

Deteksi Penyakit Daun Tomat *Real-Time* pada Platform Android Berbasis *Convolutional Neural Network*

Eprisa Nova Rahmawati¹, Tito Pinandita^{*2}, Maulida Ayu Fitriani³, Elindra Ambar Pambudi⁴

^{1,2,3,4}Teknik Informatika, Fakultas Teknik dan Sains, Universitas Muhammadiyah Purwokerto,
Indonesia

Email: ¹epriarahma@gmail.com, ²titop@ump.ac.id, ³maulidaayuf@ump.ac.id,
⁴elindraambarpambudi@ump.ac.id

Abstrak

Tanaman tomat (*Solanum lycopersicum*) merupakan komoditas hortikultura dengan kerentanan tinggi terhadap infeksi patogen pada daun, berdampak signifikan pada kualitas dan produktivitas. Identifikasi dini penyakit daun tomat menjadi krusial untuk mencegah kerugian ekonomi, namun metode konvensional secara visual dinilai kurang efektif karena bersifat subjektif dan membutuhkan waktu lama. Penelitian ini bertujuan mengembangkan aplikasi mobile untuk mengidentifikasi penyakit daun tomat secara *real-time*, akurat, dan dapat mengetahui hasil langsung di lokasi penanaman. Sistem dikembangkan menggunakan pendekatan CNN dengan arsitektur MobileNetV2V2 yang di optimasi untuk perangkat mobile. Model dilatih menggunakan 9.600 citra daun tomat mencakup enam kategori penyakit, dikonversi ke format ONNX dan diimplementasikan ke platform Android melalui Unity dengan framework Barracuda. Evaluasi model menunjukkan performa yang sangat baik dengan akurasi pelatihan 95%. Pengujian pada 60 sampel di lingkungan nyata menghasilkan akurasi deteksi *real-time* 88,33%, dengan precision 87,5%, recall 88,3%, dan F1-score 87,9%. Aplikasi ini menawarkan solusi praktis bagi petani untuk identifikasi penyakit tanpa bergantung pada koneksi internet, memungkinkan penanganan dini yang tepat, mengurangi penggunaan pestisida berlebihan, dan berpotensi meningkatkan produktivitas tanaman tomat melalui pengendalian penyakit yang lebih efektif.

Kata kunci: Android, CNN, Klasifikasi *real-time*, MobileNetV2V2, Penyakit Daun Tomat

Real-Time Tomato Leaf Disease Detection on Android Platform Based on Convolutional Neural Network

Abstract

Tomato (*Solanum lycopersicum*) is a horticultural commodity with a high susceptibility to leaf pathogen infection, significantly impacting quality and productivity. Early identification of tomato leaf diseases is crucial to prevent economic losses, but conventional visual methods are considered less effective because they are subjective and take a long time. This research aims to develop a mobile application to identify tomato leaf diseases in *real-time*, accurately, and can know the results directly at the planting location. The system was developed using CNN approach with MobileNetV2V2 architecture optimized for mobile devices. The model was trained using 9,600 tomato leaf images covering six disease categories, converted to ONNX format and implemented to Android platform through Unity with Barracuda framework. Model evaluation showed excellent performance with 95% training accuracy. Testing on 60 samples in a real environment resulted in *real-time* detection accuracy of 88.33%, with precision of 87.5%, recall of 88.3%, and F1-score of 87.9%. This application offers farmers a practical solution for disease identification without relying on an internet connection, enabling proper early treatment, reducing the use of excessive pesticides, and potentially increasing tomato crop productivity through more effective disease control.

Keywords: Android, CNN, MobileNetV2V2, *real-time* classification, tomato leaf disease.

1. PENDAHULUAN

Sektor agraria memegang fungsi vital dalam sistem ekonomi berbagai negara sebagai penyuplai kebutuhan esensial termasuk nutrisi, farmasi, tekstil, dan menciptakan lapangan kerja bagi masyarakat[1]. Salah satu komoditas hortikultura unggulan yang banyak dibudidayakan adalah tomat (*Solanum lycopersicum*) yang memiliki nilai ekonomi dan banyak dimanfaatkan dalam sektor pangan, farmasi, hingga kosmetik [2][3]. Data Badan Pusat

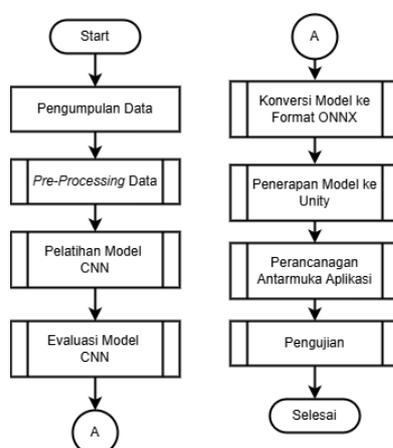
Statistik menunjukkan penurunan produksi tomat sebesar 2,14% dari 1,16 juta ton (2022) menjadi 1,14 juta ton (2023), sementara konsumsi rumah tangga meningkat 6,93%[4]. Tanaman tomat sangat rentan terhadap berbagai faktor, salah satunya adalah serangan penyakit, yang menjadi penyebab utama penurunan produksi dan dapat mengurangi hasil panen hingga 50–70%[5]. Identifikasi penyakit tanaman tomat secara konvensional melalui pengamatan visual memiliki kelemahan mendasar, seperti tingginya subjektivitas akibat ketergantungan pada pengalaman individu sehingga berisiko menghasilkan diagnosis yang keliru serta proses diagnosis yang cenderung lambat dan kurang efisien. Selain itu, kesalahan dalam penanganan dapat memperparah penyebaran penyakit di lahan pertanian. Kondisi ini menunjukkan urgensi pengembangan sistem deteksi penyakit yang objektif, cepat, dan akurat untuk diterapkan langsung di lapangan.

