness of their pets' health and help them quickly assess their cats' condition, enabling faster and more accurate treatment.

Keywords: CNN, Cats, Health, VGG-16, Website

Abstrak

Pemilik kucing sering mengalami kesulitan dalam mengenali kondisi kesehatan hewan peliharaan secara dini karena keterbatasan pengetahuan dan akses terhadap layanan dokter hewan. Kondisi tersebut dapat menyebabkan keterlambatan penanganan penyakit. Untuk mengatasi masalah ini, penelitian ini mengembangkan sistem klasifikasi kesehatan kucing berbasis web yang memanfaatkan Convolutional Neural Network (CNN) dengan arsitektur VGG-16. Sistem dirancang untuk mengidentifikasi dua kelas, yaitu kucing sehat dan kucing sakit. Dataset diperoleh melalui teknik web scraping dengan total 2.096 gambar, terdiri dari 1.672 gambar kucing sehat dan sisanya gambar kucing sakit. Data dibagi menjadi 80% data latih, 10% data validasi, dan 10% data uji, kemudian diproses melalui tahap preprocessing untuk menyesuaikan ukuran, format, serta normalisasi piksel. Model CNN dilatih dan dievaluasi menggunakan confusion matrix dan classification report. Hasil penelitian menunjukkan bahwa model mencapai akurasi 97% serta nilai precision, recall, dan F1-score yang tinggi pada kedua kelas, sehingga mampu melakukan klasifikasi dengan tingkat kesalahan yang rendah. Selain itu, pengujian sistem menggunakan metode Blackbox Testing memastikan seluruh fitur berjalan sesuai kebutuhan. Sistem berbasis web ini memungkinkan pengguna mengunggah gambar kucing secara langsung tanpa instalasi tambahan. Penelitian ini menunjukkan bahwa sistem yang dikembangkan dapat membantu pemilik kucing melakukan deteksi dini kesehatan hewan peliharaan secara cepat, praktis, dan akurat.

Kata kunci: CNN, Kucing, Kesehatan, VGG-16, Website.

Submitted : 21-11-2025 | Reviewed : 02-12-2025 | Accepted : 10-12-2025

1. Pendahuluan

Kucing merupakan salah satu hewan peliharaan yang paling banyak dipelihara di berbagai belahan dunia [1]. Popularitas ini didukung oleh sifat kucing yang jinak, menggemaskan, serta mampu beradaptasi dengan lingkungan manusia [2]. Namun demikian, kucing tetap rentan terhadap berbagai masalah kesehatan, baik ringan maupun penyakit serius yang dapat membahayakan keselamatannya [3].

Permasalahan utama yang sering terjadi adalah kurangnya pengetahuan pemilik dalam mengenali gejala awal penyakit. Banyak pemilik baru menyadari perubahan kondisi kesehatan kucing ketika gejala sudah parah, sehingga memperumit proses pengobatan dan meningkatkan risiko komplikasi [4]. Kunjungan ke dokter hewan umumnya dilakukan ketika kondisi sudah memburuk, yang tidak hanya menambah biaya perawatan tetapi juga menurunkan peluang kesembuhan. Oleh karena itu, diperlukan suatu sistem yang dapat membantu pemilik mendeteksi kondisi kesehatan kucing sejak dini [5], [6].

Kemajuan teknologi Artificial Intelligence (AI) telah memberikan kontribusi signifikan dalam bidang kesehatan, termasuk kedokteran hewan [7]. Berbagai penelitian menunjukkan bahwa Convolutional Neural Network (CNN) efektif dalam menganalisis citra dan mengklasifikasikan objek dengan tingkat akurasi tinggi [8], [9]. Namun, penelitian terkait klasifikasi kondisi kesehatan kucing berbasis citra masih terbatas, terutama yang terintegrasi ke dalam platform web sehingga dapat digunakan secara langsung oleh masyarakat. Inilah research gap yang menjadi dasar penelitian ini.

