

## Implementasi Machine Learning untuk Identifikasi Jamur pada Aplikasi Android Menggunakan Algoritma *Convolutional Neural Network*

Reyhan Afif Fahri<sup>\*1</sup>, Kemal Farouq Mauladi<sup>\*2</sup>, Muhammad Hasan Wahyudi<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Fakultas Sains Teknologi, Universitas Islam Lamongan, Indonesia  
Email: <sup>1</sup>[reyhan.raf73@gmail.com](mailto:reyhan.raf73@gmail.com), <sup>2</sup>[kemalfarouq@unisla.ac.id](mailto:kemalfarouq@unisla.ac.id), <sup>3</sup>[hasanwahyudi@unisla.ac.id](mailto:hasanwahyudi@unisla.ac.id)

### Abstrak

Kurangnya pengetahuan masyarakat dalam mengenali jamur beracun di Indonesia masih menjadi masalah serius yang dapat menyebabkan kasus keracunan, terutama di daerah yang memiliki kebiasaan mengonsumsi jamur liar. Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis Android yang dapat membantu masyarakat dalam mengidentifikasi jenis jamur secara otomatis sebagai upaya pencegahan keracunan. Metode yang digunakan adalah *Convolutional Neural Network (CNN)* dengan arsitektur MobileNetV2, yang diimplementasikan menggunakan TensorFlow Lite agar dapat dijalankan secara efisien pada perangkat mobile. Dataset terdiri dari 2000 citra jamur yang dibagi dalam rasio 80% untuk pelatihan dan 20% untuk validasi, dengan pelatihan dilakukan selama 30 epoch. Model yang dihasilkan mencapai akurasi pelatihan sebesar 83% dan akurasi pengujian sebesar 85%. Aplikasi yang dikembangkan mampu melakukan identifikasi secara cepat dan akurat, serta memiliki potensi besar untuk meningkatkan kesadaran masyarakat terhadap keamanan pangan. Penelitian ini memberikan kontribusi penting dalam pengembangan teknologi berbasis kecerdasan buatan di bidang kesehatan dan keamanan pangan.

**Kata kunci:** jamur, CNN, android, identifikasi.

## *Implementation of Machine Learning for Fungus Identification in Android Applications Using Convolutional Neural Network Algorithm*

### Abstract

The lack of public knowledge in recognizing poisonous mushrooms in Indonesia is still a serious problem that can cause poisoning cases, especially in areas that have a habit of consuming wild mushrooms. This research aims to develop an Android-based application that can help people in identifying types of mushrooms automatically as an effort to prevent poisoning. The method used is *Convolutional Neural Network (CNN)* with MobileNetV2 architecture, which is implemented using TensorFlow Lite to run efficiently on mobile devices. The dataset consists of 2000 mushroom images divided in a ratio of 80% for training and 20% for validation, with training performed for 30 epochs. The resulting model achieved a training accuracy of 83% and a testing accuracy of 85%. The developed application is able to perform identification quickly and accurately, and has great potential to increase public awareness of food safety. This research makes an important contribution to the development of artificial intelligence-based technology in the field of health and food safety.

**Keywords:** fungus, CNN, android, identification.

## 1. PENDAHULUAN

Jamur merupakan salah satu organisme dengan jumlah spesies terbanyak di dunia [1]. Di Indonesia, menurut laporan Lembaga Ilmu Pengetahuan Indonesia (LIPI) tahun 2019, jumlah spesies fungi yang telah teridentifikasi baru mencapai 2.273 jenis, atau sekitar 0,15% dari total estimasi spesies jamur di dunia [2]. Meskipun banyak jenis jamur dapat dikonsumsi dan memiliki nilai gizi tinggi, terdapat pula sejumlah spesies yang bersifat beracun dan berbahaya bagi manusia. Konsumsi jamur liar masih sering ditemukan, terutama di daerah pedesaan, sayangnya pengetahuan masyarakat tentang perbedaan antara jamur yang dapat dikonsumsi dan yang beracun masih terbatas. Hal ini dapat meningkatkan risiko keracunan makanan. Dalam kurun waktu sepuluh tahun terakhir, tercatat 76 kasus keracunan akibat konsumsi jamur liar di Indonesia, dengan jumlah korban mencapai 550 orang dan 9 di antaranya meninggal dunia [3]. Sayangnya, hingga kini Indonesia belum memiliki sistem atau database nasional yang secara khusus mencatat insiden keracunan jamur liar secara terstruktur dan berkelanjutan [4]. Hal

ini menunjukkan perlunya upaya preventif melalui edukasi dan teknologi untuk membantu masyarakat dalam mengidentifikasi jenis jamur yang aman dikonsumsi.