Convolutional Neural Network (CNN) merupakan jenis dari *deep learning* yang secara khusus dirancang untuk menganalisis data dalam bentuk gambar. CNN menawarkan keunggulan dalam pengolahan citra melalui kemampuannya untuk secara otomatis mempelajari serta mengekstraksi fitur penting dari gambar, tanpa memerlukan pra-pemrosesan manual yang rumit[6]. CNN terdiri dari beberapa lapisan, seperti *convolutional layer*, *pooling layer*, dan *fully connected layer*, yang secara bertahap mengekstrak fitur dari gambar *input* untuk melakukan klasifikasi[7]. CNN dikenal sangat efektif dalam pengenalan pola visual dan telah banyak digunakan dalam bidang pengolahan citra, termasuk untuk klasifikasi penyakit tanaman berdasarkan gambar daun, karena kemampuannya mengekstraksi fitur visual kompleks serta menghasilkan akurasi tinggi dalam membedakan jenis penyakit, seperti penelitian oleh [8] yang mengembangkan sistem klasifikasi penyakit *early blight* dan *late blight* berbasis CNN yang diimplementasikan dalam platform web menggunakan Flask dengan akurasi hingga 99%. Sejalan dengan hal tersebut, penelitian oleh [9] menerapkan model CNN untuk klasifikasi empat jenis penyakit daun tomat dan mengembangkan aplikasi desktop yang menampilkan hasil prediksi secara praktis. Meskipun demikian, aplikasi tersebut masih terbatas penggunaannya pada perangkat komputer dan belum mendukung pengolahan secara *real-time*. Sementara itu, penelitian yang dilakukan oleh [10] memanfaatkan DenseNet121 sebagai *ekstraktor* fitur untuk klasifikasi sepuluh jenis penyakit daun tomat dengan menggunakan data set berisi 10.000 gambar dan mencapai akurasi validasi sebesar 93%. Model ini dijalankan pada platform *Google Colab* namun belum diimplementasikan secara *real-time* pada perangkat *mobile*. Hasil tinjauan menunjukkan bahwa sistem klasifikasi penyakit tanaman saat ini belum mengoptimalkan pengolahan citra secara *real-time* pada perangkat *mobile*. Kondisi ini menghambat perkembangan pertanian digital di Indonesia yang memerlukan solusi praktis, portabel, dan cepat untuk mendukung pengambilan keputusan petani di lapangan. Celah penelitian ini penting untuk diisi, khususnya dalam pengembangan teknologi yang dapat digunakan secara langsung oleh petani.

Penelitian ini bertujuan mengembangkan aplikasi Android untuk klasifikasi penyakit daun tomat secara *real-time* menggunakan kamera perangkat. Sistem memanfaatkan arsitektur *Convolutional Neural Network* (CNN) untuk menganalisis bentuk, warna, dan bercak pada daun guna mengidentifikasi jenis penyakit secara cepat dan akurat di lapangan. Aplikasi ini diharapkan dapat membantu petani dalam pengambilan keputusan serta mendorong adopsi teknologi digital dalam praktik pertanian di Indonesia.

2. METODE PENELITIAN

Penelitian ini mengembangkan sistem klasifikasi penyakit daun tomat berbasis *Convolutional Neural Network* untuk aplikasi Android. Tahapan alur penelitian dirancang untuk memberikan struktur dan panduan yang jelas dalam proses pelaksanaan, sehingga tujuan yang diinginkan dapat tercapai secara sistematis. Alur penelitian tersebut disajikan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1. Persiapan Alat dan Data

Penelitian ini memanfaatkan sejumlah perangkat keras, perangkat lunak, serta himpunan data (data set) untuk mendukung proses pengembangan dan pengujian aplikasi

a. Perangkat keras

Penelitian dilakukan menggunakan *smartphone* Oppo A12 yang memiliki spesifikasi kamera utama 13MP, RAM sebesar 4GB, penyimpanan internal 64 GB, serta ditenagai oleh *chipset* Mediatek MT6765 Helio P35 (12nm).

b. Perangkat lunak

Beberapa perangkat lunak yang digunakan dalam penelitian ini antara lain:

- *Jupyter Notebook* sebagai platform pemrograman interaktif untuk pelatihan model,
- *Unity Editor* versi 2021.3.45f1 untuk pengembangan aplikasi,
- *Barracuda* versi 3.0.0 sebagai pustaka *inference* model *neural network* di *Unity*

c. Data set

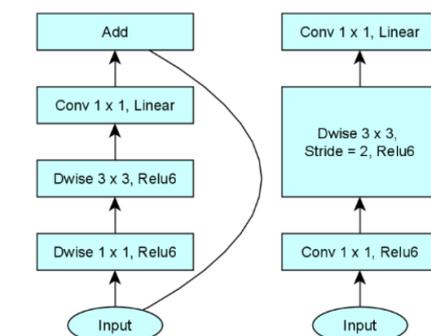
Data set yang digunakan bersumber dari platform *Kaggle* dan terdiri atas gambar daun tomat dalam kondisi sehat maupun yang mengalami infeksi berbagai penyakit. Keseluruhan citra yang digunakan mencapai 9.600 spesimen yang terbagi ke dalam enam kategori, meliputi: Bercak Bakteri Tomat, Busuk Daun Awal, Busuk Daun Akhir, Embun Tepung Tomat, Virus Keriting Daun Kuning Tomat, dan Tomat Sehat.

