

Analisis Perbandingan Ketahanan Gangguan Gambar Terhadap Arsitektur Model MobileNetV2 dan EfficientNet-Lite0 dalam Mengklasifikasikan Jenis Jamur

Muhammad Reno Afrido Arwinto¹, Yasir Galuh Adi Saputro², Bayu Ade Prasetya³, Rajnaparamitha Kusumastuti⁴

¹²³⁴Prodi S1 Teknik Informatika, Prodi Teknologi Informasi STMIK Amikom Surakarta

¹²³⁴Sukoharjo Indonesia

Email: ¹mreno6682@gmail.com, ²yasirgaluh@gmail.com,
³bayuade858@gmail.com, ⁴rajna@dosen.amikomsolo.ac.id

Abstract

Convolutional Neural Network (CNN) models designed for mushroom classification often encounter significant performance degradation when deployed in real-world scenarios. This issue stems from the disparity between "clean" (ideal) training data and field-acquired data, which is frequently subject to visual distortions such as uneven lighting (brightness), blur, and viewpoint variations (rotation). This study conducts a comparative analysis to evaluate the robustness of two widely used lightweight architectures, MobileNetV2 and EfficientNet-Lite0, against these three types of distortions. The study utilizes the Mushroom Classification Dataset, comprising 9 mushroom classes, for testing purposes. The models were trained for 50 epochs and systematically evaluated using test data subjected to varying levels of brightness (25% to 150%), Gaussian blur (sigma 1 to 5), and fixed rotation (20° to 300°). The results indicate that EfficientNet-Lite0 achieved the highest accuracy of 73% on clean test data without augmentation. EfficientNet-Lite0 demonstrated significantly superior robustness against variations in lighting and rotation. This model maintained stable accuracy under low-light conditions and across various object orientations, whereas MobileNetV2 suffered substantial performance declines. However, both architectures exhibited high vulnerability to Gaussian blur. Increased blur intensity resulted in severe performance degradation, indicating a strong reliance on sharp textural features for classification. The study concludes that EfficientNet-Lite0 is a more robust and reliable architecture for mushroom classification applications in unpredictable field conditions.

Keywords: Data Augmentation, Convolutional Neural Network (CNN), EfficientNet-Lite0, Model Robustness, Fungal Classification

Abstraksi

Model Convolutional Neural Network (CNN) untuk klasifikasi jamur sering menghadapi penurunan performa drastis ketika diimplementasikan di dunia nyata. Hal ini disebabkan oleh kesenjangan antara data latih yang "bersih" (ideal) dan data akuisisi lapangan yang sering mengalami gangguan visual, seperti pencahayaan tidak merata (brightness), keburaman (blur), dan variasi sudut pengambilan gambar (rotation). Penelitian ini melakukan analisis perbandingan untuk mengevaluasi ketahanan (robustness) dua arsitektur ringan yang sering digunakan, MobileNetV2 dan EfficientNet-

Lite0 terhadap tiga jenis gangguan tersebut. Data uji yang digunakan yaitu Mushroom Classification Dataset yang terdiri dari 9 kelas jenis jamur. Model dilatih dengan 50 epoch dan diuji kinerjanya secara berulang menggunakan data uji yang telah diberi gangguan brightness (25% hingga 150%), gaussian blur (sigma 1 hingga 5), dan rotasi tetap (20° hingga 300°). Hasil penelitian dengan akurasi tertinggi mencapai 73% menggunakan EfficientNet-Lite0 dengan data uji yang bersih tanpa augmentasi. EfficientNet-Lite0 secara signifikan lebih unggul dalam hal ketahanan terhadap variasi pencahayaan dan rotasi. Model ini mampu mempertahankan akurasi yang stabil pada kondisi minim cahaya dan pada berbagai orientasi objek, di mana MobileNetV2 mengalami penurunan kinerja yang substansial. Namun kedua arsitektur menunjukkan kerentanan yang tinggi terhadap gaussian blur. Peningkatan intensitas blur menyebabkan degradasi performa yang parah, mengindikasikan ketergantungan tinggi pada fitur tekstural yang tajam. Penelitian ini menyimpulkan bahwa EfficientNet-Lite0 merupakan arsitektur yang lebih robust dan andal untuk aplikasi klasifikasi jamur di kondisi lapangan yang tidak dapat diprediksi.

Kata Kunci: Augmentasi Data, Convolutional Neural Network (CNN), EfficientNet-Lite0, Ketahanan Model (Robustness), Klasifikasi Jamur

1. PENDAHULUAN

Kecerdasan buatan terus berevolusi dan semakin berperan dalam kehidupan manusia, seiring dengan kemajuan teknologi komputer. Kecerdasan buatan (AI) adalah cabang ilmu komputer yang berkonsentrasi pada pengembangan sistem komputer yang mampu menjalankan tugas selayaknya manusia. Tujuan kecerdasan buatan yaitu membuat mesin mampu belajar, memahami, merencanakan, dan beradaptasi sehingga mampu menyelesaikan tugas secara mandiri [1]. Pertumbuhan ini juga mendorong peningkatan jumlah penelitian di bidang kecerdasan buatan. Kini, kecerdasan buatan telah terbagi menjadi banyak sub-bidang dan banyak digunakan untuk memecahkan masalah-masalah kemanusiaan [2]. Di antara sub-bidang tersebut, pendekatan *deep learning* mampu memperluas kapabilitas teknologi *machine learning* untuk penerapan praktis. Sebagai contoh, *Convolutional Neural Network (CNN)*, yang merupakan algoritma *deep learning* yang sangat menjanjikan, memanfaatkan lapisan konvolusi yang berfungsi sebagai kernel. Berbagai arsitektur CNN tersebut dapat dimanfaatkan untuk memproses serta mengklasifikasikan citra masukan dan *feature maps* yang dihasilkan [3].

