

Implementasi *Convolutional Neural Network* dalam Mendeteksi Jenis Ras Kucing

Implementation of Convolutional Neural Network for Detecting Cat Breed Types

David Febrianto ^{*,1}, Nova Tri Romadloni²

^{1,2} Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Karanganyar

*Penulis Korespondensi

Email: febriantod273@gmail.com

Abstrak. Klasifikasi ras kucing dari citra digital merupakan salah satu tantangan dalam bidang visi komputer karena adanya kemiripan fitur visual antar ras serta keterbatasan jumlah data pada setiap kelas. Penelitian ini bertujuan untuk mengimplementasikan *Convolutional Neural Network* (CNN) dalam proses klasifikasi ras kucing berbasis citra digital. Dataset diproses melalui beberapa tahap pra-pemrosesan, meliputi resize citra, normalisasi, dan augmentasi data untuk meningkatkan variasi data pelatihan. Model CNN dibangun menggunakan *framework* Keras dan *TensorFlow* dengan beberapa lapisan konvolusi, pooling, dan *fully connected*. Selain model CNN kustom, penelitian ini juga mengeksplorasi pendekatan *transfer learning* menggunakan arsitektur VGG16, ResNet50, dan MobileNetV2. Evaluasi model dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*, serta didukung visualisasi Grad-CAM untuk membantu interpretasi prediksi model. Hasil eksperimen menunjukkan bahwa performa model CNN pada klasifikasi multi-kelas masih belum optimal akibat keterbatasan jumlah data dan tingginya kemiripan fitur visual antar ras kucing. Nilai evaluasi yang rendah menunjukkan bahwa model belum mampu melakukan generalisasi secara baik pada sebagian besar kelas. Penelitian ini menunjukkan bahwa ukuran dataset, distribusi data, dan jumlah sampel per kelas sangat memengaruhi performa CNN dalam klasifikasi citra multi-kelas. Oleh karena itu, penelitian selanjutnya disarankan menggunakan dataset yang lebih besar dan seimbang serta menerapkan optimasi arsitektur model dan fine-tuning untuk meningkatkan performa klasifikasi.

Kata kunci: *Convolutional Neural Network*, *Deep learning*, Klasifikasi Citra, Ras Kucing, *Transfer learning*

Abstract. Cat breed classification from digital images is a challenging problem in computer vision due to the similarity of visual characteristics between breeds and the limited number of samples in each class. This study aims to implement a *Convolutional Neural Network* (CNN) for cat breed classification using digital images. The dataset was processed through several preprocessing stages, including image resizing, normalization, and data augmentation to increase training data variability. The CNN model was developed using the Keras and TensorFlow frameworks with multiple convolutional, pooling, and fully connected layers. In addition to a custom CNN model, this study also explored *Transfer learning* approaches using VGG16, ResNet50, and MobileNetV2 architectures. Model evaluation was conducted using accuracy, *precision*, *recall*, and *F1-score* metrics, supported by Grad-CAM visualization to assist in interpreting model predictions. The experimental results indicate that the CNN model still has limitations in multi-class cat breed classification due to limited data and similarities in visual characteristics between breeds. The low evaluation scores indicate that the model has not been able to generalize effectively across most classes. This study demonstrates that dataset size, data distribution, and the number of

samples per class significantly affect CNN performance in multi-class image classification. Therefore, future research is recommended to use larger and more balanced datasets, as well as apply model architecture optimization and fine-tuning techniques to improve classification performance..

Keywords: Cat breeds, Convolutional Neural Network, Deep learning, Image Classification, Transfer learning

1. Pendahuluan

Kemajuan teknologi digital dan kecerdasan buatan telah membuka peluang baru dalam pengolahan citra digital, khususnya pada bidang klasifikasi gambar berbasis *deep learning*. Salah satu tantangan dalam klasifikasi citra adalah identifikasi ras kucing melalui gambar digital. Permasalahan ini menjadi kompleks karena adanya kemiripan karakteristik visual antar ras kucing, seperti warna bulu, pola, bentuk tubuh, serta variasi posisi dan pencahayaan pada citra, sehingga proses klasifikasi secara manual menjadi sulit dilakukan (Linda *et al.*, 2024; Rahayu, 2023).

Kebutuhan terhadap sistem klasifikasi ras kucing secara otomatis semakin meningkat, terutama dalam bidang medis veteriner, peternakan, dan pengembangan aplikasi cerdas berbasis *Internet of Things* (IoT) untuk mendukung pemeliharaan hewan secara lebih optimal (Aliefah Syalma Ratsdea Mufti & Yovi Litanianda, 2024; Eichstädt & Spieker, 2024).

Convolutional Neural Network (CNN) merupakan salah satu metode *deep learning* yang banyak digunakan dalam klasifikasi citra karena kemampuannya dalam mengekstraksi fitur spasial dan hierarkis secara otomatis dari data gambar. CNN telah menunjukkan performa yang baik pada berbagai bidang, seperti pengenalan objek, deteksi wajah, klasifikasi hewan, hingga analisis citra medis (Fikry & Nurhasanah, 2023). Dalam konteks klasifikasi ras kucing, berbagai penelitian telah memanfaatkan arsitektur CNN seperti VGG16, ResNet50, MobileNetV2, dan Xception untuk meningkatkan akurasi klasifikasi (Agusniar & Adelia, 2024; Fawwaz *et al.*, 2021; Wijaya Kusuma *et al.*, 2023). Selain itu, teknik augmentasi data dan *transfer learning* juga terbukti mampu meningkatkan kemampuan generalisasi model pada proses klasifikasi citra (Dede Husen, 2024).

