Terakreditasi Peringkat 4
Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi, No. 225/E/KPT/2022
Masa berlaku dari Volume 8 Nomor 1 Tahun 2022 sampai dengan Volume 12 Nomor 2 Tahun 2026

Published online at: **https://ejournal.raharja.ac.id/**

JOURNAL CERITA:
Creative Education of Research in Information Technology and Artificial informatics

Vol. 12 No. 2 (2026) 312-327 e-ISSN: 2655 - 2574

Perbandingan MobileNet V2, EfficientNet Lite, Dan DenseNet-169 Dalam Pengenalan Jamur Berbasis Android

Mahardika Virgo Wuryantoro[*1], Achmad Junaidi[2], Hendra Maulana[3]
Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia
Email: **21081010077@student.upnjatim.com[*1]**; achmadjunaidi.if@upnjatim.ac.id[2]; hendra.maulana.if@upnjatim.ac.id[3]

Wuryantoto, M. V., Junaidi, A., & Maulana, H. (2026). Perbandingan MobileNet V2, EfficientNet Lite, Dan DenseNet-169 Dalam Pengenalan Jamur Berbasis Android. *Journal Cerita: Creative Education of Research in Information Technology and Artificial Informatics*, 12(2), 312-327

DOI: https://doi.org/10.33050/cerita.v12i2.3913

ABSTRAK

Kasus keracunan jamur sering terjadi akibat salah identifikasi antara jamur beracun dan jamur yang aman untuk dikonsumsi. Teknologi pengenalan citra berbasis *Convolutional Neural Network* atau CNN merupakan salah satu solusi yang dapat digunakan untuk mengatasi permasalahan ini. Namun, penelitian sebelumnya umumnya masih mengandalkan perangkat dengan spesifikasi tinggi atau memerlukan koneksi internet dalam implementasinya, sementara implementasi praktis pada perangkat seluler atau ponsel belum banyak dibahas. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi Android praktis yang mampu mengidentifikasi jamur beracun, serta membandingkan performa tiga arsitektur CNN ringan, yaitu MobileNet V2, DenseNet-169, dan EfficientNet. Hasil penelitian menunjukkan bahwa ketiga model memiliki akurasi tinggi dalam mengidentifikasi jamur, tetapi menunjukkan perbedaan signifikan dalam efisiensi penggunaan sumber daya perangkat serta ketahanan dalam berbagai kondisi pengambilan gambar. Penelitian ini diharapkan menjadi solusi praktis untuk pengenalan jamur bagi masyarakat sekaligus menjadi acuan bagi pengembangan aplikasi serupa di masa depan.

Kata kunci: deteksi jamur beracun, pengenalan citra, jaringan syaraf tiruan, aplikasi android, Convolutional Neural Network

Received: 20-02-2026 | Accepted: 20-06-2026 | Published Online: 08-08-2026

ABSTRACT

Mushroom poisoning cases often occur due to the misidentification of poisonous and edible mushroom. Image recognition technology based on Convolutional Neural Networks or CNN is one of the solutions that can be used to address this problem. However, previous studies generally rely on high-end devices or require internet connection in its implementation, while practical deployment on mobile devices has received little attention. This research aims to develop a practical Android application capable of identifying poisonous mushrooms, as well as to compare the performance of three lightweight CNN architecture, namely MobileNet V2, DenseNet-169, and EfficientNet Lite. The results indicate that all three models have high accuracy in identifying various mushroom species, but exhibit significant differences in resource usage efficiency and robustness under varying image capture conditions. This research is expected to provide a practical solution for mushroom recognition for the general public and serve as reference for the development of similar application in the future.

Keywords: *Poisonous mushroom detection, image recognition, neural networks, android application, Convolutional Neural Network*

I. PENDAHULUAN

"Jamur" dalam penyebutan sehari-hari merupakan badan buah dari beberapa genus dalam kerajaan fungi. Jamur umum ditemukan pada berbagai macam habitat serta sering kali menjadi bahan utama dalam menu masakan masyarakat. Misalnya, *Termitomyces microcarpus* merupakan salah satu spesies jamur liar yang bisa dimakan dan umum dikenal di sebagian besar wilayah Indonesia sebagai jamur rayap atau jamur cepaki (Sibero dkk., 2021).

Namun, tidak semua jenis jamur dapat dikonsumsi dengan aman sebab diperkirakan bahwa dari 100.000 jenis jamur yang telah dikenali, sekitar 100 di antaranya dapat menyebabkan gejala keracunan bagi manusia (Graeme, 2014). Kendati demikian, pengetahuan mengenai jamur beracun masih terbatas di kalangan ahli biologi dan mikologi, sehingga masyarakat umum sering kali salah dalam membedakan jamur yang aman dikonsumsi dan yang beracun. Sebagai contoh, Putra (2022), di Indonesia sendiri hampir tidak ada pusat informasi khusus untuk menampung informasi mengenai spesies-spesies jamur beracun yang umum ditemui, pemanfaatan jamur, serta kasus-kasus keracunan jamur.

Kesalahan identifikasi jamur sering kali menjadi penyebab kasus keracunan jamur. Salah satu contoh kesalahan identifikasi jamur yang paling terkenal adalah *Amanita phalloides*, yang merupakan salah satu jamur paling mematikan bagi manusia, dengan *Amanita caesarea* (jamur Caesar) dan *Volvariella volvacea* (jamur merang) yang aman untuk dikonsumsi. Dalam sebuah laporan kasus oleh Putra dan Hermawan (2021), seorang pemuda di Gresik, Indonesia mengalami keracunan sebab memakan jamur liar yang salah diidentifikasi sebagai jamur rayap (*Termitomyces microcarpus*) yang telah disebutkan sebelumnya. Korban disebutkan mengalami gejala antara lain mual, muntah, serta pusing. Laporan lain menyebutkan bahwa di Perancis dalam kurun waktu 2023-2024 terjadi kurang lebih 1.800 kasus keracunan jamur, di mana 28 di antaranya berujung pada gejala serius (Lyall, 2024). Sementara itu, lembaga kesehatan masyarakat Amerika Serikat, CDC (*Centers for Disease Control and Prevention)*, melaporkan bahwa dalam kurun waktu 2016-2018 terdapat rata-rata 1.328 kasus keracunan jamur setiap tahunnya. Sekitar 8,6% dari pasien-pasien tersebut mengalami gejala serius (Gold dkk., 2021). Di Indonesia sendiri, selama kurun waktu 2010-2020, dilaporkan 85 kasus keracunan jamur, yang berakibat pada 550 korban dan 9 korban jiwa (Amelya dkk., 2023).

Mengingat risiko tersebut, diperlukan solusi identifikasi yang cepat dan akurat. Salah satu pendekatan yang menjanjikan adalah penggunaan teknologi *deep learning*, khususnya arsitektur *Convolutional Neural Network* (CNN). CNN merupakan salah satu teknologi populer dalam hal pengenalan citra, mampu mengenali berbagai macam benda dengan kecepatan dan akurasi tinggi. Selain itu, keunggulan CNN juga terletak pada kemampuannya untuk mengekstraksi fitur pada gambar dan mempelajarinya secara otonom (Zhao dkk., 2024).

Teknologi ini telah diterapkan dalam berbagai macam kasus, seperti pada penelitian oleh Therry dkk. (2024), yang menggunakan CNN dan MediaPipe untuk menerjemahkan Bahasa Isyarat Indonesia, menunjukkan fleksibilitas metode ini dalam kehidupan sehari-

hari. Di bidang biologi, CNN juga dapat dimanfaatkan untuk mendiagnosis penyakit tanaman, sebagaimana ditunjukkan dalam penelitian oleh Nisa' dkk. (2020) yang menggunakan arsitektur InceptionV3 untuk klasifikasi penyakit daun apel. Secara lebih spesifik, penelitian oleh Mauludy dkk. (2024) dan Zahan dkk. (2021) menunjukkan bahwa CNN memang efektif dalam mengidentifikasi jamur beracun, ditunjukkan dengan tingginya tingkat akurasi identifikasi.

Namun begitu, penelitian-penelitian tersebut masih terbatas pada sarana implementasinya, yang sering kali masih membutuhkan perangkat dengan spesifikasi yang memadai atau setidaknya koneksi internet. Pertanyaannya, mungkinkah teknologi CNN ini diimplementasikan sedemikian rupa sehingga siap digunakan sewaktu-waktu di lapangan? Bagaimana potensi implementasi CNN pada perangkat yang bisa di bawa dan di gunakan secara langsung di lapangan, misalnya perangkat Android?

