

Aplikasi Klasifikasi Penyakit Pada Tanaman Jambu Air Menggunakan Metode CNN

Antonius Oktavian Jethro¹, Tony Darmanto², Riyadi J. Iskandar³

^{1,3}Informatika, Sistem Informasi², Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak
e-mail: 121421414_antonius_o_j@widyadharm.ac.id, ²tony.darmanto@yahoo.com, ³riyadi@widyadharm.ac.id

Abstract

Diseases in water apple plants can significantly reduce the quality and quantity of the harvest, thus an accurate early classification system is needed. This research aims to develop an automatic classification system to assist in the early diagnosis of diseases in water apple plants. This system uses a Convolutional Neural Network (CNN) model with the MobileNetV2 architecture optimized through fine-tuning techniques. The dataset was obtained from Kaggle, Roboflow, and direct data collection, with a distribution of 80 percent training data, 10 percent validation data, and 10 percent testing data. The model was trained to recognize nine different conditions from images of leaves and fruits, then integrated into a web application. Testing results showed very good performance, with a final accuracy of 94 percent and a balanced F1-Score of 93 percent on unseen test data. However, the model faces challenges in distinguishing diseases with high visual similarity, especially in the class of leaves with brown spots. Overall, this research successfully produced an effective and accurate classification system. The developed application has high practical potential as an early diagnosis aid to improve plant health management and reduce the potential for crop loss.

Keywords: Artificial Intelligence, Image Classification, Water Apple, Plant Diseases, MobileNetV2

Abstrak

Penyakit pada tanaman jambu air dapat secara signifikan mengurangi kualitas dan kuantitas panen, sehingga diperlukan sistem klasifikasi dini yang akurat. Penelitian ini bertujuan mengembangkan sistem klasifikasi otomatis untuk membantu diagnosis dini penyakit pada tanaman jambu air. Sistem ini menggunakan model Convolutional Neural Network (CNN) dengan arsitektur MobileNetV2 yang dioptimalkan melalui teknik fine-tuning. Dataset diperoleh dari situs Kaggle, Roboflow, dan pengumpulan data langsung, dengan distribusi 80% data pelatihan, 10 persen data validasi, dan 10 persen data pengujian. Model dilatih untuk mengenali sembilan kondisi berbeda dari gambar daun dan buah, kemudian diintegrasikan ke dalam aplikasi web. Hasil pengujian menunjukkan performa sangat baik, dengan akurasi akhir 94 persen dan F1 score seimbang 93 persen pada data uji yang belum pernah dilihat. Namun, model menghadapi tantangan dalam membedakan penyakit dengan kemiripan visual tinggi, terutama pada kelas daun dengan bercak coklat. Secara keseluruhan, penelitian ini berhasil menghasilkan sistem klasifikasi yang efektif dan akurat. Aplikasi yang dikembangkan memiliki potensi praktis tinggi sebagai alat bantu diagnosis dini untuk meningkatkan pengelolaan kesehatan tanaman dan mengurangi potensi kehilangan hasil panen.

Kata kunci: Kecerdasan Buatan, Klasifikasi Citra, Jambu Air, Penyakit Tanaman, MobileNetV2

1. PENDAHULUAN

Jambu air (*Syzygium aqueum*) merupakan salah satu tanaman buah tropis yang memiliki peran penting dalam sektor pertanian Indonesia. Tanaman ini banyak dibudidayakan karena nilai ekonominya yang tinggi, didukung oleh permintaan pasar yang terus meningkat baik di dalam maupun luar negeri. Menurut data Badan Pusat Statistik (BPS) Indonesia (2024), produksi jambu air nasional mengalami peningkatan signifikan sebesar 27 ribu ton dari tahun 2021 hingga 2023^[1]. Di Kalimantan Barat, produksi jambu air juga menunjukkan tren positif dengan kenaikan sebesar 427 ton dalam periode yang sama^[2]. Namun, di balik capaian tersebut, beberapa wilayah seperti Kapuas Hulu justru mengalami penurunan produksi sebesar 30 ton pada kurun waktu yang sama^[3]. Penurunan ini dipengaruhi oleh berbagai faktor, salah satunya adalah serangan penyakit tanaman yang dapat mengganggu pertumbuhan dan hasil panen.

Jambu air dikenal sebagai tanaman yang relatif mudah dibudidayakan karena perawatannya tidak terlalu rumit, terutama jika terhindar dari penyakit^[4]. Namun, serangan penyakit seperti bercak daun, busuk buah, atau infeksi lainnya dapat menyebabkan kerugian besar bagi petani, baik dari segi kuantitas maupun kualitas hasil panen. Klasifikasi dini penyakit tanaman menjadi sangat penting untuk mencegah penyebaran lebih lanjut dan meminimalkan dampak ekonomi. Sayangnya, metode konvensional yang masih mengandalkan observasi

langsung oleh petani atau ahli pertanian sering kali memerlukan keahlian khusus, waktu yang lama, dan tidak selalu efektif dalam mendeteksi penyakit pada tahap awal. Oleh karena itu, diperlukan solusi inovatif berbasis teknologi yang mampu mengidentifikasi penyakit tanaman secara otomatis, cepat, dan akurat.

Kemajuan teknologi kecerdasan buatan, khususnya dalam bidang visi komputer, telah membuka peluang baru untuk mengatasi tantangan tersebut. Convolutional Neural Network (CNN), salah satu algoritma deep learning, terbukti efektif dalam mengenali pola visual yang kompleks, termasuk dalam identifikasi penyakit tanaman berdasarkan citra daun dan buah. Dengan memanfaatkan CNN, sistem dapat menganalisis gambar secara mendalam untuk mendeteksi jenis penyakit dengan tingkat akurasi tinggi. Untuk mempermudah penerapan teknologi ini, perancangan platform berbasis

website menjadi solusi yang praktis dan mudah diakses. Website ini memungkinkan petani atau pengguna umum untuk mengunggah gambar daun atau buah jambu air yang diduga terinfeksi penyakit. Sistem kemudian memproses gambar tersebut menggunakan model CNN dan memberikan hasil klasifikasi secara instan, termasuk rekomendasi tindakan pengendalian yang sesuai. Dengan adanya platform ini, petani dapat mengambil keputusan yang lebih cepat dan tepat, sehingga meningkatkan efisiensi pengelolaan kesehatan tanaman, mengurangi potensi kerugian panen, dan mendukung keberlanjutan produksi jambu air di Indonesia.

