

# Rancang Bangun Aplikasi Deteksi Penyakit Tanaman Jagung Melalui Citra Daun Berbasis Android Menggunakan Algoritma *Convolutional Neural Network*

\*<sup>1</sup>Ulfa Khaira, <sup>2</sup>Indra Weni, dan <sup>3</sup>Weni Wilia

<sup>1,2,3</sup>Program Studi Sistem Informasi, Universitas Jambi, Jalan Jambi – Muara Bulian No.KM. 15, Mendalo Darat, Kec. Jambi Luar Kota, Kabupaten Muaro Jambi, Jambi

e-mail: \*<sup>1</sup>[ulfa.ilkom@gmail.com](mailto:ulfa.ilkom@gmail.com), <sup>2</sup>[indraweni@unja.ac.id](mailto:indraweni@unja.ac.id), <sup>3</sup>[weni.wilia@unja.ac.id](mailto:weni.wilia@unja.ac.id)

**Abstract** — Corn is a staple product that is important to Indonesia's development and economy. Corn disease is a major source of yield loss and lower earnings in the corn industry. Each disease demands a distinct intervention, but not all farmers know the type of disease that allows for mistakes in disease management. Convolutional Neural Network (CNN) and transfer learning can provide the best way for recognizing corn leaf disease, as both methods are known to classify objects with high accuracy. Transfer learning was chosen because it allows features and weights from prior training sessions to be reused. As a result, computation time can be reduced while accuracy is raised. In this study, an android app was developed using the CNN pre-trained models to identify corn leaf disease. The dataset includes 2600 images of corn leaf samples from three categories (Bacterial Leaf Blight, Common Rust, and healthy leaves). The model used is MobileNetV2. From the experimental results, the MobileNetV2 produced accuracy of 99% for Bacterial Leaf Blight, 100 % for Common Rust, and 100% for healthy leaves. Performance efficiency testing showed that the highest CPU and memory usage were 15.8% and 178 MB, respectively. This app is compatible with all Android versions, from 8.0 (Oreo) to 13 (Tiramisu).

**Keywords:** Convolutional Neural Network; Corn Leaf Disease; Image Classification; MobilenetV2

## 1. PENDAHULUAN

Jagung (*Zea mays*) merupakan salah satu komoditas utama di Indonesia yang memainkan peran penting dalam memenuhi kebutuhan pangan nasional dan industri pakan ternak. Tanaman ini memiliki nilai gizi tinggi dan dapat diolah menjadi berbagai produk bernilai tambah, menjadikannya sumber pendapatan vital bagi banyak petani di seluruh negeri. Seiring dengan pertumbuhan penduduk dan meningkatnya kebutuhan pangan, permintaan jagung pun terus meningkat. Hal ini mendorong kebutuhan akan stok jagung yang lebih tinggi sepanjang tahun [1]. Namun, dalam upaya meningkatkan produksi jagung, berbagai kendala dihadapi, salah satunya adalah serangan Organisme Penyakit Tumbuhan (OPT) atau hama dan penyakit tanaman. Serangan hama ini tidak hanya merusak fisik tanaman tetapi juga dapat mengakibatkan penurunan hasil panen dan bahkan gagal panen [2], yang berpotensi berdampak pada produksi nasional secara keseluruhan [3]. Tantangan ini diperburuk oleh keterbatasan akses petani terhadap teknologi pertanian modern dan modal, serta perubahan iklim yang semakin tidak menentu.

Menyadari hal tersebut, pemerintah telah melakukan berbagai upaya untuk meningkatkan produksi jagung, salah satunya melalui gerakan pengendalian OPT. Gerakan ini merupakan upaya responsif yang dilakukan secara bersama-sama untuk menurunkan populasi atau serangan OPT di lapangan. Pengendalian hama tanaman memang memiliki peran penting dalam upaya peningkatan produksi pertanian. Pelaksanaan gerakan pengendalian OPT didasarkan pada hasil pengamatan di lapangan dan rekomendasi dari Pengendali Organisme Pengganggu Tumbuhan (POPT) setempat. Tugas POPT sangatlah krusial, yaitu untuk mengenali sejak dini gejala-gejala penyakit yang menyerang tanaman[4]. Namun, jumlah petugas POPT di Indonesia masih sangat terbatas. Menurut data dari DJPT pada tahun 2022, hanya terdapat 3.548 orang POPT-PHP yang tersebar di 6.669 kecamatan di Indonesia. Idealnya, minimal ada satu POPT di setiap kecamatan, kenyataannya satu

petugas harus menangani lebih dari satu kecamatan. Kekurangan tenaga pengamat hama dan penyakit ini menjadi hambatan dalam proses pengendalian hama dan penyakit. Penyakit pada tanaman jagung dapat diamati melalui daunnya. Penyakit yang disebabkan oleh jamur, bakteri, dan virus dapat menimbulkan pola atau bercak tertentu pada daun. Namun, tidak semua petani mampu mengidentifikasi pola tersebut. Seringkali, mereka terlambat menyadari dan mengira tanamannya dalam kondisi sehat. Dari permasalahan tersebut, muncul ide untuk mengembangkan sistem yang dapat mengidentifikasi penyakit tanaman jagung dengan cepat dan mudah. Sistem ini diharapkan dapat membantu petani dalam mendeteksi penyakit pada tanaman jagung mereka secara dini, sehingga dapat dilakukan tindakan pengendalian yang tepat dan efektif.