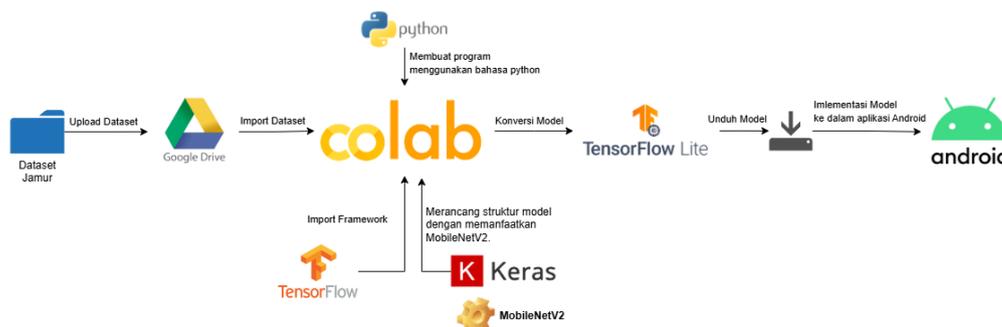
Kemajuan teknologi kecerdasan buatan telah membuka peluang baru dalam upaya identifikasi spesies jamur secara akurat. Salah satu pendekatan yang banyak digunakan adalah metode *machine learning*, khususnya *Convolutional Neural Networks* (CNN), yang terbukti efektif dalam klasifikasi objek berbasis gambar. CNN mampu mengenali pola visual kompleks sehingga sangat potensial diterapkan dalam klasifikasi jamur berdasarkan morfologi. Beberapa penelitian telah membuktikan efektivitas metode ini. Misalnya, penelitian oleh Ahmad Rizal Imaduddin (2023) menunjukkan bahwa model CNN mampu mengidentifikasi jamur beracun dengan tingkat akurasi mencapai 96% [5]. Selain itu, pengembangan aplikasi mobile berbasis CNN memungkinkan proses identifikasi dilakukan secara cepat dan mudah oleh masyarakat umum. Penerapan teknologi ini dapat menjadi solusi praktis dalam meningkatkan kesadaran dan pengetahuan masyarakat terhadap risiko keracunan jamur liar.

Penelitian ini bertujuan untuk mengembangkan aplikasi Android berbasis *machine learning* dengan menggunakan metode *Convolutional Neural Network* (CNN) untuk identifikasi jenis jamur. Aplikasi ini dirancang untuk membantu masyarakat dalam mengenali perbedaan antara jamur yang aman dan beracun, sehingga dapat mengurangi risiko keracunan akibat konsumsi jamur liar. Selain itu, aplikasi ini diharapkan dapat menjadi media edukasi yang efektif dalam meningkatkan literasi masyarakat terkait keamanan konsumsi jamur.

## 2. METODE PENELITIAN

### 2.1. Pengembangan Aplikasi Berbasis Machine Learning

Penelitian ini menggunakan pendekatan pengembangan aplikasi berbasis *machine learning*, dengan menekankan pada penggunaan algoritma *Convolutional Neural Network* (CNN) yang diimplementasikan melalui library TensorFlow dan Keras, serta memanfaatkan arsitektur MobileNetV2 guna menghasilkan model dengan tingkat akurasi yang optimal [6]. Selanjutnya, model yang telah dibangun dikonversi ke dalam format .tflite untuk mendukung proses deployment pada perangkat Android (*on-device*). Alur kerja penelitian secara keseluruhan disajikan pada Gambar 1.



Gambar 1. Diagram Alur Proses Pelatihan dan Implementasi Model Klasifikasi

Proses pengembangan model diawali dengan pengumpulan dan pengunggahan dataset gambar jamur ke Google Drive. Selanjutnya, dataset tersebut diimpor ke Google Colab untuk diproses. Di dalam Google Colab, peneliti membuat program menggunakan bahasa pemrograman Python dan mengimpor framework TensorFlow serta Keras untuk merancang struktur model dengan memanfaatkan arsitektur MobileNetV2. Setelah arsitektur model selesai dirancang, model dilatih menggunakan dataset yang telah disiapkan. Setelah proses pelatihan dan validasi selesai serta model mencapai performa yang diinginkan, model dikonversi ke dalam format TensorFlow Lite (.tflite), yang lebih ringan dan sesuai untuk digunakan di perangkat mobile [7]. Model .tflite kemudian diunduh dan diimplementasikan ke dalam aplikasi Android agar dapat dijalankan secara efisien pada perangkat tersebut.

#### 2.1.1. Dataset

Dalam penelitian ini, dataset yang dipakai mencakup 10 kategori label dengan total sebanyak 2000 gambar secara keseluruhan. Data tersebut diperoleh oleh peneliti dari kawasan hutan Mantup, Kabupaten Lamongan. Pengumpulan data dilakukan melalui metode observasi, sehingga memungkinkan diperolehnya informasi mendalam mengenai karakteristik visual jamur, baik yang beracun maupun yang tidak, seperti warna, bentuk,

tekstur, dan pola [8]. Dataset ini kemudian dibagi menjadi dua bagian, yaitu 1600 gambar (80%) untuk pelatihan dan 400 gambar (20%) untuk validasi. Rincian lebih lanjut mengenai dataset dapat dilihat pada Tabel 1.