2. METODE PENELITIAN

Penulis melakukan pengembangan dan pengujian sistem dengan mempelajari literatur-literatur yang relevan dengan perancangan sistem. Informasi dan data yang dikumpulkan berupa teori-teori yang mendasari permasalahan dan bidang yang diteliti. Teknik analisis sistem yang digunakan adalah teknik berorientasi objek dengan alat pemodelan *Unified Modeling Language* (UML) untuk menggambarkan proses kerja sistem yang dikembangkan. Teknik perancangan sistem dalam penelitian ini menggunakan bahasa pemrograman Python sebagai lingkungan untuk membangun sistem.

2.1 Metode Penelitian

Penulis melakukan pengembangan dan pengujian sistem dengan mempelajari literatur-literatur yang relevan dengan perancangan sistem. Informasi dan data dikumpulkan dari buku ilmiah, jurnal, skripsi, dan sumber terpercaya lainnya yang dipublikasikan di berbagai media. Tahapan penelitian dilakukan secara sistematis, mulai dari studi literatur, pengumpulan *dataset*, *preprocessing* data, pengembangan model *deep learning*, evaluasi performa model, hingga implementasi ke dalam sistem berbasis web. Penelitian ini bertujuan membangun sistem klasifikasi citra untuk identifikasi penyakit tanaman jambu air melalui citra daun dan buah menggunakan arsitektur MobileNetV2. Teknik analisis sistem yang digunakan adalah teknik berorientasi objek dengan *Unified Modeling Language* (UML) sebagai alat pemodelan sistem yang akan dibangun. Tahap penelitian ini melibatkan langkah-langkah sebagai berikut:

2.1.1 Studi Literatur

Tahap awal melibatkan kajian literatur terkait penyakit tanaman jambu air, klasifikasi citra, dan penerapan *Convolutional Neural Network* (CNN) dalam pengenalan pola visual. Referensi diambil dari jurnal ilmiah, skripsi, buku ilmiah, dan sumber terpercaya lainnya.

2.1.2 Pengumpulan Dataset

Dataset terdiri dari sembilan kelas kondisi tanaman jambu air, dengan total 5883 citra, dikumpulkan dari sumber terbuka seperti Kaggle, Roboflow, dan pengumpulan langsung. *Dataset* dibagi menjadi 80 persen data pelatihan, 10 persen data validasi, dan 10 persen data pengujian.

2.1.3 Preprocessing Data

Citra diseragamkan menjadi ukuran 256×256 *pixel* dan dinormalisasi ke skala 0 hingga 1 dengan metode $rescale=1.0/255$. Augmentasi data dilakukan menggunakan *ImageDataGenerator* dari Keras, dengan teknik rotasi, *zoom*, *shifting*, *shear*, pengaturan kecerahan, dan *flipping* horizontal untuk meningkatkan generalisasi model dan mencegah *overfitting*.

2.1.4 Pengembangan Model

Model kecerdasan buatan untuk identifikasi penyakit tanaman jambu air dikembangkan menggunakan teknologi *deep learning* dengan arsitektur CNN, khususnya MobileNetV2, yang dioptimalkan melalui teknik *fine-tuning* untuk meningkatkan akurasi klasifikasi.

2.1.5 Evaluasi Model

Performa model dievaluasi menggunakan *Confusion Matrix* dengan metrik seperti akurasi, presisi, *recall*, dan *F1-Score* untuk memastikan keandalan sistem dalam mengenali penyakit.

2.1.6 Implementasi Sistem

Model yang telah dilatih disimpan dalam format .h5, kemudian dikonversi ke format .tflite untuk diintegrasikan ke dalam sistem berbasis web menggunakan *library* TensorFlow, memungkinkan *User* mengunggah citra untuk analisis instan.

2.2 Landasan Teori

2.2.1 Kecerdasan Buatan

Kecerdasan Buatan (AI) adalah cabang ilmu komputer yang berfokus pada pengembangan sistem dan teknologi yang dapat meniru fungsi otak manusia, seperti berpikir, belajar, membuat keputusan, dan memecahkan masalah. AI bertujuan untuk menciptakan perangkat lunak dan perangkat keras yang dapat otomatis melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Dengan demikian, AI berperan dalam otomatisasi perilaku cerdas dan pengembangan sistem yang dapat beradaptasi dan mengambil keputusan secara mandiri^{[5][6]}.

2.2.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) adalah jenis khusus dari jaringan saraf tiruan mendalam (*deep neural network*) yang dirancang untuk memproses data gambar. Terinspirasi oleh cara kerja sistem visual manusia, metode utamanya adalah menggunakan operasi konvolusi untuk secara otomatis dan efektif mengekstraksi fitur-fitur penting dari sebuah gambar, seperti tepi, tekstur, dan bentuk, sehingga unggul dalam tugas analisis dan pengenalan citra^{[7][8]}.

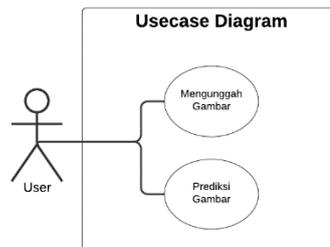
2.2.3 MobileNetV2

MobileNetV2 adalah penyempurnaan dari MobileNet dengan menggunakan struktur residual terbalik dengan lapisan *bottlenecking* tipis, konvolusi *depthwise* ringan, dan menghilangkan non-linearitas pada lapisan sempit untuk menjaga kekuatan representasi. Model ini memiliki parameter lebih sedikit, mendukung *input* di atas 32 x 32, dan performa lebih baik dengan gambar besar, sehingga efisien untuk perangkat dengan sumber daya terbatas^{[9][10]}.

2.2.4 Unified Modeling Language

Unified Modeling Language (UML) adalah standar pemodelan yang digunakan dalam pengembangan perangkat lunak berbasis *Object-Oriented* (OO). UML menyediakan notasi grafis untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem. Dengan pendekatan berbasis objek, UML membantu dalam merancang dan memahami struktur serta perilaku sistem secara lebih sistematis dan terstandarisasi^{[11][12]}.

Use case diagram merupakan representasi visual dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara *User* dan sistem dalam mencapai tujuan tertentu. Pada Gambar 1 *User* memiliki dua fungsi inti yang dilakukan secara berurutan: pertama mengunggah gambar ke dalam sistem dan kedua secara eksplisit memulai proses prediksi terhadap gambar tersebut. Diagram ini menegaskan bahwa alur kerja sistem dirancang agar lugas



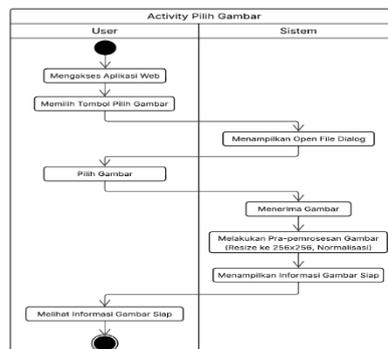
Gambar 1. Use Case Diagram Aplikasi Klasifikasi Jambu Air

dan sepenuhnya dikontrol oleh *User*, dimana prediksi hanya akan dihasilkan setelah *User* melakukan kedua tindakan utama.