Salah satu bidang yang dapat memanfaatkan kemampuan klasifikasi citra CNN adalah mikologi, khususnya untuk identifikasi jamur. Jamur didefinisikan sebagai organisme tanpa klorofil yang tubuhnya tersusun dari hifa dan dinding selnya mengandung kitin. Identifikasi jamur secara akurat sangat penting, mengingat diperkirakan, 93% spesies jamur di seluruh dunia masih belum teridentifikasi, sementara jumlah spesies yang sudah dikenal saat ini berkisar antara 2,2 hingga 3,8 juta [4]. Penggunaan arsitektur CNN di lapangan diharapkan mampu membantu proses klasifikasi yang kompleks ini.

Namun, penerapan CNN di lapangan menghadapi tantangan. Model CNN biasanya dilatih menggunakan *dataset* citra yang “bersih” (ideal), yaitu gambar yang diambil dalam

kondisi terkontrol, fokus tajam, dan pencahayaan baik. Pada kenyataannya, data citra yang diperoleh di dunia nyata, misalnya dari kamera smartphone seringkali tidak sempurna. Citra tersebut kerap mengalami berbagai gangguan visual, seperti tangkapan yang kabur (*blur*), pencahayaan yang terlalu terang atau gelap (*brightness*), perbedaan sudut pengambilan (*rotation*).

Perbedaan antara data latih yang bersih dan data uji (dunia nyata) yang penuh gangguan ini dapat menyebabkan penurunan performa model secara drastis [5]. Hal ini menimbulkan pertanyaan penelitian yang krusial, seberapa tangguh model CNN modern yang ringan dijalankan dalam perangkat *mobile* terhadap gangguan visual tersebut. Penelitian ini berfokus pada *robustness* di arsitektur populer yang dikenal efisien untuk perangkat mobile. Di dalam penelitian [6] menyatakan secara eksplisit bahwa desain *MobileNetV2* cocok digunakan untuk inferensi pada perangkat dan aplikasi berdaya rendah. Hal tersebut mendukung klaim efisiensi dan kesesuaian untuk deployment ringan. Kemudian pada penelitian [7] menyebutkan bahwa *EfficientNet-Lite family* mempunyai ukuran file dan jumlah operasi yang relatif kecil, sehingga cocok dijalankan langsung pada ponsel atau perangkat *edge* tanpa butuh server kuat.

Oleh karena itu, tujuan dari penelitian ini adalah untuk menguji *robustness* model CNN, khususnya *MobileNetV2* dan *EfficientNet-Lite0*, terhadap beberapa jenis gangguan gambar yang sering terjadi di dunia nyata. Dengan tujuan tersebut, penelitian ini berkontribusi untuk memberikan opsi model CNN yang cocok untuk diimplementasikan ke perangkat *mobile* berdasarkan *robustness*-nya terhadap gangguan gambar yang telah diujikan.

2. TINJAUAN PUSTAKA

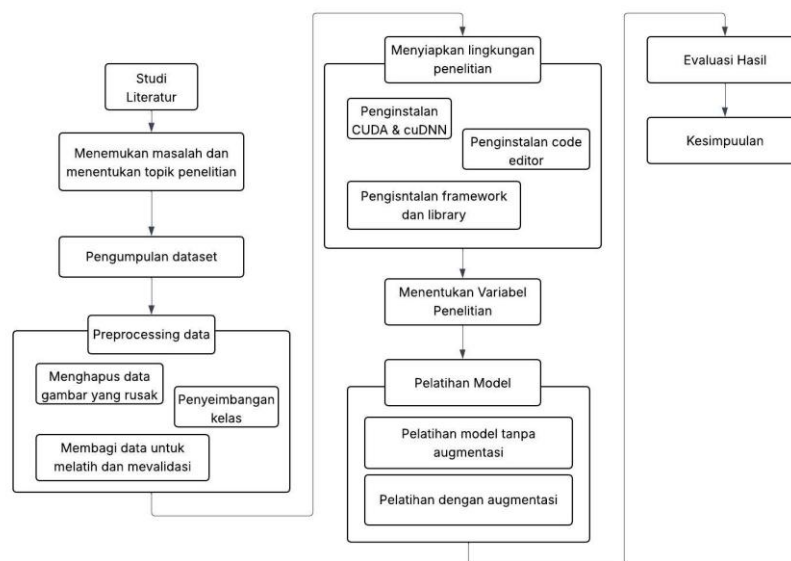
Penelitian yang serupa [8] yang mana menggunakan *dataset* yang sama kemudian dilakukan *preprocessing* memberikan hasil rata-rata *precision* 23,80%, *recall* 22,94%, dan akurasi mencapai 82,69%. Penelitian tersebut masih menggunakan fitur manual dengan SVM menghasilkan *recall* sangat rendah, yang mana membuktikan bahwa kombinasi warna dan bentuk sederhana tidak cukup untuk membedakan spesies jamur yang kompleks. Pada penelitian [9] menyarankan perlunya perbaikan pada *background* citra karena mempengaruhi akurasi, namun tidak ada studi kuantitatif mengenai seberapa besar dampak gangguan visual tersebut. Penelitian ini akan secara sistematis menganalisis degradasi performa model ketika dihadapkan pada gangguan gambar buatan untuk mensimulasikan kondisi pengambilan gambar di lapangan. Selain itu dalam penelitian [10] yang menggunakan 1200 data dengan perbandingan data latih dan uji 70:30 memberikan akurasi terbaik 89% untuk training dan 82% untuk validasi. Penelitian tersebut belum menguji keandalan model terhadap penurunan kualitas gambar. Penelitian ini akan mengisi celah tersebut dengan mengukur seberapa jauh performa *MobileNetV2* bertahan ketika diberi gangguan gambar buatan. Penelitian lain membahas augmentasi yang memperbaiki *robustness* seperti penelitian [11] kebanyakan metode augmentasi diuji pada model besar dan belum cukup bukti tentang bagaimana arsitektur