VGG-16 dipilih sebagai model utama karena memiliki arsitektur yang stabil, mudah diimplementasikan, dan telah terbukti memberikan performa yang baik dalam berbagai penelitian klasifikasi gambar [10], [11], [12]. Dibandingkan arsitektur modern seperti ResNet atau EfficientNet, VGG-16 lebih sederhana dan cocok digunakan pada dataset dengan tingkat kompleksitas sedang, seperti gambar kucing dari internet. Untuk memastikan sistem dapat diakses dengan mudah tanpa instalasi aplikasi tambahan, penelitian ini mengimplementasikan model ke dalam platform berbasis website. Website memungkinkan pengguna mengunggah gambar langsung melalui browser, menjadikannya solusi yang praktis dan ramah pengguna [13], [14].

Dengan adanya sistem klasifikasi kesehatan kucing berbasis web ini, pemilik kucing diharapkan dapat mendeteksi masalah kesehatan lebih cepat, mengurangi keterlambatan penanganan, dan meningkatkan kesejahteraan hewan. Berdasarkan permasalahan tersebut, tujuan penelitian ini adalah mengembangkan sistem klasifikasi kesehatan kucing berbasis web menggunakan model VGG-16 yang mampu mengidentifikasi kondisi kesehatan kucing secara akurat berdasarkan gambar yang diunggah pengguna.

Berdasarkan masalah yang telah dijelaskan, tujuan dari penelitian ini adalah untuk mengembangkan sistem klasifikasi kesehatan kucing berbasis web dengan menggunakan model VGG-16. Sistem ini diharapkan mampu mengidentifikasi kondisi kesehatan kucing dengan akurasi tinggi berdasarkan gambar yang diunggah oleh pengguna, sehingga dapat menjadi alat bantu yang efektif bagi pemilik kucing dalam menjaga kesehatan hewan peliharaan mereka.

2. Metode Penelitian

Metode penelitian ini terdiri dari dua bagian utama, yaitu pengembangan model CNN dan pengembangan aplikasi web, namun keduanya saling terintegrasi melalui proses deployment model. Revisi ini memberikan penjelasan yang lebih terstruktur, hubungan antar-metode, serta detail teknis yang sebelumnya belum dijelaskan, dapat dilihat pada Gambar 1 sebagai alur penelitian [15].

Gambar 1. Alur Penelitian

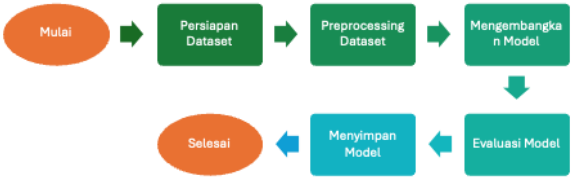
Berikut adalah penjelasan mengenai alur penelitian yang terdapat pada Gambar 1:

2.1 Pengumpulan Data

Pada tahap ini, data dikumpulkan melalui studi literatur, observasi, dan scraping dari internet. Dataset terdiri dari dua kelas: kucing sehat dan kucing sakit. Semua data diolah menjadi format gambar dengan ukuran seragam untuk keperluan pelatihan model CNN.

2.2 Pengembangan Model CNN

Pengembangan model dilakukan menggunakan Google Colab sebagai platform notebook dengan bahasa Python serta framework TensorFlow/Keras. Pada tahap ini, peneliti mengembangkan model CNN menggunakan arsitektur VGG-16, mengikuti alur kerja (pipeline) Machine Learning modern. Proses pengembangan model CNN ditunjukkan pada Gambar 2.



Gambar 2. Pengembangan Model

Submitted : 21-11-2025 | Reviewed : 02-12-2025 | Accepted : 10-12-2025

Adapun penjelasan dari pengembangan CNN pada Gambar 2 sebagai berikut [16]:

a. Persiapan *Dataset*

Dalam penelitian ini, peneliti menggunakan 2 jenis kelas yaitu kucing sehat dan kucing sakit yang di kumpulkan dengan cara scraping data pada media internet. Data dibagi menjadi:

1. 70% data latih
2. 15% data validasi
3. 15% data uji

b. *Preprocessing* Dataset

Pada tahap ini, peneliti melakukan preprocessing data, seperti mengubah ukuran gambar sesuai kebutuhan atau memutar gambar menggunakan *library ImageDataGenerator* yang tersedia di TensorFlow. Dataset yang telah terkumpul kemudian dibagi menjadi tiga bagian: data latih, data uji, dan data validasi. Preprocessing mencakup:

1. Resize gambar menjadi 224×224 pixel
2. Normalisasi pixel (0–1)
3. Augmentasi (rotation, zoom, flip) menggunakan ImageDataGenerator.

c. Mengembangkan Model

Pada tahap ini, peneliti membangun model CNN dengan menggunakan arsitektur VGG-16. Model ini akan dilatih dengan menggunakan dataset yang terdiri dari data latih dan data validasi untuk mengenali fitur-fitur yang membedakan kucing sehat dan kucing sakit. Parameter pelatihan:

1. Epoch: 25
2. Batch size: 32
3. Optimizer: Adam (lr = 0.0001)
4. Loss function: categorical crossentropy
5. Hardware: Google Colab GPU (T4)

d. Evaluasi Model

Model yang telah dibangun akan dievaluasi untuk menghasilkan confusion matrix dengan menggunakan data validasi. Evaluasi dilakukan dengan metrik:

1. Akurasi
2. Precision (macro & micro)
3. Recall (macro & micro)
4. F1-score
5. Confusion matrix
6. Classification report lengkap

e. Menyimpan Model

Model disimpan dalam format .h5 untuk digunakan pada aplikasi web, namun karena PHP tidak dapat meload file tersebut secara langsung, maka dibutuhkan komponen bridging berupa API berbasis Python (misalnya Flask atau FastAPI) yang berfungsi sebagai perantara untuk melakukan pemanggilan model; PHP akan mengirimkan data gambar ke API tersebut, Python memprosesnya menggunakan model .h5, kemudian hasil prediksi dikembalikan ke PHP dalam bentuk JSON sehingga model tetap dapat digunakan pada website tanpa harus di-load langsung oleh PHP.

2.3 Integrasi Model CNN ke Website

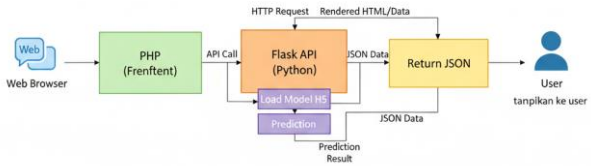
Sebelumnya tidak terdapat penjelasan mengenai hubungan CNN dan website. Pada revisi ini, integrasi dijelaskan secara eksplisit.

a. Masalah Teknis
   1) PHP tidak dapat meload model TensorFlow (.h5) secara langsung. Solusi yang dapat digunakan:
      1. PHP mengirim gambar → Flask
      2. Flask memuat model .h5 → melakukan prediksi
      3. Flask mengembalikan hasil ke PHP dalam format JSON
   2) Alternatif: TensorFlow.js (TFJS)
      1. Model dikonversi dari .h5 ke .json
      2. Prediksi dilakukan langsung di browser

Penelitian ini menggunakan pendekatan Flask sebagai backend prediksi.

2.4 Arsitektur Sistem

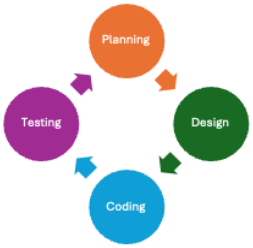
Arsitektur sistem tersebut menunjukkan alur bagaimana aplikasi web berbasis PHP memanfaatkan model deep learning (.h5) yang hanya dapat dijalankan di Python melalui Flask API. Prosesnya dimulai dari Web Browser, di mana pengguna mengirim permintaan (misalnya upload gambar atau input data). Permintaan tersebut diterima oleh PHP sebagai frontend/backend awal, lalu PHP melakukan API Call ke Flask API menggunakan protokol HTTP. Setelah itu, Flask akan memuat model .h5 dan menjalankan proses prediction untuk menghasilkan hasil klasifikasi. Output prediksi tersebut dibungkus dalam format JSON dan dikirim kembali ke PHP. Terakhir, PHP menerima JSON tersebut, mengolahnya jika perlu, dan kemudian menampilkan hasilnya kembali ke user dalam bentuk informasi yang mudah dipahami, berikut Gambar 3 arsitekturnya:



Gambar 3. Arsitektur Sistem

2.5 Pengembangan Website *Extreme Programming*

Peneliti mengembangkan website dengan menggunakan model pengembangan sistem *Extreme Programming*, yang terdiri dari beberapa tahapan yang dapat dilihat pada Gambar 4 [17].