Activity diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan alur kerja atau proses dalam suatu sistem secara berurutan. Diagram ini menggambarkan langkah-langkah aktivitas dan transisi yang dilakukan oleh aktor dan sistem. Pada Gambar 2 dan Gambar 3 *activity diagram* mengilustrasikan proses mulai dari *User* mengakses aplikasi, mengunggah gambar, sistem memprosesnya, hingga *User* menerima hasil klasifikasi. Diagram ini menggunakan elemen seperti titik awal (*start*), aksi (*action*), titik akhir (*end*), dan lajur renang (*swimlanes*) untuk memisahkan tanggung jawab antara *User* dan sistem. Dengan struktur yang jelas, *activity diagram* ini membantu memastikan bahwa alur kerja aplikasi logis dan sesuai dengan kebutuhan fungsional.

Sequence diagram adalah jenis diagram *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan urutan interaksi antara aktor dan objek atau komponen dalam sistem secara kronologis. Pada Gambar 4 Diagram urutan Pilih Gambar menggambarkan interaksi antara *User*, situs web Jambu Air, dan model sistem dalam fungsionalitas pemilihan gambar. Proses dimulai ketika *User* membuka situs web, lalu situs web Jambu Air berkomunikasi dengan model untuk menyiapkannya agar siap digunakan. Setelah model siap, situs web Jambu Air akan memberitahu *User* bahwa sistem sudah siap untuk menerima masukan berupa gambar. Selanjutnya, *User* memilih gambar, yang kemudian memicu situs web Jambu Air untuk melakukan pra-pemrosesan gambar dengan mengubah ukurannya menjadi 256 x 256 piksel dan normalisasi, sebelum akhirnya situs web Jambu Air mengonfirmasi kepada *User* bahwa gambar telah siap untuk langkah berikutnya.

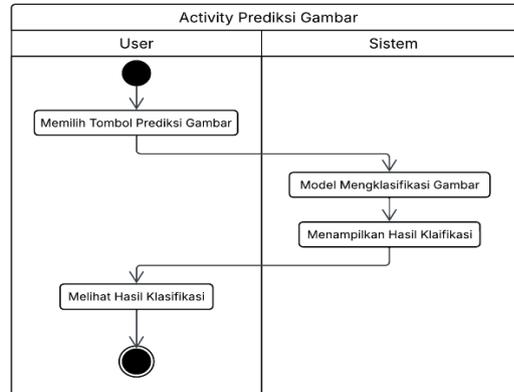
Pada Gambar 5 menjelaskan alur interaksi saat *User* melakukan prediksi gambar menggunakan *Website* Jambu Air yang terhubung dengan Model klasifikasi. Setelah gambar dipilih dan siap, proses dimulai ketika *User* memilih Prediksi Gambar, yang memicu *Website* Jambu Air untuk menginstruksikan Model agar Mengklasifikasi Gambar. Setelah klasifikasi selesai, Model mengirimkan Hasil Klasifikasi Gambar ke *Website* Jambu Air, yang kemudian menyampaikan Informasi Hasil Klasifikasi tersebut kepada *User*.



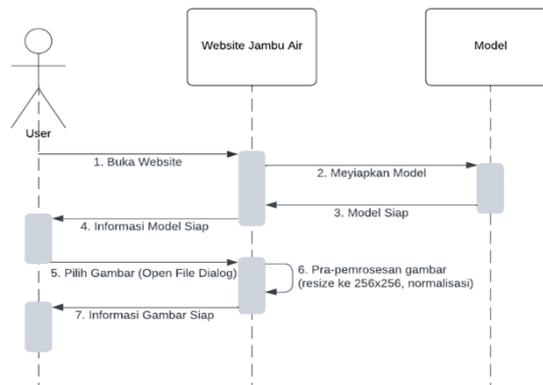
Gambar 2. Activity Diagram Pilih Gambar

2.2.5 Dataset

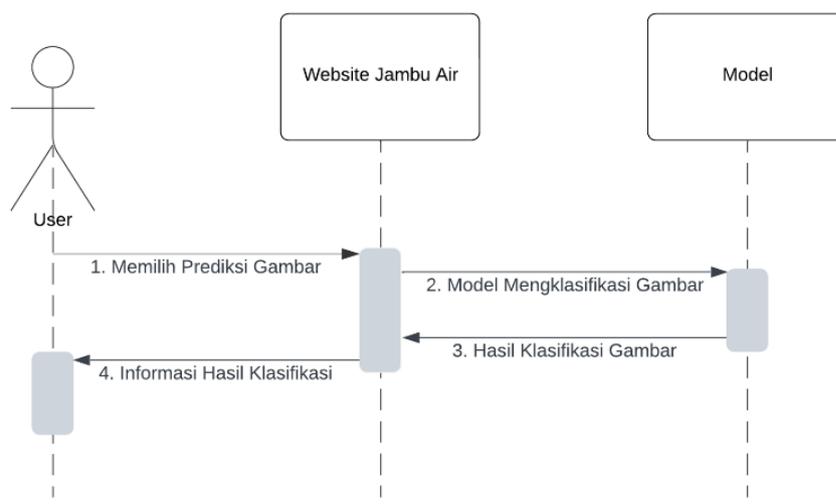
Dataset adalah kumpulan data yang berisi satu atau lebih catatan dan dapat disusun dalam bentuk tabel, dimana setiap kolom mewakili variabel dan setiap baris merepresentasikan entri data. *Dataset* menyimpan informasi yang dapat dianalisis untuk menghasilkan wawasan yang berguna^{[13][14]}. *Dataset* yang digunakan dalam penelitian ini disusun ke dalam sembilan (9) kelas yang berbeda, yang dirancang untuk mencakup kondisi visual yang paling umum dan relevan. Kelas-kelas tersebut adalah daun sehat, daun bercak hitam, daun bercak coklat, daun bercak putih, daun bercak merah, daun berlubang, buah busuk, buah sehat dan *no class*.