yang ringan merespon gangguan spesifik. Kemudian di penelitian lain yang berfokus pada *robustness* yang membahas tentang analisis lengkap bentuk ketahanan dan trade-off [12]. Namun, tidak berfokus pada perbandingan arsitektur ringan seperti *MobileNetV2* dan *EfficientNet-Lite0* secara konsisten.

Dari celah-celah penelitian yang telah dipaparkan diatas maka pada penelitian ini akan dibuktikan bagaimana model arsitektur yang populer dan ringan diuji dengan *dataset* klasifikasi jenis jamur. Penggunaan model *EfficientNet-Lite0* dan *MobileNetV2* difokuskan pada *robustness* terhadap gangguan gambar buatan yang disimulasikan selayaknya seperti cara menangkap gambar di lapangan.

3. METODE PENELITIAN

Penelitian ini dilakukan dengan melalui 9 tahap inti yaitu dimulai dari tahap studi literatur, kemudian menemukan masalah dan menentukan topik penelitian yang mana sudah dijelaskan di bagian pendahuluan, mencari *dataset*, *preprocessing dataset*, menyiapkan lingkungan penelitian, menentukan variabel penelitian, kemudian melakukan pelatihan model, dari hasil pelatihan model tersebut diambil hasil evaluasi kemudian disimpulkan. Lebih jelasnya digambarkan *flowchart* tahapan penelitian pada gambar 1 di bawah ini.



Gambar 1. Tahapan Penelitian

3.1. Mencari Dataset

Dataset yang digunakan yaitu *dataset* yang diunduh dari situs www.kaggle.com. *Dataset* mentah memiliki 11 kelas namun disini yang dipakai dan difokuskan pada kelas untuk mengklasifikasikan jenis jamur. Sehingga dua kelas yang mengklasifikasikan jamur yang boleh dimakan dan tidak boleh dimakan dihapus. Jumlah sampel 6.687 gambar kemudian dibagi menjadi 9 kelas yaitu *Agaricus* (353 gambar), *Amanita* (750 gambar), *Boletus* (1073 gambar), *Cortinarius* (836 gambar), *Entoloma* (364 gambar), *Hygrocybe*

(316 gambar), *Lactarius* (1536 gambar), *Russula* (1148 gambar), *Suillus* (311 gambar). Setiap ukuran resolusi gambar berbeda-beda satu sama lain. Format gambar yang dipakai yaitu .jpg secara seragam.

3.2. Preprocessing Data

Dataset yang masih mentah kemudian diberlakukan *preprocessing* data. Harapannya lewat tahapan ini, data yang ingin digunakan sudah bersih dan mampu memberikan akurasi yang tinggi. Dari data yang berjumlah total 6.687 gambar terdiri dari 9 kelas dengan jumlah gambar yang berbeda-beda, setelah dilakukan *preprocessing* data menjadi berjumlah setiap kelasnya terdiri dari 311 sampel gambar. *Preprocessing* dilakukan dengan cara undersampling karena data mentah yang tidak seimbang. Hal tersebut akan membuat model menjadi belajar lebih baik pada kelas yang *sample*-nya lebih banyak dibandingkan kelas yang *sample*-nya lebih sedikit. Gambar yang rusak dan memiliki kemiripan dihapus, kemudian seluruh kelas disetarakan jumlah sampelnya. Sehingga total data keseluruhan menjadi 2799 sampel gambar. Kemudian data dibagi menjadi dua bagian yaitu data untuk melatih dan data untuk melakukan validasi. Perbandingannya yaitu 80:20. Ukuran resolusi sampel gambar yang berbeda-beda diubah menjadi 224 x 224 *pixel* secara seragam.