Teknologi klasifikasi citra dan computer vision telah berkembang pesat dan kini dimanfaatkan di berbagai bidang, seperti kedokteran, bisnis, pertanian, pertahanan, dan kelautan. Dalam bidang kedokteran, *computer vision* digunakan untuk menganalisis gambar medis seperti *Magnetic Resonance Imaging (MRI)*, *Computerized Tomography (CT) scan*, dan *rontgen* untuk mendeteksi penyakit seperti kanker, penyakit jantung, dan kelainan lainnya. Algoritma berbasis AI mampu meningkatkan akurasi diagnosis dan membantu dokter dalam pengambilan keputusan klinis [5]. Di bidang pertanian, *computer vision* digunakan untuk memantau kesehatan tanaman, mendeteksi hama, dan mengoptimalkan pemupukan serta irigasi. *Drone* yang dilengkapi dengan kamera dan sensor dapat mengambil gambar lahan pertanian dan menggunakan algoritma untuk menganalisis kondisi tanaman secara *real-time* [6]. Di bidang kelautan untuk pemantauan lingkungan laut dan identifikasi spesies ikan [7]. Klasifikasi citra memungkinkan pengelompokan gambar atau foto ke dalam kategori tertentu. *Computer vision*, ibarat memberikan kemampuan manusia kepada komputer untuk memahami informasi dari gambar digital. Dengan demikian, komputer dapat mengenali objek dalam gambar seperti yang dilakukan manusia [8]. Identifikasi penyakit tanaman dapat dilakukan dengan mengamati perubahan fisik tanaman yang sakit. Oleh karena itu, *computer vision (CV)* dapat diterapkan untuk mendapatkan informasi tersebut. *Computer vision* merupakan metode yang efektif, efisien, murah, mudah digunakan, dan tidak merusak tanaman. Sistem yang dihasilkan dari *computer vision* ini dapat dijalankan pada komputer, sehingga sangat mudah digunakan oleh masyarakat untuk mendeteksi penyakit tanaman [9].

Penelitian ini akan mengenali pola penyakit tanaman jagung berdasarkan morfologi yang terjadi pada daun berdasarkan gambar digital. Penelitian ini berfokus pada bagaimana mengklasifikasikan gambar daun tanaman jagung yang terinfeksi menurut jenis penyakitnya. Beberapa penelitian terkait penelitian identifikasi penyakit tanaman pangan telah dilakukan. Penelitian yang telah dilakukan meliputi penggunaan ciri tekstur, bentuk, dan warna dalam mengidentifikasi penyakit blas, hawar daun, dan bercak daun pada tanaman padi memperoleh akurasi sebesar 70% [10]. Deteksi penyakit tanaman jagung menggunakan metode ekstraksi *Color Moment* sebagai ekstraksi ciri warna dan GLCM, algoritma KNN digunakan sebagai metode klasifikasi, dan penelitian ini dapat mengklasifikasikan daun netral (tidak berpenyakit), daun penyakit hawar daun, penyakit bercak daun, dan penyakit karat daun pada daun tanaman jagung dengan tingkat akurasi sistem terbaik yaitu 89,375% [11]. Dari penelitian deteksi penyakit tanaman berbasis citra yang telah dilakukan, perlu dilakukan segmentasi dan ekstraksi karakteristik citra sebelum memasuki tahap klasifikasi. Dalam pengolahan data yang kompleks seperti gambar dan suara, umumnya dilakukan ekstraksi fitur untuk mengubah data tersebut menjadi bentuk yang dapat dipahami dengan metode pembelajaran. Namun proses ini sangat memakan waktu dan cenderung kurang mampu menggambarkan nilai informasi data secara keseluruhan.

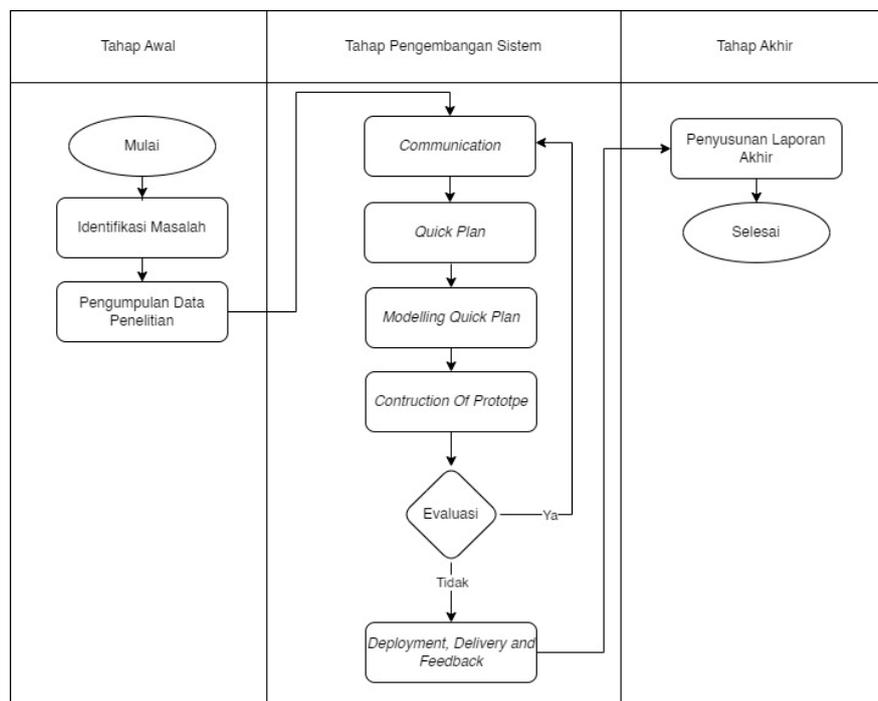
Pendekatan yang sering digunakan dalam domain *computer vision* adalah dengan menggunakan *Artificial Neural Networks (ANN)* [12]. ANN merupakan algoritma kecerdasan buatan yang dapat belajar dari data dan tidak memerlukan waktu yang lama untuk membuat modelnya. Namun ANN mempunyai kelemahan yaitu memerlukan proses segmentasi dan ekstraksi citra untuk menghasilkan kinerja yang optimal. Salah satu pengembangan ANN yang dapat menangani permasalahan tersebut adalah *Convolutional Neural Network (CNN)* [13]. CNN merupakan salah satu metode *deep learning (DL)* yang dapat diimplementasikan dalam mendeteksi suatu objek dari citra digital. Arsitektur CNN memiliki keunikan dibandingkan metode lain, termasuk segmentasi, ekstraksi fitur, dan klasifikasi dalam satu modul pemrosesan. CNN tidak memerlukan proses ekstraksi fitur khusus dan umumnya hanya memerlukan *preprocessing* dasar untuk normalisasi data [14]. Membangun model deteksi dengan algoritma CNN dapat dilakukan dengan dua teknik: 1) Membangun