Gambar 3. Activity Diagram Prediksi Gambar



Gambar 4. Sequence Diagram Pilih Gambar



Gambar 5. Sequence Diagram Prediksi Gambar



Gambar 6. Contoh Gambar Pada Dataset

Gambar 6 merupakan visualisasi dari *dataset* yang mencakup sembilan (9) kelas berbeda yaitu buah busuk, buah sehat, daun bercak coklat, daun bercak hitam, daun bercak putih, daun berlubang, daun normal, dan *no class*. Sebanyak 1800 gambar atau 80 persen dari keseluruhan *dataset* digunakan untuk melatih model agar dapat mengenali pola-pola dari setiap kelas, 225 gambar atau 10 persen dari keseluruhan *dataset* untuk divalidasi, dan 225 gambar atau 10 persen dari keseluruhan *dataset* untuk diuji. Pada *dataset* setiap kelas memiliki ukuran gambar yang berbeda-beda sehingga perlu dilakukan *resize* ke ukuran 256×256 agar model dapat mengenali pola visual dengan lebih baik.

2.3 Pengembangan Model

Pengembangan program untuk aplikasi klasifikasi penyakit pada tanaman jambu air akan dibangun menggunakan *library* Tensorflow-Keras yang kemudian di konversi ke Tensorflow-Lite agar model dapat dijalankan di web *browser*. Keseluruhan *dataset* dibagi menjadi tiga himpunan data yang berbeda, yaitu dengan proporsi pembagian masing-masing sebesar 80 persen data latih, 10 persen data validasi, dan 10 persen data uji. Data latih dimanfaatkan sepenuhnya untuk melatih bobot dan bias pada model penyakit tanaman jambu air.

Selanjutnya pada tahap model arsitektur, menggunakan MobileNetV2 dengan ukuran citra *input* adalah $(256 \times 256 \times 3)$ artinya ukuran citra dengan resolusi 256×256 dan tiga (3) saluran warna (RGB). Kemudian pada tahap augmentasi, dibuat menggunakan fungsi yang tersedia di Tensorflow-Keras yaitu *ImageDataGenerator*. Augmentasi yang diterapkan adalah *rotation range* 0.4, *width shift range* 0.4, *height shift range* 0.4, *zoom range* 0.4, *horizontal flip* menjadi *true*, *shear range* 0.4, *brightness range* 0.8 hingga 1.2, dan *fill mode nearest* untuk normalisasi gambar. Augmentasi yang digunakan berguna untuk meningkatkan variasi dan jumlah *dataset* sehingga membuat model dapat belajar lebih baik.

Selanjutnya adalah *preprocessing* data dengan normalisasi nilai piksel pada setiap gambar. Hal ini dilakukan untuk mengubah skala nilai piksel dari rentang aslinya $[0, 255]$ menjadi rentang $[0, 1]$. Proses ini diterapkan pada seluruh set data (latih, validasi, dan uji) untuk memastikan konsistensi *input* dan membantu mempercepat konvergensi model selama pelatihan.

Setelah itu, pada tahap pelatihan model, lapisan-lapisan awal dari model dasar dibekukan selama tahap pelatihan awal untuk mempertahankan representasi fitur umum dari *dataset*, sementara bagian *head* dari model dilakukan pengaturan untuk menyesuaikan dengan tugas klasifikasi biner. Head dari model terdiri atas lapisan *GlobalAveragePooling2D*, dilanjutkan dengan *Dense layer* berjumlah 256 unit dan aktivasi ReLU, serta penerapan *kernel_regularizer L2 0.0001* untuk mengurangi kompleksitas model. Selanjutnya, *Batch Normalization* digunakan untuk menstabilkan dan mempercepat proses pelatihan, dan *Dropout 0.5* diterapkan sebagai teknik regularisasi untuk mencegah *overfitting*. Lapisan *output* terdiri atas satu unit neuron dengan aktivasi *sigmoid*, yang menghasilkan probabilitas 0 hingga 1.

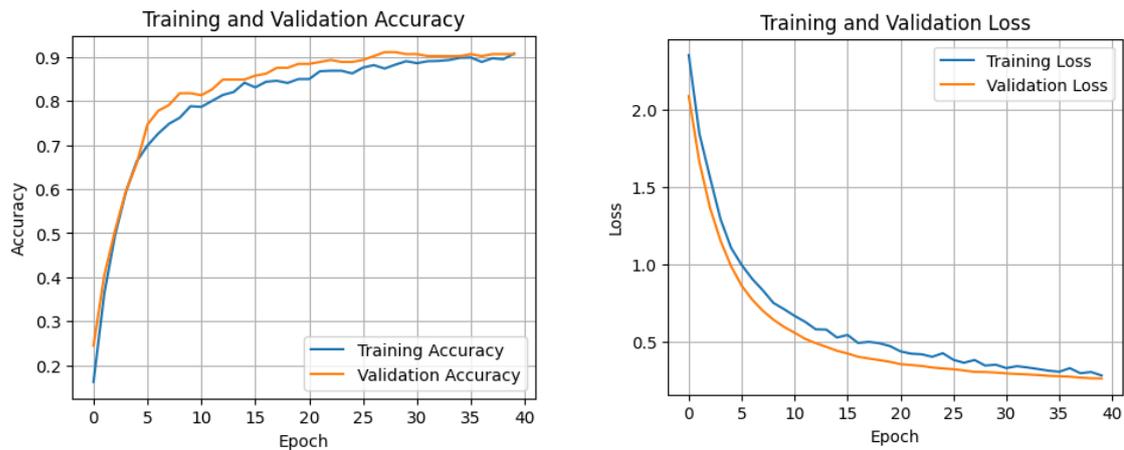
Model dikompilasi menggunakan *optimizer Adam* dengan *learning rate* sebesar 0.00001, serta fungsi *loss binary_crossentropy*, yang sesuai untuk klasifikasi biner. Pelatihan awal model dilakukan menggunakan *dataset training* yang telah diaugmentasi melalui *ImageDataGenerator* dengan *batch_size* 32 selama maksimal 20 *epoch*. Saat pelatihan dimulai, *callback EarlyStopping* akan selalu memantau metrik *val_accuracy* dengan *patience 5*, *ReduceLROnPlateau* untuk menurunkan laju pembelajaran jika *val_loss* tidak membaik, serta *ModelCheckpoint* untuk menyimpan bobot model terbaik berdasarkan nilai *val_accuracy*. Setelah pelatihan awal selesai, maka dilanjutkan pelatihan *fine-tuning* dengan membuka lapisan yang dibekukan dari MobileNetV2 dan model dikompilasi ulang dengan pengaturan yang sama. Pada tahap ini akan dilakukan pelatihan selama maksimal 20 *epoch* untuk menyesuaikan bobot pada fitur-fitur yang lebih spesifik terhadap data. Selama *fine tuning* dimulai metode *EarlyStopping*, *ReduceLROnPlateau* dan *ModelCheckpoint* juga akan diterapkan dengan pengaturan yang sama seperti pada saat pelatihan awal.