3.3. Menyiapkan Lingkungan Penelitian

Tahap ini menyiapkan alat dan bahan pendukung dalam melakukan pelatihan model. Di sini penelitian menggunakan komputer lokal dengan spesifikasi *CPU AMD Ryzen 5700x* dengan *RAM 16 GB* dan menggunakan *GPU Nvidia RTX 3060 12 GB*. Perangkat yang digunakan yaitu *Software PyCharm Community Edition 2025* dengan menginstal *library Tensorflow GPU* versi 2.10.0. *CUDA Toolkit* yang digunakan versi 11.2 dan *cuDNN* yang digunakan yaitu versi 8.1. Bahasa pemrograman yang digunakan disini menggunakan bahasa pemrograman *python* versi 3.10.

3.4. Menentukan Variabel Penelitian

Agar memberikan perbandingan yang adil dan setara maka dibuat variabel-variabel yang mendukung penelitian.

3.4.1. Variabel independen

Menggunakan arsitektur model *MobileNetV2* dan *EfficientNet-Lite0* tanpa *fine-tuning* sebagai variabel yang ditetapkan. Pelatihan model menggunakan *transfer-learning* yang mana kedua model tidak dilatih dari awal, tetapi menggunakan *weights* yang sudah dipelajari sebelumnya dari *dataset ImageNet*. Pada bagian *feature extractor* dibekukan sedangkan di bagian *head classification* saja yang dilatih menggunakan *dataset* penelitian.

3.4.2. Variabel dependen

Variabel yang diukur disini yaitu persentase akurasi.

3.4.3. Variabel kontrol

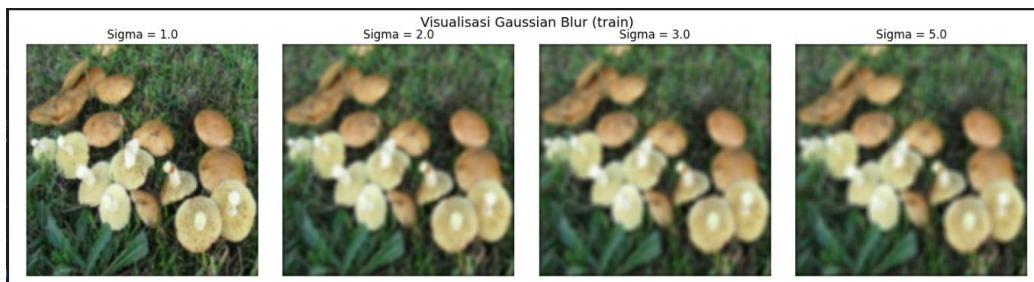
Agar perbandingan setara maka harus ada variabel yang disamakan. Disini input data dari pembagian data dan resolusi gambar disamakan. Input data dengan 2799 sampel gambar yang terdiri dari 9 kelas. Resolusi gambar diseragamkan menjadi 224x224 *pixel*. Kemudian metode untuk menormalisasikan *pixel* juga dibuat sama yaitu dengan skala $[0,1]$. *Optimizer* yang digunakan kedua arsitektur yaitu *optimizer Adam*. *Learning rate* disamakan dan diatur menjadi $1e-4$. *Learning rate schedule* yang digunakan yaitu *ReduceLROnPlateau* dengan pengaturan yang sama. *Loss function* diatur memakai *CategoricalCrossentropy*. Batch size berjumlah 32 dan *epoch* berjumlah 50.

Augmentasi data dengan parameter yang identik, terdiri dari mengatur kecerahan dengan 4 tingkat yaitu sangat gelap dengan tingkat *brightness* 25%, gelap dengan tingkat *brightness* 50%, terang dengan tingkat *brightness* 125%, dan sangat terang dengan tingkat *brightness* 150%, seperti yang digambarkan dalam gambar 2 berikut ini.



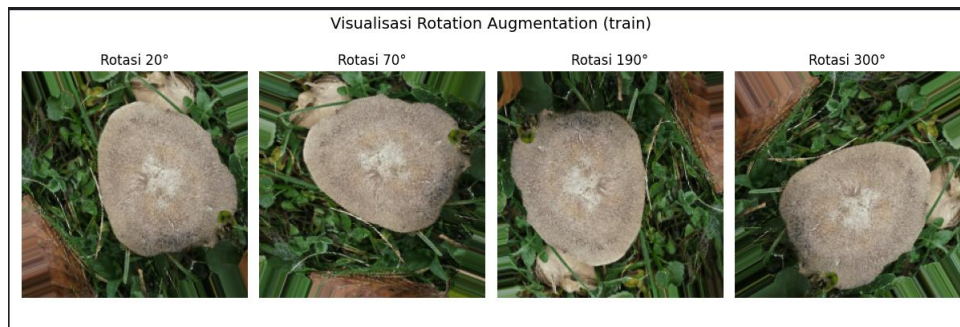
Gambar 2. Visualisasi Data Gambar dengan Augmentasi Brightness

Kemudian mengatur augmentasi data dengan *gaussian blur* yang terdiri dari 4 tingkat yaitu kabur ringan dengan nilai *sigma* = 1.0, kabur sedang dengan nilai *sigma* = 2.0, kabur berat dengan nilai *sigma* = 3.0, dan kabur ekstrem dengan nilai *sigma* = 5.0. Untuk lebih jelasnya digambarkan dalam gambar 3 di bawah ini.



