

Klasifikasi Citra Ras Kucing Berbasis CNN dengan Metode MobileNet-V2

Ramadhan Anugrah Hermawan¹, Ichsan Taufik², Yana Aditia Gerhana³

^{1,2,3}Teknik Informatika, UIN Sunan Gunung Djati Bandung, Indonesia

1207050101@student.uinsgd.ac.id

Info Artikel

Sejarah artikel :

Diterima Mei 2025

Direvisi Juni 2025

Disetujui Juni 2025

Diterbitkan Juni 2025

ABSTRACT

This study aims to evaluate the performance of Convolutional Neural Network with MobileNet-V2 architecture in classifying images of four cat breeds. Object classification is an important field in computer vision that focuses on visual information from digital images that can be processed and manipulated by computers. The main problem is the lack of public understanding of the differences in the physical characteristics of each cat breed, which is caused by the large number of cats resulting from interbreeding with similar visual appearances. The model training process was carried out using several epochs, with each epoch tested 10 times to observe the training accuracy and validation accuracy as well as the effectiveness and stability of the model. The dataset used in this study was obtained from Kaggle and the website fiveweb.org, with a total of 4,140 images. The test results showed that epoch 100 produced a validation accuracy of 94.49%. Although this result is quite good, optimization is needed to reduce overfitting and improve the generalization ability of the model. This research contributes to the development of automated systems that can be utilized in the field of education.

Keywords : Accuracy; Deep Learning; Epoch; Cat Classification; MobileNet-V2.

ABSTRAK

Penelitian ini bertujuan untuk mengevaluasi kinerja Convolutional Neural Network dengan arsitektur MobileNet-V2 dalam mengklasifikasikan citra empat ras kucing. Klasifikasi objek merupakan bidang penting dalam computer vision yang berfokus pada informasi visual dari citra digital yang dapat diproses dan dimanipulasi oleh komputer. Permasalahan utama kurangnya pemahaman masyarakat terhadap perbedaan karakteristik fisik setiap ras kucing, yang disebabkan oleh banyaknya kucing hasil perkawinan antar ras dengan kemiripan visual yang sama. Proses pelatihan model dilakukan menggunakan beberapa epoch, dengan setiap epoch diuji sebanyak 10 kali untuk mengamati nilai akurasi pelatihan dan akurasi validasi serta efektivitas dan kestabilan model. Dataset yang digunakan dalam penelitian ini diperoleh dari Kaggle dan situs fiveweb.org dengan total 4.140 citra. Hasil pengujian menunjukkan bahwa epoch 100 menghasilkan akurasi validasi sebesar 94,49%. Meskipun hasil tersebut sudah cukup baik, optimasi diperlukan untuk mengurangi overfitting dan meningkatkan kemampuan generalisasi model. Penelitian ini berkontribusi dalam pengembangan sistem otomatis dapat dimanfaatkan dalam bidang edukasi.

Kata Kunci : Akurasi; Deep Learning; Epoch; Klasifikasi Kucing; MobileNet-V2

PENDAHULUAN

Kucing merupakan salah satu hewan peliharaan yang populer di seluruh dunia. Kurangnya pengetahuan mengenai jenis ras kucing menjadi permasalahan, apalagi banyak kucing saat ini merupakan hasil kawin persilangan. Untuk mengidentifikasi ras kucing secara akurat, diperlukan sistem yang mampu mengenali ciri-ciri fisiknya. Pemahaman ras kucing penting untuk kesehatan dan perawatan. Sistem klasifikasi otomatis berbasis gambar membantu pengambilan keputusan oleh pemilik dan dokter hewan. [1]. Kucing dengan garis keturunan tercatat disebut ras murni (pure breed), dengan sekitar 32 ras yang diakui secara internasional, seperti Anggora, Persia, Himalaya, dan Kampung [2]. Mengutip secara langsung "Penerapan Convolutional Neural Network pada Klasifikasi Jenis Ras Kucing Menggunakan ResNet50V2". Penelitian ini mengklasifikasikan 12 ras kucing menggunakan arsitektur CNN ResNet50 dengan optimasi stochastic gradient descent (SGD). Model mencapai akurasi training 99,17% dan akurasi validasi 88,13%, dengan loss masing-masing 0,3760 (training) dan 0,7857 (validasi). Evaluasi menggunakan confusion matrix, precision, recall, f1-score, dan support menunjukkan rata-rata akurasi 88% [3]. Mengutip secara langsung "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network(CNN)". Penelitian ini menyimpulkan bahwa model Xception dengan Transfer Learning dan Fine Tuning memberikan performa terbaik, mencapai akurasi 93,75%, precision 93,74%, recall 93,56%, dan f1-score 93,64% [4].

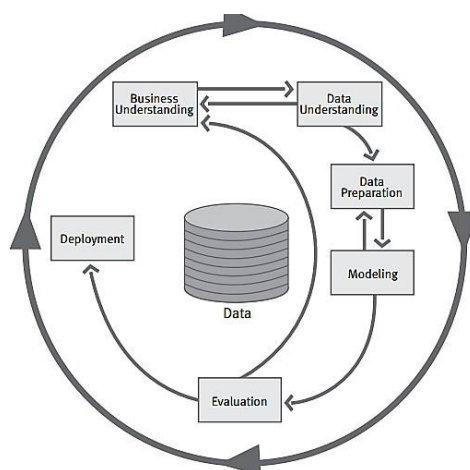
Dalam beberapa tahun terakhir, kemajuan teknologi kecerdasan buatan (AI) telah membawa perubahan signifikan di berbagai sektor, termasuk di bidang akademik dan pendidikan [5]. Google Lens adalah aplikasi inovatif dari Google yang memanfaatkan teknologi pengenalan gambar dan kecerdasan buatan untuk mengenali objek, termasuk tanaman, melalui foto dari kamera ponsel. Sementara itu, QR Code (Quick Response Code) adalah kode dua dimensi yang dapat membantu perangkat seluler untuk mengarahkan pengguna ke informasi yang relevan, seperti video tutorial dan artikel tentang manfaat tanaman [6].