2.2. Pre-Processing Data

Pra-pemrosesan data diterapkan untuk menjamin bahwa data orisinal yang dimanfaatkan dalam proses pelatihan model berada dalam keadaan yang optimal dan konsisten[1 1]. Tahapan ini meliputi proses integrasi data, pelabelan ulang, transformasi format, serta pembersihan dari data yang tidak valid[12]. Data set yang digunakan dibagi menjadi tiga bagian, yakni 80% untuk pelatihan (7.680 gambar), 10% untuk validasi (960 gambar), dan 10% untuk pengujian (960 gambar), dengan distribusi yang merata di antara enam kelas penyakit, pembagian ini bertujuan menjaga proporsi yang seimbang dan representatif dalam setiap tahap pelatihan dan evaluasi model[13]. Untuk meningkatkan keanekaragaman data dan mengurangi risiko *overfitting*, dilakukan teknik *augmentasi* gambar. Proses *augmentasi* ini melibatkan beberapa transformasi seperti perubahan warna, pencerminan horizontal, rotasi, translasi, pergeseran sudut (*shearing*), serta perbesaran (*zooming*) gambar[14].

2.3. Arsitektur Model

Sistem klasifikasi dibangun menggunakan arsitektur *MobileNetV2V2*, yang memiliki struktur sederhana, efisien dalam penggunaan memori, dan dirancang khusus untuk aplikasi seluler[15]. *MobileNetV2V2* memanfaatkan teknik *depthwise separable convolution* untuk mengurangi kompleksitas dan jumlah parameter tanpa mengorbankan kinerja klasifikasi[16]. Berdasarkan data perbandingan dari Keras API[17], *MobileNetV2V2* menunjukkan keunggulan signifikan dalam hal efisiensi dibandingkan model CNN lain yang sering digunakan sebagai *baseline*. Dengan ukuran hanya 14MB (dibandingkan 88MB Xception dan 528MB VGG16), *MobileNetV2V2* mampu mencapai akurasi Top-1 71.3% dan Top-5 90.1%, setara dengan VGG16 namun dengan jumlah parameter jauh lebih sedikit yakni hanya 3.5M dibandingkan dengan 22.9M pada Xception dan 138.4M pada VGG16. Keunggulan lainnya terlihat pada waktu inferensi CPU yang hanya 25.9ms, jauh lebih cepat dibandingkan Xception (109.4ms) dan VGG16 (69.5ms). Kemampuan *MobileNetV2V2* dalam mengoptimalkan penggunaan memori memungkinkan waktu komputasi yang lebih singkat, sehingga proses klasifikasi dapat dilakukan dengan lebih cepat. Ukurannya yang kecil juga menjamin aplikasi tetap ringan dan efisien saat dijalankan pada perangkat *mobile*. Proses kerja dari arsitektur *MobileNetV2V2* ditunjukkan pada Gambar 2[18].



Gambar 2. Proses Kerja *MobileNetV2V2*

2.4. Pelatihan Model

Model dikembangkan dengan TensorFlow/Keras menggunakan pendekatan sequential, di mana *MobileNetV2V2* berperan sebagai ekstraksi fitur yang dilanjutkan dengan lapisan *GlobalAveragePooling2D* untuk efisiensi komputasional. Selanjutnya, ditambahkan lapisan Dense berisikan 128 unit dengan fungsi aktivasi ReLU, serta *Dropout* 0,5 untuk mengatasi *overfitting*. Lapisan keluaran terdiri dari 6 *neuron* dengan aktivasi *softmax* untuk kategorisasi enam jenis kondisi daun tomat. Kompilasi model menggunakan pengoptimal Adam dengan *learning rate* 0.0001 dan fungsi *loss categorical_crossentropy*, sementara akurasi dijadikan metrik evaluasi. Proses pembelajaran berlangsung hingga 50 *epoch* dengan *batch size* 32, memanfaatkan *callback EarlyStopping* untuk menghentikan proses ketika akurasi validasi tidak menunjukkan perkembangan, serta *ModelCheckpoint* untuk menyimpan bobot optimal. Model akhir disimpan dalam format .h5.

2.5. Konversi Model dan Integrasi

Pasca pelatihan, model dikonversi ke standar *Open Neural Network Exchange* (ONNX) untuk integrasi ke dalam Unity. Pemilihan Unity didasarkan pada kemampuannya menangani input visual dinamis berbasis kamera[19], memungkinkan sistem deteksi *real-time* yang memberikan hasil klasifikasi segera saat kamera diarahkan pada daun tomat. Format ONNX dipilih berdasarkan kompatibilitasnya dengan ekosistem .NET dan bahasa pemrograman C#, serta kemampuannya mengeksekusi inferensi dengan efisiensi optimal. ONNX menyediakan kompatibilitas lintas *framework* seperti PyTorch, TensorFlow, dan scikit-learn, sehingga konversi model dapat dilakukan tanpa menurunkan akurasi[20]. Format tersebut juga dikenal karena memiliki waktu respons yang singkat dan tingkat efisiensi yang optimal, menjadikannya alternatif yang sesuai untuk penerapan di berbagai platform, termasuk untuk aplikasi dalam lingkungan Unity.

Integrasi model ONNX ke Unity dilakukan melalui Barracuda, kerangka inferensi jaringan saraf yang mendukung eksekusi multi-platform pada CPU dan GPU dengan kompatibilitas penuh terhadap format ONNX[21][22][23]. Barracuda dipilih karena karakteristiknya yang meliputi efisiensi dalam proses implementasi, kapabilitas kinerja yang optimal, serta fleksibilitas adaptasi dalam konteks pengembangan sistem berbasis AI dengan fungsionalitas kamera terintegrasi[24]. Dengan sistem ini, gambar dari kamera dikonversi menjadi tensor sebagai *input* model, dan hasil klasifikasi ditampilkan langsung pada antarmuka aplikasi.