Kemudian pada tahap evaluasi model, performa model dievaluasi dengan menggunakan metrik seperti presisi, *recall*, akurasi, *F1-Score*, dan *Confusion Matrix*. Langkah terakhir, model disimpan dalam format file berjenis .h5 (HDF5). Penyimpanan ini bertujuan untuk mempertahankan hasil pelatihan model agar dapat digunakan kembali tanpa perlu melakukan pelatihan ulang.

3. HASIL DAN PEMBAHASAN

Berdasarkan Gambar 7, dapat disimpulkan bahwa proses pelatihan awal model selama 20 *epoch* pertama menunjukkan hasil yang sangat positif dan sehat. Model menunjukkan kemampuan belajar yang efektif, yang dibuktikan dengan peningkatan performa yang konsisten pada data latih maupun data validasi. Secara spesifik, akurasi pada data validasi mengalami kenaikan yang stabil, dimulai dari 24.44 persen dan berhasil melampaui 90 persen, sementara tingkat kesalahan (*validation loss*) terus menurun secara konsisten hingga mencapai titik terendahnya pada *epoch* ke-40. Pergerakan metrik yang selaras antara data latih dan data validasi ini menjadi indikator kuat bahwa model memiliki

kemampuan generalisasi yang baik dan tidak menunjukkan tanda-tanda *overfitting* yang signifikan. Celah performa yang tidak melebar antara kedua set data tersebut menegaskan bahwa model benar-benar mempelajari pola-pola yang relevan dari data, bukan sekadar menghafal.

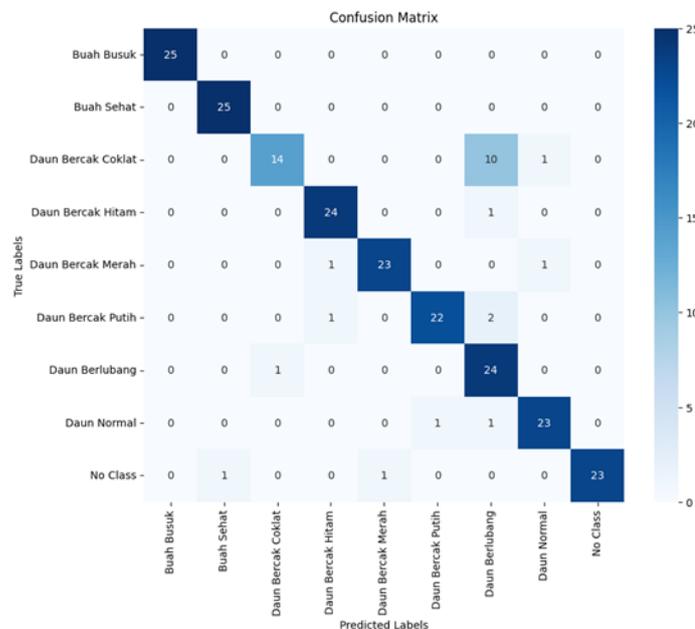


Gambar 7. Evaluasi Training dan Validation

Tabel 1 secara spesifik merangkum kinerja model pada data validasi, yang merupakan data yang digunakan untuk memonitor dan mengoptimalkan model selama proses pelatihan. Dengan nilai sebesar 0.2133 yang tergolong rendah serta metrik akurasi yang mencapai 92.4 persen, hasil ini menunjukkan bahwa model telah berhasil mencapai kondisi *good fit* pada akhir tahap pelatihannya. Keseimbangan yang sangat baik antara *precision* berada di 92 persen, sementara *recall* dan *F1-score* berada di angka 90 persen, lebih lanjut mengindikasikan bahwa performa model solid dan tidak memiliki bias terhadap metrik tertentu.

Tabel 1. Hasil Evaluasi Model Pada Data Validasi

Loss	Accuracy	Precision	Recall	F1-Score
0.2133	92.4%	92%	90%	90%



Gambar 8. Confusion Matrix Data Validasi

Berdasarkan Gambar 8 yang disajikan terhadap data validasi, dapat dianalisis bahwa model menunjukkan performa klasifikasi yang cukup baik. Dari total 225 sampel validasi, model berhasil mengklasifikasikan 212 sampel dengan benar, menghasilkan akurasi keseluruhan sebesar 92.4 persen. Angka ini mengindikasikan keseimbangan yang solid antara presisi dan *recall* di seluruh kelas. Keunggulan paling menonjol terlihat pada kelas Buah Busuk, Buah Sehat, dan No Class, yang berhasil diidentifikasi secara sempurna tanpa adanya kesalahan klasifikasi (masing-masing 25 sampel

benar). Ini juga ditunjukkan dengan nilai recall 1.00 untuk kelas-kelas tersebut.

Meskipun demikian, terdapat tantangan spesifik pada beberapa kelas. Kelas Daun Bercak Coklat menunjukkan *recall* terendah. *Confusion matrix* menjelaskan bahwa dari 25 sampel aktual Daun Bercak Coklat, 10 sampel keliru diklasifikasikan sebagai Daun Berlubang dan satu (1) sampel keliru diklasifikasikan sebagai Daun Normal. Ini menunjukkan bahwa 11 sampel dari Daun Bercak Coklat salah klasifikasi, dengan 14 sampel yang terklasifikasi benar.

Pola kebingungan dengan kelas daun berlubang juga terjadi pada kelas bercak lainnya, menandakan adanya kemiripan visual yang menjadi tantangan. Contohnya, Daun Bercak Hitam memiliki satu (1) sampel yang salah diklasifikasikan sebagai Daun Berlubang, Daun Bercak Merah memiliki satu (1) sampel salah sebagai Daun Berlubang, dan Daun Bercak Putih memiliki dua (2) sampel salah sebagai Daun Berlubang. Namun, secara keseluruhan, jumlah kesalahan ini relatif kecil dibandingkan total 225 sampel data validasi.

Tabel 2 menampilkan hasil pengujian yang merupakan tolok ukur sesungguhnya dari kemampuan model, yaitu performanya pada data uji yang sepenuhnya baru dan belum pernah dilihat sebelumnya. Pada tahap ini, model menunjukkan kinerja cukup baik, dengan nilai akurasi 92 persen dan nilai *loss* 0.2458. Peningkatan performa pada data uji ini adalah sebuah temuan yang sangat positif dan signifikan.