Klasifikasi objek adalah aspek penting dalam Computer Vision, yang fokus pada pengembangan sistem untuk memahami informasi visual [7]. Citra digital adalah gambar yang dapat diproses oleh komputer. Gambar ini dapat dimodifikasi melalui berbagai metode, seperti pengambilan fitur, perubahan sudut pandang, atau penyesuaian ukuran. Salah satu cara untuk memanfaatkan data gambar adalah melalui klasifikasi menggunakan algoritma Convolutional Neural Network (CNN) [8]. MobileNet adalah arsitektur CNN yang efisien dan ringan, cocok untuk perangkat dengan keterbatasan daya komputasi [9]. MobileNetV2, yang dikembangkan oleh peneliti Google, adalah versi lebih lanjut dari MobileNet, menawarkan efisiensi yang lebih tinggi dengan menggunakan blok sisa terbalik dan fitur bottleneck untuk mengurangi kompleksitas jaringan [10].

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi ras kucing berbasis Convolutional Neural Network (CNN) dengan menggunakan arsitektur MobileNet-V2. Data dilatih menggunakan bahasa pemrograman Python di Google Collaboratory. Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum dan dirilis pada 1991. Bahasa ini populer karena sintaksnya yang sederhana, bersifat open source, serta memiliki library lengkap dan dukungan komunitas yang kuat. Python juga multifungsi, termasuk untuk Machine Learning dan Deep Learning. Kode Python dapat ditulis menggunakan IDE seperti VS Code, Sublime Text, PyCharm, atau platform online seperti Jupyter Notebook dan Google Collaboratory [11]. Aplikasi yang dihasilkan akan memudahkan identifikasi ras kucing secara otomatis dengan tingkat akurasi dan efisiensi yang tinggi, terutama pada perangkat mobile.

METODE

Metode penelitian ini menggunakan metode *Cross-Industry Standard Process for Data Mining (CRISP-DM)* dirancang untuk menyediakan panduan berupa tahapan-tahapan dalam proses pengumpulan data. Tahapan tersebut terdiri dari enam langkah, yaitu pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan penerapan [12]. Dilihat pada gambar 1.

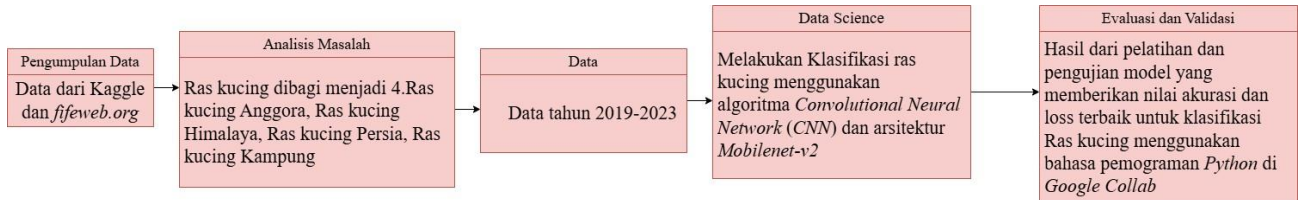


Gambar 1. Tahapan CRISP-DM

Sumber : Salma Navisa, Luqman Hakim ,Aulia Nabilah [12].

Bussines Understanding

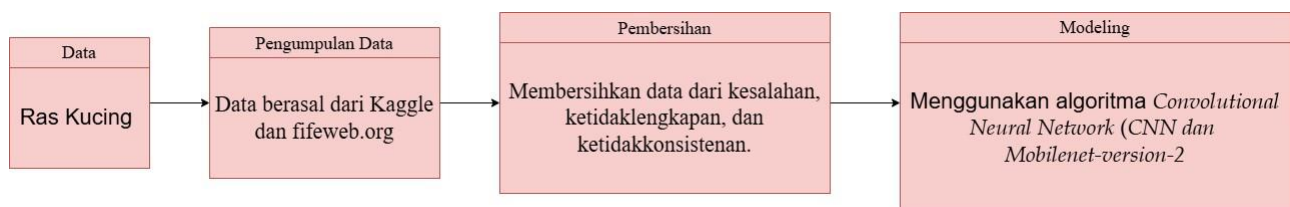
Pada tahap ini, pemahaman tentang inti dari aktivitas data yang akan dilaksanakan dan kebutuhan dari sudut pandang bisnis [13]. Dilihat pada gambar 2.



Gambar 2. Tahapan Bussines Understanding

Data Understanding

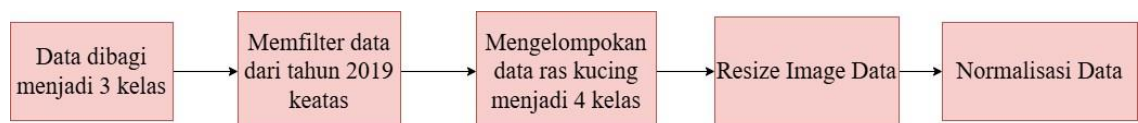
Pada tahap ini, fase di mana melibatkan pengumpulan data awal, mempelajari data untuk memahami karakteristiknya, mengidentifikasi masalah [13]. Dilihat pada Gambar 3.