dan melatih model CNN dari awal; dan 2) Menggunakan metode *transfer learning* menggunakan model yang sudah ada (siap pakai) tanpa pelatihan ulang. *Transfer learning* adalah metode yang populer digunakan dalam membangun model karena mengurangi waktu dan tenaga pengembang dalam membangun model yang diinginkan. Metodenya adalah dengan menggunakan kembali pengetahuan dari model terlatih yang ada untuk tugas lain. Metode *transfer learning* ini dapat digunakan untuk klasifikasi, regresi, dan *clustering*.

Berdasarkan penjelasan sebelumnya, penelitian ini menawarkan solusi klasifikasi penyakit daun jagung menggunakan CNN dengan *pretrained model* MobileNetV2 pada aplikasi berbasis Android. Terdapat tiga jenis gambar daun jagung yang diklasifikasi yaitu penyakit Hawar Daun, Karat Daun, dan Daun Sehat. Evaluasi kinerja akan dilakukan untuk melihat cara terbaik dalam mengenali penyakit daun jagung. Selain itu, pengujian efisiensi performa aplikasi juga dilakukan menggunakan Firebase Test Lab dengan Robo Test untuk mengetahui performa CPU dan memori pada aplikasi guna memastikan kualitas dan kinerja aplikasi.

## 2. METODOLOGI PENELITIAN

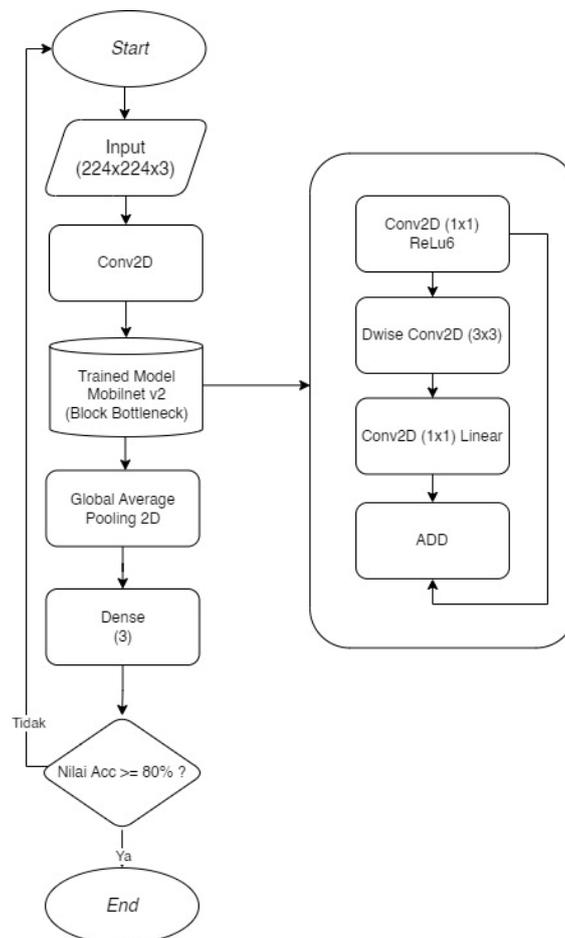
Pengembangan aplikasi yang efektif memerlukan metodologi yang terarah dan sistematis. Salah satu metode yang umum digunakan adalah *Systems Development Life Cycle* (SDLC). Metode ini memungkinkan perubahan desain dan adaptasi pada setiap tahap pengembangan. Dalam penelitian ini menggunakan model *Prototype* [15]. Model ini melibatkan interaksi aktif antara peneliti, Badan Penyuluhan Pertanian Telanaipura, dan petani dalam proses pengembangan aplikasi. Keuntungan model ini adalah mengidentifikasi risiko dan masalah awal, serta mengurangi risiko kegagalan [16]. Metode ini memberikan pengalaman sebelumnya kepada pengguna untuk perangkat lunak yang akan digunakan, serta meningkatkan dan melengkapi dengan umpan balik yang diberikan oleh pengguna. Tahapan penelitian disajikan pada Gambar 1.



Gambar 1. Tahapan Penelitian.