Penelitian sebelumnya yang menggunakan dataset Oxford-IIIT Pet Dataset dengan jumlah data yang besar menunjukkan bahwa model CNN mampu mencapai performa tinggi. (Fawwaz *et al.*, 2021) melaporkan bahwa arsitektur *Xception* memperoleh akurasi sebesar 93,75% dalam klasifikasi ras kucing. Penelitian lain juga menunjukkan bahwa kombinasi strategi augmentasi data, pemilihan arsitektur yang tepat, dan penggunaan *transfer learning* terbukti efektif dalam menangani tantangan pengenalan pola visual kompleks, termasuk dalam konteks klasifikasi ras kucing (Aliefah Syalma Ratsdea Mufti & Yovi Litanianda, 2024; Eichstädt & Spieker, 2024).

Berbeda dengan penelitian sebelumnya yang umumnya menggunakan dataset publik berukuran besar dengan jumlah kelas terbatas, penelitian ini berfokus pada eksplorasi implementasi CNN pada dataset ras kucing berukuran kecil dengan jumlah kelas yang relatif banyak. Penelitian ini tidak hanya mengevaluasi performa klasifikasi model CNN, tetapi juga menganalisis pengaruh keterbatasan dataset terhadap kemampuan generalisasi model pada klasifikasi multi-kelas dengan karakteristik visual antar kelas yang mirip.

Kontribusi utama penelitian ini adalah mengevaluasi kemampuan *Convolutional Neural Network* (CNN) pada skenario klasifikasi multi-kelas dengan jumlah data yang sangat terbatas pada setiap kelas. Penelitian ini juga menganalisis pengaruh keterbatasan dataset terhadap kemampuan generalisasi model dalam membedakan ras kucing yang memiliki kemiripan

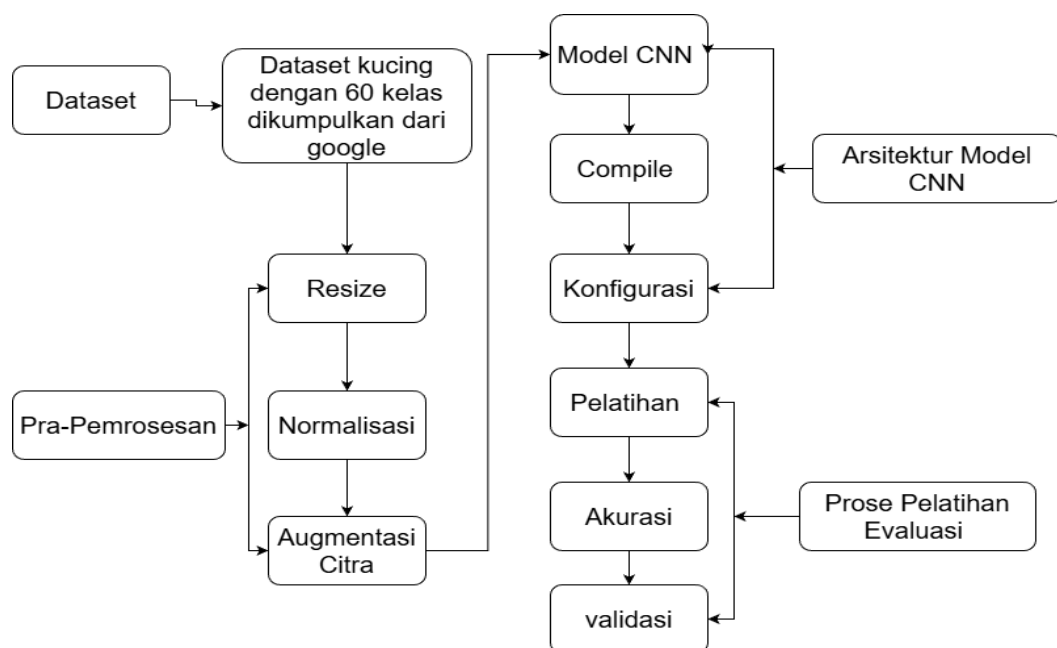
karakteristik visual. Selain itu, penelitian ini memanfaatkan visualisasi Grad-CAM dan *Feature CAM* untuk membantu interpretasi fokus prediksi model terhadap citra masukan.

2. Metode

Bagian ini menjelaskan tahapan penelitian yang dilakukan untuk mengimplementasikan *Convolutional Neural Network* (CNN) pada klasifikasi ras kucing berbasis citra digital. Metode penelitian disusun secara sistematis agar proses eksperimen dapat direplikasi pada penelitian selanjutnya. Tahapan penelitian meliputi pengumpulan dataset, pra-pemrosesan citra, perancangan arsitektur model CNN, proses pelatihan, serta evaluasi performa model.

Seluruh eksperimen dilakukan menggunakan perangkat komputer dengan spesifikasi minimum prosesor Intel Core i5, RAM 8GB, dan sistem operasi Windows 10. Implementasi model dilakukan menggunakan bahasa pemrograman Python dengan *framework TensorFlow* dan Keras. Dataset dikumpulkan secara manual melalui pencarian citra ras kucing dari Google Images, kemudian dilakukan proses seleksi dan validasi data untuk mengurangi citra yang tidak sesuai kategori, duplikat, atau memiliki kualitas visual rendah.

Penelitian ini berfokus pada analisis performa CNN pada kondisi dataset terbatas dengan jumlah kelas yang relatif banyak. Oleh karena itu, hasil eksperimen digunakan untuk mengevaluasi tantangan generalisasi model pada klasifikasi citra multi-kelas dengan keterbatasan jumlah sampel pada setiap kategori. Seluruh tahapan penelitian, mulai dari pengumpulan data, pra-pemrosesan citra, pelatihan model, hingga evaluasi performa, dirangkum dalam diagram alur penelitian pada gambar berikut.



Gambar 1. Alur Proses Metode Penelitian

2.1 Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari 60 kelas ras kucing dengan total 300 citra, di mana setiap kelas memiliki 5 gambar. Seluruh citra dikumpulkan secara manual melalui pencarian menggunakan Google Images dan tidak berasal dari repositori standar seperti Kaggle

maupun Oxford-IIIT Pet Dataset. Pengumpulan data dilakukan untuk memperoleh variasi visual dari berbagai ras kucing sesuai kebutuhan penelitian.