Tabel 1. Dataset Jamur

Nama Jamur	Jenis Jamur	Jumlah Data	Training	Validasi
Jamur Barat	Tidak Beracun	200	160	40
Jamur Kuping	Tidak Beracun	200	160	40
Jamur Lentinus Squarrosulus	Tidak Beracun	200	160	40
Jamur Merang	Tidak Beracun	200	160	40
Jamur Tiram	Tidak Beracun	200	160	40
Jamur Amanita phalloides	Beracun	200	160	40
Jamur Coprinopsis lagopus	Beracun	200	160	40
Jamur Leucoagaricus leucothites	Beracun	200	160	40
Jamur Mycena leaiana	Beracun	200	160	40
Jamur Chlorophyllum molybdites	Beracun	200	160	40
<b>Total</b>		<b>2000</b>	<b>1600</b>	<b>400</b>

### 2.1.2. Pemrosesan Data

Preprocessing data adalah tahap penting dalam pengembangan model machine learning maupun deep learning, karena bertujuan untuk mengolah data mentah agar layak digunakan dalam proses pelatihan model. Dalam proses ini, dilakukan beberapa langkah penting seperti menyesuaikan ukuran gambar agar seragam (resizing), serta mengubah nilai piksel ke skala tertentu melalui normalisasi[9]. Pada tahap ini, dilakukan beberapa langkah penting, antara lain:

- Resize: Seluruh gambar diubah ukurannya ke dimensi tertentu (misalnya 128x128 piksel) menggunakan fungsi `resize()`, agar memiliki ukuran yang seragam sehingga dapat diproses secara konsisten oleh model.
- Konversi Format: Gambar dikonversi ke format RGB untuk memastikan semua gambar memiliki tiga saluran warna yang sama.
- Normalisasi: Untuk mengubah nilai piksel gambar ke dalam skala tertentu (biasanya antara 0 hingga 1) untuk mempercepat proses pelatihan dan meningkatkan stabilitas model.
- Augmentasi data: proses memperbanyak variasi data dengan teknik seperti rotasi, flipping, zooming, atau perubahan pencahayaan, guna meningkatkan kemampuan generalisasi model terhadap data baru.

### 2.1.3. Pengembangan Model

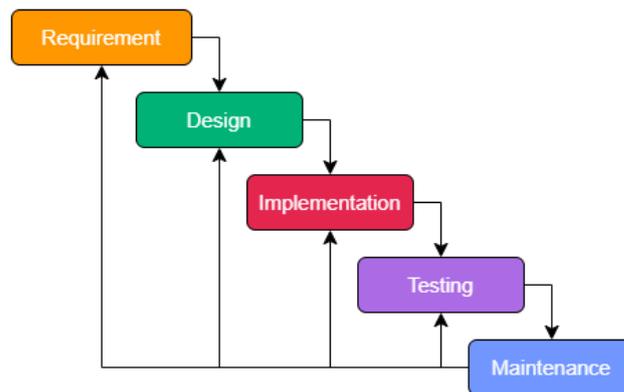
Model yang dikembangkan dalam penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN) dengan pendekatan transfer learning melalui arsitektur MobileNetV2 sebagai fondasi utama.[10]. Pemilihan *MobileNetV2* dilatarbelakangi oleh efisiensi komputasi yang tinggi serta kinerjanya yang unggul dalam klasifikasi citra. Arsitektur ini dipakai sebagai dasar karena sebagian besar parameternya telah melalui proses pelatihan awal pada tugas-tugas umum dalam bidang penglihatan komputer (*computer vision*), sehingga memungkinkan proses pelatihan ulang berjalan lebih cepat dan tidak memerlukan dataset berukuran besar untuk memperoleh hasil yang optimal[11].

Model yang dihasilkan dari proses ini dirancang untuk mengenali dan mengelompokkan gambar jamur ke dalam 10 kelas, yakni: Jamur Barat, Jamur Kuping, *Lentinus squarrosulus*, Jamur Merang, Jamur Tiram, *Amanita phalloides*, *Coprinopsis lagopus*, *Leucoagaricus leucothites*, *Mycena leaiana*, dan *Chlorophyllum molybdites*. Klasifikasi dilakukan berdasarkan ciri visual yang diperoleh dari data training.

Dalam proses pelatihan, model dioptimalkan dengan menggunakan 30 epoch dan batch size sebesar 32. Dengan memanfaatkan arsitektur *MobileNetV2* yang telah melalui proses *pre-training*, model mampu mencapai akurasi yang tinggi serta memberikan kemampuan generalisasi yang baik dalam mendeteksi jenis jamur. Setelah model melalui tahap evaluasi dan terbukti memiliki tingkat kepercayaan yang tinggi, langkah selanjutnya adalah mengonversi model ke dalam format *TensorFlow Lite* dengan menggunakan *TF Lite Converter*. Konversi ini mengubah format asli TensorFlow menjadi format `.tflite`, yang dapat dijalankan secara efisien di berbagai perangkat, termasuk ponsel Android, perangkat iOS, maupun platform tertanam (*embedded system*) lainnya[12].

## 2.2. Metode Pengembangan Aplikasi

Pada penelitian ini, aplikasi dikembangkan menggunakan metode *waterfall*. Metode *Waterfall* merupakan salah satu model dalam pengembangan perangkat lunak yang tergolong ke dalam siklus hidup klasik (*classic life cycle*)[13]. Metode ini mengutamakan proses yang berjalan secara sistematis dan berurutan. Konsep pengembangannya dapat diilustrasikan seperti aliran air terjun, di mana setiap tahap dilakukan satu per satu dari awal hingga akhir tanpa melewati atau melompati tahap sebelumnya. Penamaan *waterfall* merujuk pada urutan langkah-langkah yang harus diselesaikan secara bertahap. Adapun tahap-tahap yang termasuk dalam metode Waterfall dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Metode *Waterfall*

### 1. Analisis Kebutuhan (*Requirement*)

Tahap awal ini berfungsi untuk mengidentifikasi dan mengumpulkan kebutuhan sistem yang akan dikembangkan. Informasi diperoleh melalui proses wawancara, survei, serta kajian literatur guna memperoleh pemahaman mendalam mengenai kebutuhan pengguna dan sistem.