Pada tahapan penelitian (Gambar 1), dimulai dengan identifikasi masalah, pengumpulan data penelitian. Setelah itu, tahap pengembangan sistem menggunakan model *Prototype* yang mana model ini memiliki sifat iteratif/ berulang dimana pengguna secara aktif terlibat dalam pembuatan gambaran dan konsep sistem yang

akan diimplementasikan. Pada penelitian ini model *Prototype* memiliki beberapa tahap yaitu (1) *Communication* (Komunikasi), mencakup proses mengidentifikasi masalah petani jagung, mengumpulkan *dataset* penyakit, dan menganalisis kebutuhan sistem; (2) *Quick Plan* (Perencanaan Cepat), tahap untuk menetapkan teknologi yang akan digunakan; (3) *Modelling Quick Design* (Model Perancangan Cepat); pada tahap ini dilakukan perancangan sistem meliputi cara kerja aplikasi, pengembangan antarmuka, dan perancangan algoritma CNN yang alur kerjanya disajikan pada Gambar 2; (4) *Construction of Prototype* (Konstruksi Prototipe), tahap ini mengimplementasikan desain ke dalam kode program; (5) *Deployment, Delivery and Feedback* (Penyebaran, Pengiriman, dan Umpan Balik), *prototype* diserahkan kepada pengguna, dan diuji menggunakan metode *Blackbox Testing* dan *Test Lab Firebase*. Adapun perangkat lunak yang digunakan meliputi *Visual Studio Code*, bahasa pemrograman *Dart* dan *Python*, serta integrasi dengan *Jupyter Notebook*. Selain itu, dalam penelitian ini juga digunakan emulator *Android* dan *framework Flutter*.



Gambar 2. Perancangan Algoritma CNN.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 *Communication*

Tahapan pertama dalam metode *Prototype* adalah *Communication* (Komunikasi). Tahap ini merupakan proses pengumpulan data dan informasi yang diperlukan untuk menentukan kebutuhan program aplikasi yang akan dibangun, melalui konsultasi dan komunikasi dengan BPP Telanaipura dan petani di wilayah Kota Jambi. Pada

proses ini dilakukan wawancara terhadap petani kebun jagung untuk mendapatkan informasi penyakit tanaman jagung. Dari hasil tahapan komunikasi diketahui bahwa kurangnya pengetahuan petani mengenai hama dan penyakit pada tanaman jagung berdampak pada penanganan yang mereka lakukan. Hal ini dapat mengakibatkan meluasnya penyakit pada tanaman sehingga menyebabkan produksi jagung menurun. Hasil dari tahapan komunikasi didapatkan kesimpulan kebutuhan sistem yang disajikan pada Tabel 1.

Tabel 1. Kebutuhan Fungsional Sistem.

No	Kode	Deskripsi Kebutuhan
1	SR-1	Pengguna dapat mengenali jenis penyakit tanaman jagung dengan foto yang diambil baik dari galeri atau kamera <i>handphone</i> yang digunakan.
2	SR-2	Pengguna dapat melihat informasi rinci tentang jenis penyakit tanaman jagung yang telah diidentifikasi dan langkah-langkah yang dapat dilakukan untuk mencegah terjadinya penyakit.
3	SR-3	Pengguna dapat melihat daftar jenis penyakit tanaman jagung yang dapat dideteksi oleh sistem.

(Sumber: Hasil Tahap *Communication*)

### 3.2 Quick Plan

Tahapan kedua adalah *Quick Plan*, yaitu menetapkan teknologi yang akan digunakan. Aplikasi dikembangkan berbasis Android dengan menggunakan *framework* Flutter dengan bahasa pemrograman Dart. Pada aplikasi yang dikembangkan tertanam di dalamnya model tflite (TensorFlow Lite). Model tflite yang ditanamkan di dalam aplikasi menggunakan bahasa pemrograman Python. Algoritma klasifikasi yang digunakan adalah *Convolutional Neural Network* (CNN) dengan *pretrained model* MobileNetV2 yang diterapkan melalui teknologi *computer vision*. Algoritma tersebut diharapkan dapat memberikan solusi untuk mengatasi masalah tersebut dengan membuat aplikasi Android menggunakan metode *Prototype*.

Proses pengambilan dan pengumpulan data citra daun jagung dilakukan di kebun Jagung kelompok tani Dewa Buluran Kota Jambi. Pada tahap ini juga dilakukan validasi citra penyakit pada daun jagung oleh Bapak Burhani, A.Md selaku tenaga ahli di BPP Telanaipura. Setelah dikumpulkan, data tersebut diberi label dan dijumlahkan berdasarkan beberapa jenis kondisi daun jagung yang teridentifikasi, seperti yang dapat dilihat pada Tabel 2 berikut.

Tabel 2. Data Penelitian.

No	Jenis Daun	Gambar
1.	Sehat	

---

No	Jenis Daun	Gambar
2.	Karat Daun	
3.	Hawar Daun	

---

(Sumber: Diolah peneliti, 2022)