Gambar 3. Visualisasi Data Gambar dengan Augmentasi Blur

Parameter *augmentasi* terakhir yaitu sudut rotasi tetap gambar. Terdiri dari rotasi kecil dengan dirotasi tetap sebesar 20° , rotasi sedang yang dirotasikan tetap sebesar 70° , kemudian rotasi kuat dengan dirotasikan tetap sebesar 190° , terakhir rotasi ekstrem dengan dirotasikan tetap sebesar 300° . Divisualisasikan dalam gambar 4 berikut ini.



Gambar 4. Visualisasi Data Gambar dengan Augmentasi Rotation

3.5. Skenario Penelitian

Pengujian dilakukan dengan total 26 kali *training* kedua model arsitektur CNN. Setiap model di-*training* dengan *augmentasi* yang sudah ditetapkan dengan 13 kali *training* setiap arsitektur modelnya. Dari 13 kali *training* model tersebut skenario pengujiannya yaitu pada satu model di-*training* dengan data uji bersih satu kali terlebih dahulu. Kemudian dilanjutkan dengan melakukan training menggunakan augmentasi berjenis *brightness* dengan empat parameter brightness tanpa kembali. Setelah itu model akan di-*training* dengan data uji augmentasi *blur* dengan empat parameter *blur*. Terakhir model dilatih dengan augmentasi *rotation* yang mana terdapat empat kali *training* dengan empat parameter *rotation*. Semua model di-*training* ulang untuk setiap skenario gangguan gambar. Terakhir hasil *training* dievaluasi dan dibandingkan yang kemudian menghasilkan kesimpulan penelitian.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian

Dari penelitian yang telah dilakukan memberikan hasil dan temuan sebagai berikut.

4.1.1. Performa Data Uji Bersih

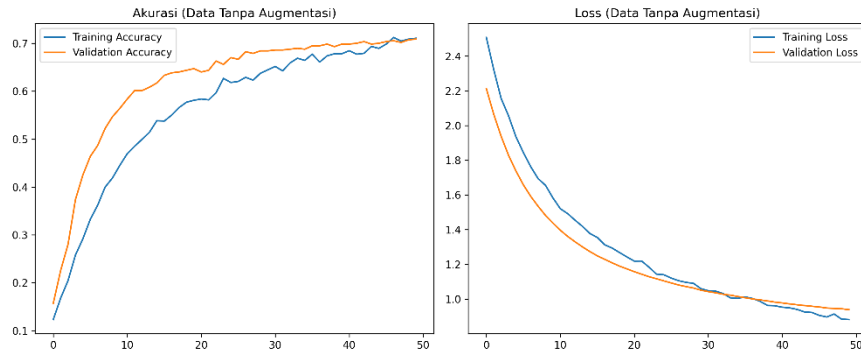
Sebelum diuji pada data dengan gangguan gambar, kedua model dievaluasi dengan menggunakan data uji bersih untuk mendapatkan akurasi optimalnya. Performa *baseline* ini menjadi acuan untuk mengukur seberapa besar penurunan performa akibat gangguan. Hasil pengujian disajikan di dalam tabel 1 berikut ini.

Tabel 1. Perbandingan Baseline Model pada Data Uji Bersih

Model	Akurasi
MobileNetV2	71%
EfficientNet-Lite0	73%

Dalam tabel 1 diatas menunjukkan bahwa model *EfficientNet-Lite0* memiliki akurasi yang lebih besar daripada model *MobileNetV2* pada pengujian data bersih. Model *EfficientNet-Lite0* mencapai akurasi 73% sedangkan model *MobileNetV2* 2% lebih rendah,

yaitu 71%. Hal tersebut menunjukkan bahwa kedua model hampir setara dalam mengenali citra jamur secara normal.



Gambar 5. Grafik Akurasi-Loss pada Model *MobileNetV2* dengan Data Bersih

Pada gambar 5 dapat dilihat grafik perkembangan *akurasi-loss* model *MobileNetv2* dengan data yang bersih. Kurva akurasi dan *loss* menunjukkan bahwa model tidak *overfitting*, namun masih berpotensi mengalami penurunan performa saat diuji pada data dengan kondisi data gambar yang berbeda. Hasilnya cenderung merefleksikan kemiripan distribusi antara data latih dan validasi, bukan kemampuan untuk mengenali gambar dengan murni.