Proses seleksi data dilakukan dengan mempertimbangkan kesesuaian label ras, kualitas visual gambar, serta keberagaman sudut pengambilan citra. Gambar yang buram, duplikat, mengandung watermark berlebihan, atau tidak sesuai dengan kategori ras dieliminasi dari dataset. Meskipun demikian, jumlah data pada masing-masing kelas masih relatif terbatas sehingga representasi fitur visual setiap ras belum optimal untuk proses pelatihan model CNN.

Keterbatasan jumlah data tersebut berpotensi memengaruhi kemampuan generalisasi model dalam melakukan klasifikasi ras kucing, terutama karena banyaknya jumlah kelas yang digunakan dalam penelitian ini. Oleh karena itu, penelitian selanjutnya disarankan menggunakan dataset publik dengan jumlah data yang lebih besar dan distribusi kelas yang lebih seimbang untuk meningkatkan performa model klasifikasi.

Tabel 1. Dataset kelas ras kucing

Ras Kucing	Jumlah citra
abyssinian	5
aegean	5
american_bobtail	5
american_curl	5
american_ringtail	5
american_shorthair	5
american_wirehair	5
australian_mist	5
balinese	5
bambino	5
bengal	5
birman	5
bombay	5
brambile	5
brazillian_shorthair	5
british_longhair	5
british_shorthair	5
burmese	5
burmilla	5
california_sprangled	5
chartreux	5
chausie	5
cheetoh	5
cornish_rex	5
cymric	5
desert_lynx	5
devon_rex	5
donskoy	5
exotic_shorthair	5
foldex	5
german_rex	5
havana_brown	5
highlander	5
japanese	5

korat	5
lambkin	5
laperm	5
lykoy	5
mandalay	5
manx	5
minskin	5
mojave	5
munchkin	5
nebelung	5
ocicat	5
pantherette	5
persia	5
peterbald	5
poodle	5
ragamuffin	5
ragdoll	5
savannah	5
serengeti	5
siamese	5
skookum	5
snowshoe	5
sokoke	5
somali	5
sphynx	5
suphalax	5

Struktur dataset disusun dalam direktori `dataset_ras_kucing`, di mana setiap folder merepresentasikan satu kelas ras kucing, seperti abyssinian, persian, dan ragdoll. Penamaan folder dilakukan sesuai label kelas agar kompatibel dengan fungsi `ImageDataGenerator` pada *framework* Keras, sehingga proses pelabelan dan pemrosesan data dapat dilakukan secara otomatis berdasarkan struktur direktori (Aliefah Syalma Ratsdea Muftti & Yovi Litanianda, 2024). Contoh sampel citra dari salah satu kelas dataset ditunjukkan pada Gambar 2.



Gambar 2. Contoh Sampel Data Citra Ras Abyssinian

2.2 Pra-pemrosesan

Sebelum digunakan dalam proses pelatihan, seluruh citra pada dataset melalui beberapa tahapan pra-pemrosesan untuk memastikan kesesuaian input terhadap model *Convolutional Neural Network* (CNN). Tahap pertama dilakukan dengan mengubah ukuran seluruh citra menjadi 224×224 piksel agar sesuai dengan dimensi input standar pada arsitektur CNN modern, seperti DenseNet121 (Wijaya Kusuma *et al.*, 2023). Selanjutnya, nilai piksel dinormalisasi ke dalam rentang 0 hingga 1 untuk membantu mempercepat proses konvergensi dan meningkatkan stabilitas pelatihan model (Dede Husen, 2024).

Untuk mengatasi keterbatasan jumlah data dan meningkatkan variasi citra, diterapkan teknik augmentasi data melalui transformasi seperti rotasi acak, horizontal flipping, zoom, dan shearing. Teknik augmentasi digunakan untuk menambah keragaman data pelatihan secara sintetik sehingga model memperoleh variasi pola visual yang lebih beragam (Dede Husen, 2024) (Dede Husen, 2024; Rahma & Suadaa, 2023). Setelah proses augmentasi, dataset dibagi menjadi data latih sebesar 80% dan data validasi sebesar 20%. Pembagian data dilakukan menggunakan parameter `validation_split` pada fungsi `ImageDataGenerator` dari *framework* Keras. Meskipun augmentasi data dapat membantu meningkatkan variasi citra pelatihan, keterbatasan jumlah data asli pada setiap kelas masih berpotensi memengaruhi kemampuan generalisasi model dalam proses klasifikasi multi-kelas (Dede Husen, 2024; Rahma & Suadaa, 2023; Wijaya Kusuma *et al.*, 2023).

2.3 Arsitektur Model CNN

Model *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini terdiri dari beberapa tahapan utama, yakni lapisan konvolusi (Conv2D), pooling (MaxPooling2D), dan fully-connected (Dense). Seluruh model diimplementasikan menggunakan *framework* Keras dengan backend TensorFlow, yang menyediakan kemudahan dalam membangun dan mengatur arsitektur jaringan. Model custom CNN disusun dengan kombinasi beberapa lapisan konvolusi untuk mengekstraksi fitur visual, dilanjutkan dengan pooling untuk mereduksi dimensi spasial, serta lapisan Flatten dan Dense sebagai classifier akhir. Untuk mengurangi risiko overfitting, diterapkan teknik regularisasi berupa dropout pada beberapa lapisan. Optimasi dilakukan dengan berbagai optimizer seperti Adam dan RMSprop yang terbukti efektif dalam meningkatkan performa CNN, termasuk pada arsitektur VGG-19 (Ramadhan & Hernadi, 2025)