### 2. Perancangan Sistem (*Design*)

Pada tahap ini, dilakukan penyusunan rancangan sistem secara menyeluruh. Aktivitas yang dilakukan meliputi perancangan antarmuka pengguna serta pemetaan alur kerja sistem agar sesuai dengan kebutuhan yang telah dianalisis sebelumnya.

### 3. Penerapan (*Implementation*)

Tahapan ini mencakup proses pengembangan aplikasi Android menggunakan bahasa pemrograman Kotlin, dengan mengintegrasikan model machine learning berbasis *Convolutional Neural Network* (CNN) menggunakan arsitektur MobileNetV2. Model yang telah dilatih sebelumnya kemudian dikonversi ke format TensorFlow Lite (.tflite) agar dapat dijalankan secara efisien di perangkat Android. Integrasi dilakukan melalui TensorFlow Lite Interpreter, yang memuat dan menjalankan model langsung di perangkat. Gambar jamur yang diambil dari kamera atau galeri diproses terlebih dahulu, lalu diklasifikasikan oleh model untuk menentukan jenis jamur berdasarkan ciri visualnya. Hasil klasifikasi kemudian ditampilkan kepada pengguna melalui antarmuka aplikasi. Dengan pendekatan ini, proses klasifikasi dapat dilakukan secara offline, cepat, dan efisien tanpa memerlukan koneksi ke server eksternal.

### 4. Pengujian (*Testing*)

Pengujian dilakukan untuk memastikan bahwa sistem yang dikembangkan telah sesuai dengan rancangan, berfungsi dengan baik, dan memiliki kinerja yang optimal. Dalam penelitian ini, dilakukan dua jenis pengujian, yaitu:

- Pengujian Aplikasi (Black-Box Testing)

pengujian ini dilakukan untuk mengevaluasi fungsionalitas aplikasi berdasarkan masukan dan keluaran tanpa melihat proses internal. Penguji memberikan berbagai input melalui antarmuka aplikasi, seperti gambar jamur dari kamera atau galeri, lalu memeriksa apakah hasil klasifikasi yang ditampilkan sesuai dengan ekspektasi. Metode ini digunakan untuk memastikan semua fitur berjalan sesuai dengan spesifikasi.

- Pengujian Model Machine Learning

Pengujian model dilakukan langsung melalui perangkat smartphone Android untuk menilai kemampuan klasifikasi model dalam kondisi nyata (on-device). Evaluasi dilakukan dengan memberikan

beberapa gambar jamur yang telah diketahui jenisnya, kemudian mencatat jumlah prediksi yang benar dan salah.

5. Pemeliharaan (Maintenance)

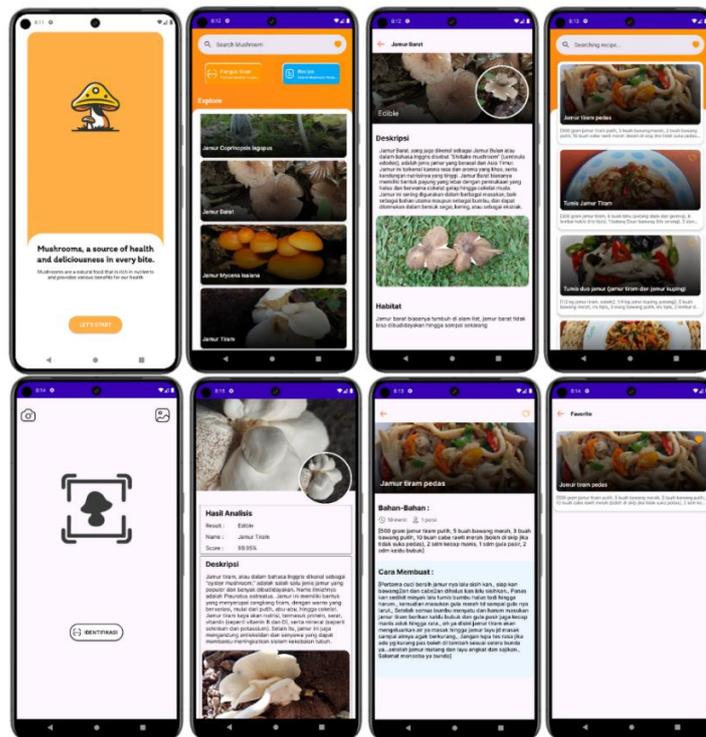
Tahap terakhir ini mencakup aktivitas perawatan sistem secara rutin, perbaikan jika ditemukan kesalahan, peninjauan kembali terhadap performa sistem, serta peningkatan atau pembaruan perangkat lunak sesuai kebutuhan di masa mendatang.

3. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini berupa aplikasi Android yang dirancang untuk mendeteksi jenis jamur. Aplikasi ini memfasilitasi pengguna dalam melakukan identifikasi berbagai jenis jamur dengan mudah.

3.1. Hasil Pengembangan Aplikasi Android

Aplikasi ini dikembangkan dengan Android Studio sebagai platform utama dalam proses pengembangannya. Android Studio dipilih karena menawarkan dukungan optimal dalam pembuatan aplikasi Android. Dibangun di atas IntelliJ IDEA sebuah IDE terkenal untuk bahasa pemrograman Java dan Kotlin. Android Studio tidak hanya berfungsi sebagai editor kode dan alat bantu pemrograman, tetapi juga dilengkapi dengan beragam fitur serta mendukung proses pembangunan, pengujian, dan distribusi aplikasi Android secara efisien. Gambar 4 menampilkan tampilan hasil rancangan UI/UX dari aplikasi Android untuk mendeteksi jenis jamur. Perancangan UI/UX ini difokuskan pada penciptaan antarmuka yang *user-friendly*, menarik secara visual, dan mudah dipahami oleh pengguna. Selama tahap ini, berbagai komponen visual seperti susunan tampilan, skema warna, ikon, serta interaksi antar elemen dirancang secara detail guna memberikan pengalaman pengguna yang optimal.



Gambar 3. Desain UI/UX Aplikasi

Berdasarkan Gambar 4, aplikasi ini terdiri dari beberapa halaman utama. Pertama, terdapat halaman pembuka (*Start Page*) yang menjadi tampilan awal saat aplikasi dibuka (gambar paling kiri atas). Selanjutnya adalah halaman beranda (*Home Page*) yang menampilkan berbagai jenis jamur beserta gambarnya (gambar kedua dari kiri atas). Jika salah satu jamur dipilih, pengguna akan diarahkan ke halaman detail jamur (*Detail Page*) yang berisi informasi lengkap mengenai deskripsi dan habitat jamur tersebut (gambar ketiga dari kiri atas). Aplikasi ini juga menyediakan halaman resep (*Recipe Page*) yang memuat berbagai olahan makanan berbahan dasar jamur (gambar keempat dari kiri atas). Salah satu fitur utama aplikasi ini adalah halaman pemindaian (*Scan Page*), yang memungkinkan pengguna mengidentifikasi jenis jamur menggunakan kamera atau gambar dari galeri (gambar

pertama dari kiri bawah). Hasil identifikasi kemudian ditampilkan di halaman hasil (*Result Page*), yang menyajikan nama jamur, tingkat kecocokan, serta deskripsinya secara rinci (gambar kedua dari kiri bawah). Selain itu, terdapat halaman resep detail (*Recipe Detail Page*) yang menjelaskan bahan-bahan dan langkah-langkah memasak jamur secara lengkap (gambar ketiga dari kiri bawah). Terakhir, pengguna dapat menyimpan resep favorit mereka, yang kemudian ditampilkan di halaman favorit (*Favorite Page*) (gambar keempat dari kiri bawah).

### 3.1.1. Pengujian Blackbox

Setelah proses pengembangan Aplikasi Android untuk mendeteksi jenis jamur berbasis Machine Learning selesai, langkah berikutnya adalah melakukan proses pengujian. Dalam penelitian ini, pengujian dilakukan menggunakan metode Blackbox Testing. *Blackbox Testing* merupakan metode yang menitikberatkan pada pengujian fungsi aplikasi tanpa memerlukan informasi mengenai struktur kode maupun proses internalnya [14]. Penguji hanya berfokus pada data masukan (*input*) yang diberikan dan hasil keluaran (*output*) yang dihasilkan, tanpa mengetahui bagaimana data diproses di dalam sistem. Metode ini tergolong sederhana untuk diterapkan karena hanya memerlukan batas minimum dan maksimum dari data yang diharapkan [11]. Rangkuman hasil pengujian menggunakan metode blackbox disajikan pada data berikut.

Tabel 2. Hasil Pengujian *Blackbox Testing*

No	Nama Fitur	Skenario Pengujian	Hasil yang diharapkan	Hasil Uji	Keterangan
1.	Start Page (Tombol Start)	User menekan tombol "Mulai"	Berpindah ke halaman Home	Valid	Berfungsi dengan baik
2.	Home Page (List Jamur)	User memilih salah satu jenis jamur dari daftar	Muncul halaman detail jamur yang dipilih	Valid	Menampilkan informasi yang sesuai
3.	Search (Home Page)	User memasukkan kata kunci "Tiram"	Menampilkan hasil yang mengandung kata "Tiram"	Valid	Fitur pencarian bekerja dengan benar
4.	Detail Jamur	User membuka detail jamur "Jamur Barat"	Muncul deskripsi dan habitat jamur tersebut	Valid	Konten tampil sesuai harapan
5.	Resep Jamur	User membuka halaman Resep	Menampilkan daftar resep berbahan jamur	Valid	Tampilan dan isi sesuai
6.	Search (Recipe Page)	User memasukkan kata "Tiram" di kolom pencarian resep	Menampilkan resep yang menggunakan jamur tiram	Valid	Resep terkait muncul
7.	Scan Page	User menekan tombol "Identifikasi" tanpa input gambar	Muncul pesan error/peringatan input kosong	Valid	Validasi input berhasil
8.	Scan Page (galeri)	User mengunggah gambar jamur tiram dari galeri	Menampilkan hasil analisis dengan nama dan akurasi prediksi	Valid	Sistem mengenali jenis jamur
9.	Scan Page(kamera)	User mengunggah gambar jamur tiram dari kamera	Menampilkan hasil analisis dengan nama dan akurasi prediksi	Valid	Sistem mengenali jenis jamur
10.	Result Page	User memeriksa hasil identifikasi jamur	Menampilkan nama, deskripsi, dan skor prediksi	Valid	Detail informasi lengkap
11.	Resep Detail	User menekan resep "Jamur Tiram Pedas"	Menampilkan bahan dan langkah pembuatan	Valid	Informasi tampil sesuai
12.	Tambah Favorit	User menekan ikon favorit pada salah satu resep	Resep tersimpan di halaman Favorite	Valid	Resep masuk ke daftar favorit

