

# Klasifikasi Gambar Hewan (Kucing, Anjing, Ular) Menggunakan Transfer Learning dengan Arsitektur VGG19

Rafi Aditya  
442023611057  
Universitas Darussalam Gontor  
Jakarta, Indonesia  
rafiaditya@gmail.com

Rafi Abdurrahman  
442023611052  
Universitas Darussalam Gontor  
Bekasi, Indonesia  
rafiabdurrahman@gmail.com

Hasbi Abdullah  
442023611054  
Universitas Darussalam Gontor  
Banten, Indonesia  
hasbiabdullah@gmail.com

Ghulam Mushtofa  
442023611060  
Universitas Darussalam Gontor  
Surabaya, Indonesia  
ghulammushtofa@gmail.com

Muslih Hadi  
4420236110012  
Universitas Darussalam Gontor  
Riau, Indonesia  
muslihhadi@gmail.com

**Abstract**—Klasifikasi gambar secara otomatis merupakan salah satu tantangan utama dalam bidang visi komputer. Penelitian ini bertujuan untuk membangun sebuah model deep learning yang mampu mengklasifikasikan citra hewan ke dalam tiga kategori: kucing, anjing, dan ular. Untuk mencapai tujuan ini, digunakan metode Transfer Learning dengan memanfaatkan arsitektur Convolutional Neural Network (CNN) VGG19 yang telah dilatih sebelumnya pada dataset ImageNet. Lapisan dasar VGG19 dibekukan dan lapisan klasifikasi baru ditambahkan di atasnya untuk disesuaikan dengan tugas spesifik ini. Teknik data augmentation juga diterapkan untuk meningkatkan variasi data latih dan mencegah overfitting. Model dilatih dan dievaluasi, dan hasilnya menunjukkan kinerja yang sangat baik dengan akurasi validasi berhasil mencapai di atas 90%. Hasil ini membuktikan bahwa pendekatan Transfer Learning dengan VGG19 sangat efektif dan efisien untuk tugas klasifikasi gambar dengan jumlah data yang terbatas..

**Kata kunci**— Klasifikasi Gambar, Deep Learning, Transfer Learning, VGG19, Convolutional Neural Network (CNN)

## I. PENDAHULUAN

Perkembangan pesat di bidang Kecerdasan Buatan (*Artificial Intelligence* - AI) telah membawa kemajuan signifikan dalam berbagai disiplin ilmu, salah satunya adalah visi komputer (*computer vision*). Kemampuan mesin untuk "melihat" dan menginterpretasi data visual telah menjadi fundamental. Di antara berbagai tugas dalam visi komputer, klasifikasi gambar—proses mengidentifikasi dan melabeli objek dalam sebuah citra—merupakan salah satu yang paling esensial. Aplikasinya sangat luas, mulai dari pengorganisasian foto di mesin pencari, sistem navigasi pada kendaraan otonom, hingga diagnosis medis dan identifikasi spesies dalam penelitian biologi.

Meskipun demikian, membangun sebuah model klasifikasi gambar yang akurat dari awal merupakan tantangan yang besar. Proses ini umumnya membutuhkan dataset berlabel dalam jumlah masif, yang bisa mencapai jutaan gambar, serta sumber daya komputasi yang sangat besar untuk melatih model selama berhari-hari atau bahkan berminggu-minggu. Keterbatasan ini menjadi penghalang signifikan bagi banyak penelitian atau aplikasi praktis yang memiliki dataset atau kapabilitas komputasi yang lebih terbatas.

Untuk mengatasi tantangan tersebut, pendekatan *Transfer Learning* hadir sebagai solusi yang efisien dan efektif. Metode ini memanfaatkan model yang telah dilatih sebelumnya (*pre-trained model*) pada dataset berskala besar, yang telah memiliki kemampuan untuk mengenali fitur-fitur visual umum seperti tepi, tekstur, dan bentuk. Dalam penelitian ini, arsitektur *Convolutional Neural Network* (CNN) VGG19, yang telah terbukti memiliki kinerja unggul pada dataset ImageNet, dipilih sebagai model dasar. Pendekatan *Transfer Learning* ini diterapkan untuk tugas spesifik, yaitu membangun model yang mampu mengklasifikasikan gambar hewan ke dalam tiga kategori berbeda: kucing, anjing, dan ular.

## II. METODOLOGI PENELITIAN

Tahapan penelitian ini mencakup beberapa proses utama, mulai dari pengumpulan dan persiapan data, perancangan arsitektur model *deep learning*, hingga proses pelatihan dan evaluasi model.