4.1.2. Performa Data Uji *Brightness*

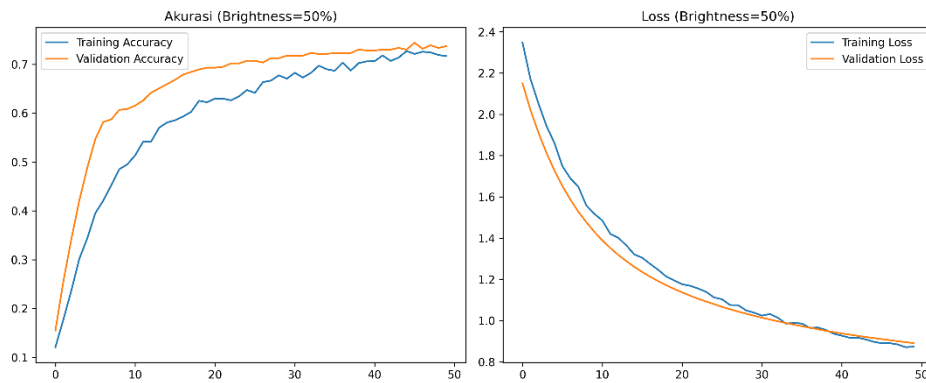
Tabel 2. Hasil Akurasi Perbandingan Model pada Data Uji *Brightness*

Model	Intensitas <i>Brightness</i>			
	25%	50%	125%	150%
<i>MobileNetV2</i>	56%	65%	70%	70%
<i>EfficientNet-Lite0</i>	74%	74%	73%	71%

Pada tabel 2 diatas dapat dilihat hasil akurasi dari percobaan *training* menggunakan *augmentasi brightness* yang intensitasnya sudah ditentukan. Pada model *MobileNetV2* kinerja meningkat seiring peningkatan *brightness* dari 25% sampai 125%, namun ketika intensitas *brightness* ditingkatkan ke 150% tidak memberikan peningkatan akurasi. Akurasi tertinggi adalah 70% pada *brightness* 125%. Hal tersebut menunjukkan bahwa model *MobileNetV2* relatif sensitif terhadap pencahayaan rendah, tetapi stabil di pencahayaan tinggi. Sedangkan model *EfficientNet-Lite0* memberikan akurasi yang relatif konsisten tinggi di semua tingkat *brightness*. Akurasinya mampu melebihi hasil akurasi pada percobaan *training* dengan data yang bersih. Performa penurunan sebesar 2% saat *brightness* melebihi 125%. Hal ini menunjukkan bahwa *EfficientNet-Lite0* memiliki ketahanan yang baik terhadap variasi pencahayaan.

Gambar 6 di bawah adalah grafik pelatihan model *EfficientNet-Lite0* dengan data yang diaugmentasi *brightness* sebesar 50% menghasilkan performa optimal dan stabil,

dengan akurasi sekitar 74% dan loss konvergen di kisaran 0.9. Model menunjukkan kemampuan generalisasi yang baik tanpa *overfitting*, serta mampu mempertahankan performa serupa dengan *brightness* 25%.



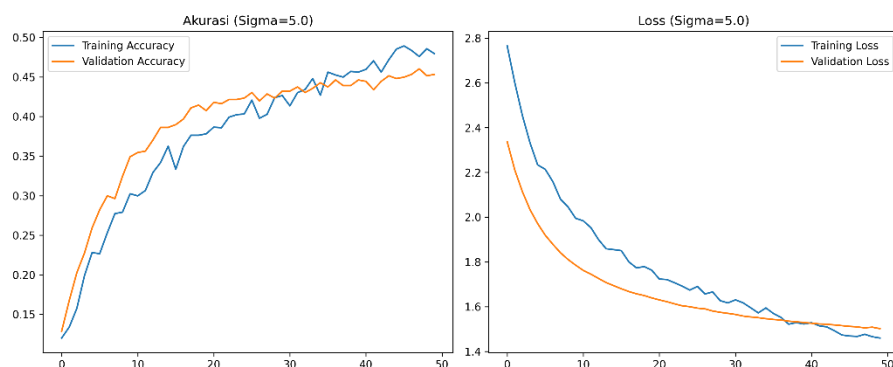
Gambar 6. Grafik Akurasi-Loss pada Model EfficientNet-Lite0 dengan *Brightness* 50%

4.1.3. Performa Data Uji *Blur*

Tabel 3. Hasil Akurasi Perbandingan Model pada Data Uji *Gaussian Blur*

Model	Nilai <i>Sigma Blur</i>			
	<i>sigma</i> = 1	<i>sigma</i> = 2	<i>sigma</i> = 3	<i>sigma</i> = 5
MobileNetV2	66%	58%	52%	45%
EfficientNet-Lite0	66%	56%	46%	40%

Pada tabel 3 diatas *MobileNetV2* memiliki ketahanan yang lebih baik terhadap gangguan *gaussian blur* dibandingkan *EfficientNet-Lite0*. Meskipun kedua model memiliki performa awal dengan akurasi yang sama yaitu 66%, *MobileNetV2* mampu mempertahankan akurasi lebih tinggi pada seluruh tingkat *gaussian blur* berikutnya. Hal ini menunjukkan bahwa *MobileNetV2* lebih adaptif terhadap degradasi detail visual, sedangkan *EfficientNet-Lite0* lebih optimal untuk citra beresolusi dan ketajaman tinggi.