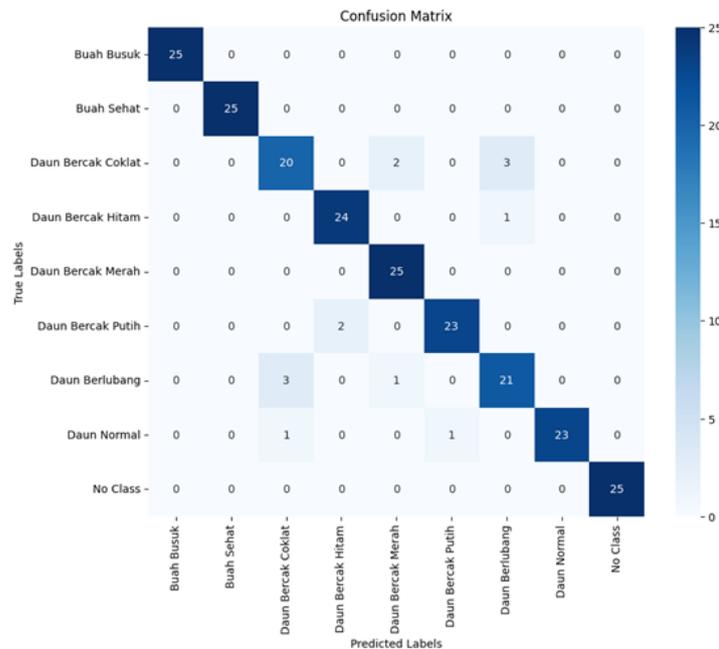
Tabel 2. Hasil Evaluasi Model Pada Data Uji

Loss	Accuracy	Precision	Recall	F1-Score
0.2458	92%	94%	94%	94%

Confusion matrix Gambar 9 terhadap data uji baru untuk mengonfirmasi kemampuan model dalam memprediksi data yang belum pernah dilihat sebelumnya. Hasil pengujian ini disajikan dalam *confusion matrix* dan menunjukkan performa yang baik, terlihat dari nilai-nilai yang sebagian besar terkonsentrasi di sepanjang diagonal utama. Secara spesifik, kelas Buah Busuk, Buah Sehat, dan No Class berhasil diklasifikasikan secara sempurna, tanpa ada satu pun kesalahan prediksi.

Meskipun demikian, teridentifikasi beberapa titik kebingungan kecil. Kesalahan klasifikasi yang paling signifikan terjadi pada kelas Daun Berlubang, di mana tiga (3) sampel keliru diprediksi sebagai Daun Bercak Coklat dan satu (1) sampel sebagai Daun Bercak Merah. Selain itu, terdapat juga kesalahan prediksi dua (2) sampel Daun Bercak Putih sebagai Daun Bercak Merah. Sementara itu, Daun Bercak Coklat pada gambar ini hanya memiliki kesalahan kecil yaitu dua (2) sampel salah diprediksi sebagai Daun Berlubang dan tiga (3) sampel sebagai Daun Normal.

Walaupun ada beberapa kesalahan klasifikasi ini, performa model secara keseluruhan menegaskan bahwa model memiliki kemampuan untuk bekerja dengan baik pada data yang belum pernah dilihat sebelumnya, serta efektif dan akurat saat dihadapkan pada data baru.

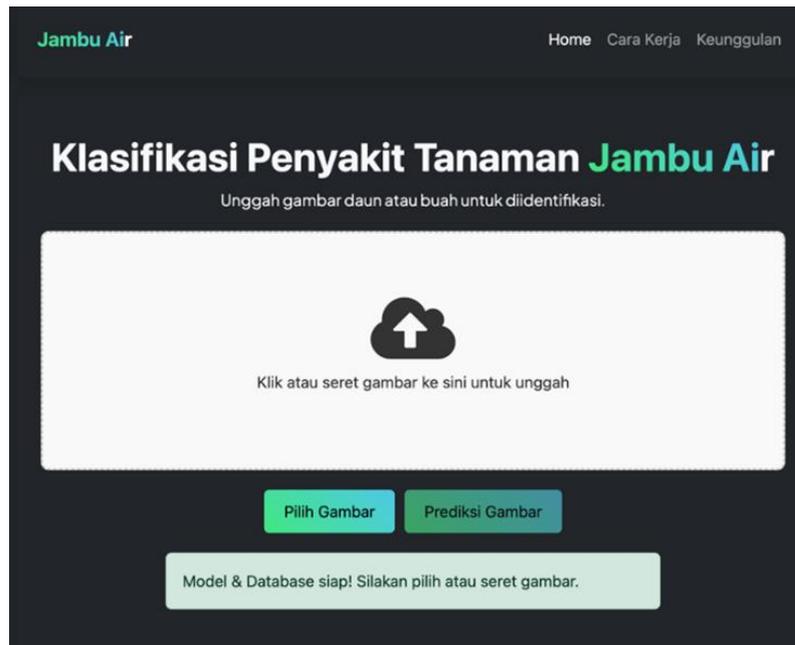


Gambar 9. Confusion Matrix Data uji

Gambar 10 menyajikan antarmuka utama (*main user interface*) dari aplikasi berbasis web yang dirancang untuk Klasifikasi Penyakit Tanaman Jambu Air. Antarmuka ini didesain secara minimalis dan intuitif untuk memandu *User* dalam proses identifikasi penyakit secara mandiri. Pada bagian tengah layar, terdapat komponen *input* utama berupa area unggah (*upload*) yang ditandai dengan ikon awan dan instruksi "Klik atau seret gambar ke sini untuk unggah".

Komponen ini menyediakan dua mekanisme interaksi standar bagi *User* untuk memasukkan gambar daun atau buah yang akan dianalisis: melalui dialog pemilihan file yang diaktifkan oleh tombol "Pilih Gambar", atau dengan metode seret-dan-lepas (*drag-and-drop*) yang lebih modern. Di bawah area unggah, terdapat dua tombol aksi

utama yang merepresentasikan alur kerja sistem: "Pilih Gambar" untuk memulai proses dan "Prediksi Gambar"