Gambar 4. Tahapan *Extreme Programming*

Submitted : 21-11-2025 | Reviewed : 02-12-2025 | Accepted : 10-12-2025

Penjelasan mengenai setiap tahap dalam perancangan menggunakan metode *Extreme Programming* adalah sebagai berikut:

a. *Planning*

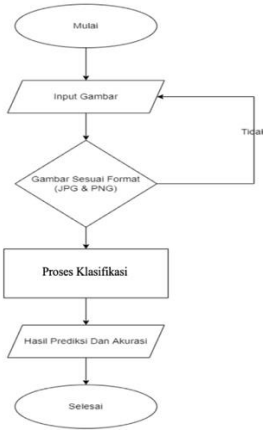
Pada tahap ini, dilakukan analisis untuk menentukan kebutuhan fungsional dan non-fungsional yang diperlukan dalam pengembangan sistem. Adapun analisis kebutuhan fungsional adalah sebagai berikut: user dapat mengakses website dan website dapat menampilkan hasil prediksi dan hasil akurasi pada user. Sedangkan untuk analasis kebutuhan non-fungsional dapat dilihat pada tabel 1.

Tabel 1. Kebutuhan Non-fungsional

No	Kebutuhan Software	Kebutuhan Hardware
1.	Windows 11 64 bit	Laptop ASUS X550Z
2.	Visual Studio dan Google Collab sebagai editor	
3.	Google Chrome sebagai browser	

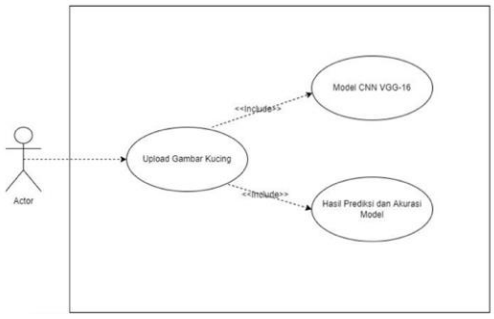
b. *Design*

Tahap ini melibatkan pembuatan flowchart dan diagram use case untuk menggambarkan alur kerja serta fungsi utama sistem. Desain flowchart dari sistem dapat dilihat pada Gambar 5.



Gambar 5. Flowchart sistem

Adapun diagram *use case* dapat dilihat pada Gambar 6.



Gambar 6. Diagram *Use case*

c. *Coding*

Pada tahap ini, website dikembangkan dengan menggunakan bahasa pemrograman PHP untuk mewujudkan desain yang telah disusun.

d. *Testing*

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah dirancang, dengan menggunakan metode *Blackbox Testing*.

2.6 Kesimpulan

Bagian kesimpulan berisi ringkasan hasil penelitian yang mencakup capaian akhir pelatihan CNN, kinerja model berdasarkan metrik evaluasi, serta keberhasilan integrasi model dengan website melalui arsitektur *Flask* + PHP.

3. Hasil dan Pembahasan

3.1 Implementasi Model CNN

3.1.1 Persiapan Dataset

Pada tahap ini, dataset yang diperoleh melalui proses scraping dari internet disimpan di Google Drive, yang kemudian dihubungkan dengan Google Colaboratory. Hasil pengumpulan dataset dapat dilihat pada Gambar 7.

Gambar 7. Pengumpulan Dataset

Pada gambar 7 menampilkan contoh dataset yang berhasil dikumpulkan, terdiri dari gambar kucing sehat dan kucing sakit dengan jumlah data sebanyak 2096.

3.1.2 Preprocessing Dataset

Data gambar kucing dibagi menjadi tiga bagian: 80% untuk data train, 10% untuk data test, dan 10% untuk data valid. Pada data yang digunakan dalam dataset, yaitu 'kucing sehat', dan 'kucing sakit'. Hasil pada pembagian data penelitian dapat dilihat di Tabel 2.