### 3.3 Modelling Quick Design

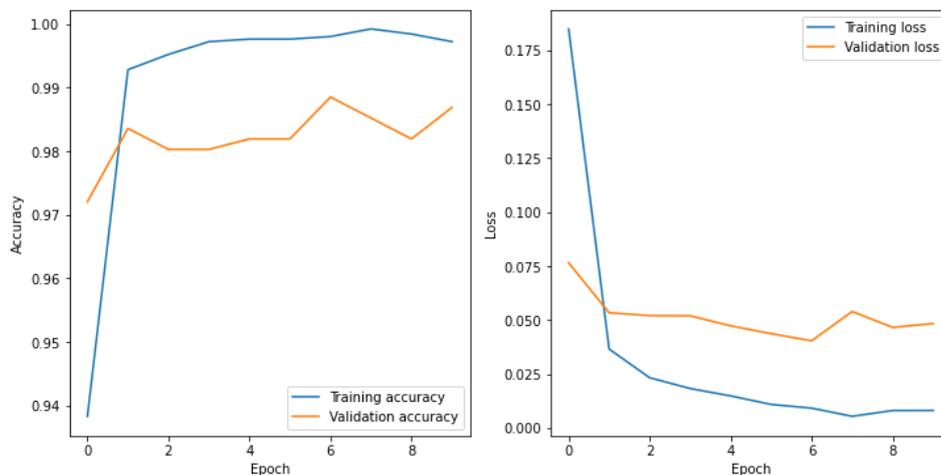
Tahapan ketiga dalam metode *Prototype* adalah *Modelling Quick Design*. Pada tahapan ini dilakukan perancangan algoritma CNN. *Dataset* secara keseluruhan berjumlah 650 citra dan dilakukan augmentasi gambar dengan pengaturan rotasi 90, 180 dan 270. Setelah proses augmentasi maka jumlah data menjadi 2600 citra. Model CNN yang ditanamkan pada aplikasi menggunakan arsitektur MobilNetV2.

*Convolutional Neural Network* (CNN) adalah salah satu jenis algoritma pembelajaran mendalam (*deep learning*) yang banyak digunakan untuk memproses, mengidentifikasi, mendeteksi, dan mengkategorikan data dalam gambar. Kemampuan CNN terletak pada pengenalan fitur penting secara otomatis tanpa bantuan manusia. CNN merupakan pengembangan lebih lanjut dari jaringan saraf tiruan (*neural network*) biasa. Arsitektur CNN terdiri dari lapisan konvolusi (*convolutional layer*) dan lapisan terhubung penuh (*fully connected layer*). Lapisan konvolusi juga memiliki beberapa lapisan *pooling* yang efektif mengurangi dimensi spasial dari input gambar.

Rancangan algoritma CNN yang digunakan dapat dilihat pada Gambar 2. Penelitian ini menggunakan *transfer learning*, teknik yang disebut *transfer learning* memanfaatkan CNN dengan model terlatih awal (*pre-trained model*) yang sudah diproses menggunakan data dalam jumlah besar. Model ini kemudian dapat mengenali tugas baru tanpa perlu pelatihan ulang dari awal [17]. Dalam *transfer learning*, beberapa lapisan memiliki arsitektur yang unik untuk mengakomodasi data baru, sementara beberapa lainnya menggunakan arsitektur yang sudah terlatih dari sumber eksternal.

Penelitian ini menggunakan model dasar yang sudah terlatih untuk mengkategorikan kumpulan data yang relatif kecil [18]. Pelatihan ulang dilakukan pada lapisan baru dengan skema pembobotan yang paling sesuai. Pada penelitian ini diterapkan *pretrained model* MobileNetV2. MobileNetV2 dirancang dengan mempertimbangkan efisiensi komputasi. Model ini menggunakan *depthwise separable convolutions* dan *linear bottlenecks* yang secara signifikan mengurangi jumlah operasi komputasi dan parameter model. Hal ini memungkinkan MobileNetV2 untuk dijalankan dengan lancar pada perangkat mobile dengan daya pemrosesan yang terbatas, seperti *smartphone* dan *tablet*. Meskipun memiliki desain yang efisien, MobileNetV2 tetap mampu menghasilkan akurasi yang tinggi dalam berbagai tugas, seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. Hasilnya, model CNN pada penelitian berhasil mencapai tingkat akurasi *training* dan

validasi 99% (Gambar 3). Dari hasil percobaan, MobileNetV2 menghasilkan akurasi sebesar 99% untuk penyakit Hawar Daun, 100 % untuk penyakit Karat Daun, dan 100% untuk Daun Sehat. Grafik perbandingan akurasi dan *loss* pada *training* dan *validation* MobileNetV2 tersedia pada Gambar 3.