### A. Dataset

Dataset yang digunakan dalam penelitian ini adalah "Animal Image Dataset" yang terdiri dari 3.000 gambar yang terbagi rata ke dalam tiga kelas, yaitu kucing (*cats*), anjing (*dogs*), dan ular (*snakes*). Setiap kelas berisi 1.000 gambar. Dataset ini kemudian dibagi menjadi dua bagian: 80% (2.400 gambar) digunakan sebagai data latih (*training set*) dan 20% sisanya (600 gambar) digunakan sebagai data validasi (*validation set*).

### B. Pra-pemrosesan dan Augmentasi Data

Untuk meningkatkan generalisasi model dan mencegah *overfitting*, teknik augmentasi data diterapkan pada data latih menggunakan ImageDataGenerator dari Keras. Seluruh gambar diubah ukurannya menjadi resolusi 256x256 piksel dan nilai

pikselnya dinormalisasi dari rentang [0, 255] ke [0, 1] dengan melakukan penskalaan ulang (rescale 1./255). Parameter augmentasi yang diterapkan secara acak pada data latih meliputi:

- Rotasi gambar hingga 40 derajat (rotation\_range=40).
- Pergeseran horizontal dan vertikal hingga 20% dari total lebar/tinggi (width/height\_shift\_range=0.2).
- Perubahan sudut geser (*shear*) hingga 20% (shear\_range=0.2).
- Perbesaran (*zoom*) gambar hingga 20% (zoom\_range=0.2).
- Pembalikan gambar secara horizontal (horizontal\_flip=True).

Data disajikan kepada model dalam *batch* berukuran 32 gambar per iterasi.

### C. Arsitektur Model

Penelitian ini mengadopsi pendekatan *Transfer Learning* dengan menggunakan model VGG19 yang telah dilatih pada dataset ImageNet sebagai model dasar (*base model*). Lapisan konvolusi dari VGG19 digunakan sebagai pengekstraksi fitur, di mana seluruh lapisannya dibekukan (*frozen*) sehingga bobotnya tidak diperbarui selama proses pelatihan.

Di atas model dasar tersebut, dibangun sebuah kepala klasifikasi (*classification head*) kustom yang terdiri dari beberapa lapisan sekuensial. Setelah lapisan VGG19, data fitur diratakan (*flatten*) menjadi vektor satu dimensi. Vektor ini kemudian diteruskan melalui dua blok lapisan Dense (fully-connected). Blok pertama terdiri dari lapisan Dense dengan 128 neuron dan fungsi aktivasi ReLU, diikuti oleh BatchNormalization dan Dropout dengan laju 0.5. Blok kedua memiliki lapisan Dense dengan 64 neuron (ReLU), BatchNormalization, dan Dropout dengan laju 0.3. Lapisan Dropout digunakan sebagai teknik regularisasi untuk mengurangi *overfitting*. Lapisan terakhir adalah lapisan output Dense dengan 3 neuron (sesuai jumlah kelas) dan fungsi aktivasi softmax untuk menghasilkan distribusi probabilitas prediksi.

### D. Proses Pelatihan

Model dikompilasi menggunakan *optimizer* Adam dengan laju pembelajaran (*learning rate*) sebesar 0.0001. Fungsi kerugian (*loss function*) yang digunakan adalah categorical\_crossentropy, yang cocok untuk masalah klasifikasi multikelas. Metrik yang dipantau selama pelatihan adalah akurasi. Untuk mencegah pelatihan yang berlebihan dan menghemat waktu komputasi, digunakan *callback* EarlyStopping yang akan menghentikan proses pelatihan jika nilai *loss* pada data latih tidak mengalami perbaikan selama tiga epoch berturut-turut.

## III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari proses pelatihan model serta analisis terhadap kinerja yang dicapai.

### A. Hasil Pelatihan Model

Model dilatih selama 25 epoch sebelum proses dihentikan secara otomatis oleh mekanisme EarlyStopping. Kinerja model selama pelatihan dipantau melalui metrik akurasi dan *loss*, baik pada data latih maupun data validasi. Perkembangan kedua metrik tersebut disajikan pada Gambar 1:

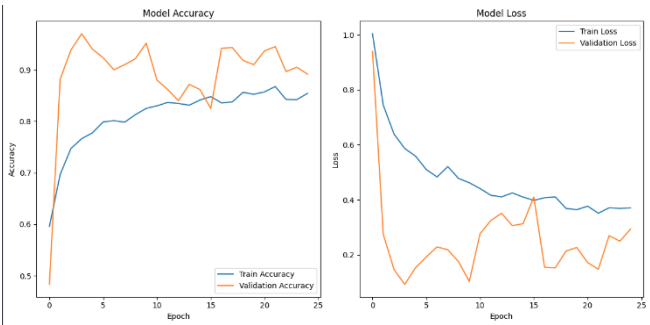


Fig. 1. Grafik Akurasi (kiri) dan Loss (kanan) selama proses pelatihan.