Gambar 3. Tahapan Data Understanding

Data Preparation

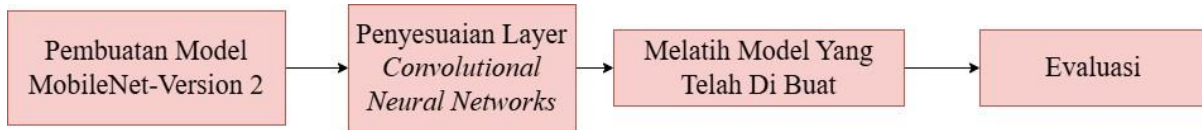
Pada tahap ini, sering disebut sebagai fase yang membutuhkan banyak tenaga kerja [13]. Aktivitas yang dilakukan melibatkan pembersihan, transformasi, dan pemilihan data yang akan digunakan untuk pelatihan. Dilihat pada Gambar 4.



Gambar 4. Tahapan Data Preparation

Modeling

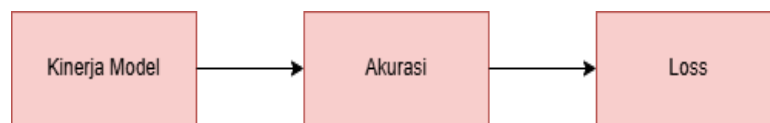
Pada tahap ini, fase menentukan tehnik yang digunakan [13]. Dilihat pada Gambar 5.



Gambar 5. Modeling

Evaluation

Pada tahap ini, mencakup analisis data yang dihasilkan dari proses pemodelan yang telah dilakukan pada fase sebelumnya [13]. Dilihat pada Gambar 6.



Gambar 6. Evaluation

Melakukan Resize Image

Proses Resize Image dilakukan dengan menyesuaikan dimensinya menjadi 224×224 piksel [17]. Diubah secara otomatis menggunakan *Source Code* yang diuji dengan *Google Collaboratory*. Dengan begitu, seluruh gambar dalam dataset akan memiliki ukuran dimensi piksel yang sama, sehingga mempermudah tahap pemrosesan data dan analisis berikutnya. *Source Code* dapat dilihat dibawah ini.

Source Code :

```

for i, class_name in enumerate ( classes ) :
    class_path = os.path.join ( dataset_dir, class_name )
    for file_name in os.listdir ( class_path ) :
        image_path = os.path.join ( class_path,
        file_name )
        image = Image.open (
        image_path ).convert ( RGB )
        image = image.resize (( 224,224 )
        x_data.append ( np.array(
        image ))
        y_data.append ( i )
  
```

Melakukan Normalisasi Data

Normalisasi data dilakukan dengan mengubah nilai X dan Y. Nilai X disesuaikan agar berada dalam rentang 0 hingga 1, sementara nilai Y dikonversi menjadi angka dari 0 hingga 7 sesuai dengan jumlah kelas yang tersedia [17]. *Source Code* dapat dilihat dibawah ini.

Source Code :

```

x_train = preprocess_input ( x_train )
x_test = preprocess_input ( x_test )
y_train = to_categorical ( y_train, 4 )
y_test = to_categorical ( y_test, 4 )
print(f"y_train shape: { y_train.shape }")
print(f"y_test shape: { y_test.shape }")

```

Membangun Model *MobileNet - Version 2*

Dalam pengujian penelitian ini, model dikembangkan menggunakan arsitektur *MobileNetV2* di *Google Collaboratory*. Model ini digunakan sebagai *feature extractor* dalam sistem *deep learning* untuk klasifikasi gambar.

Source Code :

```

input_shape = (224, 224, 3)
base_model = MobileNetV2 ( include_top=False, weights='imagenet',
input_shape=input_shape )
base_model.summary ()

```

Penyesuaian *Layer Convolutional Neural Networks*

Setelah membangun model menggunakan arsitektur *MobileNetV2*, langkah selanjutnya adalah menyesuaikan layer *Convolutional Neural Networks (CNN)* untuk meningkatkan performa dalam klasifikasi gambar. Kode berikut digunakan untuk membangun model yang telah dimodifikasi dengan penyesuaian layer agar lebih optimal.

Source Code :

```

inputs = Input ( input_shape )
x = base_model ( inputs, training = False ) x = Flatten() ( x )
x = Dense ( 256, activation = 'relu',
kernel_regularizer=tf.keras.regularizers.l2 ( 0.001 )) ( x )
x = Dropout ( 0.5 ) ( x )
outputs = Dense ( 4, activation = 'softmax' ) ( x )
model = Model ( inputs, outputs )
model.summary()

```

Melatih Model *MobileNet - Version 2*

Setelah melakukan *input layer* pada *Convolutional Neural Network (CNN)* sesuai kebutuhan, langkah berikutnya adalah melatih model *MobileNetV2* yang telah dibangun sebelumnya. Proses pelatihan ini menggunakan berbagai teknik optimasi untuk meningkatkan akurasi dan menekan nilai loss.