Beberapa penelitian menyebutkan bahwa terdapat model-model yang cukup ringan sehingga efektif untuk penerapan pada perangkat *mobile*. Penelitian oleh Haksoro dan Setiawan (2021) dan DemiRel dan DemiRel (2023) misalnya, menyebutkan mengenai model MobileNet, terutama versi dua, yang memiliki arsitektur yang terbilang ringan dan andal dalam mengidentifikasi jamur.

Penelitian lain oleh Mauludy dkk. (2024) menyebutkan bahwa EfficientNet merupakan pilihan yang baik dalam hal identifikasi jamur serta memiliki arsitektur yang ringan, meskipun belum disebutkan implementasinya pada perangkat *mobile*. Hampir serupa dengan penelitian tersebut, Kousis dkk. (2022) berhasil mengimplementasikan model EfficientNet Lite pada perangkat *mobile*, meskipun pada kasus pengenalan kanker kulit.

Alternatif lain ditunjukkan oleh penelitian yang dilakukan oleh Yulita dkk. (2023). Penelitian tersebut mengembangkan aplikasi berbasis Android yang menggunakan arsitektur DenseNet untuk mendeteksi penyakit pada daun tomat.

Beberapa penelitian terdahulu menyebutkan potensi penerapan CNN pada perangkat *mobile*, khususnya Android, dengan efisien serta memberikan hasil yang memuaskan. Model-model yang menjadi perhatian antara lain MobileNet, DenseNet, dan EfficientNet. Berangkat dari informasi tersebut, penulis bermaksud untuk mengembangkan sebuah aplikasi Android yang mampu melakukan tugas identifikasi jamur di lapangan. Selain itu, penelitian ini juga bertujuan untuk membandingkan ketiga model tersebut dalam hal performa identifikasi serta efisiensi penggunaan sumber daya perangkat. Harapannya, penelitian ini dapat menjadi referensi implementasi model CNN pada perangkat *mobile,* khususnya Android, terutama dalam hal identifikasi jamur.

II. METODE PENELITIAN

Penelitian ini berjenis *comparative research* atau penelitian komparatif yang bertujuan untuk menguji dan membandingkan kinerja tiga model CNN yaitu MobileNet V2, DenseNet-169, dan EfficientNet Lite pada tugas pengenalan jamur di perangkat Android. Pendekatan yang digunakan adalah pendekatan kuantitatif, di mana ketiga model akan dibandingkan berdasarkan metrik-metrik performa seperti akurasi, *precision, recall, F1-score*, serta penggunaan sumber daya perangkat. Penelitian ini akan dilakukan dalam beberapa tahap, antara lain pengumpulan data, *preprocessing* dan augmentasi data, pelatihan model, pengembangan aplikasi Android, serta uji coba. Gambar 1 memuat diagram yang menjelaskan langkah-langkah dalam penelitian ini.

A. Pengumpulan Data

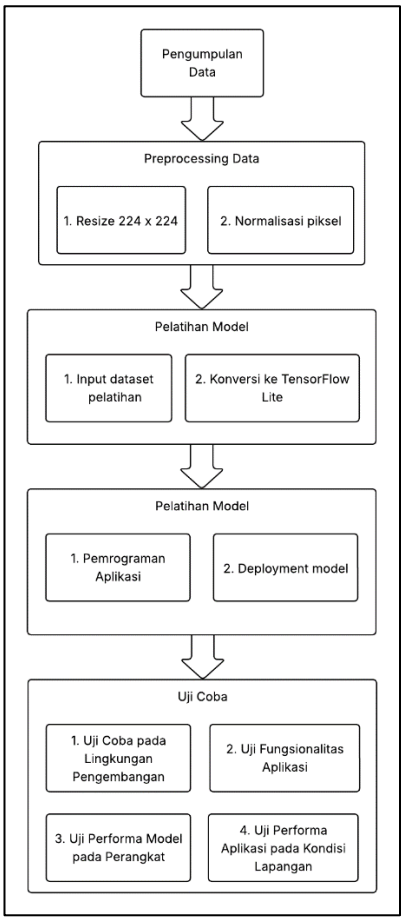
Spesies jamur yang akan digunakan dalam penelitian ini terdiri dari 12 spesies jamur, dengan enam spesies di antaranya beracun dan enam spesies tidak beracun. Daftar spesies jamur ini dapat dilihat pada Tabel 1. Sementara itu, contoh gambar jamur yang digunakan dapat dilihat pada Gambar 2.

Dataset jamur-jamur tersebut dikumpulkan dari berbagai sumber publik yang tersedia pada platform seperti Kaggle, GBIF (*Global Biodiversity Information Facility*), dan iNaturalist. Platform-platform ini dipilih karena menyediakan kelengkapan dan jumlah data yang cukup untuk melatih sebuah model CNN. Selain itu, data-data yang disediakan oleh GBIF dan iNaturalist telah diverifikasi secara urun daya oleh komunitas. Sebanyak 12.283 gambar jamur berhasil dikumpulkan yang kemudian dibagi sesuai dengan prinsip Paretto dimana rasio

dataset pelatihan dan dataset pengujian yaitu 80:20.

B. Preprocessing dan Augmentasi Data

Tahap ini dilakukan sehingga citra yang akan diproses oleh model sesuai dengan kriteria masukan model. Ketiga model membutuhkan citra masukan berukuran $224 \times 224 \times 3$ piksel. Ukuran ini berarti citra memiliki tinggi dan lebar sebesar 224 piksel serta memiliki tiga saluran warna yaitu merah, hijau, dan biru atau berformat RGB. Nilai piksel dalam citra kemudian dinormalisasi sehingga berada pada rentang 0 hingga 1.



Gambar 1. Diagram Tahapan Penelitian
Sumber: diolah dari data primer

Selain *preprocessing*, dilakukan juga augmentasi data yang bertujuan untuk memperkaya dataset serta membantu mengurangi *overfitting*. *Overfitting* sendiri merupakan suatu kondisi di mana model terlalu menghafal data pelatihan sehingga menyebabkan model tidak dapat menggeneralisasi data baru dan memiliki kinerja yang buruk. Kondisi ini umumnya disebabkan oleh dataset pelatihan yang terlalu homogen. Dalam proses augmentasi data, citra dimodifikasi sedemikian rupa sehingga menghasilkan citra baru dengan variasi kondisi. Teknik-teknik yang dilakukan dalam proses ini antara lain:

- Rotasi: memutar citra sebanyak derajat tertentu
- *Flip*: membalik citra secara horizontal maupun vertical
- *Zoom*: memperbesar atau memperkecil citra
- *Translation* atau pergeseran: menggeser citra ke arah horizontal atau vertical

Tabel 1. Daftar Jamur Beracun dan Tidak Beracun

Beracun	Tidak Beracun
Amanita phalloides	*Boletus edulis*
Amanita muscaria	*Lentinula edodes*
Galerina marginata	*Auricularia auricula*
Chlorophyllum molybdites	*Pleurotus ostreatus*
Psilocybe cubensis	*Termitomyces microcarpus*
Cortinarius rubellus	*Agaricus bisporus*

Sumber: diolah dari data sekunder

Gambar 2. Contoh Gambar Jamur

C. Pelatihan Model

Ketiga model yang dilatih merupakan model pra-latih yang dibangun menggunakan bobot awal dari pelatihan dengan dataset ImageNet. Dari model-model pra-latih ini, beberapa lapisan awal dibekukan sehingga model dapat memanfaatkan informasi yang telah dipelajari untuk mengenali pola-pola umum. Lapisan-lapisan sisanya kemudian dilatih ulang menggunakan dataset jamur sehingga dapat lebih mengenali fitur-fitur yang lebih khas dan spesifik dari dataset.

Model dilatih dengan learning rate 0.0001 selama maksimum 60 *epoch* dengan *early stopping*, *batch size* 32, dan optimizer Adam. Fungsi *loss* yang digunakan adalah *categorical cross-entropy* dengan metrik evaluasi akurasi. Pengunaan *early stopping* memungkinkan pelatihan model untuk berhenti secara otomatis apabila metrik pelatihan tidak menunjukkan perkembangan selama beberapa *epoch*. Untuk mengatasi ketidakseimbangan jumlah citra pada tiap-tiap kelas dalam dataset, digunakan strategi pembobotan kelas. Metode ini digunakan untuk mencegah model terlalu menghafal kelas dengan jumlah citra yang lebih banyak. Berdasarkan jumlah citra yang dimiliki, suatu kelas akan diberikan bobot yang besar apabila jumlah citranya sedikit dan bobot yang lebih kecil apabila jumlah citranya banyak. Dengan demikian, pengaruh tiap kelas pada model akan seimbang.