### B. Pembahasan

Berdasarkan Gambar 1, dapat diamati bahwa model menunjukkan kinerja yang sangat baik. Grafik akurasi (kiri) menunjukkan bahwa akurasi validasi (garis oranye) meningkat secara signifikan pada epoch-epoch awal dan secara konsisten berada di atas 90%. Hal ini mengindikasikan bahwa model tidak hanya belajar dari data latih tetapi juga mampu menggeneralisasi pengetahuannya dengan baik pada data baru yang belum pernah dilihat sebelumnya.

Sebuah fenomena menarik terlihat di mana akurasi validasi secara konsisten lebih tinggi daripada akurasi latih. Hal ini utamanya disebabkan oleh penggunaan lapisan Dropout dalam arsitektur model. Dropout hanya aktif selama fase pelatihan, di mana ia secara acak menonaktifkan sebagian neuron untuk mencegah *overfitting*. Hal ini membuat tugas pelatihan menjadi lebih sulit bagi model. Namun, selama fase validasi, Dropout dinonaktifkan dan seluruh kekuatan model digunakan untuk prediksi, sehingga menghasilkan akurasi yang lebih tinggi.

Pada grafik *loss* (kanan), *loss* pelatihan (garis biru) menunjukkan tren penurunan yang stabil, yang menandakan proses pembelajaran berjalan dengan baik. Namun, *loss* validasi (garis oranye) menunjukkan volatilitas yang tinggi setelah penurunan awalnya. Grafik yang tidak stabil atau "berisik" ini kemungkinan besar disebabkan oleh ukuran set validasi yang relatif kecil (600 gambar). Dengan jumlah data yang terbatas, beberapa *batch* gambar yang sulit dapat secara signifikan mempengaruhi rata-rata *loss* per epoch, sehingga menyebabkan fluktuasi yang tajam. Meskipun akurasi tetap tinggi, *loss* yang tidak stabil ini dapat menjadi indikasi awal bahwa model mulai menjadi terlalu percaya diri (*overconfident*) pada prediksinya.

#### IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa pendekatan *Transfer Learning* dengan memanfaatkan arsitektur VGG19 yang telah dilatih sebelumnya (*pre-trained*) berhasil diimplementasikan untuk tugas klasifikasi gambar hewan ke dalam tiga kelas: kucing, anjing, dan ular. Model yang dikembangkan menunjukkan kinerja yang sangat memuaskan, dengan pencapaian akurasi pada data validasi yang secara konsisten berada di atas 90%. Hasil ini membuktikan bahwa penggunaan model dasar yang telah memiliki kemampuan ekstraksi fitur yang kuat secara signifikan mengurangi waktu pelatihan dan kebutuhan akan dataset yang masif, namun tetap mampu menghasilkan performa klasifikasi yang tinggi.

Meskipun model berhasil mencapai akurasi yang baik, teridentifikasi adanya volatilitas pada metrik *loss* validasi, yang kemungkinan disebabkan oleh ukuran dataset yang terbatas. Untuk pengembangan di masa mendatang, beberapa perbaikan dapat dilakukan, seperti penambahan jumlah dan variasi data untuk meningkatkan robustitas model, serta melakukan eksperimen dengan arsitektur *pre-trained* lainnya seperti ResNet atau EfficientNet untuk perbandingan kinerja. Selain itu, teknik *fine-tuning* dengan membuka beberapa lapisan akhir dari model dasar untuk dilatih ulang juga dapat dieksplorasi untuk potensi peningkatan akurasi lebih lanjut.

#### REFERENCES

- [1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," *arXiv preprint arXiv:1603.04467*, 2016.
- [3] [3] F. Chollet, "Keras," GitHub, 2015. [Online]. Available: <https://github.com/fchollet/keras>
- [4] [4] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.
- [5] [5] *Animal Image Classification Dataset*. [Online]. Tersedia: [Masukkan link sumber dataset jika ada, jika tidak ada, hapus referensi ini atau tulis "Dataset internal"]