Tabel. 2 Pembagian dataset

Kelas	Train	Test	Valid	Jumlah Dataset
Kucing Sehat Kucing Sakit	1337	167	168	1672

Selanjutnya dataset yang telah dikumpulkan kemudian diproses melalui tahap preprocessing, sebagaimana ditampilkan pada Gambar 8.

Gambar 8. Pre Processing Dataset

Pada Gambar 8 menampilkan kode dan hasil preprocessing yang diterapkan meliputi perubahan ukuran gambar menjadi 128×128 piksel dengan tiga kanal warna (RGB) dan penggunaan ukuran batch sebesar 64. Selain itu, dilakukan rescaling dengan faktor 1/255 untuk menormalkan nilai piksel dari rentang 0–255 menjadi 0–1, yang bertujuan untuk meningkatkan kinerja model. Mode klasifikasi yang digunakan adalah categorical, di mana label dikonversi ke dalam format one-hot encoding, sehingga sesuai untuk klasifikasi multi-kelas. Data train dan data test juga diacak (shuffle) untuk meningkatkan generalisasi model.

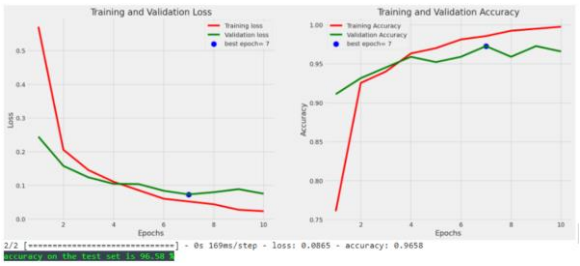
3.1.3 Mengembangkan Model CNN

Dalam penelitian ini, penulis menggunakan arsitektur model CNN, yaitu VGG-16 yang dikembangkan dapat dilihat pada Gambar 9.

Gambar 9. Model CNN

Pada Gambar 9 menunjukkan pengembangan model dengan base model VGG-16 sebagai fitur ekstraktor. Setelah itu, ditambahkan Conv2D (32 filter, 3×3, ReLU), MaxPooling2D (2×2) untuk reduksi dimensi, serta Dropout (0.5) untuk mencegah overfitting. Model dikompilasi dengan Adam optimizer, categorical_crossentropy, dan metrik akurasi.

Model yang telah dikembangkan kemudian dilatih menggunakan dataset yang terbagi menjadi tiga bagian, yaitu 80% untuk data latih, 10% untuk data uji, dan 10% untuk data validasi. Pembagian ini merupakan langkah penting dalam machine learning untuk memastikan bahwa model tidak hanya terlatih dengan baik, tetapi juga dievaluasi secara optimal guna menghindari overfitting. Hasil dari proses pelatihan ditampilkan pada Gambar 10.



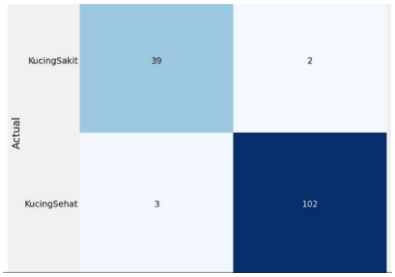
Gambar 10. Hasil Latih Model

Gambar 10 menunjukkan hasil pelatihan model melalui dua grafik, yaitu Training and Validation Loss (kiri) dan Training and Validation Accuracy (kanan). Grafik pertama menggambarkan perubahan nilai loss selama pelatihan, di mana garis merah menunjukkan training loss yang menurun tajam di awal dan terus berkurang, menandakan bahwa model semakin belajar dari data latih. Sementara itu, validation loss (garis hijau) juga menurun tetapi cenderung stabil setelah beberapa epoch, dengan titik biru menandai epoch terbaik (ke-7) sebagai kemungkinan titik optimal sebelum overfitting terjadi.

Pada grafik kedua, training accuracy (garis merah) meningkat pesat hingga mendekati 100%, sedangkan validation accuracy (garis hijau) juga meningkat meskipun sedikit fluktuatif setelah epoch ke-7, menunjukkan potensi kecil overfitting. Secara keseluruhan, model mencapai akurasi 96,58% pada test set, dengan nilai loss akhir sebesar 0.0865, yang menunjukkan performa klasifikasi yang sangat baik dengan kesalahan prediksi yang minim.