Model dikompilasi menggunakan optimizer Adam, dengan fungsi loss berupa *categorical_crossentropy*, dan evaluasi dilakukan menggunakan metrik akurasi. Selain model CNN yang dirancang dari awal (custom), penelitian ini juga mengimplementasikan beberapa arsitektur pre-trained populer melalui pendekatan *transfer learning*, antara lain VGG16, ResNet50, dan MobileNetV2. Pemanfaatan model prelatih ini dilakukan dengan memanfaatkan bobot yang telah dilatih sebelumnya pada dataset *ImageNet*, yang kemudian disesuaikan ulang (*fine-tuning*) untuk klasifikasi citra kucing. Strategi ini terbukti efektif dalam meningkatkan akurasi klasifikasi, sebagaimana didukung oleh berbagai penelitian sebelumnya seperti yang dilakukan (Pane & Sihombing, 2023; Ramayanti *et al.*, 2022; Wijaya Kusuma *et al.*, 2023). Penelitian menunjukkan bahwa fine-tuning arsitektur VGG16 mampu meningkatkan akurasi klasifikasi 11 ras kucing hingga 27,7% dibandingkan model dasar (Fikry & Nurhasanah, 2023). Menunjukkan bahwa penggunaan MobileNetV2 dengan penyesuaian *transfer learning* dan hyperparameter mampu meningkatkan performa klasifikasi citra medis secara signifikan, mendukung pendekatan serupa dalam penelitian ini (Somoal & Dzirkillah, 2025)

2.4 Proses pelatihan dan Evakuasi

Proses pelatihan model dilakukan dalam rentang 15 hingga 50 *epoch*, bergantung pada kompleksitas arsitektur CNN yang digunakan dan hasil evaluasi awal selama pelatihan. Penentuan jumlah *epoch* didasarkan pada studi oleh Wasil *et al.* mengenai pengaruh *epoch* terhadap akurasi model CNN dalam klasifikasi gambar, di mana peningkatan jumlah *epoch* menunjukkan korelasi terhadap peningkatan akurasi hingga titik tertentu sebelum *overfitting* terjadi (Wasil *et al.*, 2022). Data pelatihan dan validasi disiapkan menggunakan *ImageDataGenerator* dari Keras, dengan pengaturan *batch_size* sebesar 32, yang memungkinkan proses augmentasi dan normalisasi dilakukan secara real-time selama pelatihan. Model dikompilasi dan dioptimasi menggunakan parameter *loss function categorical_crossentropy* dan *optimizer Adam*, serta metrik evaluasi berupa akurasi.

Evaluasi performa model dilakukan dengan menghitung sejumlah metrik penting seperti akurasi, *precision*, *recall*, dan *F1-score*, yang memberikan gambaran komprehensif terhadap kualitas klasifikasi pada setiap kelas. Selain itu, dilakukan visualisasi hasil pelatihan berupa grafik akurasi dan loss per *epoch*, yang digunakan untuk memantau kestabilan dan kecenderungan model terhadap *overfitting* atau *underfitting*. Untuk memperdalam analisis hasil prediksi, juga diterapkan metode visualisasi prediksi benar dan salah menggunakan citra output, serta eksplorasi teknik visualisasi Grad-CAM dan pengembangannya Teknik visualisasi seperti Integrated Grad-CAM, dan *Feature CAM* digunakan untuk memberikan interpretasi yang lebih akurat dan sensitif terhadap fokus perhatian model pada citra input. Selain itu, juga menekankan pentingnya seleksi fitur dan evaluasi menyeluruh terhadap performa model klasifikasi dalam konteks medis, yang relevan dalam menilai kualitas hasil klasifikasi pada domain visual seperti deteksi ras kucing (Supriyanti *et al.*, 2024). Dengan pendekatan ini, tidak hanya performa numerik model yang dapat dievaluasi, tetapi juga aspek interpretabilitas dari hasil prediksi model dapat dianalisis secara mendalam (Clement *et al.*, 2024; Sattarzadeh *et al.*, 2021).

3. Hasil dan Pembahasan

Bagian ini menyajikan hasil penelitian, baik dalam bentuk teks, tabel, atau gambar. Penggunaan foto sangat dibatasi pada hasil yang jelas. Setiap gambar dan tabel diberi nomor secara berurut dan harus diacu pada naskah.

3.1 Arsitektur Model CNN dan Hasil Pelatihan Selama 15 Epoch

Pada tahap ini, dilakukan perancangan arsitektur model *Convolutional Neural Network* (CNN) yang digunakan untuk melakukan klasifikasi citra ras kucing. Arsitektur model dirancang untuk mengekstraksi fitur-fitur penting dari citra masukan melalui beberapa lapisan konvolusi dan pooling, kemudian dilanjutkan dengan lapisan *fully connected* untuk melakukan proses klasifikasi. Setelah arsitektur model ditetapkan, proses pelatihan dilakukan menggunakan data latih selama 15 *epoch* dengan tujuan untuk mempelajari pola dan karakteristik setiap kelas ras kucing berdasarkan nilai akurasi dan *loss* pada data pelatin serta data validasi.

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_9 (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_10 (Conv2D)	(None, 29, 29, 64)	18,496
max_pooling2d_10 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_11 (Conv2D)	(None, 12, 12, 128)	73,856
max_pooling2d_11 (MaxPooling2D)	(None, 6, 6, 128)	0
flatten_3 (Flatten)	(None, 4608)	0
dense_6 (Dense)	(None, 128)	589,952
dropout_3 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 60)	7,740

Total params: 690,940 (2.64 MB)
Trainable params: 690,940 (2.64 MB)
Non-trainable params: 0 (0.00 B)