Source Code :

```

model.compile ( loss = 'categorical_crossentropy',
optimizer=tf.keras.optimizers.Adam (learning_rate = 0.0001 ), metrics =
['accuracy'])
# Callbacks
early_stop = EarlyStopping ( monitor = 'val_loss', patience = 20,
mode = 'min')
reduce_lr = ReduceLROnPlateau ( monitor ='val_loss', factor = 0.2,
patience = 10, min_lr = 0.00001, mode = 'min' ) # Train the model
history = model.fit (
datagen.flow ( x_train, y_train, batch_size = 32),
epochs = 1,
validation_data = ( x_test, y_test ),
callbacks = [ early_stop, reduce_lr ],
verbose = 1)

```

Deployment

Pada tahap ini, menyajikan hasil dari model yang telah diterapkan dalam proses data mining, termasuk penyampaian informasi yang diperoleh selama analisis agar mudah dipahami oleh pengguna [12]. Tahap ini mencakup pembuatan laporan setelah mengevaluasi penelitian terkait model klasifikasi Ras kucing.

HASIL DAN PEMBAHASAN

Bussines Understanding

Pada tahap ini, tujuan, persyaratan, dan batasan penelitian ditetapkan melalui analisis mendalam guna merumuskan permasalahan yang dapat diukur [14]. Banyak orang mengalami kesulitan dalam membedakan berbagai ras kucing yang ada karena jumlahnya yang sangat beragam. Dengan memanfaatkan algoritma *Convolutional Neural Network*, diharapkan sistem ini dapat menyajikan informasi yang akurat mengenai jenis kucing berdasarkan data yang telah diuji, sehingga membantu masyarakat dalam mengenali ras kucing dengan lebih mudah.

Data Understanding

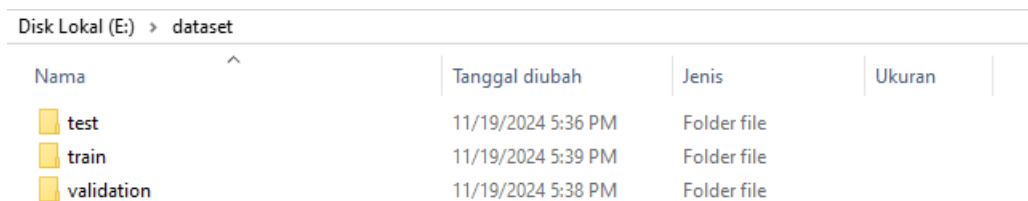
Tahap ini dimulai dengan pengumpulan data, deskripsi data, serta evaluasi kualitasnya [15]. Dalam penelitian ini, data diperoleh dari platform *Kaggle* berjumlah 2.140 Gambar dan situs *fifeweb.org* berjumlah 2.000 Gambar. Data yang digunakan merupakan kumpulan informasi yang relevan, dengan fokus pada data yang dirilis sejak tahun 2019 dan seterusnya, guna memastikan kesesuaian serta kualitas dalam proses analisis.

Data Preparation

Fase Data Preparation mencakup seluruh aktivitas yang diperlukan untuk membangun dataset yang akan digunakan sebagai input dalam pemodelan aplikasi dari data mentah [16]. Proses pembersihan data dilakukan agar data siap diolah pada tahap implementasi. Langkah-langkah yang dilakukan meliputi:

1. Membagi Data

Membagi Data ke dalam tiga kategori, yaitu Data *Train*, Data *Test*, dan Data *Validation*. Dilihat pada Gambar 7.

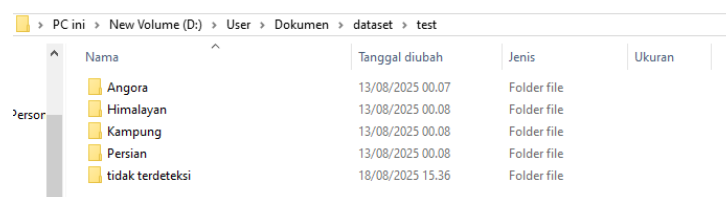


Nama	Tanggal diubah	Jenis	Ukuran
test	11/19/2024 5:36 PM	Folder file	
train	11/19/2024 5:39 PM	Folder file	
validation	11/19/2024 5:38 PM	Folder file	

Gambar 7. Membagi Data Ke Dalam Tiga Kategori

2. Mengelompokkan Jenis Kucing

Dataset dibagi menjadi lima subkelas, yang terdiri atas empat kelas utama, serta satu kelas tambahan, yaitu Tidak Terdeteksi. Kelas terakhir ini berisi sejumlah gambar hewan lain yang digunakan untuk membantu model mengenali perbedaan antara kucing dan objek non-kucing.. Dilihat pada Gambar 8.



Nama	Tanggal diubah	Jenis	Ukuran
Angora	13/08/2025 00:07	Folder file	
Himalayan	13/08/2025 00:08	Folder file	
Kampung	13/08/2025 00:08	Folder file	
Persian	13/08/2025 00:08	Folder file	
tidak terdeteksi	18/08/2025 15:36	Folder file	

Gambar 8. Dataset Dibagi Menjadi 5 Kelas

3. Data Pengujian

Data Yang Dipakai 4.140 data yang digunakan untuk pengujian.

4. Melakukan Resize Image

Proses Resize Image dilakukan dengan menyesuaikan dimensinya menjadi 224×224 piksel [17]. Diubah secara otomatis menggunakan *Source Code* yang diuji dengan

Google Collaboratory. Dengan begitu, seluruh gambar dalam dataset akan memiliki ukuran dimensi piksel yang sama, sehingga mempermudah tahap pemrosesan data dan analisis berikutnya.

5. Melakukan Normalisasi Data

Normalisasi data dilakukan dengan mengubah nilai X dan Y. Nilai X disesuaikan agar berada dalam rentang 0 hingga 1, sementara nilai Y dikonversi menjadi angka dari 0 hingga 7 sesuai dengan jumlah kelas yang tersedia [17].