3.1.4 Evaluasi Model

Setelah pengujian model dilakukan menggunakan data valid, evaluasi kinerja model dilakukan menggunakan *confusion matrix* yang dapat dilihat pada Gambar 11.



Gambar 11. *Confusion Matrix*

Gambar 11 di atas menunjukkan *confusion matrix* hasil klasifikasi model terhadap dua kelas: Kucing Sakit dan Kucing Sehat. Model berhasil mengklasifikasikan 39 kucing sakit dan 102 kucing sehat dengan benar. Namun, terdapat 2 kesalahan di mana kucing sehat diklasifikasikan sebagai kucing sakit (False Positive) dan 3 kesalahan di mana kucing sakit diklasifikasikan sebagai kucing sehat (False Negative). Dengan total kesalahan hanya 5 dari 146 sampel, model menunjukkan akurasi tinggi dengan presisi dan recall yang seimbang, menjadikannya cukup andal untuk klasifikasi serupa.

Submitted : 21-11-2025 | Reviewed : 02-12-2025 | Accepted : 10-12-2025

Setelah mendapatkan confusion matrix di atas, langkah berikutnya adalah menghasilkan classification report untuk mengevaluasi performa model secara lebih rinci mencakup akurasi, presisi, recall, dan F1-score, yang digunakan untuk menilai kinerja model dalam melakukan klasifikasi. Hasil dari classification report dapat dilihat pada Gambar 12.

```
Classification Report:
----------------------
              precision    recall  f1-score   support

 KucingSakit       0.93      0.95      0.94        41
 KucingSehat       0.98      0.97      0.98       105

    accuracy                           0.97       146
   macro avg       0.95      0.96      0.96       146
weighted avg       0.97      0.97      0.97       146
```

Gambar 12. Classification Report

Berdasarkan Gambar 12, model memiliki akurasi sebesar 97%, menunjukkan kinerja yang sangat baik dalam mengklasifikasikan kondisi kucing. Untuk kelas Kucing Sakit, *precision* sebesar 0.93, *recall* 0.95, dan *F1-score* 0.94, yang berarti model cukup baik dalam mengidentifikasi kucing yang sakit dengan sedikit kesalahan prediksi. Sementara itu, kelas Kucing Sehat memiliki *precision* 0.98, *recall* 0.97, dan *F1-score* 0.98, menunjukkan performa yang sangat baik dalam mengidentifikasi kucing sehat. Nilai rata-rata (*macro avg* dan *weighted avg*) juga menunjukkan skor yang tinggi, menandakan keseimbangan model dalam mengklasifikasikan kedua kelas dengan baik.

3.1.5 Saving Model

Model yang telah dibuat disimpan dalam format .h5 dan diimplementasikan ke dalam website untuk memungkinkan prediksi langsung melalui antarmuka web.

3.2 Implementasi Website

Aplikasi berbasis web ini memiliki dua halaman utama, yaitu Halaman Utama dan Halaman Prediksi.

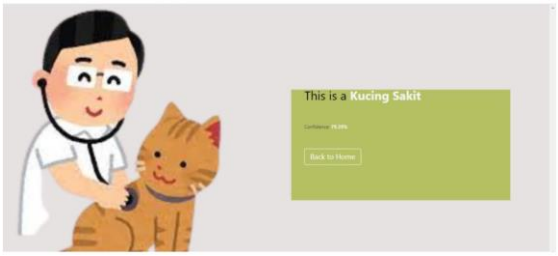
3.2.1 Halaman Utama

Halaman ini digunakan mengunggah gambar kucing melalui kotak unggah berwarna putih yang bertuliskan "*Upload an image*". Pengguna dapat memilih gambar dari perangkat mereka untuk diproses lebih lanjut oleh sistem. Tampilan Halaman Utama dapat dilihat pada Gambar 13.

Gambar 13. Tampilan Halaman Utama

3.2.2 Halaman Prediksi

Setelah pengguna mengunggah gambar dan menekan tombol "Classify", sistem akan menampilkan hasil prediksi beserta penjelasan mengenai apakah kucing dalam gambar tersebut termasuk dalam kategori sehat atau sakit, serta tingkat akurasinya. Hasil prediksi ini didasarkan pada model yang telah dilatih sebelumnya. Tampilan Halaman Prediksi dapat dilihat pada Gambar 14.