2.6. Perancangan Antarmuka Aplikasi

Pengembangan antarmuka aplikasi mengadopsi metodologi *user-centered design* (UCD), yang mengoptimalkan pengalaman *end-user* dengan menitikberatkan pada kebutuhan pengguna sesuai *behavior* mereka dalam penggunaan produk[25]. Implementasi mengutamakan kemudahan akses dan penggunaan bagi berbagai pengguna, khususnya petani tomat, melalui desain minimalis yang berfokus pada fungsionalitas inti dan navigasi intuitif agar pengguna dapat langsung memahami cara penggunaannya tanpa perlu mempelajarinya secara mendalam.

2.7. Pengujian

Evaluasi kinerja model *Convolutional Neural Network* (CNN) dilakukan melalui analisis *confusion matrix* untuk mengukur akurasi klasifikasi pada masing-masing kelas. *Confusion matrix* merupakan sebuah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan menampilkan jumlah prediksi yang benar dan salah terhadap data uji dalam bentuk matriks dua dimensi[26]. Matriks ini menyajikan perbandingan antara label aktual dan label prediksi, sehingga memudahkan dalam mengidentifikasi performa model secara keseluruhan serta kesalahan klasifikasi yang terjadi antar kelas. Matriks ini juga digunakan untuk menghitung metrik lanjutan seperti *precision*, *recall*, dan *F1-score* pada masing-masing kelas[27].

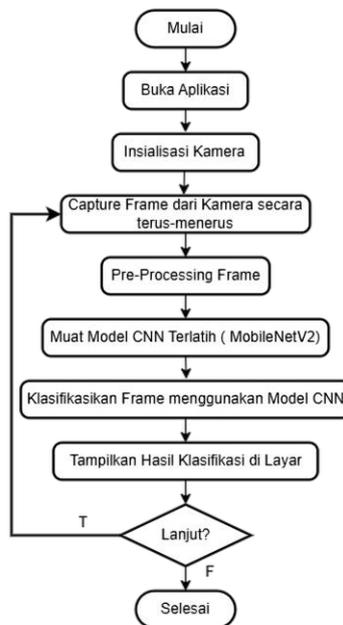
Pengujian sistem *real-time* dilakukan di perangkat seluler menggunakan 60 citra uji, dengan masing-masing 10 gambar untuk setiap kelas penyakit daun tomat. Citra uji diperoleh dari dua sumber, yaitu data sekunder dari internet dan data primer berupa gambar daun tomat yang diambil langsung dari kebun, sebagaimana ditampilkan pada Gambar 3. Proses pengujian memanfaatkan kamera internal perangkat tanpa melalui proses unggah manual, sehingga memungkinkan penilaian terhadap efektivitas sistem dalam mendeteksi penyakit daun secara langsung di lingkungan nyata.



Gambar 3. Dataset Uji Sistem

2.7. Alur Penggunaan Aplikasi

Alur kerja aplikasi dimulai dari pengambilan citra daun tomat melalui kamera, dilanjutkan dengan konversi citra menjadi *tensor*, proses inferensi menggunakan model CNN, hingga hasil klasifikasi ditampilkan pada antarmuka aplikasi, seperti ditunjukkan pada Gambar 4.



Gambar 4. Alur Penggunaan Aplikasi

3. HASIL DAN PEMBAHASAN

Hasil implementasi model pada aplikasi Android menunjukkan potensi signifikan dalam konteks praktis bagi petani. Kemampuan identifikasi *real-time* memungkinkan deteksi dini penyakit tanaman, memberikan kesempatan bagi petani untuk mengambil tindakan preventif sebelum penyakit menyebar lebih luas. Hal ini dapat mengurangi penggunaan pestisida berlebihan dan kerugian ekonomi akibat gagal panen.

3.1. Hasil Pelatihan Model

3.1.1. Confusion Matrix dan Metrik Performa per Kelas



Gambar 5. Confusion Matrix

Confusion Matrix pada Gambar 5, mengilustrasikan performa model CNN dalam melakukan identifikasi enam klasifikasi daun tomat yang berbeda yaitu Bercak Bakteri, Busuk Daun Akhir, Busuk Daun Awal, Embun Tepung, Keriting Kuning, dan kategori Sehat. Evaluasi menunjukkan bahwa model berhasil mencapai tingkat akurasi total 95%. Analisis mendalam terhadap confusion matrix mengungkapkan pola klasifikasi sebagai berikut:

- Kategori Sehat dan Keriting Kuning menunjukkan hasil klasifikasi sempurna dengan akurasi 100%, dengan semua 170 sampel Keriting Kuning dan 139 sampel Sehat terklasifikasi dengan benar, mengindikasikan bahwa kedua kelas ini memiliki fitur visual yang sangat distingtif dan mudah dikenali oleh model.
- Kategori Embun Tepung memperoleh akurasi 96%, dengan 168 dari 174 sampel terklasifikasi dengan benar, dan 6 sampel mengalami kesalahan klasifikasi (3 sebagai Bercak Bakteri dan 3 sebagai Busuk Daun Akhir).
- Busuk Daun Awal menunjukkan tantangan klasifikasi yang lebih signifikan dengan 132 dari 143 sampel terklasifikasi dengan benar, sementara beberapa sampel salah diidentifikasi sebagai Busuk Daun Akhir (4 sampel) dan Bercak Bakteri (4 sampel).
- Terdapat pola kesalahan klasifikasi yang konsisten antara Bercak Bakteri dan Busuk Daun Akhir, dimana dari 170 sampel Bercak Bakteri, 157 terklasifikasi dengan benar (92%), dan dari 164 sampel Busuk Daun Akhir, 146 terklasifikasi dengan benar (89%) dengan 6 sampel salah diidentifikasi sebagai Bercak Bakteri.