Gambar 7. Grafik Akurasi-Loss pada model *MobileNetV2* dengan Nilai *Sigma* 5

Gambar 7 di atas adalah grafik *training* dengan model *MobileNetV2* pada data uji *gaussian blur* dengan nilai *sigma* 5. Akurasi *training* dan *validation* meningkat secara bertahap sekitar *epoch* ke-40, kemudian mulai stabil di kisaran 0.45-0.48. Pada tahap awal *validation accuracy* naik lebih cepat dibanding *training accuracy*, menunjukkan model

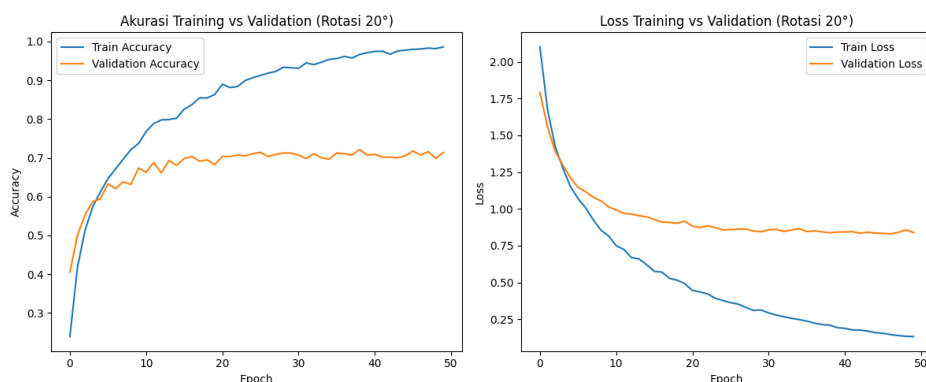
cukup cepat beradaptasi pada data validasi awal. Setelah sekitar *epoch* ke-30, kedua kurva semakin mendekat, menandakan bahwa model tidak mengalami *overfitting* berat, namun tetap belum mencapai generalisasi yang tinggi.

4.1.4. Performa Data Uji Rotation

Tabel 4. Hasil Akurasi Perbandingan Model pada Data Uji Rotasi

Model	Nilai Rotasi			
	20°	70°	190°	300°
MobileNetV2	71%	64%	64%	65%
EfficientNet-Lite0	80%	72%	74%	74%

Tabel 4 diatas menunjukkan bahwa *EfficientNet-Lite* memiliki ketahanan lebih baik terhadap perubahan orientasi dibandingkan *MobileNetV2*. Meskipun kedua model mengalami penurunan akurasi pada rotasi besar, *EfficientNet-Lite0* tetap stabil di atas 70%, sedangkan *MobileNetV2* turun hingga 64%. Hal tersebut menandakan bahwa arsitektur *EfficientNet-Lite0* lebih mampu menangani perubahan orientasi objek dibandingkan *MobileNetV2* yang kurang tahan terhadap variasi orientasi gambar.



Gambar 8. Grafik Akurasi-Loss pada Model *EfficientNet-Lite0* dengan Rotasi Tetap 20°

Gambar 8 di atas menyajikan grafik akurasi-loss pada model *EfficientNet-Lite0* dengan rotasi tetap 20 derajat. Model mengalami *overfitting*, menjadi semakin pintar pada data latih, tetapi tidak bertambah pintar pada data validasi. Hal tersebut ditandai dengan adanya kesenjangan garis biru dan oranye yang semakin melebar setelah *epoch* ke-15. Performa generalisasi optimal model dicapai di sekitar *epoch* 15-20 dan pelatihan lebih lanjut setelah titik tersebut memberikan peningkatan performa pada data validasi, melainkan hanya memperburuk *overfitting*.

4.2. Pembahasan

Untuk menjawab dan memberikan solusi pada masalah penelitian, maka hasil penelitian yang sudah disajikan dievaluasi pada pembahasan berikut ini.

4.2.1. Dampak Gangguan Visual pada Model

Dalam pengujian *brightness*, terdapat perbedaan yang terlihat. *MobileNetV2* terbukti sangat sensitif terhadap cahaya. Performanya menurun drastis setelah kondisi

citra yang gelap menghasilkan akurasi 56% dan baru stabil saat kondisi terang. Sebaliknya, *EfficientNet-Lite0* sangat tangguh, mampu mempertahankan akurasi yang stabil di persentase 71%-74% baik dalam kondisi kurang cahaya maupun pencahayaan berlebih.

Pada pengujian gangguan *blur*, kedua model sama-sama mengalami kesulitan fatal. Kinerja keduanya menurun seiring hilangnya detail citra, membuktikan bahwa ketajaman tekstur adalah kunci utama bagi kedua model untuk mengenali jamur. Meskipun *MobileNetV2* sedikit lebih *robust* pada kondisi kabur ekstrem dengan hasil akurasi 45% dibandingkan *EfficientNet-Lite0* yang menghasilkan akurasi 40%. Keduanya tetap mengalami penurunan performa yang parah.

Terakhir, pengujian *rotation* memberikan tantangan berat yang menyebabkan kedua model mengalami *overfitting*. Meskipun hasil ini menunjukkan keterbatasan penelitian, *EfficientNet-Lite0* terbukti lebih unggul secara fundamental karena sempat mencapai akurasi 80% pada rotasi tetap 20° sebelum mengalami *overfitting*, menunjukkan kemampuan belajar yang lebih cepat dibandingkan *MobileNetV2*.