Setelah model selesai dilatih, model kemudian dikonversi ke format TensorFlow Lite sehingga dapat diimplementasikan pada perangkat Android. Dalam proses ini, model berbasis Keras, yakni MobileNet V2 dan DenseNet-169, dapat langsung dikonversi. Sementara itu, model EfficientNet Lite yang berbasis PyTorch harus dikonversi ke format ONNX (*Open Neural Network Exchange*) terlebih dahulu sebelum dapat dikonversi ke format TensorFlow Lite.

D. Pengembangan Aplikasi

Aplikasi Android untuk pengenalan jamur ini dikembangkan dengan mempertimbangkan beberapa *use case* berikut:

1. Mengambil gambar jamur menggunakan kamera perangkat
2. Mengambil gambar jamur dari galeri
3. Menampilkan informasi prediksi spesies jamur beserta informasi racun dalam jamur tersebut
4. Menyimpan hasil klasifikasi
5. Melihat kembali hasil klasifikasi yang disimpan
6. Menghapus hasil klasifikasi
7. Menampilkan daftar jamur yang dapat dikenali oleh aplikasi

Aplikasi ini dikembangkan menggunakan bahasa pemrograman Kotlin. Untuk membantu implementasi model pada aplikasi, digunakan pustaka TensorFlow Lite dari Google.

Tabel 2. Skenario Pengujian Kondisi Lapangan

No.	Kategori Pengujian	Skenario Pengujian
1.	Pengambilan gambar dengan berbagai jenis pencahayaan	Pencahayaan gelap
		Pencahayaan sangat terang
2.	Pengambilan gambar dengan sudut beragam	Sudut atas
		Sudut samping
		Sudut bawah
3.	Jarak pengambilan gambar yang berbeda	± 10 cm
		± 30 cm
		± 50 cm
4.	Latar belakang jamur yang beragam	Latar tanah
		Latar daun-daunan
		Latar rumput
5.	Jamur tertutup sebagian oleh objek lain	Tertutup daun
		Tertutup ranting

Sumber: diolah dari data primer

E. Uji Coba

Tahap uji coba dilakukan untuk menilai performa model-model yang telah dilatih serta menguji fungsionalitas aplikasi yang telah dikembangkan. Beberapa pengujian yang dilakukan antara lain uji coba pada lingkungan pengembangan, uji coba fungsionalitas aplikasi, uji coba performa model pada perangkat Android, serta uji coba kinerja aplikasi dalam kondisi dunia nyata.

Pengujian di lingkungan pengembangan dilakukan pada platform pengembangan yang digunakan, dalam hal ini Google Colab, untuk mendapatkan gambaran awal mengenai performa model sebelum diimplementasikan pada perangkat Android. Berbeda dengan tahap pelatihan, uji coba ini akan menggunakan dataset pengujian yang belum pernah dilihat oleh model sebelumnya. Hasil dari tahap ini menjadi tolok ukur kemampuan dasar setiap arsitektur model. Beberapa metrik yang diuji meliputi akurasi, *precision*, *recall*, *F1-score*, *loss*, dan waktu inferensi.

Pengujian fungsionalitas aplikasi dilakukan untuk memastikan bahwa seluruh fitur yang telah dikembangkan berjalan sesuai dengan yang diharapkan. Pada dasarnya, uji coba ini menjadi tolok ukur sejauh mana rancangan *use case* yang telah disusun berhasil diimplementasikan.

Pengujian kemudian dilanjutkan untuk menilai performa model pada perangkat Android. Tahap ini bertujuan untuk mengukur performa model setelah diimplementasikan serta penggunaan sumber daya perangkat oleh model. Beberapa metrik yang akan diuji antara lain akurasi klasifikasi, penggunaan memori, penggunaan ruang penyimpanan, serta lama waktu inferensi. Hasil dari pengujian ini digunakan untuk membandingkan efisiensi tiap model dalam menggunakan sumber daya perangkat.

Terakhir, pengujian performa model pada kondisi lapangan dilakukan untuk menilai keandalan model dalam mengidentifikasi jamur dengan berbagai variasi kondisi pengambilan gambar. Uji coba ini dilakukan dengan beberapa skenario seperti yang diuraikan pada Tabel 2. Untuk setiap skenario, model akan diuji dengan empat spesies jamur, yaitu *Agaricus bisporus*, *Auricularia auricula*, *Lentinula edodes*, dan *Pleurotus ostreatus*. Keempat spesies jamur ini dipilih karena mudah didapatkan dan tidak beracun sehingga tidak berbahaya untuk

ditangani. Model akan diberi empat gambar terpisah untuk setiap skenario, sehingga pengujian memiliki variasi visual yang cukup untuk mengevaluasi respons model. Jumlah ini dipilih dengan mempertimbangkan efisiensi waktu dan sumber daya. Jumlah prediksi benar kemudian dihitung untuk menentukan akurasi model. Hasil dari pengujian ini digunakan untuk menilai apakah model dapat tetap akurat ketika dihadapkan dengan situasi penggunaan sehari-hari yang mungkin ditemui pengguna. Selain itu, hasil pengujian ini juga digunakan sebagai pembanding antara model-model yang telah dilatih apakah tetap dapat stabil dengan berbagai kondisi pengambilan gambar.

F. Metrik Pengujian Performa Model

Penghitungan metrik-metrik ini sangat erat kaitannya dengan konsep *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat jenis data ini dihitung secara terpisah untuk setiap kelas dengan pendekatan *one-vs-all*. *True Positive* mengacu pada jumlah data dari suatu kelas yang berhasil diprediksi benar sebagai kelas tersebut. *False Positive* adalah jumlah data dari kelas lain yang salah diprediksi sebagai kelas tersebut. Sebaliknya, *False Negative* menunjukkan jumlah data dari kelas tersebut yang salah diprediksi sebagai kelas lain. Sementara itu, *True Negative* adalah jumlah data dari kelas lain yang berhasil diprediksi bukan sebagai kelas tersebut.

Berdasarkan nilai TP, FP, FN, dan TN, metrik-metrik pengujian dihitung untuk mengukur performa model. Akurasi merupakan rasio dari jumlah prediksi benar terhadap seluruh data uji dan dihitung menggunakan rumus:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision mengukur seberapa akurat prediksi positif model terhadap kenyataan. Artinya, *precision* menunjukkan proporsi data yang diprediksi sebagai suatu kelas dan memang benar termasuk dalam kelas tersebut. *Precision* dihitung menggunakan rumus:

$$Precision = \frac{TP}{TP + FP}$$

Precision dihitung untuk masing-masing kelas dengan cara memperlakukan kelas tersebut sebagai "positif" dan sisanya sebagai "negatif".

Recall mengukur kemampuan model dalam menemukan seluruh data yang memang termasuk dalam suatu kelas. Sederhananya, metrik ini menghitung seberapa banyak data dari suatu kelas yang berhasil dikenali oleh model. *Recall* dihitung dengan rumus:

$$Recall = \frac{TP}{TP + FN}$$

F1-score menghitung rata-rata harmonis dari *precision* dan *recall* sebagai nilai tunggal. F1-score seimbang dalam mempertimbangkan kesalahan *false positive* dan *false negative* dan dihitung dengan rumus:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

III. HASIL DAN PEMBAHASAN

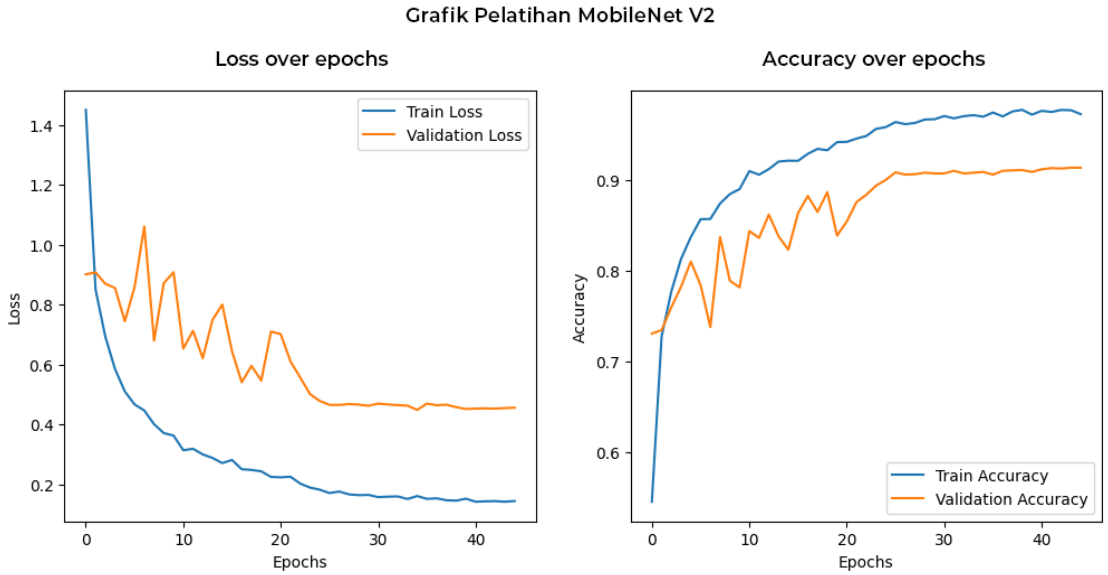
A. Hasil Pelatihan Model

Hasil pelatihan ketiga model dievaluasi berdasarkan metrik *loss* dan akurasi terhadap dataset pelatihan dan validasi seperti yang dapat dilihat pada