Tabel 1. Classification Report

Kategori Penyakit	Precision	Recall	F1-Score
Bercak Bakteri	0.92	0.92	0.92
Busuk Daun Akhir	0.92	0.89	0.91
Busuk Daun Awal	0.94	0.92	0.93
Embun Tepung	0.96	0.97	0.96
Keriting Kuning	1.00	1.00	1.00
Sehat	0.95	1.00	0.98

Berdasarkan metrik pada Tabel 1, analisis komprehensif performa model untuk setiap kategori penyakit dapat dijabarkan sebagai berikut:

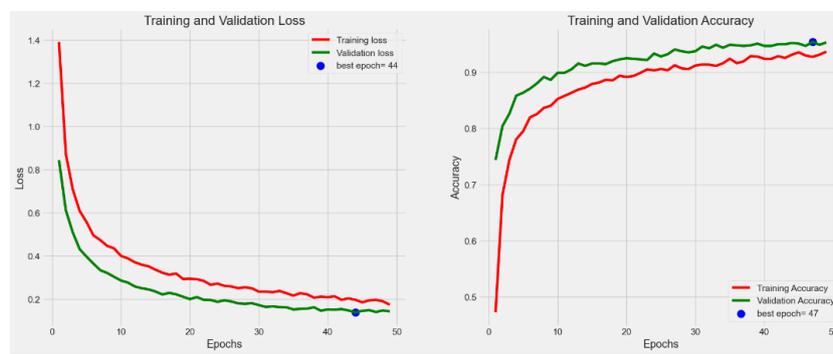
- Keriting Kuning menunjukkan performa sempurna pada semua indikator (precision 1.00, recall 1.00, F1-score 1.00), yang mengindikasikan bahwa fitur visual pada kategori ini sangat distingtif berupa daun yang menguning dan berkerut secara khas, sehingga mudah dibedakan oleh model dari kategori lainnya.

- Sehat menunjukkan performa sangat baik dengan *recall* sempurna 1.00, yang berarti model dapat mengidentifikasi seluruh sampel daun sehat tanpa ketinggalan. Nilai *precision* 0.95 menunjukkan adanya sejumlah kecil *false positive*, yaitu beberapa daun sakit ringan yang salah diklasifikasikan sebagai sehat. Secara keseluruhan, *F1-score* 0.98 menandakan keseimbangan yang sangat baik antara *precision* dan *recall*.
- Embun Tepung menunjukkan performa sangat baik dengan *precision* 0.96, *recall* 0.97, dan *F1-score* 0.96. Karakteristik visual berupa lapisan tepung putih pada permukaan daun memberikan fitur yang cukup distingtif untuk dikenali model dengan konsisten.
- Busuk Daun Awal memiliki *precision* 0.94, *recall* 0.92, dan *F1-score* 0.93. Nilai *recall* yang sedikit lebih rendah mengindikasikan tantangan dalam mengenali semua kasus positif, terutama pada tahap awal penyakit dimana gejala belum terlalu jelas.
- Bercak Bakteri dan Busuk Daun Akhir, keduanya menunjukkan performa yang relatif lebih rendah dibandingkan kategori lainnya namun tetap dalam rentang memuaskan dengan *F1-score* >0.90. Busuk Daun Akhir memiliki nilai *recall* terendah (0.89) di antara semua kategori, yang menandakan bahwa model terkadang gagal mengidentifikasi beberapa sampel positif untuk kategori ini.

Pola kesalahan klasifikasi ini dapat dijelaskan melalui kemiripan karakteristik visual antara kondisi patologis tersebut, terutama pada tahap perkembangan awal penyakit dimana gejala belum sepenuhnya spesifik. Secara khusus, Busuk Daun Awal dan Busuk Daun Akhir memiliki manifestasi visual yang berkembang secara progresif, sehingga pada beberapa kasus tanda-tanda visual dapat tumpang tindih. Begitu pula Bercak Bakteri dan Busuk Daun Akhir keduanya menampilkan pola lesi nekrotik yang pada beberapa kondisi pencahayaan dan sudut pengambilan gambar dapat terlihat serupa.

Meskipun terdapat kesalahan klasifikasi antar kategori tertentu, secara umum model mendemonstrasikan konsistensi performa yang baik dan kemampuan generalisasi yang memadai terhadap data set pengujian dengan semua kategori mencapai *F1-score* di atas 0.90.

3.1.2. Training dan Loss Accuracy



Gambar 6. Training dan Loss Accuracy

Grafik *Training dan Loss Accuracy* pada Gambar 6, menampilkan kurva *loss* dan akurasi model selama proses pelatihan dengan 50 *epoch*. Model CNN menunjukkan performa yang baik, ditandai dengan penurunan *loss* yang konsisten dan peningkatan akurasi yang signifikan pada data pelatihan maupun data validasi[28]. Analisis kurva akurasi dan *loss* selama proses pelatihan memberikan gambaran menyeluruh mengenai stabilitas dan efektivitas model:

- Penurunan Loss: Grafik sebelah kiri menunjukkan penurunan *training loss* (merah) dan *validation loss* (hijau) secara konsisten hingga sekitar epoch ke-30, setelah itu penurunan berlangsung dengan laju yang lebih lambat. Titik minimum *validation loss* tercapai pada epoch ke-44 dengan nilai 0.128, sementara *training loss* terus menurun hingga mencapai 0.086 pada epoch terakhir. Gap yang relatif kecil antara *training loss* dan *validation loss* (0.042) mengindikasikan bahwa model tidak mengalami *overfitting* yang signifikan.
- Peningkatan Akurasi: Grafik di sebelah kanan memperlihatkan peningkatan bertahap pada akurasi pelatihan dan validasi. Akurasi validasi mengalami peningkatan signifikan pada 15 *epoch* pertama (dari 70.5% menjadi 91.2%), kemudian meningkat dengan laju yang lebih moderat hingga mencapai puncaknya pada epoch ke-47 (95.0%). Sementara itu, akurasi pelatihan meningkat lebih cepat dan mencapai 97.2% pada epoch terakhir. Perbedaan antara akurasi pelatihan dan validasi yang relatif kecil (2.2%) kembali menegaskan bahwa model memiliki kemampuan generalisasi yang baik.