Gambar 14. Tampilan Halaman Prediksi

3.3 *Blackbox Testing*

Sistem diuji untuk memastikan bahwa aplikasi beroperasi sesuai dengan persyaratan dan desain yang telah ditentukan. Hasil blackbox testing pada website yang telah dikembangkan dapat dilihat dalam Tabel 3.

Tabel 3. Blackbox Testing

No	Fitur	Skenario Pengujian	Output yang diharapkan	Output Aktual	Hasil
1.	Halaman Utama	Pengguna mengakses halaman utama website	Halaman website ditampilkan	Halaman website ditampilkan	Berhasil
2.	Upload gambar	Pengguna mengunggah gambar	Gambar berhasil di unggah	Gambar berhasil di unggah	Berhasil
3.	Melakukan klasifikasi	Setelah mengunggah gambar, pengguna menekan tombol "Classify"	Menampilkan hasil klasifikasi	Menampilkan hasil klasifikasi	Berhasil

4. Kesimpulan

Penelitian ini berhasil membangun sistem klasifikasi kesehatan kucing berbasis web menggunakan model CNN dengan arsitektur VGG-16. Sistem dapat menganalisis gambar kucing dan memberikan prediksi kondisi sehat atau sakit secara cepat, sehingga membantu pemilik kucing melakukan deteksi dini. Model dilatih menggunakan dataset hasil scraping dan dievaluasi menggunakan confusion matrix serta classification report. Hasilnya, model mencapai akurasi 97% dengan precision, recall, dan F1-score yang tinggi sehingga cukup andal dalam membedakan kucing sehat dan sakit.

Submitted : 21-11-2025 | Reviewed : 02-12-2025 | Accepted : 10-12-2025

Meskipun demikian, penelitian ini masih memiliki keterbatasan, terutama pada dataset yang tidak seimbang dan kualitas gambar yang bervariasi. Hal ini berpotensi mempengaruhi generalisasi model. Untuk pengembangan ke depan, model dapat ditingkatkan menggunakan arsitektur yang lebih modern seperti EfficientNet, serta menambah jumlah dan kualitas dataset. Pengujian dengan data nyata dari pengguna juga perlu dilakukan untuk memastikan keandalan sistem dalam kondisi sebenarnya. Sistem berbasis web yang dibangun memudahkan pengguna karena dapat diakses tanpa instalasi aplikasi tambahan, dan diharapkan dapat menjadi langkah awal dalam penerapan AI untuk mendukung kesehatan hewan peliharaan.

Daftar Rujukan

[1] P. D. Hadi, D. A. Widhining K, and F. A. Fiolana, "Identifikasi Jenis Ras Pada Kucing Menggunakan Algoritma Support Vector Machine (SVM)," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 6, no. 1, pp. 77–86, 2024, doi: 10.26905/jasiek.v6i1.10989.

[2] N. Puspitasari, R. Rizwar, J. Jarulis, D. Darmi, and A. H. Putra, "Studi Kesejahteraan Kucing Peliharaan di Beberapa Toko Hewan Peliharaan (Pet Shop)," *BIOEDUSAINSJurnal Pendidik. Biol. dan Sains*, vol. 5, no. 2, pp. 382–390, 2022, doi: 10.31539/bioedusains.v5i2.2352.

[3] A. S. Aqila, D. C. Budinuryanto, and M. Wijaya, "Penerapan Kesejahteraan Hewan oleh Staf pada Kucing yang Dirawat Inap di Klinik Hewan di Kota Bandung," *Indones. Med. Veterinus*, vol. 9, no. 5, pp. 773–786, 2020, doi: 10.19087/imv.2020.9.5.773.

[4] D. Azura, M. Nabila, and A. S. H. Damanik, "Analisis Dampak Perilaku Sterilisasi Terhadap Kesehatan Kucing Betina dan Jantan," *J. Biol.*, vol. 1, no. 1, pp. 1–10, 2023, doi: 10.47134/biology.v1i1.1925.