Gambar 3. Arsitektur Model *Convolutional Neural Network*

Gambar tersebut menyajikan ringkasan arsitektur model *Convolutional Neural Network* (CNN) yang disusun secara sekuensial untuk memproses data citra. Arsitektur ini terdiri dari tiga blok utama lapisan konvolusi (Conv2D) yang diikuti oleh lapisan penyusutan (MaxPooling2D) untuk mengekstraksi fitur secara hierarkis, mulai dari 32 hingga 128 filter. Setelah fitur diekstraksi, data diratakan melalui lapisan Flatten dan diteruskan ke lapisan saraf padat (Dense) yang dilengkapi dengan *Dropout* untuk mencegah *overfitting*. Dengan total mencapai 690.940 parameter yang dapat dilatih, model ini dirancang untuk melakukan klasifikasi akhir ke dalam 60 kategori berbeda, sebagaimana terlihat pada jumlah unit di lapisan keluaran terakhir.

```

Epoch 1/15
8/8 ----- 6s 414ms/step - accuracy: 0.0161 - loss: 4.1326 - val_accuracy: 0.0000e+00 - val_loss: 4.0919
Epoch 2/15
8/8 ----- 3s 399ms/step - accuracy: 0.0145 - loss: 4.0953 - val_accuracy: 0.0185 - val_loss: 4.0951
Epoch 3/15
8/8 ----- 3s 411ms/step - accuracy: 0.0191 - loss: 4.0908 - val_accuracy: 0.0185 - val_loss: 4.0945
Epoch 4/15
8/8 ----- 4s 469ms/step - accuracy: 0.0203 - loss: 4.0906 - val_accuracy: 0.0185 - val_loss: 4.0948
Epoch 5/15
8/8 ----- 3s 419ms/step - accuracy: 0.0176 - loss: 4.0944 - val_accuracy: 0.0185 - val_loss: 4.0913
Epoch 6/15
8/8 ----- 3s 388ms/step - accuracy: 0.0199 - loss: 4.0892 - val_accuracy: 0.0185 - val_loss: 4.0834
Epoch 7/15
8/8 ----- 6s 526ms/step - accuracy: 0.0269 - loss: 4.0710 - val_accuracy: 0.0185 - val_loss: 4.0743
Epoch 8/15
8/8 ----- 3s 379ms/step - accuracy: 0.0174 - loss: 4.0681 - val_accuracy: 0.0185 - val_loss: 4.0683
Epoch 9/15
8/8 ----- 3s 388ms/step - accuracy: 0.0505 - loss: 4.0506 - val_accuracy: 0.0185 - val_loss: 4.0701
Epoch 10/15
8/8 ----- 3s 390ms/step - accuracy: 0.0394 - loss: 4.0086 - val_accuracy: 0.0185 - val_loss: 4.0468
Epoch 11/15
8/8 ----- 5s 361ms/step - accuracy: 0.0489 - loss: 3.9264 - val_accuracy: 0.0185 - val_loss: 4.0532
Epoch 12/15
8/8 ----- 3s 371ms/step - accuracy: 0.0369 - loss: 3.9312 - val_accuracy: 0.0185 - val_loss: 4.3456
Epoch 13/15
8/8 ----- 3s 364ms/step - accuracy: 0.0612 - loss: 3.9823 - val_accuracy: 0.0370 - val_loss: 4.0229
Epoch 14/15
8/8 ----- 4s 533ms/step - accuracy: 0.0826 - loss: 3.8507 - val_accuracy: 0.0185 - val_loss: 4.0702
Epoch 15/15
8/8 ----- 3s 355ms/step - accuracy: 0.0551 - loss: 3.7382 - val_accuracy: 0.0370 - val_loss: 4.0279

```

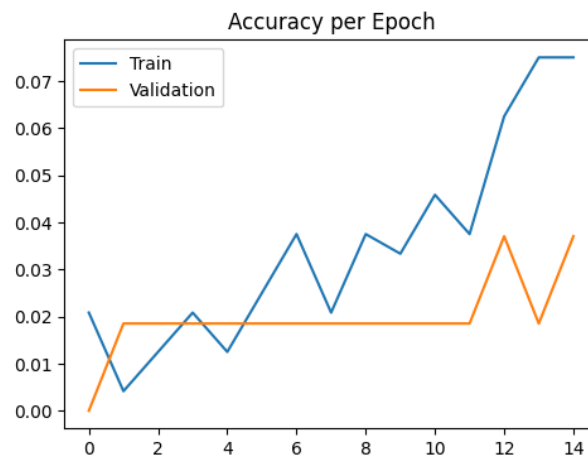
Gambar 4. Log Proses Pelatihan Model CNN

Gambar tersebut menunjukkan log performa model selama 15 *epoch* pelatihan, yang mencakup metrik akurasi (accuracy) dan kerugian (loss) baik pada data latih maupun data validasi. Terlihat bahwa nilai loss mengalami penurunan bertahap dari 4.1326 pada *epoch* pertama menjadi 3.7382 pada *epoch* terakhir, sementara akurasi meningkat secara perlahan namun masih berada di angka yang cukup rendah (sekitar 5.5%). Stagnasi pada nilai val_accuracy di angka 0.0185 pada sebagian besar *epoch* menunjukkan bahwa model mungkin sedang mengalami kesulitan untuk

belajar secara efektif atau membutuhkan durasi pelatihan yang lebih lama serta penyesuaian hyperparameter agar dapat melakukan generalisasi dengan lebih baik pada data validasi.

3.2 Visualisasi Performa Model

Evaluasi terhadap keberhasilan proses pelatihan model dilakukan dengan memantau pergerakan metrik akurasi pada setiap iterasi. Tahapan ini sangat krusial untuk mengidentifikasi sejauh mana arsitektur CNN yang telah dirancang mampu mengenali pola fitur pada dataset sekaligus mendeteksi adanya potensi masalah pada proses generalisasi. Dengan membandingkan metrik antara data latih (*training*) dan data validasi secara visual, kita dapat mengambil keputusan teknis terkait perlunya penyesuaian parameter atau arsitektur lebih lanjut. Adapun rincian mengenai fluktuasi akurasi selama proses eksperimen dapat dilihat pada grafik di bawah ini.