Modelling

MobileNetV2 merupakan arsitektur yang dikembangkan untuk meningkatkan efisiensi dalam pemrosesan citra dengan memanfaatkan inverted residual block dan bottleneck layer. Komponen ini memungkinkan pengurangan jumlah perhitungan dibandingkan dengan versi sebelumnya, sehingga membuat model lebih ringan dan cepat dalam melakukan klasifikasi gambar [18]. Kemampuan ini membuat model sangat cocok digunakan pada perangkat dengan keterbatasan daya komputasi, seperti ponsel atau perangkat IoT.

Membangun Model *MobileNet - Version 2*

Dalam pengujian penelitian ini, model dikembangkan menggunakan arsitektur *MobileNetV2* di *Google Collaboratory*. Model ini digunakan sebagai *feature extractor* dalam sistem *deep learning* untuk klasifikasi gambar.

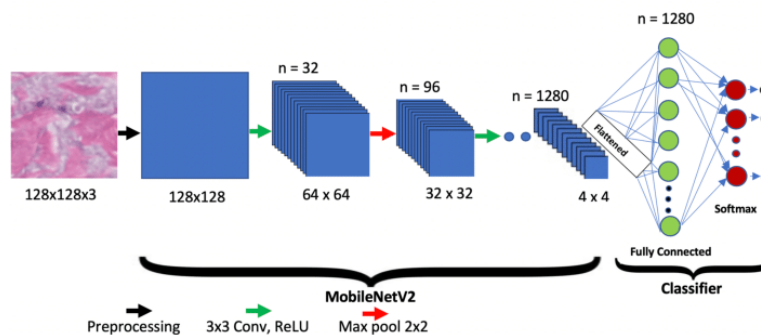
1. `input_shape = (224, 224, 3)` : Menentukan ukuran input gambar (224 × 224, 3 channel warna (RGB)).
2. `base_model = MobileNetV2 (include_top = False, weights = imagenet, input_shape = input_shape)` :
 - a. *MobileNetV2* : Memuat arsitektur model.
 - b. `include_top = False` : Menghapus lapisan *fully connected* (FC) agar model berfungsi sebagai ekstraktor fitur.
 - c. `weights = imagenet` : Menggunakan bobot pralatih dari ImageNet.
 - d. `input_shape = input_shape` : Menyesuaikan ukuran input.
 - e. `base_model.summary ()` : Menampilkan detail arsitektur model.

Penyesuaian *Layer Convolutional Neural Networks*

Setelah membangun model menggunakan arsitektur *MobileNetV2*, langkah selanjutnya adalah menyesuaikan layer *Convolutional Neural Networks* (CNN) untuk meningkatkan performa dalam klasifikasi gambar. Kode berikut digunakan untuk membangun model yang telah dimodifikasi dengan

penyesuaian layer agar lebih optimal.

1. *Input Layer* : Menerima gambar berwarna (224×224 , RGB).
2. *Base Model* : MobileNetV2 digunakan sebagai ekstraktor fitur dengan bobot dibekukan (*training* = False).
3. *Flatten* : Mengubah feature map 2D menjadi vektor 1D untuk proses klasifikasi.
4. *Dense Layer* : Dense (256, activation = relu) Belajar pola dari fitur yang diekstrak.
5. *kernel_regularizer* = `tf.keras.regularizers.l2` (0.001) Mencegah overfitting dengan L2 regularization.
6. *Dropout* : Dropout (0.5) Menonaktifkan 50% neuron secara acak untuk meningkatkan generalisasi.
7. *Output Layer* : Dense (4, activation = softmax) Menghasilkan probabilitas dari 4 kelas.
8. Model Akhir : Menggabungkan seluruh layer dan menampilkan arsitektur dengan `summary` ().



Gambar 9. Layer MobileNet-V2

Melatih Model *MobileNet* - Version 2

Setelah melakukan *input layer* pada *Convolutional Neural Network* (CNN) sesuai kebutuhan, langkah berikutnya adalah melatih model *MobileNetV2* yang telah dibangun sebelumnya. Proses pelatihan ini menggunakan berbagai teknik optimasi untuk meningkatkan akurasi dan menekan nilai *loss*.

1. *Mengompilasi Model*
Model dikompilasi menggunakan *loss categorical_crossentropy* untuk klasifikasi multi-kelas, dengan *optimizer* Adam (learning rate 0.0001) dan metrik evaluasi akurasi.
2. *Callbacks* (Penghentian Dini dan Penyesuaian Learning Rate)

EarlyStopping menghentikan pelatihan jika *val_loss* tidak membaik selama 20 epoch berturut-turut, sementara *ReduceLROnPlateau* menurunkan *learning rate* jika *val_loss* stagnan selama 10 epoch.

3. Pelatihan Model
4. Model dilatih menggunakan *data augmentation* dengan *datagen.flow()*, hanya dalam 1 epoch sehingga belajar sekali dari seluruh dataset. Validasi dilakukan dengan (*x_test*, *y_test*), serta menggunakan *callbacks early_stop* dan *reduce_lr*.