4.2.2. Perbandingan Ketahanan Model

Secara keseluruhan, model *EfficientNet-Lite0* terbukti menjadi model yang secara signifikan lebih tahan terhadap gangguan gambar yang diberikan dibandingkan model *MobileNetV2*, terutama dalam menghadapi variasi pencahayaan yang memberikan hasil dengan sangat stabil dan variasi rotasi yang mana dapat mengungguli secara signifikan. Namun, kedua model memiliki kelemahan fundamental dalam menangani hilangnya detail gambar akibat *gaussian blur*, menunjukkan ketergantungan yang tinggi pada fitur tekstural yang tajam.

5. KESIMPULAN

Penelitian ini menegaskan bahwa *EfficientNet-Lite0* merupakan arsitektur yang lebih andal untuk tugas klasifikasi jamur di lapangan yang tidak ideal. Model tersebut mampu memberikan akurasi 73% pada tahap pengujian data yang bersih dan bisa ditingkatkan lagi melalui tahap *augmentasi* data. Kemudian model ini juga memiliki ketahanan pada pengujian data *brightness*. Ditandai dengan stabilitas performa yang tinggi, terutama dalam mempertahankan akurasinya pada kondisi pencahayaan minim. Kedua arsitektur terdegradasi secara drastis seiring meningkatnya intensitas blur. Temuan ini mengindikasikan bahwa tekstur halus dan ketajaman tepi merupakan komponen krusial yang dipelajari kedua model untuk melakukan klasifikasi. Meskipun keduanya mengalami *overfitting* pada pengujian data rotasi tetap yang mana menjadi keterbatasan pada penelitian ini, namun *EfficientNet-Lite0* mampu mencapai akurasi validasi yang lebih tinggi sebelum mengalami stagnasi.

Arah penelitian berikutnya mengimplementasikan model *EfficientNet-Lite0* ke perangkat edge ataupun *mobile devices*. Sehingga harapannya model tersebut mampu dipakai secara langsung di lapangan untuk melakukan tugas klasifikasi jenis jamur.

DAFTAR PUSTAKA

- [1] P. Kecerdasan Buatan Dalam Mendukung Pembelajaran dan B. Karyadi, "PEMANFAATAN KECERDASAN BUATAN DALAM MENDUKUNG PEMBELAJARAN MANDIRI," vol. 8, no. 2, hlm. 253–258, doi: 10.32832/educate.v8i02.14843.
- [2] S. ORAL, İ. ÖKTEN, dan U. YÜZGEÇ, "Fungus Classification Based on CNN Deep Learning Model," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 12, no. 1, hlm. 226–241, Mar 2023, doi: 10.17798/bitlisfen.1225375.
- [3] A. Faghihi, M. Fathollahi, dan R. Rajabi, "Diagnosis of Skin Cancer Using VGG16 and VGG19 Based Transfer Learning Models," Apr 2024, [Daring]. Tersedia pada: <http://arxiv.org/abs/2404.01160>
- [4] M. A. Hakiki, N. Rachmat, dan K. Kunci-Cnn, "Klasifikasi Spesies Jamur Menggunakan Convolutional Neural Network dengan Arsitektur MobileNetV2," *Jurnal Algoritme*, vol. 5, no. 3, hlm. 274–287, 2025, doi: 10.35957/algoritme.v6i1.11077.
- [5] H. Zhang, H. Singh, M. Ghassemi, dan S. Joshi, "'Why did the Model Fail?': Attributing Model Performance Changes to Distribution Shifts." [Daring]. Tersedia pada: https://github.com/MLforHealth/expl_
- [6] O. O. Martins, C. C. Oosthuizen, dan D. A. Desai, "Evaluating unified training optimisations for MobileNetV2: Efficiency-accuracy trade-offs in fine-grained dog breed classification," *Discover Applied Sciences*, vol. 7, no. 11, hlm. 1240, Okt 2025, doi: 10.1007/s42452-025-07798-1.
- [7] W. Fabre, K. Haroun, V. Lorrain, M. Lepecq, dan G. Sicard, "From Near-Sensor to In-Sensor: A State-of-the-Art Review of Embedded AI Vision Systems," 1 Agustus 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s24165446.
- [8] Y. Yohannes, D. Udjulawa, dan T. Ivan Sariyo, "Klasifikasi Jenis Jamur Menggunakan SVM dengan Fitur HSV dan HOG," *PETIR*, vol. 15, no. 1, hlm. 113–120, Des 2021, doi: 10.33322/petir.v15i1.1101.
- [9] "IMPLEMENTASI METODE CNN DALAM KLASIFIKASI GAMBAR JAMUR PADA ANALISIS IMAGE PROCESSING."
- [10] U. Sri Rahmadhani dan N. Lysbetti Marpaung, "Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN," vol. 8, no. 2, 2023.
- [11] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, dan B. Lakshminarayanan, "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty," Feb 2020, [Daring]. Tersedia pada: <http://arxiv.org/abs/1912.02781>
- [12] D. Hendrycks dkk., "The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization," dalam *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 2021, hlm. 8320–8329. doi: 10.1109/ICCV48922.2021.00823.