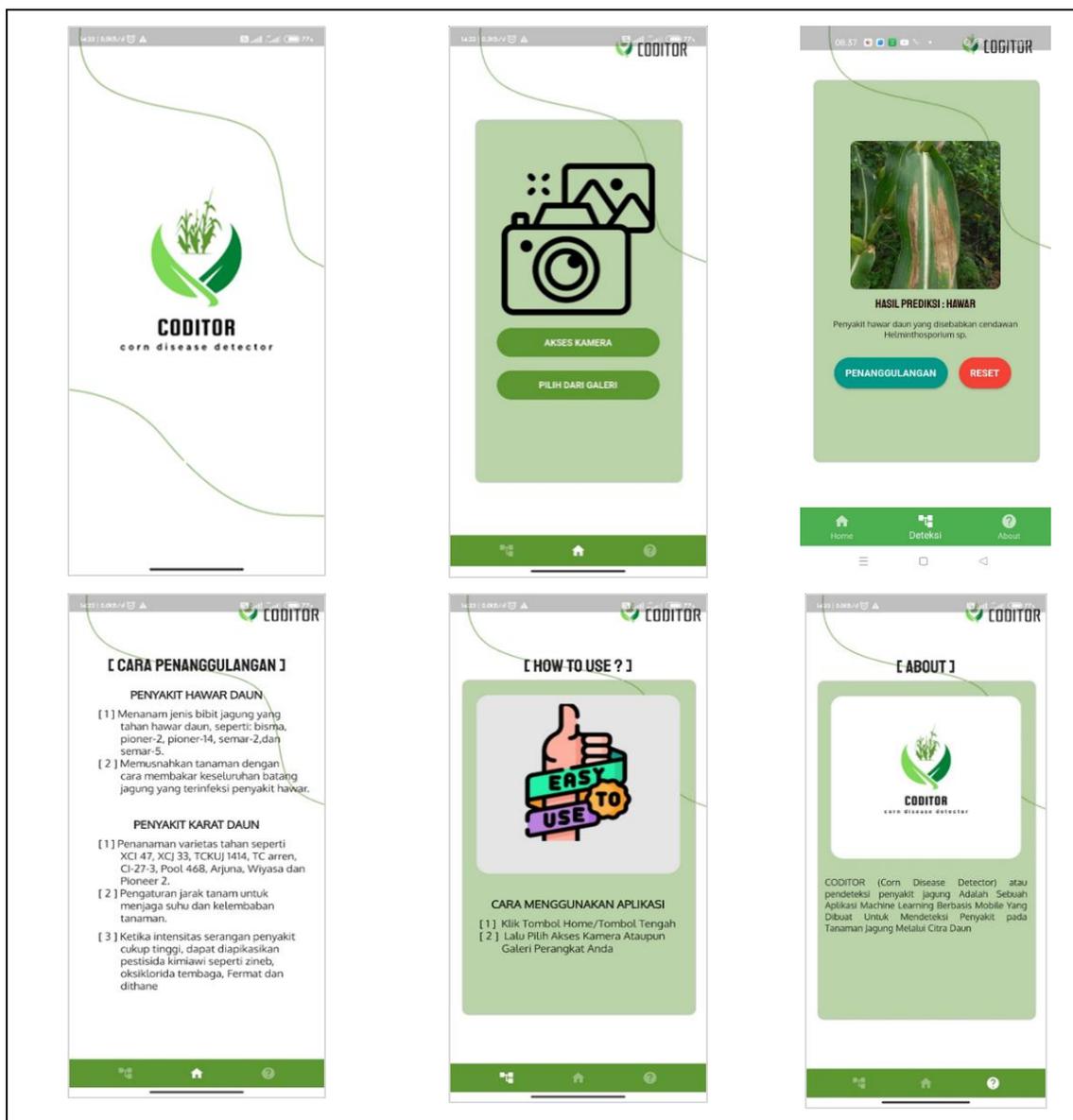


Gambar 3. Accuracy dan *loss* training MobileNetV2.

### 3.4 Construction of Prototype

Tahapan keempat adalah *Construction of Prototype*, yakni tahap mengimplementasikan aplikasi yang telah direncanakan menjadi sebuah sistem yang dapat digunakan. Model CNN terbaik yang telah diuji ditanam ke dalam aplikasi. Dalam pengembangan aplikasi ini digunakan *framework* Flutter dengan bahasa pemrograman Dart sedangkan model yang akan ditanamkan ke dalam aplikasi menggunakan *library* TensorFlow Lite dengan bahasa pemrograman Python. Aplikasi ini diberi nama Coditor (*Corn Disease Detector*) dan telah tersedia pada Google Play Store.

Aplikasi Coditor memiliki beberapa halaman, seperti *splash screen* yang menampilkan logo aplikasi, halaman *Home* yang memberikan informasi tentang penggunaan aplikasi, halaman Deteksi untuk mengunggah foto daun jagung dan memulai proses deteksi, serta halaman Hasil Klasifikasi yang menampilkan hasil deteksi penyakit. Selain itu, terdapat halaman List Penyakit yang memberikan informasi lengkap tentang penyakit jagung yang umum, serta halaman *About* yang menjelaskan tentang aplikasi secara singkat. Dengan aplikasi ini, petani jagung dapat dengan mudah mendeteksi penyakit dan mengambil tindakan yang tepat untuk menjaga kesehatan tanaman Jagung. Beberapa tampilan halaman yang tersedia di dalam aplikasi Coditor dapat dilihat pada Gambar 4.



Gambar 4. Tampilan Aplikasi Coditor.

### 3.5 Deployment, Delivery, and Feedback

Setelah pembaruan dan perbaikan aplikasi, dilakukan pengujian menggunakan metode BlackBox Testing untuk memastikan aplikasi berfungsi sesuai persyaratan fungsional.

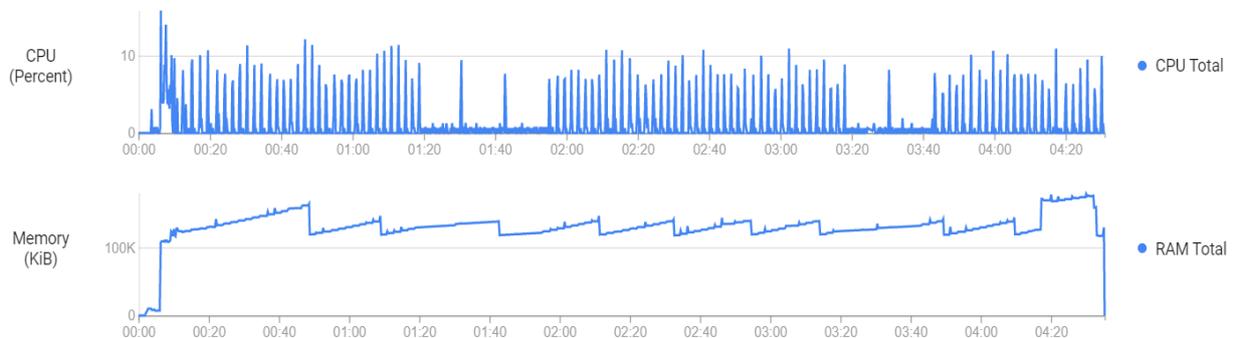
Tabel 3. Hasil Pengujian *Black-box Testing*.