- Titik Konvergensi: Model menunjukkan tanda-tanda konvergensi setelah *epoch* ke-40, dimana fluktuasi akurasi validasi dan *validation loss* menjadi minimal. Hal ini mengindikasikan bahwa learning rate dan jumlah epoch yang dipilih sudah optimal untuk arsitektur dan dataset yang digunakan.
- Stabilitas Model: Tidak terlihat adanya fluktuasi ekstrem pada kurva *validation loss* maupun akurasi validasi, yang menunjukkan stabilitas proses pembelajaran dan efektivitas strategi regularisasi yang diterapkan (seperti *dropout* dan *batch normalization*).

Pemilihan 50 epoch sebagai titik optimal didasarkan pada hasil uji coba sebelumnya, di mana jumlah epoch kurang dari 50 belum mencapai akurasi yang diharapkan, sedangkan setelah *epoch* ke-47 tidak terjadi peningkatan performa yang signifikan. Penerapan teknik *early stopping* dengan parameter *patience* = 7 membantu mengidentifikasi titik optimal pelatihan dan mencegah potensi overfitting jika pelatihan dilanjutkan terlalu lama.

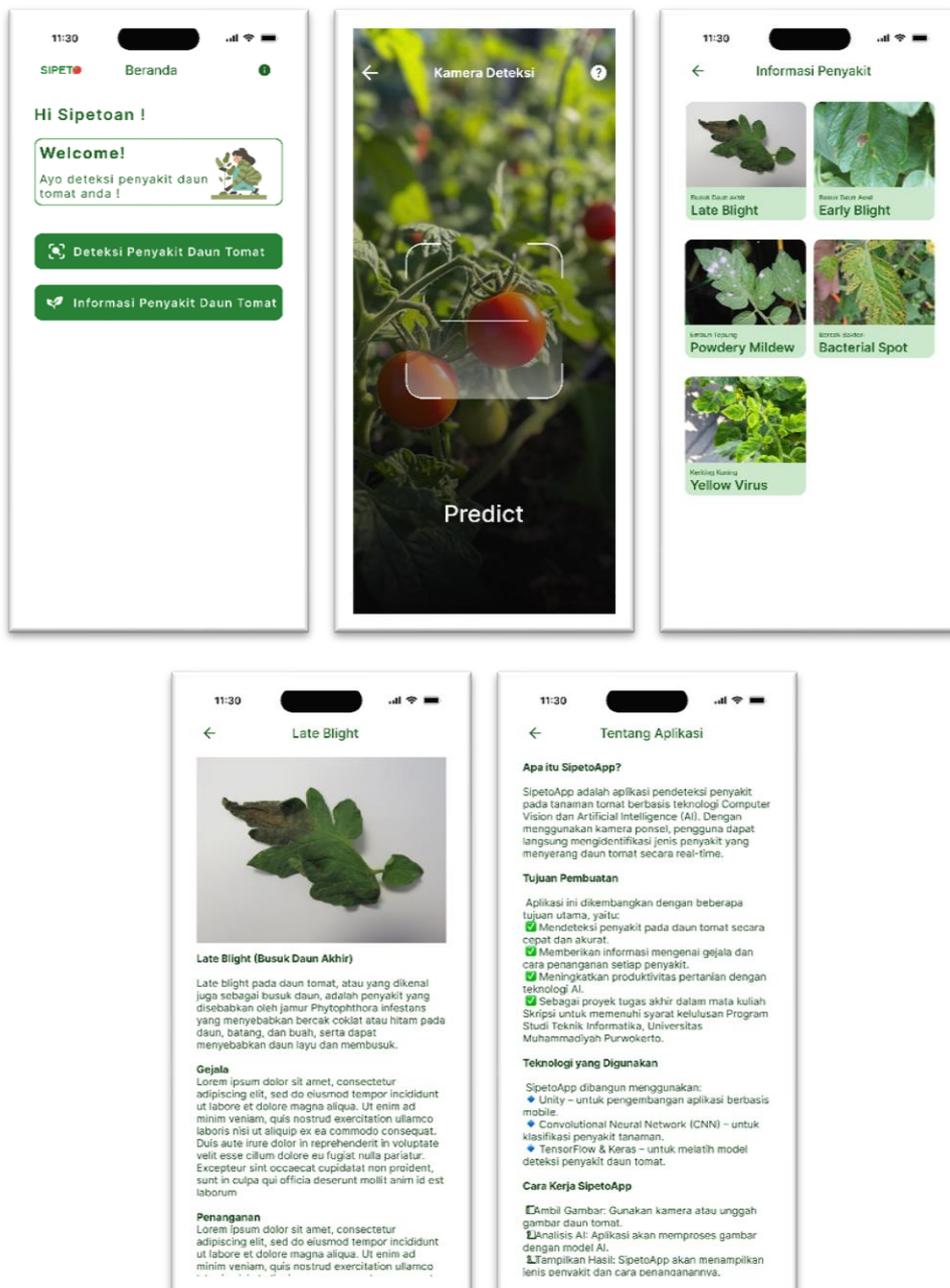
3.1.3. Perbandingan MobileNetV2V2 dengan arsitektur lainnya