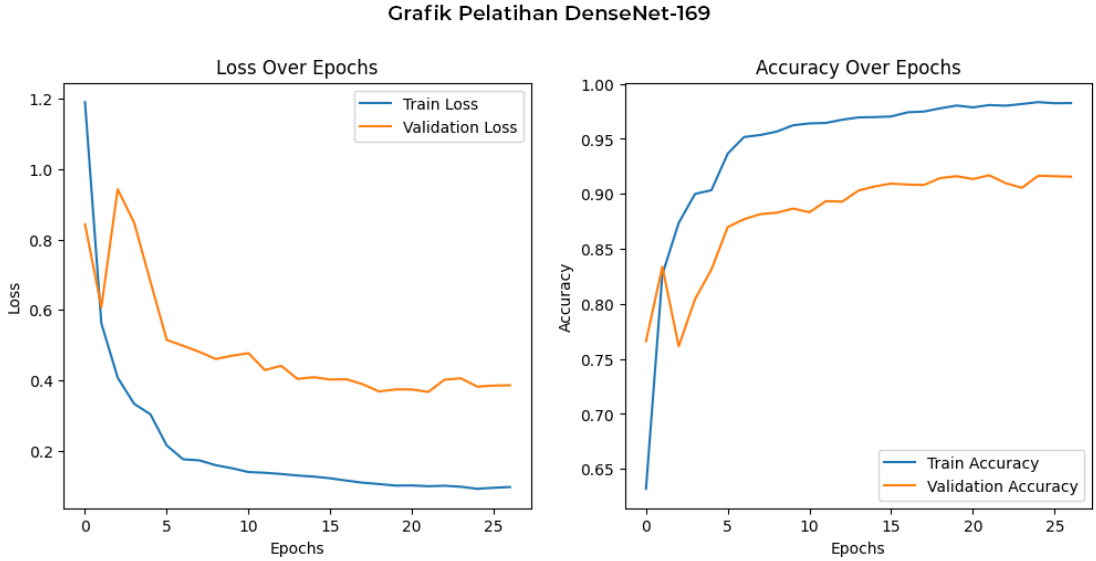
Tabel 3.

Pelatihan model MobileNet V2 ditunjukkan oleh grafik pada Gambar 3. Model ini menunjukkan peningkatan performa selama 45 *epoch*, dengan penurunan *train loss* dari 1.49 menjadi 0.14 dan *validation loss* dari 0.9 menjadi 0.45 sebagaimana yang ditunjukkan oleh grafik sebelah kiri. Grafik sebelah kanan menunjukkan peningkatan akurasi pelatihan dari 38% ke 97%, sementara akurasi validasi meningkat dari 73% ke 91%.

Pelatihan model DenseNet-169 menunjukkan hasil yang lebih baik dengan waktu pelatihan yang lebih singkat sebanyak 27 *epoch* sebagaimana yang dapat dilihat dalam grafik pada Gambar 4. Pada grafik sebelah kiri, *train loss* turun dari 1.67 ke 0.09 dan *validation loss* turun dari 0.84 ke 0.38. Sementara itu, pada grafik sebelah kanan, akurasi pelatihan meningkat dari 47% ke 98% dan akurasi validasi meningkat dari 76% ke 91%.



Gambar 3. Grafik Pelatihan MobileNet V2



Gambar 4. Grafik Pelatihan DenseNet-169

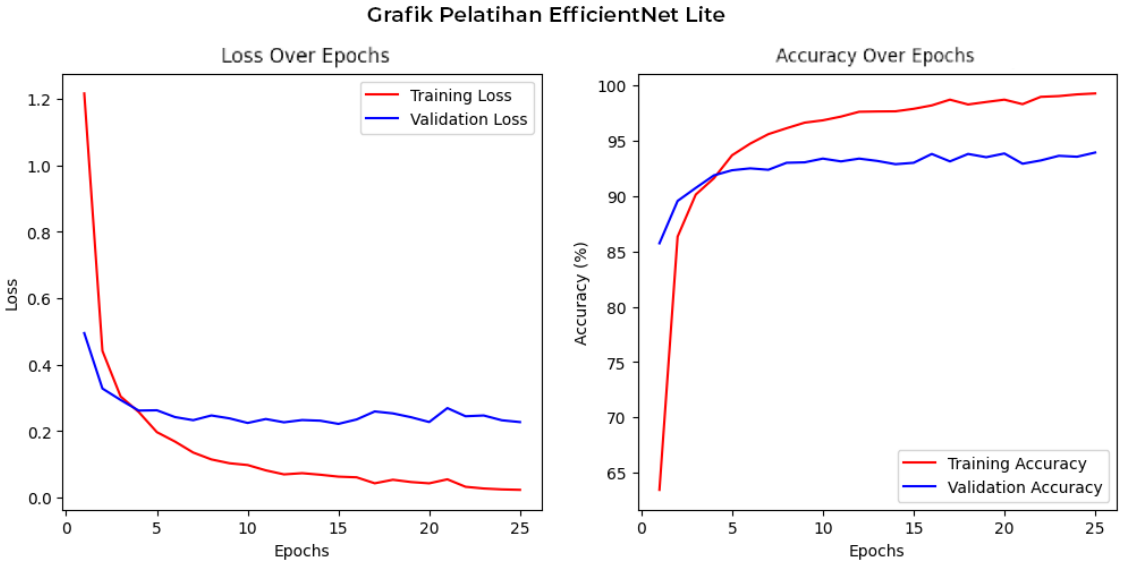
Pelatihan model EfficientNet Lite merupakan yang tercepat, hanya membutuhkan waktu 25 *epoch*. Grafik pelatihan dari model ini dapat dilihat pada Gambar 5. *Train loss* menurun dari 1.21 ke 0.02 dan *validation loss* dari 0.49 ke 0.22, sebagaimana ditunjukkan oleh grafik sebelah kiri. Pada grafik sebelah kanan dapat dilihat bahwa model ini berhasil meningkatkan akurasi pelatihan dari 63% ke 99% dan akurasi validasi dari 85% ke 93%.

Hasil pengukuran *loss* dan akurasi ketiga model setelah pelatihan terhadap data pengujian dan validasi menunjukkan bahwa ketiga model memiliki kemampuan klasifikasi yang baik. Semua model berhasil mencapai akurasi validasi di atas 90%. Hasil pengukuran pada dataset validasi memang secara umum lebih rendah dari pengukuran pada dataset pelatihan. Hal ini memang wajar sebab model bekerja pada data yang belum pernah dilihat dan masih bisa diterima. Melihat dari hasil ini, EfficientNet Lite merupakan model dengan performa terbaik, berhasil mencapai konvergensi dengan jumlah *epoch* paling sedikit, nilai *loss* awal dan akhir yang paling kecil, serta akurasi awal dan akhir yang paling tinggi baik untuk dataset pengujian maupun dataset validasi.

Tabel 3. Hasil Pelatihan Model

Model	Jumlah *epoch*	*Train loss*		*Validation loss*		*Train accuracy*		*Validation accuracy*	
		Awal	Akhir	Awal	Akhir	Awal	Akhir	Awal	Akhir
MobileNet V2	45	1.49	0.14	0.9	0.45	38%	97%	73%	91%
DenseNet-169	27	1.67	0.09	0.84	0.38	47%	98%	76%	91%
EfficientNet Lite	25	1.21	0.02	0.49	0.22	63%	99%	85%	93%

Sumber: diolah dari data primer



Gambar 5. Grafik Pelatihan EfficientNet Lite

B. Pengembangan Aplikasi

Aplikasi klasifikasi jamur ini merupakan sebuah aplikasi Android yang dikembangkan menggunakan bahasa pemrograman Kotlin dan *framework* Jetpack Compose. Aplikasi ini mendukung pengambilan gambar menggunakan kamera atau dari galeri dan penyimpanan data secara lokal. Seluruh proses klasifikasi gambar jamur dilakukan secara lokal tanpa memerlukan koneksi internet.