[5] R. H. Kiswanto, S. Bakti, and R. M. H. Thamrin, "Rancang Bangun Sistem Pakar Diagnosa Penyakit Kucing Menggunakan Metode Backward Chaining," *J. Eksplora Inform.*, vol. 11, no. 1, pp. 67–76, 2022, doi: 10.30864/eksplora.v11i1.610.

[6] Fira Kusuma Wardana, L. D. Bakti, and K. Nurwijayanti, "Sistem Pakar Diagnosa Penyakit Pada Kucing Dengan Metode Certainty Factor Berbasis Web," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 2, no. 1, pp. 20–31, 2023, doi: 10.69916/jkbti.v2i1.14.

[7] M. R. Setyawan, F. R. B. Putra, and L. J. Fakhri, "Comparison of Accuracy Level of Certainty Factor Method and Bayes Theorem on Cattle Disease," *Ilk. J. Ilm.*, vol. 16, no. 3, pp. 343–355, 2024, doi: 10.33096/ilkom.v16i3.1943.343-355.

[8] M. Jasmine Pemeena Priyadarsini *et al.*, "Lung Diseases Detection Using Various Deep Learning Algorithms," *J. Healthc. Eng.*, vol. 2023, pp. 1–13, Feb. 2023, doi: 10.1155/2023/3563696.

[9] D. Candra, G. Wibisono, M. Ayu, and M. Afrad, "Transfer Learning model Convolutional Neural Network menggunakan VGG-16 untuk Klasifikasi Tumor Otak pada Citra Hasil MRI," *LEDGER J. Inform. Inf. Technol.*, vol. 3, no. 1, pp. 11–18, 2024, doi: doi.org/10.20895/ledger.v3i1.1387.

[10] R. Andre, B. Wahyu, and R. Purbaningtyas, "Klasifikasi Tumor Otak Menggunakan Convolutional Neural Network Dengan Arsitektur Efficientnet-B3," *J. IT*, vol. 11, no. 3, pp. 55–59, 2021, doi: https://doi.org/10.24853/justit.12.3.55-59.

[11] A. Priyatama, Z. Sari, and Y. Azhar, "Deep Learning Implementation using Convolutional Neural Network for Alzheimer's Classification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 310–217, 2023, doi: 10.29207/resti.v7i2.4707.

[12] F. R. B. Putra, M. R. Setyawan, and L. J. F. Rendra Soekarta, Nabila, "Implementasi Deep Learning Menggunakan Cnn Untuk Klasifikasi Tingkat Kematangan Buah Jeruk Berbasis Android," vol. 8, no. 1, pp. 36–43, 2023, doi: 10.51544/jurnalmi.v9i2.5462.

[13] I. Erdiansyah, M. I. N. Syahputra, and M. Dian, "Pengelolaan Diagnosis Penyakit Tanaman Padi Menggunakan Metode Teorema Bayes Management of Rice Plant Disease Diagnosis Using Bayes Theorem Method," *JIKO (JURNAL Inform. DANKOMPUTER)*, vol. 8, no. 2, pp. 393–403, 2024, doi: 10.26798/jiko.v8i2.1315.

[14] I. Umami, M. Tony Adam, and Winarti, "Perancangan Sistem Informasi Pengelolaan Surat Menyurat Berbasis Web Desa Sumberkarang," *J. Ilm. Multidisiplin*, vol. 1, no. 9, pp. 2880–2885, 2022.

[15] R. Soekarta, N. Nurdjan, and A. Syah, "Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN)," *Insect (Informatics Secur. J. Tek. Inform.*, vol. 8, no. 2, pp. 143–151, 2023, doi: doi.org/10.33506/insect.v8i2.2356.

[16] R. Soekarta, M. Fadli Hasa, and E. S. Ode, "Klasifikasi Kematangan Buah Pisang Secara Real-Time Menggunakan Convolutional Neural Network Berbasis Android," vol. 10, no. 01, pp. 11–20, 2024.

[17] M. N. Arief, M. R. Setyawan, R. Soekarta, and P. M. Simori, "Web-Based System Information Certificate Services at Klamono District Offices," *Semesta Tek.*, vol. 26, no. 2, pp. 181–193, 2023, doi: 10.18196/st.v26i2.19573.