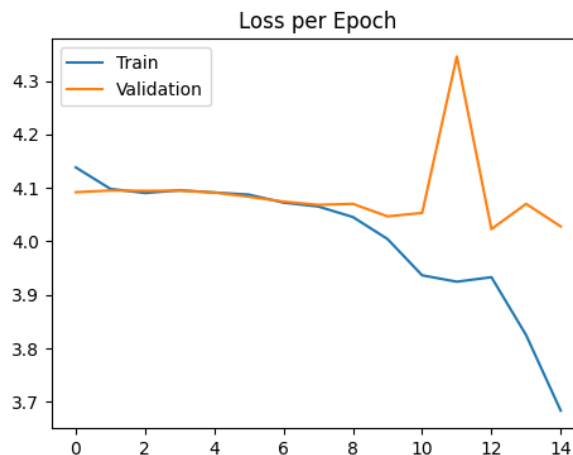
Gambar 5. Gambar Grafik Akurasi Model CNN pada Data Pelatihan dan Validasi per *Epoch*

Menunjukkan hasil visualisasi akurasi model pada data pelatihan dan validasi selama 15 *epoch*. Dari grafik tersebut, terlihat bahwa akurasi pada data pelatihan mengalami peningkatan secara bertahap, dimulai dari sekitar 0,01 pada *epoch* pertama dan mencapai lebih dari 0,07 pada *epoch* ke-14 dan ke-15. Hal ini menunjukkan bahwa model memiliki kemampuan untuk belajar dari data pelatihan, meskipun tingkat akurasinya secara umum masih tergolong rendah. Di sisi lain, akurasi pada data validasi cenderung stagnan di sekitar angka 0,0185 dan hanya mengalami sedikit peningkatan menjadi 0,0370 pada *epoch* ke-12 dan ke-14. Kesenjangan yang cukup besar antara akurasi pelatihan dan validasi ini menunjukkan bahwa model mengalami underfitting, yaitu ketidakmampuan model dalam menangkap pola penting dari data pelatihan yang dapat digeneralisasi ke data baru.(Agusniar & Adelia, 2024)

Kondisi tersebut dapat disebabkan oleh beberapa faktor, di antaranya adalah jumlah data pelatihan yang terbatas, arsitektur model yang belum optimal untuk kompleksitas permasalahan, atau belum diterapkannya teknik augmentasi data. Augmentasi data seperti rotasi, flipping, zooming, dan shifting dapat meningkatkan keberagaman data pelatihan sehingga membantu model belajar secara lebih efektif. Selain itu, penggunaan *transfer learning* dengan memanfaatkan arsitektur *pretrained* seperti VGG16, ResNet50, atau MobileNet dapat menjadi solusi untuk meningkatkan akurasi dan performa model dalam klasifikasi gambar ras kucing secara lebih akurat dan efisien.

3.3 Evaluasi Model Berdasarkan Nilai Loss

Selain metrik akurasi, indikator penting lainnya dalam menilai keberhasilan model *deep learning* adalah nilai loss atau fungsi kerugian. Nilai ini merepresentasikan seberapa besar selisih antara prediksi yang dihasilkan oleh model dengan label aktual pada dataset. Pemantauan terhadap grafik loss sangat berguna untuk melihat apakah model benar-benar belajar menuju titik konvergensi atau justru mengalami deviasi yang mengarah pada masalah stabilitas pelatihan. Perbandingan tren penurunan loss antara data pelatihan dan data validasi disajikan secara rinci pada grafik berikut.



Gambar 6. Gambar Grafik Loss Model CNN pada Data Pelatihan dan Validasi per *Epoch*

Evaluasi performa model CNN juga dilakukan dengan mengamati nilai loss pada data pelatihan dan validasi, sebagaimana ditunjukkan pada Gambar 4.3. Secara umum, nilai loss pada data pelatihan menunjukkan penurunan yang konsisten dari sekitar 4.13 di *epoch* awal hingga mencapai 3.70 pada akhir pelatihan (*epoch* ke-15). Penurunan ini menandakan bahwa model semakin mampu meminimalkan kesalahan prediksi pada data pelatihan seiring bertambahnya *epoch*.

Berbeda halnya dengan data validasi, nilai loss terlihat cukup stabil di sekitar 4.09 hingga *epoch* ke-9, namun mengalami lonjakan tajam pada *epoch* ke-11 dengan nilai di atas 4.3, sebelum kembali menurun dan berfluktuasi pada rentang 4.00 hingga 4.10 di akhir pelatihan. Fluktuasi dan ketidakkonsistenan ini mengindikasikan bahwa model belum berhasil mempelajari pola umum dari data validasi secara efektif. Perbedaan tren yang signifikan antara loss pelatihan dan validasi menjadi indikasi bahwa model belum mampu melakukan generalisasi dengan baik, dan kemungkinan besar mengalami masalah overfitting ringan pada *epoch* tengah, meskipun tren akhirnya lebih menyerupai underfitting karena akurasi yang rendah.

Gambaran ini menguatkan kesimpulan sebelumnya bahwa meskipun model menunjukkan pembelajaran terhadap data pelatihan, kemampuan generalisasi terhadap data validasi masih sangat terbatas. Diperlukan langkah-langkah perbaikan seperti augmentasi data, penyesuaian arsitektur model, serta eksperimen terhadap jumlah *epoch* atau penggunaan teknik regularisasi lanjutan.

3.4 Evaluasi Kinerja Model Klasifikasi CNN

Setelah mengamati performa model melalui grafik akurasi dan loss selama proses pelatihan, tahap selanjutnya dilakukan evaluasi menggunakan data validasi untuk mengetahui kemampuan model dalam melakukan klasifikasi multi-kelas. Evaluasi dilakukan menggunakan metrik akurasi,

precision, *recall*, dan *f1-score*. Namun, interpretasi terhadap metrik tersebut perlu dilakukan secara hati-hati karena jumlah data pengujian pada setiap kelas sangat terbatas. Adapun ringkasan hasil pengujian model tersebut disajikan secara detail pada gambar berikut.

accuracy			0.0370	54
macro avg	0.0053	0.0364	0.0091	54
weighted avg	0.0054	0.0370	0.0093	54

Gambar 7. Hasil Evaluasi Model CNN

Hasil evaluasi menunjukkan bahwa sebagian besar kelas memiliki nilai *precision*, *recall*, dan *f1-score* sebesar 0.0000. Kondisi ini menunjukkan bahwa model belum mampu mengenali pola visual secara konsisten pada sebagian besar kelas ras kucing. Nilai akurasi keseluruhan yang diperoleh sebesar 0.0370 atau sekitar 3,7% mengindikasikan bahwa performa model masih sangat rendah dan mendekati prediksi acak pada klasifikasi multi-kelas dengan 60 kategori.