Proses klasifikasi gambar jamur dapat dimulai setelah pengguna mengambil gambar sebuah jamur, baik melalui kamera atau galeri. Hasil klasifikasi menampilkan nama spesies jamur, tingkat keyakinan (*confidence score*) model terhadap prediksi tersebut, status keberacunan jamur, serta informasi singkat mengenai jamur yang diprediksi. Pengguna kemudian dapat menyimpan hasil klasifikasi tersebut untuk ditinjau kembali.

Model CNN yang telah dilatih sebelumnya dan dikonversi menjadi format TensorFlow Lite dengan ekstensi .tflite diimplementasikan menggunakan pustaka TensorFlow Lite atau yang juga dikenal sebagai LiteRT (LiteRuntime). Pustaka ini menyediakan semua utilitas yang diperlukan untuk memuat model dan menjalankan proses inferensi. Sebelum gambar diberikan kepada model untuk klasifikasi, dilakukan *preprocessing* terlebih dahulu untuk memastikan gambar sesuai dengan kriteria masukan model. Tahap ini mencakup mengubah ukuran gambar menjadi 224 × 224 serta menormalisasi nilai piksel gambar sehingga berada pada rentang 0-1.

Tabel 4. Hasil Pengujian Model

Model	*Precision*	*Recall*	*F1-score*	Akurasi	*Loss*
MobileNet V2	0.92	0.92	0.92	92%	0.4
DenseNet-169	0.94	0.93	0.93	93%	0.34
EfficientNet Lite	0.96	0.96	0.96	96%	0.19

Sumber: diolah dari data primer

Tabel 5. Hasil Pengujian Fungsionalitas Aplikasi

No.	**Fitur yang Diuji**	**Status Pengujian**
1.	Pengambilan gambar	Berhasil
2.	Pemilihan gambar	Berhasil
3.	Inferensi model dan tampilan hasil	Berhasil
4.	Navigasi antar halaman	Berhasil
5.	Menyimpan hasil klasifikasi	Berhasil
6.	Melihat hasil klasifikasi	Berhasil
7.	Hapus hasil klasifikasi	Berhasil
8.	Melihat daftar jamur	Berhasil

Sumber: diolah dari data primer

C. Uji Coba di Lingkungan Pengembangan

Hasil pengujian tahap ini dapat dilihat pada

Tabel 4. MobileNet V2 mencapai akurasi 92% dengan nilai *loss* 0.4. Rata-rata *precision*, *recall*, dan *F1-score* masing-masing sebesar 0.92. Hal ini menunjukkan keseimbangan antara kemampuan model dalam mengidentifikasi kelas positif secara tepat (*precision*) dan mendeteksi seluruh data positif yang ada (*recall*). Dengan nilai *F1-score* yang seimbang di angka 0.92, MobileNet V2 menunjukkan performa yang cukup stabil di berbagai kelas. Namun, berdasarkan *confusion matrix*, model ini lemah pada beberapa kelas, seperti *Lentinula edodes*, *Agaricus bisporus,* dan *Galerina marginata.*

Model DenseNet-169 menghasilkan akurasi sebesar 93% dengan nilai *loss* 0.34. Rata-rata *precision* tercatat 0.94, lebih tinggi dibanding MobileNet V2, menunjukkan bahwa model ini lebih akurat dalam memprediksi kelas-kelas target dengan sedikit kesalahan positif. *Recall* sebesar 0.93 menunjukkan kemampuan model mendeteksi data positif cukup baik, namun masih ada beberapa kasus kelas positif yang tidak terdeteksi dengan benar. *F1-score* yang mencapai 0.93 memperlihatkan keseimbangan performa secara umum. Meski demikian, *confusion matrix* menunjukkan bahwa DenseNet-169 lemah dalam membedakan kelas *Lentinula edodes* dan *Agaricus bisporus.* Kedua kelas tersebut seringkali salah diprediksi sebagai *Pleurotus ostreatus.* Namun begitu, performa untuk kelas-kelas lain sangat baik terutama pada kelas *Amanita muscaria* dan *Auricularia auricula.*

EfficientNet Lite menunjukkan performa terbaik dengan akurasi 96% dan nilai *loss* paling rendah, yaitu 0.19. Rata-rata *precision*, *recall,* dan *F1-score* tercatat sebesar 0.96. Nilai *precision* yang tinggi menunjukkan model sangat minim dalam melakukan kesalahan prediksi positif. Sementara itu, *recall* yang juga mencapai angka 0.96 mengindikasikan bahwa model berhasil mengenali sebagian besar data positif dengan baik di hampir semua kelas. *F1-score* yang tinggi menunjukkan performa yang konsisten dan andal. Model ini mampu mengenali sebagian besar kelas dengan sangat baik, meskipun terbilang lemah pada kelas-kelas dengan fitur yang mirip. Misalnya, *Lentinula edodes* seringkali salah diprediksi sebagai *Pleurotus ostreatus* dan *Auricularia auricula*, dan *Agaricus bisporus* banyak salah diprediksi sebagai *Psilocybe cubensis* dan *Chlorophyllum molybdites*.

D. Uji Fungsionalitas Aplikasi

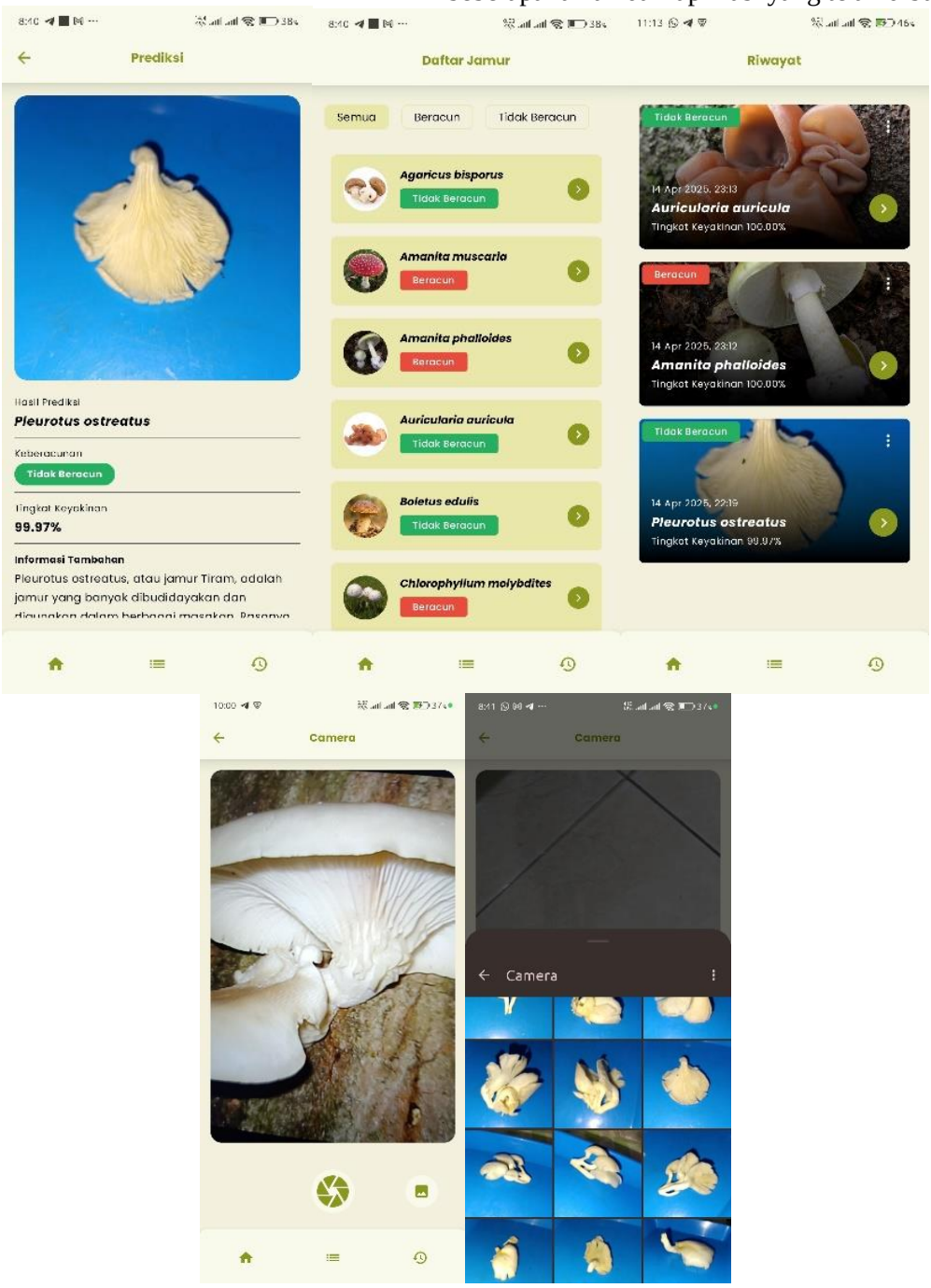
Tahap ini terdiri dari beberapa skenario dan berfungsi untuk menguji apakah semua rancangan *use case* berhasil diterapkan. Pengujian ini menggunakan pendekatan *black-box testing*, yaitu dengan menguji fungsi-fungsi aplikasi berdasarkan *input* dan *output* tanpa melihat struktur internal kode program. Tujuan dari pengujian ini adalah untuk memverifikasi bahwa seluruh proses, mulai dari pengambilan atau pemilihan gambar, hingga penyimpanan dan pengelolaan hasil

klasifikasi, dapat dilakukan secara normal oleh pengguna akhir. Daftar skenario dan hasil pengujiannya dapat dilihat pada

Tabel 5.