No	Test case	Sesuai	Tidak Sesuai
1.	Halaman <i>splashscreen</i>	10/10	0/10
2.	Halaman <i>home</i>	10/10	0/10
3.	Halaman deteksi	10/10	0/10
4.	Tombol akses kamera	10/10	0/10

No	Test case	Sesuai	Tidak Sesuai
5.	Tombol pilih dari galeri	10/10	0/10
6.	Halaman hasil klasifikasi penyakit	10/10	0/10
7.	Tombol pengulangan	10/10	0/10
8.	Tombol <i>reset</i>	10/10	0/10
9.	Halaman list penyakit	10/10	0/10
10.	<i>Card list</i> penyakit	10/10	0/10
11.	<i>Tab</i> ciri-ciri penyakit	10/10	0/10
12.	<i>Tab</i> solusi	10/10	0/10
13.	Halaman <i>about</i> aplikasi	10/10	0/10

Sumber: Hasil Pengujian *BlackBox Testing*

Berdasarkan Tabel 3, dapat disimpulkan bahwa semua fungsi pada aplikasi Coditor berjalan dengan baik dan memenuhi harapan yang telah ditetapkan. Hasil uji *black-box testing* yang positif tersebut menunjukkan bahwa aplikasi Coditor telah memenuhi persyaratan fungsional yang telah ditetapkan, sehingga dapat dikatakan bahwa aplikasi tersebut siap untuk digunakan. Namun, untuk melakukan pengujian lanjutan dan pengembangan lebih lanjut guna memastikan kualitas dan kinerja aplikasi Coditor, dilakukan pengujian Test Lab Firebase. Adapun hasil dari pengujian menggunakan Test Lab Firebase tersedia pada Gambar 5 berikut.



Gambar 5. Penggunaan CPU dan RAM Aplikasi Coditor.

Pengujian performa menunjukkan penggunaan CPU maksimum 15,8%. Dalam penelitian yang dilakukan oleh Alviyando et al [19] dibandingkan penggunaan CPU dari beberapa perangkat. Hasilnya menunjukkan bahwa rentang penggunaan CPU berkisar antara 8,45%-28,2%. Penggunaan CPU pada aplikasi Coditor yang masih tergolong rendah dan jauh dari batas maksimum yang dapat ditoleransi. Penggunaan memori maksimum 158 MB. Kompatibilitas (*Compability*) aplikasi juga diuji pada berbagai versi Android, mulai dari versi Android 8.0 (Oreo) hingga Android 13 (Tiramisu). Berdasarkan hasil pengujian kompatibilitas dapat disimpulkan bahwa aplikasi dapat berjalan di perangkat Android 8.0 Oreo hingga Android 13 Tiramisu. Namun, perlu diperhatikan bahwa performa aplikasi mungkin dapat berbeda-beda pada setiap perangkat, tergantung dari spesifikasi *hardware* dan *software* yang digunakan.

#### 4. KESIMPULAN

Aplikasi deteksi penyakit pada tanaman jagung berbasis Android menggunakan algoritma *Convolutional Neural Network* dengan *pretrained model* MobileNetV2 telah berhasil dikembangkan. Dari hasil percobaan, MobileNetV2 menghasilkan akurasi sebesar 99% untuk penyakit Hawar Daun, 100 % untuk penyakit Karat Daun, dan 100% untuk daun sehat. Metode identifikasi penyakit tanaman jagung pada penelitian ini telah diimplementasikan pada aplikasi Coditor yang berbasis *mobile*. Pengujian efisiensi kinerja menunjukkan penggunaan CPU dan memori tertinggi masing-masing sebesar 15,8% dan 178 MB. Aplikasi ini kompatibel dengan semua versi Android, dari versi 8.0 (Oreo) hingga versi 13 (Tiramisu). Penelitian ini hanya mampu untuk mengenali citra daun tanaman jagung dalam tiga kategori, pada penelitian selanjutnya dapat menambah kelas penyakit agar aplikasi dapat mengidentifikasi lebih banyak jenis penyakit pada tanaman jagung.

#### UCAPAN TERIMA KASIH

Kegiatan penelitian ini mendapat bantuan dana DIPA PNBPFakultas Sains dan Teknologi Universitas Jambi Skema Penelitian Dasar Tahun Anggaran 2022 Nomor: SP DIPA-023.17.2.677565/2022 Tanggal 17 November 2021, sesuai dengan Surat Perjanjian Kontrak Penelitian Nomor: 257/UN21.11/PT.01.05/SPK/2022 Tanggal 17 Mei 2022. Terima kasih disampaikan kepada Badan Penyuluhan Pertanian Telanaipura atas bantuan sumber data yang diberikan, sehingga penelitian terlaksana dengan baik dan lancar.