13.	Halaman Favorit	User membuka halaman Favorit	Menampilkan resep yang sudah ditandai sebagai favorit	Valid	Data tampil sesuai
14.	Hapus Favorit	User menekan ikon favorit pada salah satu resep favorit	Resep dihapus dari halaman Favorite	Valid	Resep terhapus dari daftar favorit

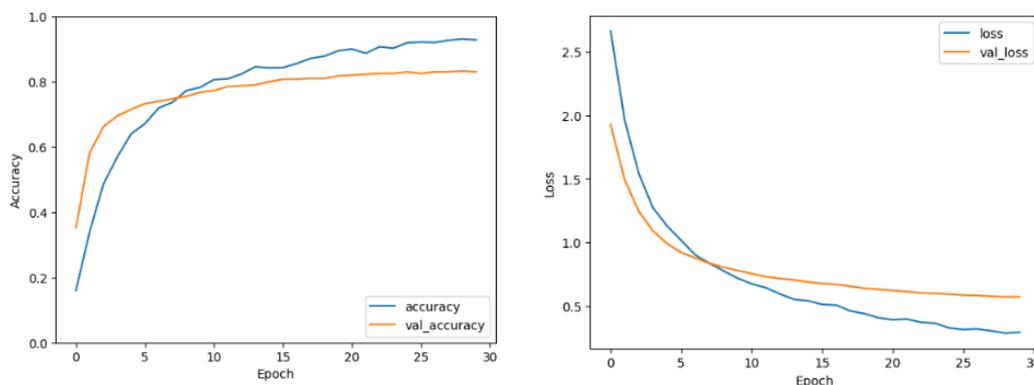
Berdasarkan hasil pengujian menggunakan metode Blackbox, seluruh fitur dan menu yang tersedia dalam aplikasi berhasil berfungsi dengan baik dan menghasilkan keluaran yang sesuai dengan harapan pengembang.

### 3.2. Hasil Pengujian Model Deteksi

Pelatihan model dilakukan selama 30 epoch, di mana satu epoch merepresentasikan satu putaran penuh pemrosesan seluruh data pelatihan oleh algoritma machine learning [15]. Dalam setiap epoch, model akan membaca dan mempelajari seluruh dataset training satu kali. Selama proses pelatihan berlangsung, model akan mengulangi proses ini berkali-kali untuk menyesuaikan serta menyempurnakan bobot dan bias agar kinerjanya semakin optimal.

#### 3.2.1. Hasil Pelatihan Model

Model yang dikembangkan telah dilatih selama 30 epoch menggunakan optimizer Adam dengan learning rate 0.0001 dan batch size 32. Dataset terdiri dari 160 gambar untuk pelatihan dan 40 gambar untuk validasi. Selama proses pelatihan, dilakukan berbagai tahapan preprocessing seperti konversi gambar ke format RGB, resize, dan konversi ke array numerik. Hasil pelatihan ditampilkan dalam Gambar 3, yang terdiri dari grafik akurasi dan loss pada data latih dan validasi.



Gambar 4. Grafik Performa Model, Akurasi dan Loss pada Data Pelatihan dan Validasi

Model yang dikembangkan telah melalui proses pelatihan selama 30 epoch dan menunjukkan performa yang cukup baik. Berdasarkan hasil pelatihan, akurasi pada data latih meningkat secara signifikan dan mencapai angka mendekati 90%, sementara akurasi pada data validasi stabil pada nilai 83%. Perbedaan antara akurasi pelatihan dan validasi yang relatif kecil menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting secara signifikan. Hal ini juga didukung oleh grafik loss yang menunjukkan penurunan yang stabil pada kedua jenis data, dengan val\_loss akhir berada di angka sekitar 0.56. Secara keseluruhan, hasil ini menunjukkan bahwa model berbasis metode Convolutional Neural Network (CNN) dengan arsitektur MobileNetV2 yang digunakan dalam penelitian ini mampu melakukan klasifikasi jenis jamur dengan tingkat akurasi dan keandalan yang baik. Performa yang ditunjukkan dari berbagai metrik evaluasi menegaskan bahwa model layak untuk diterapkan dalam aplikasi berbasis android guna mendeteksi jenis jamur.