Hasil pengujian, sebagaimana ditunjukkan pada

Tabel 5, menunjukkan bahwa seluruh fitur berhasil dijalankan tanpa kendala atau malafungsi. Dengan demikian, dapat disimpulkan bahwa aplikasi telah memenuhi seluruh kebutuhan fungsional dan telah bekerja dengan baik dalam lingkungan operasionalnya. Keberhasilan semua skenario pengujian menunjukkan bahwa aplikasi ini telah siap digunakan oleh pengguna untuk melakukan klasifikasi jamur secara langsung melalui perangkat Android. Gambar 6 menunjukkan beberapa laman dari aplikasi yang telah dibuat.



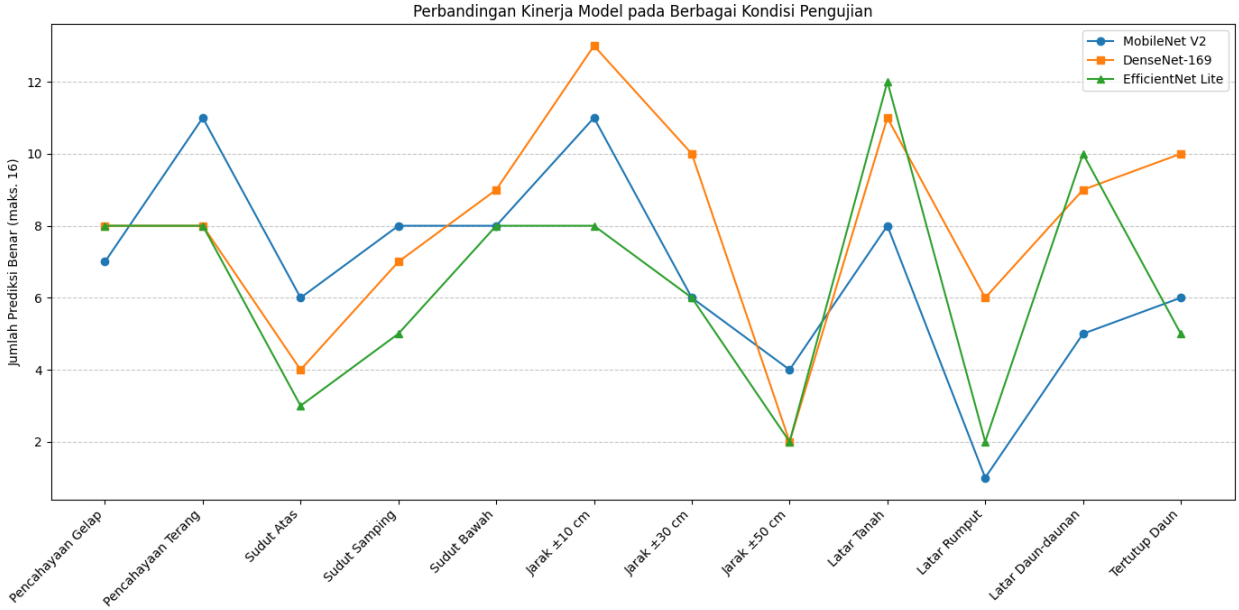
Gambar 6. Aplikasi Pengenalan Jamur: (a) Halaman Hasil Prediksi, (b) Halaman Daftar Jamur, (c) Halaman Riwayat, (d) Halaman Kamera, (e) Halaman Galeri

E. Uji Coba Performa Model di Perangkat

Uji coba ini dilakukan secara langsung pada perangkat Android dengan dataset pengujian yang sama dengan yang digunakan pada uji coba di lingkungan pengembangan. Pengujian ini dilakukan menggunakan perangkat Redmi Note 13 Pro dengan prosesor Mediatek Helio G99 Ultra dan 8 GB RAM. Hasil dari pengujian ini dapat dilihat pada Tabel 6.

EfficientNet Lite merupakan model tercepat dalam melakukan inferensi dengan hanya membutuhkan waktu 96.92 ms saja. Model ini mencapai akurasi 92% serta menggunakan rata-rata 19.35 MB memori dalam melakukan inferensi. Aplikasi pengenalan jamur dengan model ini menggunakan ruang penyimpanan sebesar 28.62 MB.

Dari hasil pengujian ini, tampak bahwa meskipun akurasi DenseNet-169 tertinggi di antara ketiga model, peningkatan tersebut tidak sebanding dengan sumber daya perangkat yang digunakannya. Model ini memerlukan sumber daya perangkat 1,5 hingga 2 kali lebih besar hanya untuk mencapai akurasi 1 – 2% lebih baik. Dalam hal efisiensi serta keseimbangan penggunaan sumber daya perangkat dan akurasi, EfficientNet Lite merupakan model terbaik dalam pengujian ini.



Gambar 7. Hasil Pengujian Pada Kondisi Lapangan

Tabel 6. Hasil Pengukuran Performa Model di Perangkat

Model	Akurasi	Penggunaan Memori (MB)	Penggunaan Ruang Penyimpanan (MB)	Waktu Inferensi (ms)
MobileNet V2	90%	17.93	27.46	110.93
DenseNet-169	93%	40.66	38.61	290.38
EfficientNet Lite	92%	19.35	28.62	96.92

Sumber: diolah dari data primer

F. Uji Coba pada Kondisi Lapangan

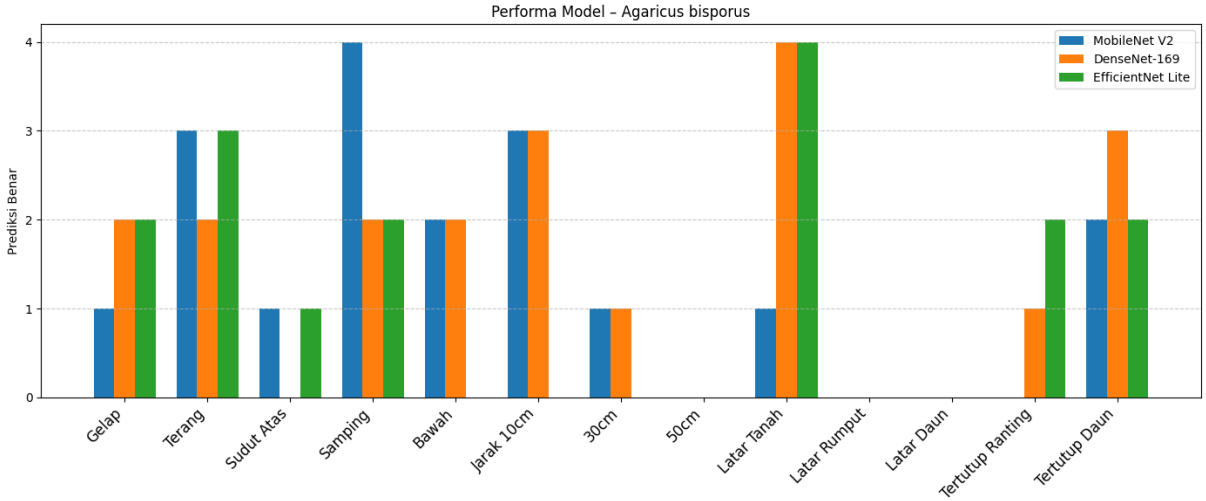
Hasil pengujian seluruh model dapat dilihat pada grafik yang ditunjukkan oleh Gambar 7. Berdasarkan grafik tersebut, tampak bahwa DenseNet-169 memiliki performa yang paling akurat serta paling konsisten di seluruh skenario dan spesies jamur. Model ini lebih mampu mengenali *Lentinula edodes*, spesies yang sulit dikenali oleh kedua model lain sebagaimana yang dapat dilihat pada Gambar 10.