Data validasi yang digunakan pada tahap evaluasi terdiri dari 54 citra yang tersebar pada 60 kelas ras kucing. Jumlah data uji yang sangat terbatas menyebabkan hasil evaluasi belum cukup representatif untuk menggambarkan performa model secara menyeluruh. Pada beberapa kelas, data pengujian hanya terdiri dari satu citra sehingga nilai *precision*, *recall*, dan *f1-score* menjadi kurang stabil untuk digunakan sebagai dasar pengambilan kesimpulan ilmiah. Oleh karena itu, hasil evaluasi pada penelitian ini lebih difokuskan untuk menunjukkan keterbatasan model ketika diterapkan pada dataset kecil dengan jumlah kelas yang banyak.

Rendahnya performa model diduga dipengaruhi oleh beberapa faktor. Pertama, jumlah citra pelatihan yang terbatas menyebabkan model kesulitan mempelajari representasi fitur visual yang memadai pada setiap kelas. Kedua, tingginya jumlah kelas dibandingkan jumlah data per kelas meningkatkan kompleksitas proses klasifikasi. Ketiga, kemiripan karakteristik visual antar ras kucing menyebabkan model sulit membedakan fitur spesifik dari masing-masing kategori.

Hasil penelitian ini berbeda dengan penelitian sebelumnya yang menggunakan dataset publik berukuran besar, seperti Oxford-IIIT Pet Dataset, yang mampu menghasilkan akurasi di atas 90% pada klasifikasi ras kucing (Fawwaz *et al.*, 2021). Perbedaan tersebut menunjukkan bahwa ukuran dataset, distribusi data, dan jumlah sampel per kelas memiliki pengaruh besar terhadap performa CNN dalam klasifikasi citra multi-kelas.

Berdasarkan hasil evaluasi, model pada penelitian ini masih mengalami underfitting karena belum mampu mempelajari pola visual secara optimal baik pada data pelatihan maupun data validasi. Oleh karena itu, peningkatan performa model pada penelitian selanjutnya dapat dilakukan melalui penambahan jumlah dataset, pengurangan jumlah kelas, penggunaan validasi silang (*cross-validation*), serta penerapan *transfer learning* dan fine-tuning menggunakan model pra-latih dengan dataset yang lebih besar dan seimbang.

3.5 Visualisasi Hasil Prediksi Model

Tahap akhir dari evaluasi eksperimen ini adalah melakukan uji coba prediksi secara langsung terhadap beberapa sampel citra untuk melihat performa model dalam kondisi riil. Visualisasi ini bertujuan untuk memvalidasi apakah label yang diprediksi oleh arsitektur CNN sesuai dengan label sebenarnya (*ground truth*) serta mengidentifikasi karakteristik visual apa yang menyebabkan model berhasil atau gagal dalam melakukan klasifikasi. Melalui observasi langsung pada hasil citra, kita dapat memahami perilaku model secara lebih intuitif dibandingkan hanya melalui angka

statistik. Contoh hasil klasifikasi pada beberapa sampel ras kucing ditampilkan pada gambar berikut.



Gambar 8. Gambar hasil prediksi klasifikasi ras kucing

Menampilkan lima contoh prediksi model CNN terhadap gambar ras kucing, yang disertai label prediksi dan label sebenarnya (*ground truth*). Dari lima gambar yang ditampilkan, tiga di antaranya berhasil diklasifikasikan dengan benar oleh model, yaitu dua gambar dengan label somali dan satu gambar bombay. Hal ini menunjukkan bahwa untuk beberapa kelas tertentu, model memiliki kemampuan membedakan fitur visual secara efektif.

Namun, terdapat dua gambar yang diklasifikasikan secara keliru oleh model. Salah satunya adalah gambar kucing ras laperm yang diprediksi sebagai *american_wirehair*. Kesalahan ini mungkin terjadi karena kemiripan visual antara kedua ras tersebut atau karena keterbatasan model dalam mengenali pola spesifik pada gambar dengan jumlah data pelatihan yang terbatas. Kesalahan seperti ini umum terjadi pada dataset yang memiliki jumlah gambar yang tidak seimbang antar kelas atau fitur visual yang saling tumpang tindih.

Visualisasi hasil prediksi seperti ini penting untuk memahami secara langsung bagaimana model berperilaku pada data nyata. Selain memberikan insight terhadap performa model, visualisasi ini juga membantu dalam proses debugging dan pengembangan lebih lanjut, terutama dalam menentukan kelas mana saja yang perlu perhatian lebih seperti penambahan data atau augmentasi.

Kesimpulan

Penelitian ini telah mengimplementasikan *Convolutional Neural Network* (CNN) untuk klasifikasi ras kucing berbasis citra digital menggunakan dataset yang terdiri dari 60 kelas ras kucing. Proses penelitian meliputi tahap pengumpulan dataset, pra-pemrosesan citra, pelatihan model CNN, serta evaluasi performa menggunakan metrik akurasi, *precision*, *recall*, dan *f1-score*.

Hasil eksperimen menunjukkan bahwa performa model klasifikasi masih belum optimal. Nilai akurasi validasi sebesar 3,7% menunjukkan bahwa model mengalami kesulitan dalam melakukan klasifikasi multi-kelas secara akurat. Rendahnya performa model dipengaruhi oleh keterbatasan jumlah data pada setiap kelas, tingginya jumlah kategori klasifikasi, serta kemiripan karakteristik visual antar ras kucing yang menyebabkan proses ekstraksi fitur menjadi kurang efektif.