Penelitian ini menggunakan MobileNetV2V2 sebagai arsitektur utama yang mendukung hasil penelitian yang dilakukan oleh[29] yang menunjukkan bahwa MobileNetV2 memiliki keunggulan berupa ukuran model yang kecil, kecepatan inferensi tinggi, dan latensi rendah, menjadikannya ideal untuk aplikasi *real-time* di perangkat dengan sumber daya terbatas. Walaupun MobileNetV2 memiliki kekurangan dalam hal akurasi yang lebih rendah dan kemampuan generalisasi yang terbatas pada dataset yang kompleks dibandingkan model lain, keunggulan dalam efisiensi komputasi dan kecepatan inferensi menjadikannya pilihan optimal untuk implementasi sistem klasifikasi penyakit tanaman pada perangkat mobile.

Berdasarkan penelitian yang dilakukan oleh[30], dalam konteks klasifikasi penyakit daun tomat, perbandingan performa menunjukkan bahwa MobileNetV2 mencapai akurasi 88% dengan penggunaan CPU 134% dan RAM 70%. Meskipun akurasi ini lebih rendah dibandingkan Inception V3 (98%) dan VGG19 (93%), MobileNetV2 memiliki keunggulan dalam penggunaan sumber daya yang lebih efisien dengan penggunaan RAM terendah dibandingkan model lainnya. Penelitian tersebut juga mencatat bahwa meskipun MobileNetV2 memiliki *precision* dan *recall* yang lebih rendah (31%), model ini tetap menjadi pilihan yang baik untuk aplikasi mobile karena ukurannya yang kecil dan kebutuhan komputasi yang ringan, menjadikannya solusi efisien untuk implementasi pada perangkat dengan sumber daya terbatas.

3.2. Implementasi Antarmuka Aplikasi

Aplikasi yang dikembangkan dalam penelitian ini dinamakan SipetoApp (Sistem Pendeteksi Daun Tomat), yang dirancang untuk membantu petani dalam mendeteksi penyakit tanaman secara cepat dan efisien. Aplikasi terdiri dari empat komponen utama, yaitu halaman beranda sebagai *landing page* dengan akses cepat ke fitur-fitur esensial aplikasi, halaman deteksi penyakit yang mengintegrasikan kamera untuk analisis *real-time* daun tomat, halaman informasi penyakit yang menyajikan database komprehensif berisi nama ilmiah, dokumentasi visual, deskripsi gejala, penyebab, serta metode penanganan dan rekomendasi pestisida, dan halaman tentang aplikasi yang memuat deskripsi singkat mengenai tujuan serta manfaat aplikasi, sebagaimana ditunjukkan pada Gambar 7.



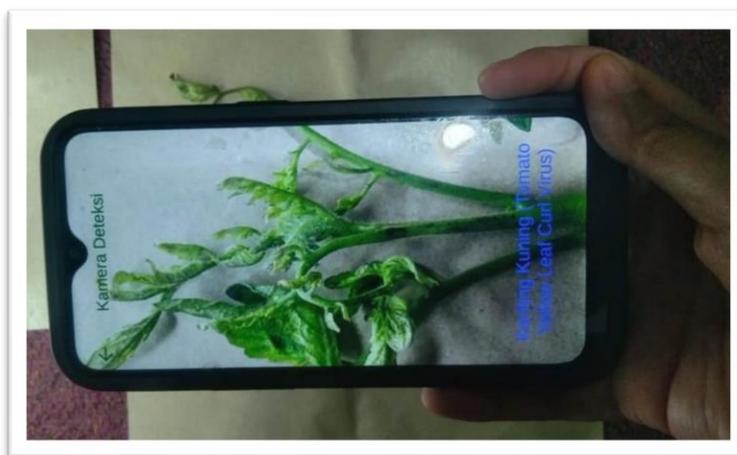
Gambar 7. Desain UI/UX Aplikasi

Pengembangan antarmuka pengguna mengimplementasikan prinsip-prinsip desain yang berfokus pada kemudahan penggunaan (*usability*) dan pengalaman pengguna (*user experience*). Warna dominan hijau dipilih untuk merefleksikan konteks pertanian dan memberi kesan alami, sementara kontras dengan warna putih mengoptimalkan keterbacaan informasi. *Layout* dirancang dengan mempertimbangkan aspek ergonomis dan efisiensi interaksi, dengan penempatan tombol navigasi yang mudah dijangkau, konsistensi desain antar halaman, dan hierarki visual yang jelas untuk membantu pengguna memahami alur aplikasi.

3.3. Pengujian Kinerja *Real-time*

Evaluasi fungsionalitas sistem diimplementasikan secara langsung pada perangkat *mobile* dengan memanfaatkan kamera bawaan, sehingga tidak memerlukan proses *upload* gambar secara manual. Proses identifikasi dapat dieksekusi dengan mengarahkan modul kamera ke spesimen daun tomat, kemudian hasil analisis ditampilkan secara otomatis pada *interface* aplikasi. Dokumentasi proses evaluasi ditunjukkan pada Gambar 8,

yang merepresentasikan tampilan antarmuka aplikasi beserta visualisasi output klasifikasi yang diproses secara *real-time* pada *display* perangkat.