#### 3.2.2. Pengujian Model

Penulis melakukan pengujian aplikasi secara langsung melalui smartphone dengan mengunggah gambar untuk dianalisis. Sebanyak 40 gambar digunakan untuk setiap kategori, di mana setiap kategori terdiri dari 4 gambar yang berbeda. Gambar-gambar ini tidak termasuk dalam dataset yang sudah ada, tetapi memiliki jenis yang sama. Hasil dari akurasi dan objek yang terdeteksi dicatat dengan teliti. Setelah pengujian selesai, penulis

mencatat hasilnya dan menghitung totalnya menggunakan rumus akurasi (*Accuracy Formula*). Berikut adalah tabel yang menunjukkan hasil dari pengujian tersebut.

Tabel 3. Hasil Pengujian Model CNN untuk Identifikasi Jamur

Nama Jamur	Jumlah Data Uji	Berhasil	Gagal
Jamur Barat	4	3	1
Jamur Kuping	4	4	0
Jamur Lentinus Squarrosulus	4	3	1
Jamur Merang	4	4	0
Jamur Tiram	4	4	0
Jamur Amanita phalloides	4	3	1
Jamur Coprinopsis lagopus	4	3	1
Jamur Leucoagaricus leucothites	4	3	1
Jamur Mycena leaiana	4	4	0
Jamur Chlorophyllum molybdites	4	3	1
<b>Total</b>	<b>40</b>	<b>34</b>	<b>6</b>

Merujuk pada Tabel 3, model mampu mengklasifikasikan 34 sampel secara tepat, sementara terdapat 6 sampel yang tidak teridentifikasi dengan benar. Nilai akurasi diperoleh dengan menerapkan rumus perhitungan akurasi sebagai berikut;

$$Accuracy = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Prediksi}} \times 100\% \quad (1)$$

$$Accuracy = \frac{34}{40} \times 100\%$$

$$\frac{34}{40} = 0.85$$

$$0.85 \times 100\% = 85\%$$

Model klasifikasi jamur yang dikembangkan menggunakan algoritma *Convolutional Neural Network* dengan arsitektur MobileNetV2 dan diintegrasikan ke dalam aplikasi Android melalui TensorFlow Lite berhasil menunjukkan performa yang cukup baik dalam pengujian langsung di perangkat smartphone. Berdasarkan Tabel 3, dari total 40 sampel gambar uji, model berhasil mengklasifikasikan 34 sampel dengan benar, menghasilkan akurasi sebesar 85%. Nilai ini menunjukkan bahwa model mampu mengenali berbagai jenis jamur berdasarkan ciri visualnya dengan tingkat ketepatan yang cukup tinggi, meskipun gambar uji yang digunakan tidak termasuk dalam data pelatihan. Hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik.

Jika dibandingkan dengan penelitian sebelumnya oleh Elok Iedfitra Haksoro & Abas Setiawan (2021), model MobileNetV2 diterapkan untuk klasifikasi jamur yang dapat dikonsumsi atau beracun menggunakan metode transfer learning [6]. Mereka mencapai akurasi tertinggi sebesar 92,19% dengan pengaturan learning rate 0,00001 (pre-training) dan 0,0001 (fine-tuning). Dataset yang digunakan berisi 1.284 gambar, jauh lebih besar dari dataset dalam penelitian ini yang hanya menggunakan 200 gambar (160 pelatihan, 40 validasi). Perbedaan akurasi ini wajar mengingat adanya perbedaan kompleksitas tugas klasifikasi (dua kelas vs. sepuluh kelas), ukuran dataset, serta teknik pelatihan dan pengujian yang digunakan.

Meskipun akurasi penelitian ini lebih rendah, pencapaian akurasi sebesar 85% tetap dapat dianggap kompetitif, terutama dalam konteks pengembangan aplikasi mobile. Penggunaan MobileNetV2 memberikan efisiensi tinggi dalam hal komputasi karena arsitekturnya yang ringan dan dioptimalkan untuk perangkat dengan sumber daya terbatas. Ini menjadikannya sangat cocok untuk diimplementasikan pada aplikasi Android menggunakan TensorFlow Lite, di mana ukuran model yang kecil dan kecepatan inferensi sangat penting. Dengan demikian, MobileNetV2 tidak hanya efektif dari segi akurasi, tetapi juga efisien dalam pemrosesan, menjadikannya pilihan tepat untuk klasifikasi gambar jamur berbasis mobile.

#### 4. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi Android untuk identifikasi jamur menggunakan algoritma *Convolutional Neural Network* berbasis arsitektur MobileNetV2, yang mencapai akurasi sebesar 85% dalam pengujian terhadap gambar uji. Hasil ini menunjukkan bahwa model mampu mengklasifikasikan berbagai jenis jamur secara cukup akurat, bahkan pada gambar yang tidak termasuk dalam data pelatihan.

Secara praktis, aplikasi ini berpotensi membantu masyarakat dalam mengidentifikasi jamur beracun atau yang dapat dikonsumsi, terutama bagi mereka yang tinggal di daerah dengan risiko tinggi terhadap salah konsumsi jamur. Keunggulan MobileNetV2 yang ringan dan efisien juga menjadikannya ideal untuk digunakan pada perangkat mobile.