Hasil evaluasi juga menunjukkan bahwa model masih mengalami underfitting dan belum mampu melakukan generalisasi dengan baik terhadap data validasi. Oleh karena itu, hasil penelitian ini lebih menekankan pada analisis keterbatasan implementasi CNN pada dataset kecil dengan jumlah kelas yang banyak dibandingkan pencapaian performa klasifikasi tinggi.

Penelitian selanjutnya disarankan menggunakan dataset dengan jumlah citra yang lebih besar dan distribusi kelas yang lebih seimbang. Selain itu, penerapan *transfer learning*, fine-tuning model pra-latih, optimasi hyperparameter, dan validasi silang (cross-validation) dapat dilakukan untuk meningkatkan performa dan kemampuan generalisasi model klasifikasi ras kucing.

Daftar Pustaka

- Agusniar, C., & Adelia, D. (2024). Penerapan *Convolutional Neural Network* pada Klasifikasi Jenis Ras Kucing Menggunakan ResNet50V2. *Journal of Information Engineering and Educational Technology*, 8(1), 1–6. <https://doi.org/10.26740/jieet.v8n1.p1-6>
- Aliefah Syalma Ratsdea Mufti, & Yovi Litanianda. (2024). Implementasi Metode *Convolutional Neural Network* (CNN) untuk Klasifikasi Jenis Ras Kucing. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains Dan Informatika*, 2(2), 104–110. <https://doi.org/10.61132/uranus.v1i2.172>
- Clement, F., Yang, J., & Cheng, I. (2024). *Feature CAM: Interpretable AI in Image Classification*. 1–16.
- Dede Husen. (2024). Evaluasi Teknik Augmentasi Data Untuk Klasifikasi Tumor Otak Menggunakan Cnn Pada Citra Mri. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 5(2), 219–228. <https://doi.org/10.46764/teknimedia.v5i2.220>
- Eichstädt, T., & Spieker, S. (2024). *Deep learning*. 52 *Stunden Informatik*, (November), 391–397. https://doi.org/10.1007/978-3-658-41838-0_46
- Fawwaz, M. A. A., Ramadhani, K. N., & Sthevani, F. (2021). Klasifikasi Ras pada Kucing menggunakan Algoritma *Convolutional Neural Network* (CNN). *E-Proceeding of Engineering*, 8(1), 715–730.
- FIKRY, R. M., & NURHASANAH, Y. I. (2023). Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar. *E-Proceeding FTI*, 1–21.
- Linda, K. D., Kusrini, K., & Hartanto, A. D. (2024). Studi Literatur Mengenai Klasifikasi Citra Kucing Dengan Menggunakan *Deep learning: Convolutional Neural Network* (CNN). *Journal of Electrical Engineering and Computer (JEECOM)*, 6(1), 129–137. <https://doi.org/10.33650/jeeecom.v6i1.7480>
- Pane, Y. Y., & Sihombing, J. J. (2023). Klasifikasi Jenis Burung menggunakan Metode *Transfer learning*. *Jurnal Teknologi Terpadu*, 9(2), 89–94. <https://doi.org/10.54914/jtt.v9i2.744>
- Rahayu, M. I. (2023). Klasifikasi ras kucing menggunakan metadata dataset kaggle dengan *framework* yolo v5. *Jurnal Teknologi Informasi Dan Komunikasi*, 12(1).
- Rahma, I. A., & Suadaa, L. H. (2023). Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(6), 1329–1340. <https://doi.org/10.25126/jtiik.2023107325>
- Ramadhan, F., & Hernadi, J. (2025). Evaluasi Optimizer Adam dan RMSProp pada Arsitektur VGG-19 Klasifikasi Ekspresi Wajah Manusia. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(2), 1414–1426. <https://doi.org/10.29100/jupi.v10i2.6197>
- Ramayanti, D., Sri Dianing Asri, & Lionie Lionie. (2022). Implementasi Model Arsitektur VGG16

- dan MobileNetV2 Untuk Klasifikasi Citra Kupu-Kupu. *JSAI (Journal Scientific and Applied Informatics)*, 5(3), 182–187. <https://doi.org/10.36085/jsai.v5i3.2864>
- Sattarzadeh, S., Sudhakar, M., Plataniotis, K. N., Jang, J., Jeong, Y., & Kim, H. (2021). Integrated grad-cam: Sensitivity-aware visual explanation of deep convolutional networks via integrated gradient-based scoring. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2021-June*, 1775–1779. <https://doi.org/10.1109/ICASSP39728.2021.9415064>
- Somoal, M. G., & Dzikrillah, A. R. (2025). Komparasi MobileNETV2 dengan Kustomisasi *Transfer learning* dan Hyperparameter untuk Identifikasi Tumor Otak. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 12(1), 229–240. <https://doi.org/10.25126/jtiik.2025129582>
- Supriyanti, W., Sujalwo, Anggoro, D. A., Maryam, & Romadloni, N. T. (2024). Optimizing Cervical Cancer Diagnosis with Correlation-Based Feature Selection: A Comparative Study of Machine Learning Models. *International Journal of Advanced Computer Science and Applications*, 15(12), 698–707. <https://doi.org/10.14569/IJACSA.2024.0151270>
- Wasil, M., Harianto, H., & Fathurrahman, F. (2022). Pengaruh *Epoch* pada Akurasi menggunakan *Convolutional Neural Network* untuk Klasifikasi fashion dan Furniture. *Infotek: Jurnal Informatika Dan Teknologi*, 5(1), 53–61. <https://doi.org/10.29408/jit.v5i1.4393>
- Wijaya Kusuma, W., Rizal Isnanto, R., Fauzi, A., & Korespondensi, P. (2023). DenseNet121 Menggunakan Kerangka Kerja TensorFlow untuk Deteksi Jenis Hewan. *Jurnal Teknik Komputer*, 1(4), 141–147. <https://doi.org/10.14710/jtk.v1i4.37009>