#### DAFTAR PUSTAKA

- [1] B. Arifin, N. A. Achسانی, D. Martianto, L. K. Sari, & A. H. Firdaus, "The Future of Indonesian Food Consumption," *Jurnal Ekonomi Indonesia*, Vol. 8, No. 1, pp. 71-102, 2019, doi: 10.52813/jei.v8i1.13.
- [2] K. K. Sari, "Viral Hama Invasif Ulat Grayak (*Spodoptera frugiperda*) Ancam Panen Jagung di Kabupaten Tanah Laut Kalsel," *Jurnal Proteksi Tanaman Tropika*, Vol. 3, No. 3, pp. 244–247, 2020. doi: 10.55719/binar.v3i1.1057.
- [3] S. Syahri & R. U. Somantri, "Penggunaan Varietas Unggul Tahan Hama dan Penyakit Mendukung Peningkatan Produksi Padi Nasional," *Jurnal Penelitian dan Pengembangan Pertanian*, Vol. 35, No. 1, pp. 25-36, 2016, doi: 10.21082/jp3.v35n1.2016.p25-36..
- [4] A. Mahmud, H. Triwidodo, & A. Nurmansyah, "Analisis Kinerja Petugas Pengendali Organisme Pengganggu Tumbuhan di Kabupaten Subang," *Jurnal Ilmu Pertanian Indonesia*, Vol. 26, No. 3, pp. 392–399, 2021, doi: 10.18343/jipi.26.3.392.
- [5] R. Agustina, R. Magdalena, & N. K. C. Pratiwi, "Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, Vol. 10, No. 2, p. 446-457, 2022, doi: 10.26760/elkomika.v10i2.446.
- [6] A. Abbas, Z. Zhang, H. Zheng, M. M. Alami, A. F. Alrefaei, Q. Abbas, S. A. H. Naqvi, M. J. Rao, W. F. A. Mosa, Q. Abbas, A. Hussain, M. Z. Hassan, & L. Zhou, "Drones in Plant Disease Assessment, Efficient Monitoring, and Detection: A Way Forward to Smart Agriculture," *Agronomy*, Vol. 13, No. 6, p. 1524, 2023, doi: 10.3390/agronomy13061524.
- [7] N. Wiranda, H. S. Purba, & R. A. Sukmawati, "Survei Penggunaan Tensorflow pada Machine Learning untuk Identifikasi Ikan Kawasan Lahan Basah," *IJEIS (Indonesian J. Electronics and Instrumentations System)*, Vol. 10, No. 2, p. 179-188, 2020, doi: 10.22146/ijeis.58315.
- [8] A. Camargo & J. S. Smith, "Image Pattern Classification for the Identification of Disease Causing

- Agents in Plants,” *Computers and Electronics in Agriculture*, Vol. 66, No. 2, pp. 121-125, 2009, doi: 10.1016/j.compag.2009.01.003.
- [9] A. Vibhute & S. K. Bodhe, “Applications of Image Processing in Agriculture: A Survey,” *International Journal of Computer Applications*, Vol. 52, No. 2, pp. 34-40, 2012, doi: 10.5120/8176-1495.
- [10] G. Anthonys and N. Wickramarachchi, “An Image Recognition System for Crop Disease Identification of Paddy Fields in Sri Lanka,” in *ICIIS 2009 - 4th International Conference on Industrial and Information Systems 2009, Conference Proceedings*, 2009. doi: 10.1109/ICIINFS.2009.5429828.
- [11] I. P. Sari, B. Hidayat, & R. D. Atmaja, “Perancangan dan Simulasi Deteksi Penyakit Tanaman Jagung Berbasis Pengolahan Citra Digital Menggunakan Metode Color Moments dan GLCM,” *Prosiding SENIATI 2016*, Vol. 2, No. 2, 2016, doi: 10.36040/seniati.vi0.811.
- [12] E. M. de Oliveira, D. S. Leme, B. H. G. Barbosa, M. P. Rodarte, & R. G. F. A. Pereira “A Computer Vision System for Coffee Beans Classification Based on Computational Intelligence Techniques,” *Journal of Food Engineering*, Vol. 171, pp. 22–27, Feb. 2016, doi: 10.1016/j.jfoodeng.2015.10.009.
- [13] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, & H. Arshad, “State-of-the-art in Artificial Neural Network Applications: A Survey,” *Heliyon*, Vol. 4, no. 11, e00938, Nov. 2018, doi: 10.1016/j.heliyon.2018.e00938.
- [14] A. S. Razavian, H. Azizpour, J. Sullivan, & S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, USA, pp. 512-519, 2014. doi: 10.1109/CVPRW.2014.131.
- [15] R. Scroggins, “SDLC and Development Methodologies,” *Global Journal of Computer Science and Technology: Software & Data Engineering*, Vol. 14, No. 7, 2014.
- [16] D. Nurcahya, H. Nurfauziah, & H. Dwiatmodjo, “Comparison of Waterfall Models and Prototyping Models of Meeting Management Information Systems,” *Jurnal Mantik*, Vol. 6, No. 2, pp. 1924-1939, 2022.
- [17] A. Dalkran, A. Atakan, A. S. Rifaioğlu, M. J. Martin, R. C. Atalay, A. C. Acar, T. Dogan, & V. Atalay, “Transfer Learning for Drug-Target Interaction Prediction,” *Bioinformatics*, Vol. 39, pp. i103-i110, 2023, doi: 10.1093/bioinformatics/btad234.
- [18] M. Iman, H. R. Arabnia, & K. Rasheed, “A Review of Deep Transfer Learning and Recent Advancements,” *Technologies*, Vol. 11, No. 2, p. 40, Mar. 2023, doi: 10.3390/technologies11020040.
- [19] I. D. Alviyando, R. Munadi, & Sussi, “Integrasi Monitoring Pembayaran dan Monitoring Posisi Bis Melalui Aplikasi Android Berbasis Internet of Things,” *e-Proceedings of Engineering*, Vol. 8, No. 6, pp. 11599–11609, 2021.