Penelitian selanjutnya disarankan untuk meningkatkan akurasi model dengan menambah jumlah dan variasi dataset, serta mengoptimalkan fitur dan antarmuka pengguna (UI/UX) aplikasi agar lebih informatif dan mudah digunakan oleh masyarakat umum.

#### DAFTAR PUSTAKA

- [1] U. Sri Rahmadhani and N. Lysbetti Marpaung, "Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN," *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, vol. 8, no. 2, May 2023, doi: <https://doi.org/10.30591/jpit.v8i2.5229>.
- [2] U. : Beberapa *et al.*, "BEBERAPA JAMUR LIAR YANG BERPOTENSI MENYEBABKAN KERACUNAN DI INDONESIA," *J Biol (Denpasar)*, vol. 18, no. 1, pp. 42–56, 2025, doi: [10.15408/kauniyah.v18i1.34781](https://doi.org/10.15408/kauniyah.v18i1.34781).
- [3] I. P. Putra and R. Hermawan, "Identifikasi Jamur Beracun *Clitocybe sp.* di Gresik, Indonesia (Studi Kasus)," *Media Penelitian dan Pengembangan Kesehatan*, vol. 31, no. 2, pp. 119–124, Aug. 2021, doi: [10.22435/mpk.v31i2.4352](https://doi.org/10.22435/mpk.v31i2.4352).
- [4] I. Permana Putra Divisi Mikologi, D. Biologi, I. Pertanian Bogor, G. Biologi Jl Agatis Kampus IPB Darmaga, and J. Barat, "Kasus-Kasus Keracunan *Chlorophyllum cf. molybdites* di Indonesia," Mar. 2021. doi: <https://doi.org/10.36987/jpbn.v7i1.1984>.
- [5] A. Rizal, I. I, and T. N. Suharsono, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Citra Jamur Berbasis Mobile," *INNOVATIVE: Journal Of Social Science Research*, vol. 3, pp. 864–875, Oct. 2023, Accessed: May 14, 2025. [Online]. Available: <https://j-innovative.org/index.php/Innovative/article/view/4935>
- [6] E. Iedfitra Haksoro and A. Setiawan, "Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network," Online, May 2021. doi: <https://doi.org/10.31961/eltikom.v5i2.428>.
- [7] S. Alden and B. N. Sari, "Implementasi Algoritma CNN Untuk Pemilahan Jenis Sampah Berbasis Android Dengan Metode CRISP-DM," *Jurnal Informatika*, vol. 10, no. 1, pp. 62–71, Mar. 2023, doi: [10.31294/inf.v10i1.14985](https://doi.org/10.31294/inf.v10i1.14985).
- [8] Norfajrina, Istiqamah, and Sari Indriyani, "Jenis-Jenis Jamur (Fungi) Makroskopis Di Desa Bandar Raya Kecamatan Tamban Catur," 2021, doi: <https://doi.org/10.18592/ak.v1i1.5156>.
- [9] N. Fajria, dan Yulindon, J. Teknik Elektro, P. Negeri Padang, J. Limau Manih Padang, and C. Author, "Deteksi Jamur Beracun dan Tidak Beracun Menggunakan CNN dan YOLO," 2024, doi: [10.30630/eji.0.0.453](https://doi.org/10.30630/eji.0.0.453).
- [10] R. Juan, H. Butar-Butar, and N. Lysbetti Marpaung, "Deep Learning untuk Identifikasi Daun Tanaman Obat Menggunakan Transfer Learning MobileNetV2," vol. 8, no. 2, 2023, doi: <https://doi.org/10.30591/jpit.v8i2.5217>.
- [11] F. Paramudita and M. I. Zulfa, "Aplikasi Android Pendeteksi Kualitas Beras Berbasis Machine Learning Menggunakan Metode Convolutional Neural Network," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 3, no. 7, pp. 297–305, Aug. 2023, doi: [10.52436/1.jpti.310](https://doi.org/10.52436/1.jpti.310).
- [12] R. Roslidar, M. R. Syahputra, R. Muharar, and F. Arnia, "Adaptasi Model CNN Terlatih pada Aplikasi Bergerak untuk Klasifikasi Citra Termal Payudara," *Jurnal Rekayasa Elektrika*, vol. 18, no. 3, Sep. 2022, doi: [10.17529/jre.v18i3.8754](https://doi.org/10.17529/jre.v18i3.8754).
- [13] E. Yohanes Meol, D. Nababan, Y. P. K. Kelen, P. Studi, T. Informasi, and U. Timor, "Sistem Informasi Penjualan Ikan pada Kefamenanu Berbasis Android Menggunakan Metode Waterfall," 2023. doi: <https://doi.org/10.58982/krisnadana.v3i2.527>.

- [14] J. Shadiq, A. Safei, R. Wahyudin Ratu Loly, C. sitasi, L. Rwr, and P. Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing, “Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing,” *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 97–110, Jun. 2021, doi: <https://doi.org/10.51211/imbi.v5i2.1561>.
- [15] M. Wasil, “Pengaruh Epoch pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture,” *Jurnal Informatika dan Teknologi*, vol. 5, no. 1, p. 53, Jan. 2022, doi: [10.29408/jit.v5i1.4393](https://doi.org/10.29408/jit.v5i1.4393).