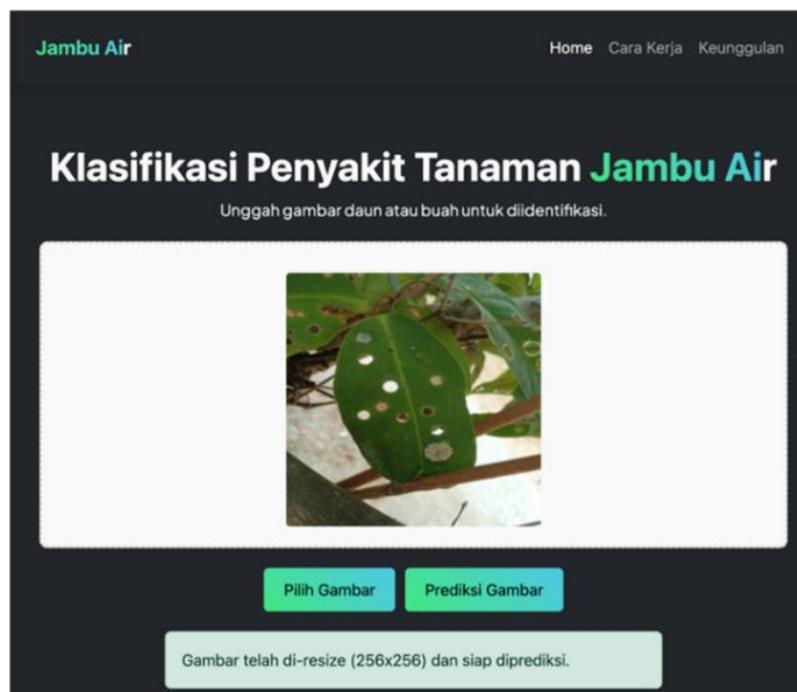


Gambar 10. Tampilan Halaman Utama Aplikasi Klasifikasi Penyakit Jambu Air

untuk mengeksekusi model klasifikasi setelah gambar berhasil dimuat.

Sebagai elemen umpan balik (*feedback*), terdapat notifikasi berwarna hijau di bagian bawah yang menyatakan "Model & Database siap! Silakan pilih atau seret gambar," yang secara efektif menginformasikan *User* bahwa sistem telah berhasil dimuat dan siap menerima *input*. Secara keseluruhan, desain antarmuka ini memprioritaskan kemudahan penggunaan dengan memfokuskan perhatian *User* pada fungsi inti, yaitu proses unggah dan prediksi gambar untuk diagnosis penyakit tanaman.

Gambar 11 secara jelas mengilustrasikan sebuah tahapan pra-pemrosesan internal yang sangat krusial. Proses ini berlangsung secara otomatis dan segera setelah *User* selesai mengunggah citra ke dalam sistem, bahkan sebelum eksekusi prediksi dapat dilakukan. Meskipun secara visual, pratinjau gambar dan tombol aksi mungkin tampak identik dengan tahapan sebelumnya, sebuah perubahan signifikan dapat diamati dengan jelas melalui pesan status yang diperbarui di bagian bawah antarmuka. Pesan status ini kini secara eksplisit menyatakan: "Gambar telah di-*resize* (256 × 256) dan siap diprediksi." Informasi ini secara lugas menginformasikan kepada *User* bahwa sistem telah berhasil melakukan langkah teknis penting, yaitu mengubah ukuran gambar yang diunggah menjadi dimensi standar 256 × 256 piksel.



Gambar 11. Tampilan Halaman Setelah Mengunggah Gambar

Standardisasi ukuran *input* ini merupakan prasyarat fundamental bagi model klasifikasi berbasis CNN, karena arsitektur model tersebut dirancang dan dilatih untuk menerima data dengan dimensi yang seragam guna memastikan konsistensi dan akurasi dalam ekstraksi fitur. Dengan memberikan umpan balik transparan mengenai proses teknis ini, sistem tidak hanya mengonfirmasi bahwa gambar siap untuk dianalisis tetapi juga secara implisit memberi edukasi mengenai salah satu aspek cara kerja model, sembari menjaga alur interaksi yang sederhana bagi *User* yang kini hanya perlu menekan tombol "Prediksi Gambar" untuk memulai klasifikasi.

Dengan memberikan umpan balik yang transparan mengenai proses teknis ini, sistem tidak hanya mengonfirmasi bahwa gambar sudah siap untuk dianalisis, tetapi juga secara implisit mengedukasi *User* mengenai salah satu aspek penting dari cara kerja model di balik layar. Ini membantu *User* memahami bahwa ada serangkaian langkah cerdas yang dilakukan sistem untuk memastikan hasil terbaik. Meskipun demikian, proses ini tetap menjaga alur interaksi yang sederhana bagi *User*, yang kini hanya perlu menekan tombol "Prediksi Gambar" untuk memulai klasifikasi. Ini menunjukkan perpaduan antara kecanggihan teknis di latar belakang dan kemudahan penggunaan di permukaan, menciptakan pengalaman yang efisien dan informatif bagi setiap *User*.



Gambar 12. Tampilan Halaman Setelah Mengklik Button Prediksi

Gambar 12 menampilkan antarmuka hasil akhir dari sistem klasifikasi, yang disajikan kepada *User* setelah proses analisis oleh model *machine learning* selesai dieksekusi. Antarmuka ini dirancang untuk menyajikan informasi secara hierarkis, dimulai dengan *output* kuantitatif primer dari model yang ditampilkan dalam kotak "Ringkasan Prediksi". Pada contoh ini, sistem mengidentifikasi citra masukan sebagai kelas daun berlubang dengan tingkat keyakinan (*confidence score*) sebesar 92.72 persen. Di bawah ringkasan tersebut, sistem menyajikan basis pengetahuan kualitatif yang terstruktur dalam sebuah kotak informasi yang detail.

Bagian yang ditampilkan berfungsi untuk menampilkan prediksi model dengan menyediakan deskripsi, kemungkinan penyebab (misalnya, serangan hama), daftar gejala yang dapat diamati secara fisik, dan yang terpenting, rekomendasi cara penanggulangan. Rekomendasi tindakan ini secara strategis dipisahkan menjadi dua kategori, yaitu pencegahan dan pengendalian, yang mengubah sistem dari alat diagnostik murni menjadi sebuah sistem pendukung keputusan (*decision-support system*) yang memberikan solusi praktis.

Terakhir, notifikasi di bagian atas layar ("Analisis selesai! Anda bisa memilih atau seret gambar lain.") secara formal mengakhiri siklus analisis saat ini dan mempersilakan *User* untuk memulai identifikasi baru. Secara keseluruhan, antarmuka hasil ini secara efektif mengintegrasikan prediksi algoritmik dengan informasi agronomis yang relevan, sehingga memberikan pengalaman kepada *User* yang komprehensif, informatif, dan dapat ditindaklanjuti.

4. KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari hasil perancangan aplikasi klasifikasi penyakit pada tanaman jambu air, yaitu sebagai berikut:

- Penerapan metode fine-tuning pada model *Convolutional Neural Network* (CNN) menunjukkan performa yang sangat efektif dalam perancangan aplikasi klasifikasi untuk kondisi kesehatan daun dan buah pada tanaman jambu air.
- Model menunjukkan adanya kesulitan dalam membedakan kelas dengan kemiripan visual yang tinggi. Hal ini

- c) terbukti pada kasus uji daun bercak hitam, dimana model mendeteksi fitur dari dua kelas sekaligus, yaitu bercak putih dan hitam.
- d) Model menunjukkan kemampuan untuk menangani gambar yang tidak relevan dengan objek penelitian. Melalui kategori no class, model berhasil mengidentifikasi dan tidak memaksakan klasifikasi pada gambar yang berada di luar cakupan kelas penyakit yang telah ditentukan.

5. SARAN

Terdapat beberapa saran pengembangan aplikasi klasifikasi penyakit pada tanaman jambu air, yaitu sebagai berikut:

- a) Memperluas dataset secara signifikan, terutama dengan menambah jumlah dan variasi gambar untuk kelas-kelas yang secara visual mirip, seperti daun bercak hitam dan daun bercak putih. Penambahan citra dengan berbagai kondisi pencahayaan, sudut pengambilan gambar, dan tingkat keparahan penyakit.
- b) Fungsionalitas sistem dapat ditingkatkan tidak hanya mengidentifikasi jenis penyakit, tetapi juga mengklasifikasikan tingkat keparahannya (misalnya, ringan, sedang, atau parah).
- c) Eksplorasi terhadap arsitektur Convolutional Neural Network (CNN) yang lebih modern atau canggih dalam mengenali pola struktur gambar daun jambu air.

6. UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih sebesar-besarnya kepada semua pihak yang telah memberikan bimbingan, petunjuk, saran, serta dorongan sehingga penelitian ini dapat diselesaikan. Ucapan terima kasih juga disampaikan kepada keluarga, terutama ayah, ibu, dan saudara-saudara, atas motivasi dan dukungan yang begitu berarti hingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik Indonesia (2024). Statistik Indonesia 2024. Badan Pusat Statistik Indonesia. <https://www.bps.go.id/publication/2024/02/28/c1bacde03256343b2bf769b0/statistik-indonesia2024.html>
- [2] Badan Pusat Statistik Kalimantan Barat (2024). Provinsi kalimantan barat dalam angka 2024. Badan Pusat Statistik Provinsi Kalimantan Barat. <https://kalbar.bps.go.id/publication/2024/02/28/866d5072b1e2ba6b2e756d07/provinsi-kalimantan-barat-dalam-angka-2024.html>
- [3] Data Luas Panen, Produksi dan Provitas Jambu Air Tahun 2019-2023. (2024). Satu Data Provinsi Kalimantan Barat. https://dev-data.kalbarprov.go.id/en_GB/dataset/data-luas-panen-produksi-dan-provitas-jambu-air-tahun-2019-2023
- [4] Aldo, Dasril. (2023). Identifikasi Dan Pengendalian Hama Tanaman Jambu Air Dengan Pendekatan Case Based Reasoning Berbasis Sistem Pakar. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(3), 491–502. <https://doi.org/10.25126/jtiik.20231036556>
- [5] Iqbal, Mohd, Januardi Nasir, Andy Supriyadi, Siti Nurhayati, Hafidzah Hafidzah, Yonky Pernando, Hurnaningsih Hurnaningsih, Feri Febria Laksana, Rusdan Rusdan, Sitti Nur Alam, Elly Agustina Julisawati, Santi Prayudani, Meryatul Husna, Yuyun Yunsida Lase, Sri Mardiyati, dan Susi Wagiyati Purtingrum. (2024). *Konsep Kecerdasan Buatan*. CV. Gita Lentera.
- [6] Yudatama, Uky, Nur Syamsiyah, Irmawati, Ade Davy Wiranata, Rahmi Imanda, Hadiansyah Ma'sum, Aris Wahyu Murdiyanto, Yahya, R. Arri Widyanto, Dhita Diana Dewi, Samuel Aleksander Mandowen, Firman Noor Hasan, Halomoan Edy Manurung, Mayko Edison Koibur, Remuz MB Kmurawak, Nur Chalik Azhar, Ivana Lucia Kharisma, Yuni Widiastuti, Kamdan, dan Rr. Isni Anisah Puspowati H. (2023). *Memahami Teknologi Informasi: Prinsip, Pengembangan, dan Penerapan*. Kaizen Media Pub
- [7] Cahyaningtyas, Christian, Mira, Candra Gudiato, Maya Sari, dan Noviyanti P. (2025). *Computer Vision untuk Pemula: Deteksi dan Analisis Ekspresi Wajah dengan CNN*. Uwais Inspirasi Indonesia.
- [8] Suparwito, Hari, Ridowati Gunawan, Iwan Binanto, Rosalia Arum Kumalasanti, dan Wiwien Widyastuti. (2023). *Pengantar pembelajaran mesin menggunakan bahasa pemrograman python*. Sanata Dharma University Press.
- [9] Keras Team. *Keras documentation: MobileNet, MobileNetV2, and MobileNetV3*. Keras. Retrieved June 1, 2025, from <https://keras.io/api/applications/mobilenet/>
- [10] Pytorch Team. *MobileNet v2 – PyTorch*. PyTorch. Retrieved June 1, 2025, from https://pytorch.org/hub/pytorch_vision_mobilenet_v2/
- [11] Rukmana, Arief Yanto, Rakhmadi Rahman, Hery Afriyadi, Dikwan Moeis, Zunan Setiawan, Nur Subchan, Lena Magdalena, Marcello Singadji, Augury El Rayeb, dan Agus Tommy Adi Prawira Kusuma. (2023). *Pengantar Sistem Informasi: Panduan praktis pengenalan sistem informasi & penerapannya*. PT. Sonpedia Publishing Indonesia.

- [12] Destriana, Rachmat, Syepry Maulana Husain, Nurdiana Handayani, Aditya Tegar Prahara Siswanto, dan Deepublish. (2022). Diagram UML Dalam membuat Aplikasi Android firebase “studi kasus Aplikasi bank sampah.” Deepublish.
- [13] Mustika, Yunita Ardilla, Abraham Manuhutu, Nazaruddin Ahmad, Imanuddin Hasbi, Guntoro, Melda Agnes Manuhutu, Mohamad Ridwan, Hozairi, Anindya Khrisna Wardhani, Syariful Alim, Ikhsan Romli, Yoga Religia, D Tri Octafian, Unggul Utan Sufandi, dan Iin Ernawati. (2021). Data Mining Dan Aplikasinya. Penerbit Widina.
- [14] Putra, Rian Rahmanda, Indra Griha Tofik Isa, dan Ahmad Bahri Joni Malyan. (2023). Buku Ajar Pengantar Deep Learning dalam Pemrosesan Citra. Penerbit NEM.