Evaluation

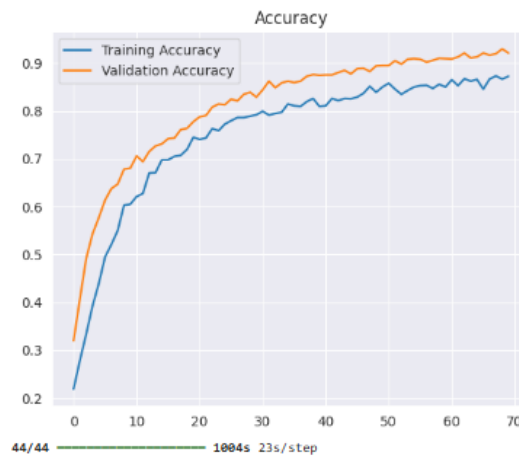
Tahap ini bertujuan untuk mengevaluasi hasil dari tahap sebelumnya, yaitu *Modelling*. Evaluasi dilakukan untuk menyesuaikan model agar lebih optimal dan sesuai dengan target yang ingin dicapai [19]. Proses pengujian dilakukan sebanyak sepuluh kali pada setiap variasi jumlah epoch, yaitu epoch 70, epoch 90, dan epoch 100. Dari seluruh pengujian tersebut kemudian dianalisis untuk menentukan kombinasi epoch dan pengujian yang memberikan performa terbaik berdasarkan nilai akurasi validasi dan nilai loss. Hasil analisis menunjukkan bahwa model dengan konfigurasi epoch 100 pada pengujian ke-1 menghasilkan kinerja paling optimal, ditandai dengan nilai akurasi validasi tertinggi dan loss terendah. Evaluasi lebih lanjut menggunakan *confusion matrix* serta metrik *precision*, *recall*, *F1-score*, dan *support* memperlihatkan bahwa model pada epoch 100 memiliki performa yang paling stabil dengan tingkat akurasi validasi sebesar 94,9% serta indikasi *overfitting* yang minimal jika dibandingkan dengan model pada epoch 70 dan epoch 90.

Tabel 1. Data Hasil Pengujian Model

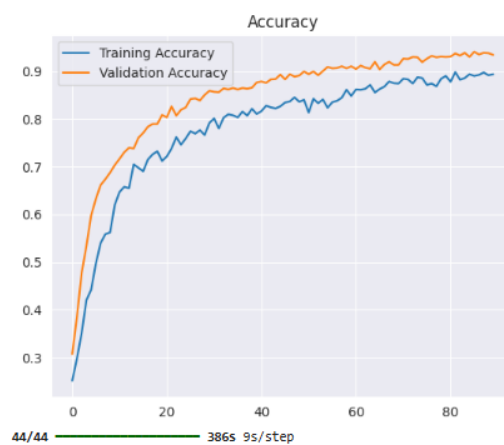
Metrik	Epoch 70	Epoch 90	Epoch 100
Akurasi Training	88,46%	89,50%	91,02%
Loss Training	31,92%	27,52%	25,37%
Akurasi Validasi	92,10%	93,48%	94,49%
Loss Validasi	23,02%	18,07%	16,33%

Grafik Accuracy Training Dan Validasi

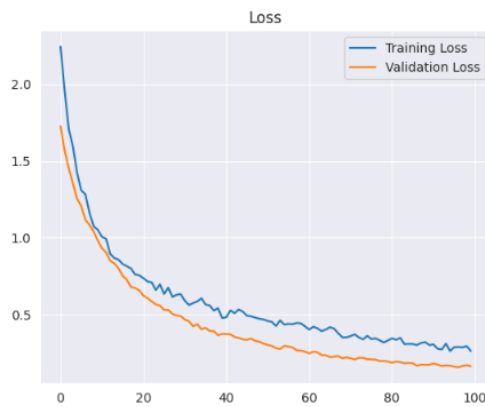
Akurasi adalah metrik yang mengukur persentase prediksi yang benar dari total prediksi yang dilakukan [20]. Visualisasi grafik hasil pelatihan pada epoch 70 ditunjukkan pada **Gambar 10**, sedangkan grafik untuk epoch 90 disajikan pada **Gambar 11**. Adapun grafik hasil pelatihan pada epoch 100 dapat dilihat pada **Gambar 12**.



Gambar 10. Grafik Accuracy Epoch 70



Gambar 11. Grafik Accuracy Epoch 90

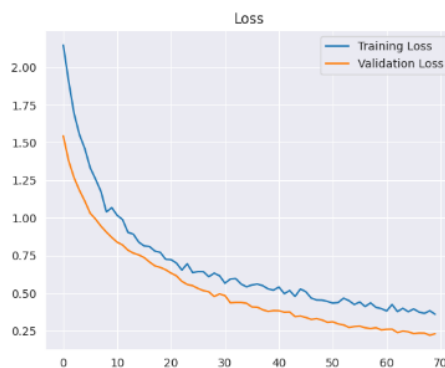


Gambar 12. Grafik Accuracy Epoch 100

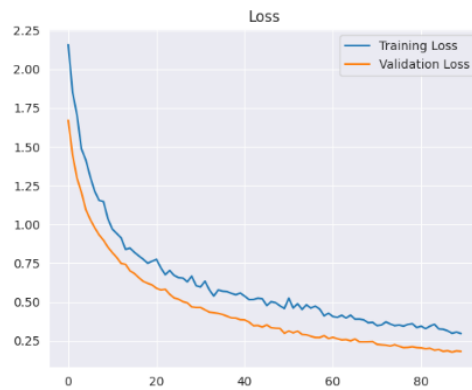
Berdasarkan Grafik ini membandingkan akurasi training (biru) dan validasi (merah). Pada epoch 70, akurasi training 88,46% mencapai dan validasi 92,10%. Epoch 90 menunjukkan akurasi training 89,50% dan validasi 93,48%, sementara epoch 100 mencatat training 91,02% dan validasi 94,49%.