Sementara itu, MobileNet V2 tampak kesulitan pada variasi latar belakang, pencahayaan rendah, dan variasi sudut. Model ini utamanya lemah pada spesies *Lentinula edodes* dan *Agaricus bisporus*, seperti ditampilkan pada Gambar 8 dan Gambar 10. MobileNet V2 cenderung bekerja lebih baik pada pencahayaan

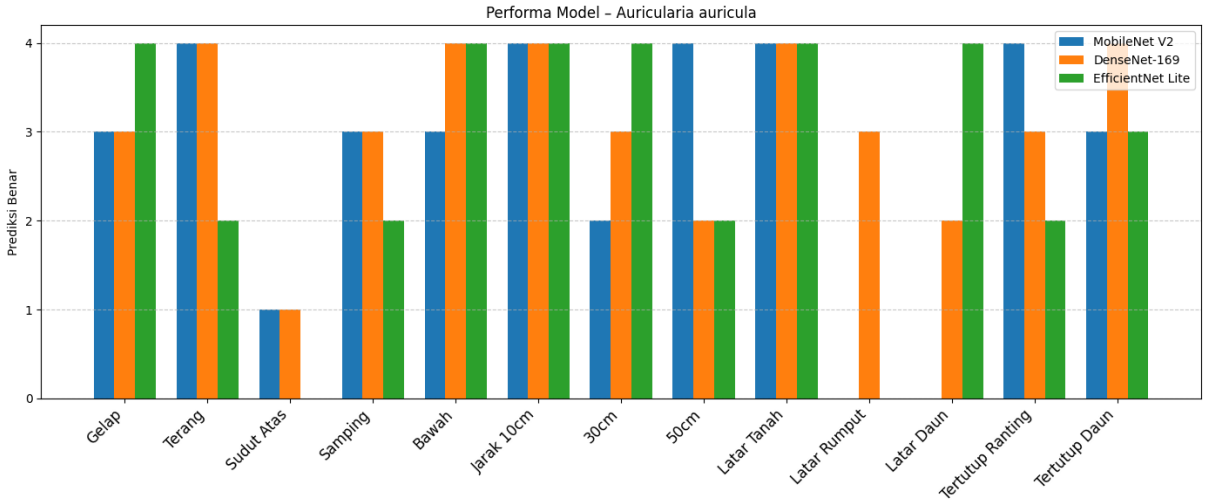
terang dan jarak pengambilan gambar yang dekat. Selain itu, model ini lebih mampu mengenali spesies *Auricularia auricula* dan *Pleurotus ostreatus* seperti yang ditunjukkan oleh Gambar 9 dan Gambar 11

Berbeda dengan dua model lain, EfficientNet Lite tampak tidak konsisten dan cenderung fluktuatif. Model ini gagal mengenali jamur pada kondisi pengambilan gambar yang relatif ideal dan justru berhasil pada kondisi yang kurang ideal. Sama seperti MobileNet V2, model ini lemah pada spesies *Lentinula edodes* dan *Agaricus bisporus* (dapat dilihat pada Gambar 8 dan Gambar 10) serta kuat pada *Auricularia auricula* dan *Pleurotus ostreatus* (dapat dilihat pada Gambar 9 dan Gambar 11).

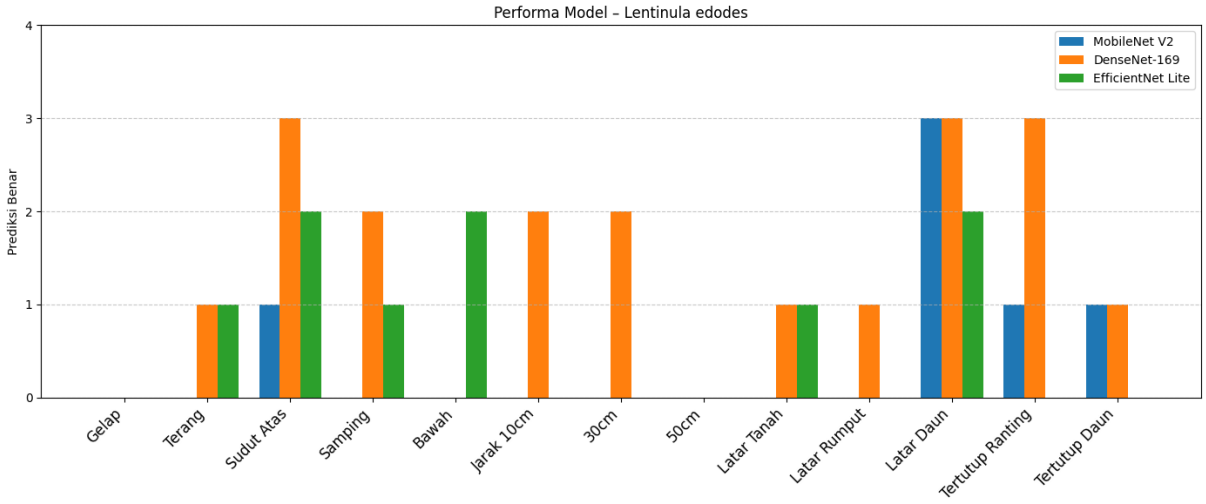
Secara umum, pada pengujian DenseNet-169 terbukti sebagai model yang paling andal dan stabil, dengan performa tinggi di berbagai kondisi pencahayaan, latar belakang, serta sudut pengambilan gambar. Kelebihan ini menjadikannya pilihan yang unggul untuk diterapkan dalam aplikasi pengenalan jamur di dunia nyata, terutama dalam konteks penggunaan oleh masyarakat umum yang mengambil gambar dalam kondisi tak terkontrol. Sementara itu, MobileNet V2 dan EfficientNet Lite memiliki keunggulan pada situasi tertentu, namun menunjukkan keterbatasan yang cukup signifikan ketika dihadapkan pada variasi kondisi lapangan. MobileNet V2 lebih sesuai untuk penggunaan di lingkungan yang terang dan jarak dekat, sementara EfficientNet Lite kurang dapat diandalkan karena performanya yang fluktuatif.



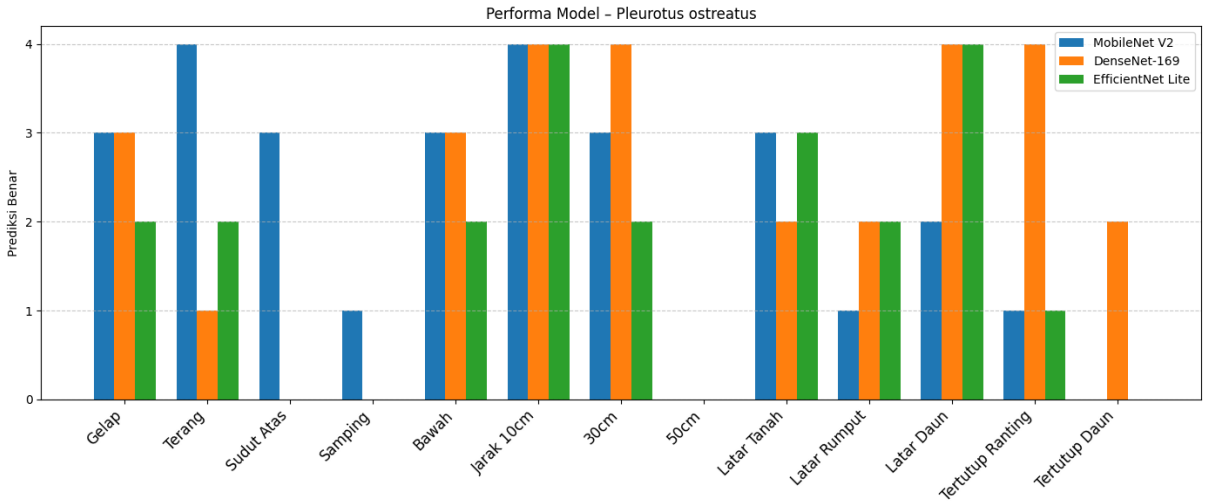
Gambar 8. Hasil Pengujian Model Pada *Agaricus bisporus*



Gambar 9. Hasil Pengujian Model pada *Auricularia Auricula*



Gambar 10. Hasil Pengujian Model pada *Lentinula edodes*



Gambar 11. Hasil Pengujian Model pada *Pleurotus ostreatus*

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, penulis menyimpulkan bahwa model CNN pengenalan jamur beracun memang dapat untuk diimplementasikan dengan hasil yang memuaskan pada perangkat *mobile*, khususnya yang berbasis Android. Model-model seperti MobileNet V2, DenseNet-169, dan EfficientNet Lite berhasil diimplementasikan secara langsung pada perangkat Android tanpa terlalu banyak menggunakan sumber daya perangkat. Ketiga model berhasil memberikan hasil yang memuaskan, dengan rata-rata akurasi di atas 90% ketika diuji menggunakan dataset pengujian.