Grafik Loss Training dan Validasi

Loss adalah metrik yang digunakan untuk menilai sejauh mana perbedaan antara hasil prediksi model dan nilai sebenarnya [20]. Pengujian pada epoch 70 dapat dilihat pada Gambar 13, sementara epoch 90 ditampilkan pada Gambar 14, dan epoch 100 pada Gambar 15.



Gambar 13. Grafik Loss Epoch 70



Gambar 14. Grafik Loss Epoch 90



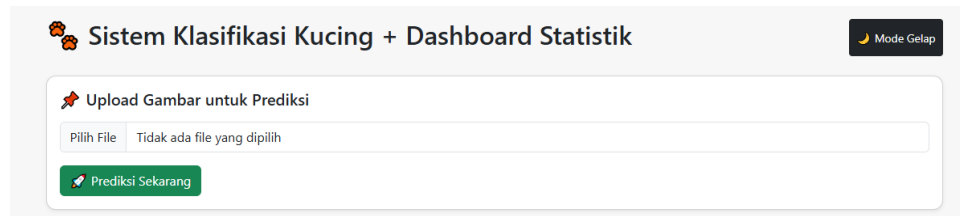
Gambar 15. Grafik Loss Epoch 100

Berdasarkan grafik, epoch 70 menunjukkan perbedaan nilai loss yang cukup besar antara proses pelatihan dan validasi, sehingga kemampuan generalisasi model masih belum optimal. Pada epoch 90, tren loss terlihat lebih konsisten dengan selisih antara *training loss* dan *validation loss* yang lebih kecil, yang mengindikasikan performa model yang lebih seimbang. Sementara itu, epoch 100 menghasilkan nilai *training loss* dan *validation loss* paling rendah dengan jarak yang minimal, yang menunjukkan tingkat overfitting yang lebih rendah serta kestabilan model yang paling baik.

Deployment


Pada tahap ini, ditampilkan hasil dari model yang telah diimplementasikan pada proses data mining. Tahap ini juga mencakup penyajian informasi yang diperoleh dari proses analisis sehingga mudah

dipahami oleh pengguna [12]. Selain itu, dilakukan pembuatan laporan berdasarkan hasil evaluasi model klasifikasi ras kucing, serta pengembangan aplikasi berbasis web sebagai media visualisasi hasil.



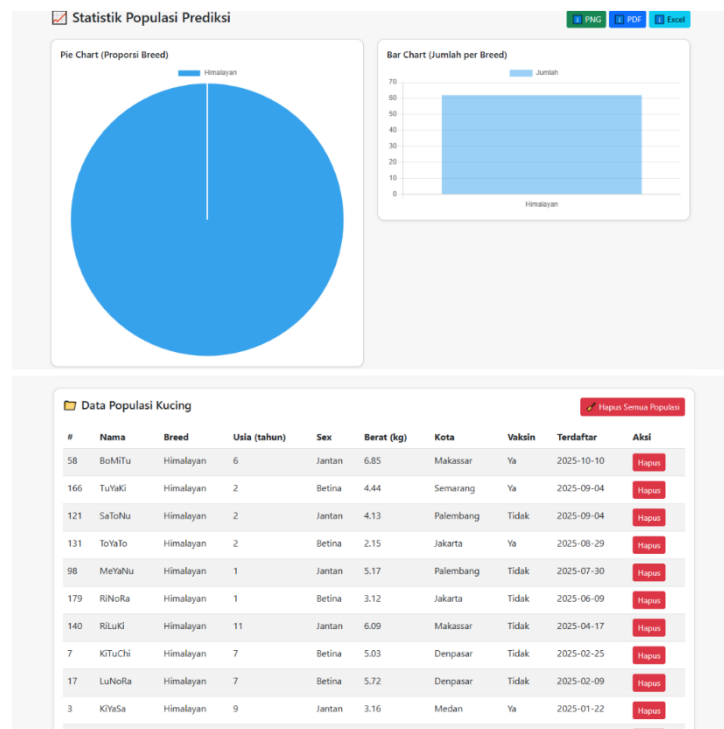
Gambar 16. Halaman Utama

Pada halaman utama sistem, pengguna dapat mengunggah gambar melalui tombol **“Pilih File”**, lalu menekan tombol **“Prediksi Sekarang”** untuk menjalankan proses klasifikasi. Sistem kemudian akan memproses gambar tersebut menggunakan model klasifikasi yang telah dilatih, dan dalam beberapa detik hasilnya akan langsung ditampilkan.

#	File	Prediksi	Preview	Waktu	Aksi
71	20251118233204_himalayan_1.png	Himalayan		2025-11-18 23:32:12	Hapus

Gambar 17. Tabel Riwayat

Setiap prediksi yang berhasil dilakukan akan otomatis muncul pada bagian Riwayat Prediksi. Pada bagian ini, pengguna dapat melihat informasi lengkap mengenai setiap proses klasifikasi, seperti nama file yang diunggah, hasil identifikasi ras kucing, tampilan mini (preview) gambar yang dianalisis, serta waktu dan tanggal prediksi dilakukan.



Gambar 18. Grafik dan Tabel Populasi

Data yang masuk juga akan otomatis digunakan untuk memperbarui grafik statistik populasi kucing dan tabel Data Populasi Kucing, sehingga pengguna dapat langsung melihat jumlah tiap ras dalam bentuk *pie chart*, *bar chart* dan *line chart*, serta daftar populasi lengkap yang tersimpan di *Database*.