Penelitian ini merupakan pengembangan dari penelitian-penelitian sebelumnya yang sebagian besar diterapkan pada perangkat non-*mobile*, seperti komputer desktop atau server berbasis *cloud*, sehingga membutuhkan spesifikasi tinggi atau koneksi internet yang stabil. Jika dibandingkan dengan sistem sejenis atau pendekatan sebelumnya, keunggulan utama dari aplikasi ini terletak pada kemampuannya untuk berfungsi secara luring tanpa koneksi internet, sehingga sangat cocok untuk penggunaan di lapangan, seperti hutan atau daerah terpencil tanpa infrastruktur digital yang memadai. Selain itu, sistem ini dirancang dengan mempertimbangkan kemudahan penggunaan dan aksesibilitas bagi masyarakat umum, termasuk mereka yang tidak memiliki latar belakang teknis di bidang biologi atau teknologi informasi. Antarmuka aplikasi yang sederhana dan intuitif memungkinkan pengguna untuk melakukan identifikasi jamur secara mandiri hanya dengan mengambil foto melalui kamera perangkat. Dengan demikian, aplikasi ini berpotensi menjadi alat bantu edukatif dan preventif dalam mencegah kasus keracunan jamur, terutama di daerah

pedesaan atau wilayah dengan akses terbatas terhadap tenaga ahli mikologi.

Dari hasil tahapan-tahapan pengujian yang telah dilakukan, ditemukan bahwa dari ketiga model DenseNet-169 menggunakan sumber daya perangkat paling banyak. Model ini merupakan model dengan efisiensi penggunaan sumber daya perangkat yang paling rendah. Namun, hal ini diimbangi dengan fakta bahwa model ini merupakan yang paling baik ketika dihadapkan dengan variasi kondisi pengambilan gambar di lapangan. Performa MobileNet V2 dapat dikatakan berada di tengah-tengah DenseNet-169 dan EfficientNet Lite. Model ini sedikit lebih ringan daripada EfficientNet Lite namun juga sedikit lebih lambat. Akurasi model ini berada di bawah EfficientNet Lite ketika diuji menggunakan dataset pengujian. Namun, performa model ini sedikit lebih baik dan konsisten ketika diuji pada kondisi lapangan.

Dengan mempertimbangkan seluruh hasil pengujian, dapat disimpulkan bahwa implementasi model CNN untuk identifikasi jamur beracun pada perangkat Android merupakan pendekatan yang efektif dan layak secara teknis. Ketiga model yang diuji, MobileNet V2, DenseNet-169, dan EfficientNet Lite, mampu dijalankan secara lokal tanpa memerlukan koneksi internet, serta memberikan waktu inferensi yang cepat dengan penggunaan sumber daya perangkat yang masih dalam batas wajar. DenseNet-169 menonjol dari segi akurasi dan ketahanan terhadap variasi kondisi lapangan, menjadikannya model yang andal meskipun dengan konsumsi sumber daya yang lebih tinggi. Di sisi lain, EfficientNet Lite unggul dari segi efisiensi dan kecepatan, namun menunjukkan kelemahan dalam hal konsistensi pada kondisi nyata. MobileNet V2 cukup seimbang antara akurasi, efisiensi, dan stabilitas performa, menjadikannya opsi alternatif yang layak untuk diterapkan dalam aplikasi mobile ringan.

Sebagai tindak lanjut dari penelitian ini penulis menyarankan beberapa potensi pengembangan yang dapat dilakukan. Pertama, aplikasi pengenalan jamur beracun ini mungkin dapat bekerja lebih baik dengan menggunakan pendekatan *ensemble learning*. Pendekatan ini menggabungkan beberapa model CNN untuk menghasilkan sebuah klasifikasi dan memungkinkan sistem untuk memanfaatkan kelebihan masing-masing model. *Ensemble learning* berpotensi untuk memberikan jalan tengah dari performa ketiga model tersebut yang mungkin dapat menghasilkan akurasi identifikasi yang baik, penggunaan sumber daya yang minimal, serta ketahanan terhadap variasi kondisi pengambilan gambar di lapangan.

Kedua, pengembangan aplikasi ini juga dapat mencakup *input* tambahan seperti lokasi geografis (GPS), musim, atau jenis substrat tempat jamur tumbuh untuk membantu sistem mengenali pola kemunculan spesies tertentu, yang jika digabung dengan *output* dari *ensemble* CNN, mungkin dapat meningkatkan akurasi secara signifikan serta membantu pengguna lebih memahami informasi yang disajikan.

DAFTAR PUSTAKA

Amelya, M., Putra, I., Hermawan, R., & Nurzakiah, R. (2023). Pemanfaatan dan Resiko Keracunan Jamur Coprinoid pada Jerami Padi di Indonesia. *Quagga: Jurnal Pendidikan dan Biologi*, *15*(1), 1–8. https://doi.org/10.25134/quagga.v15i1.5180

DemiRel, Y., & DemiRel, G. (2023). Deep Learning Based Approach for Classification of Mushrooms. *Gazi University Journal of Science Part A: Engineering and Innovation*, *10*(4), 487–498. https://doi.org/10.54287/gujsa.1355751

Gold, J. A. W., Kiernan, E., Yeh, M., Jackson, B. R., & Benedict, K. (2021). Health Care Utilization and Outcomes Associated with Accidental Poisonous Mushroom Ingestions—United States, 2016–2018. *MMWR. Morbidity and Mortality Weekly Report*, *70*(10), 337–341. https://doi.org/10.15585/mmwr.mm7010a1

Graeme, K. A. (2014). Mycetism: A Review of the Recent Literature. *Journal of Medical Toxicology*, *10*(2), 173–189. https://doi.org/10.1007/s13181-013-0355-2

Haksoro, E. I., & Setiawan, A. (2021). Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network. *Jurnal ELTIKOM*, *5*(2), 81–91. https://doi.org/10.31961/eltikom.v5i2.428

Kousis, I., Perikos, I., Hatzilygeroudis, I., & Virvou, M. (2022). Deep Learning Methods for Accurate Skin Cancer Recognition and Mobile Application. *Electronics*, *11*(9), Article 9.

https://doi.org/10.3390/electronics11091294

Lyall, A. (2024, Oktober 10). *Alert over wild mushrooms in France: 400 cases of intoxication since July*. https://www.connexionfrance.com/news/alert-over-wild-mushrooms-in-france-400-cases-of-intoxication-since-july/681910

Mauludy, M. W., Rulyana, D., & Hardjianto, M. (2024). Deteksi Jamur Beracun dengan Algoritma Convolutional Neural Network dan Arsitektur EfficientNet-B0. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *8*(1), 555–562. https://doi.org/10.30865/mib.v8i1.7276

Nisa', C., Puspaningrum, E. Y., & Maulana, H. (2020). Penerapan Metode Convolutional Neural Network untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data. *Prosiding Seminar Nasional Informatika Bela Negara*, *1*, 169–175. https://doi.org/10.33005/santika.v1i0.46

Putra, I. (2022). KASUS-KASUS KERACUNAN JAMUR LIAR DI INDONESIA / Poisoning Cases of Wild Mushrooms in Indonesia. *Jurnal Ekologi Kesehatan*, *20*, 215–230. https://doi.org/10.22435/jek.v20i3.4943

Putra, I. P., & Hermawan, R. (2021). Identifikasi Jamur Beracun Clitocybe sp. Di Gresik, Indonesia (Studi Kasus). *Media Penelitian dan Pengembangan Kesehatan*, *31*(2), 119–124. https://doi.org/10.22435/mpk.v31i2.4352

Sibero, M. T., Putra, I. P., & Murwani, R. (2021). Deskripsi dan Potensi Cendawan Makro Asal Hutan Adat Penembahen, Desa Juhar, Kabupaten Tanah Karo, Sumatera Utara. *Jurnal Mikologi Indonesia*, *5*(1), 48–54. https://doi.org/10.46638/jmi.v5i1.164

William Lijaya Therry, R., Junaidi, A., & Nugroho Sihananto, A. (2024). PROGRAM PENERJEMAH BAHASA ISYARAT INDONESIA (BISINDO) SECARA REAL TIME MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DAN MEDIAPIPE. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(6), 11642–11649. https://doi.org/10.36040/jati.v8i6.11582

Yulita, I. N., Amri, N. A., & Hidayat, A. (2023). Mobile Application for Tomato Plant Leaf Disease Detection Using a Dense Convolutional Network Architecture. *Computation*, *11*(2), Article 2. https://doi.org/10.3390/computation11020020

Zahan, N., Hasan, M. Z., Malek, M. A., & Reya, S. S. (2021). A Deep Learning-Based Approach for Edible, Inedible and Poisonous Mushroom Classification. *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 440–444. https://doi.org/10.1109/ICICT4SD50815.2021.9396845

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*(4), 99–142. https://doi.org/10.1007/s10462-024-10721-6