PENUTUP

Penelitian ini membuktikan bahwa CNN dengan MobileNet-V2 mampu mengklasifikasikan ras kucing akurasi tinggi. Epoch 100 memberikan performa optimal dengan keseimbangan terbaik antara akurasi pelatihan dan validasi. Namun, tantangan seperti overfitting dan generalisasi masih ada. Pengembangan selanjutnya dapat mencakup regulasi tambahan seperti dropout, data augmentation, atau model yang lebih kompleks. Pengujian dengan dataset lebih besar juga diperlukan untuk memastikan kinerja model dalam berbagai kondisi nyata, sehingga lebih efektif dalam sistem pengenalan gambar berbasis AI.

DAFTAR PUSTAKA

- [1] 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, Jan. 2024, doi: 10.35969/interkom.v18i4.318.
- [2] A. Nur Ramadhayani, V. Lusiana, U. Stikubank Semarang JI Tri Lomba Juang No, K. Semarang Selatan, K. Semarang, and J. Tengah, "KLASIFIKASI JENIS KUCING MENGGUNAKAN ALGORITMA PRINCIPAL COMPONENT ANALYSIS DAN K-NEAREST NEIGHBOR," *Jurnal informasi dan Komputer*, vol. 10, no. 2, 2022.
- [3] C. Agusniar and D. Adelia, "Penerapan Convolutional Neural Network pada Klasifikasi Jenis Ras Kucing Menggunakan ResNet50V2," 2024.
- [4] J. Tugas, A. Fakultas Informatika, M. Afif, A. Fawwaz, K. N. Ramadhani, and F. Sthevanie, "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network(CNN)," 2020.
- [5] S. Diantama, "PEMANFAATAN ARTIFICIAL INTELEGENT (AI) DALAM DUNIA PENDIDIKAN," 2023.
- [6] P. Annisa, "Penerapan Teknologi Google Lens dan QR Code pada Tanaman Pertanian," *dst*, vol. 3, no. 2, pp. 240-245, Nov. 2023, doi: 10.47709/dst.v3i2.3130.
- [7] A. S. Riyadi, I. P. Wardhani, D. S. Widayati, and K. Kunci, "KLASIFIKASI CITRA ANJING DAN KUCING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)," *Universitas Gunadarma Jl. Margonda Raya No*, vol. 5, no. 1, p. 12140, 2021.
- [8] A. Aziz, R. Reyhan Zhafari, and M. M. Santoni, *Klasifikasi 10 Spesies Monyet Berdasarkan Citra Menggunakan Convolutional Neural Network*. 2021.
- [9] D. Ramayanti, D. Asri, and L. Lionie, "Implementasi Model Arsitektur VGG16 dan MobileNetV2 Untuk Klasifikasi Citra Kupu-Kupu Article Info ABSTRAK," *JSAI: Journal Scientific and Applied Informatics*, vol. 5, no. 3, 2022, doi: 10.36085.
- [10] O. Virgantara Putra, M. Zaim Mustaqim, D. Muriatmoko, J. Teknik Informatika, and F. Sains dan Teknologi, "Transfer Learning untuk Klasifikasi Penyakit dan Hama Padi Menggunakan MobileNetV2 Transfer Learning for Rice Disease and Pest Classification using MobileNetV2," 2023.
- [11] M. Riziq sirfatullah Alfarizi, M. Zidan Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "PENGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING," 2023.
- [12] S. N. Luqman *et al.*, "Komparasi Algoritma Klasifikasi Genre Musik pada Spotify Menggunakan CRISP-DM," 2021.

- [13] I. Budiman, T. Prahasto, and Y. Christyono, "DATA CLUSTERING MENGGUNAKAN METODOLOGI CRISP-DM UNTUK PENGENALAN POLA PROPORSI PELAKSANAAN TRIDHARMA," 2012.
- [14] A. Abhista Hibatullah and W. Apriandari, "KLASIFIKASI KUALITAS JENIS KOPI HALUS ROBUSTA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DAN MOBILENET-V2," 2024.
- [15] M. A. Hasanah, S. Soim, and A. S. Handayani, "Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir," 2021. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [16] N. Alfiah, "Klasifikasi Penerima Bantuan Sosial Program Keluarga Harapan Menggunakan Metode Naive Bayes".
- [17] Y. P. ASTUTI, E. R. Subhiyakto, I. Wardatunizza, and E. Kartikadarma, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Jenis Tanah Berbasis Android,". *Jurnal Informatika: Jurnal pengembangan IT*, Vol.8, No.3, pp. 220-225, Sep. 2023, doi: 10.30591/jpit.v8i3.5026.
- [18] W. Hastomo, E. Hadiyanto, and D. Sutarno, "KLASIFIKASI COVID-19 CHEST X-RAY DENGAN TIGA ARSITEKTUR CNN (RESNET-152, INCEPTIONRESNET-V2, MOBILENET-V2)," 2021.
- [19] M. Y. Titimeidara and W. Hadikurniawati, "Monica Yoshe Titimeidara Implementasi Metode Naive Bayes Implementasi Metode Naive Bayes Classifier Untuk Klasifikasi Status Gizi Stunting Pada Balita."
- [20] Z. I. Nugraha, Arnita, Kana Saputra S, A. Setiawan, R. Maharani, and F. Zaharani, "Implementasi Algoritma CNN Dalam Pengembangan Website Untuk Klasifikasi Sampah Organik, Dan Non-Organik,". *Jurnal Manajemen Informatika & Sistem Informasi (MISI) Volume 8, no. 1, pp. 90-101, Jan. 2025, doi: 10.36595/misi.v5i2.*