Gambar 8. Pengujian Kamera *Real-Time*

Hasil pengujian menunjukkan sistem mampu melakukan klasifikasi dengan presisi tinggi dalam kondisi *real-time*. Dari total 60 spesimen yang dievaluasi, 53 spesimen teridentifikasi dengan tepat, sedangkan 7 spesimen mengalami misklasifikasi. Kalkulasi tingkat akurasi dilakukan menggunakan formulasi matematis yang direpresentasikan pada Persamaan (1), dan hasil lengkap dari pengujian tersebut ditunjukkan pada Tabel 2.

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

$$Akurasi = \frac{53}{60} \times 100\% = 88,33\%$$

Tabel 2. Hasil Uji Sistem

Jenis Penyakit	Jumlah Citra	Prediksi		Kategori yang Sering Tertukar
		Tepat	Tidak Tepat	
Bercak Bakteri	10	7	3	Busuk Daun Awal
Busuk Daun Awal	10	7	3	Busuk Daun Akhir, Bercak Bakteri
Busuk Daun Akhir	10	10	0	-
Embun Tepung	10	10	0	-
Keriting Kuning	10	10	0	-
Sehat	10	9	1	-
Total	60	53	7	

Kesalahan klasifikasi paling banyak terjadi pada kategori Bercak Bakteri dan Busuk Daun Awal, yang sering tertukar satu sama lain, serta dengan kategori Busuk Daun Akhir. Beberapa sampel Busuk Daun Awal diklasifikasikan sebagai Busuk Daun Akhir, dan sebaliknya Bercak Bakteri kerap disalahartikan sebagai Busuk Daun Awal. Hal ini mengindikasikan bahwa terdapat kemiripan fitur visual seperti pola bercak dan kerusakan jaringan daun pada ketiga kategori tersebut, sehingga menyulitkan model dalam membedakan secara konsisten.

3.4. Analisis Visual Fitur Penyakit

Analisis visual terhadap karakteristik masing-masing kategori penyakit daun tomat, sebagaimana ditampilkan pada Gambar 9 dibawah, dilakukan untuk memperoleh pemahaman yang lebih mendalam mengenai faktor-faktor yang mempengaruhi akurasi klasifikasi. Berdasarkan analisis visual, beberapa observasi penting dapat dikemukakan:

1. Bercak Bakteri dan Busuk Daun Awal memiliki karakteristik visual yang serupa, yaitu bercak nekrotik berwarna coklat kehitaman dengan ukuran kecil hingga sedang, terutama pada tahap awal infeksi. Perbedaan utama terletak pada pola distribusi bercak, di mana Bercak Bakteri cenderung memiliki pola yang lebih acak, sementara Busuk Daun Awal menunjukkan pola konsentris.
2. Busuk Daun Akhir memiliki lesi berwarna coklat tua hingga hitam yang lebih besar dan jelas dengan pola konsentris yang lebih terlihat. Pada fase lanjut, area nekrotik dapat mencakup sebagian besar permukaan daun.
3. Embun Tepung menampilkan pola visual yang sangat distingtif berupa lapisan putih seperti tepung pada permukaan daun, tanpa adanya bercak nekrotik, sehingga sangat mudah dibedakan dari kategori lainnya.
4. Keriting Kuning memiliki karakteristik berupa perubahan warna daun menjadi kuning dengan pola kerutan yang khas, tanpa adanya bercak atau lesi seperti pada penyakit lainnya.
5. Sehat menunjukkan daun dengan warna hijau yang seragam tanpa adanya bercak, lesi, atau perubahan morfologis yang mengindikasikan infeksi.



Gambar 9. Visual Penyakit

Analisis ini menjelaskan mengapa kategori Keriting Kuning, Embun Tepung, dan Sehat memiliki tingkat akurasi yang lebih tinggi, karena karakteristik visualnya yang sangat berbeda dari kategori lainnya. Sementara itu, kemiripan visual antara Bercak Bakteri, Busuk Daun Awal, dan Busuk Daun Akhir menyebabkan tantangan yang lebih besar dalam proses klasifikasi dan meningkatkan kesalahan klasifikasi antar kelas penyakit tersebut.

Selain akibat pola fitur visual daun, terdapat beberapa tantangan dan keterbatasan lain yang perlu diperhatikan. Salah satu kendala utama adalah ketergantungan sistem terhadap kualitas citra, di mana aspek seperti pencahayaan, ketajaman fokus, dan sudut pengambilan gambar sangat memengaruhi hasil klasifikasi. Selain itu, cakupan klasifikasi masih terbatas pada enam kategori penyakit daun tomat, sementara di lapangan terdapat lebih banyak varian penyakit yang belum terakomodasi. Sistem juga belum mampu mendeteksi infeksi ganda dalam satu gambar, sehingga tidak dapat mengenali keberadaan lebih dari satu jenis penyakit secara bersamaan. Variabilitas kondisi lapangan seperti bayangan, perubahan cuaca, serta pantulan cahaya turut menjadi tantangan dalam menjaga konsistensi performa sistem. Di samping itu, adanya mutasi patogen dan variasi gejala visual akibat perbedaan musim menuntut pembaruan model secara berkala agar tetap relevan. Tantangan dan keterbatasan tersebut memberikan arah bagi pengembangan selanjutnya, khususnya pada peningkatan ketahanan model terhadap kondisi lingkungan yang dinamis, perluasan basis data penyakit, serta implementasi kemampuan deteksi infeksi ganda secara simultan.

3.5. Implikasi bagi Petani

Sistem yang dikembangkan dalam penelitian ini memberikan manfaat nyata bagi petani di lapangan. Aplikasi ini dirancang untuk mengatasi beberapa permasalahan utama yang sering dihadapi petani tomat, seperti keterlambatan dalam identifikasi penyakit, kesulitan dalam membedakan gejala visual yang serupa, serta penggunaan pestisida yang tidak tepat sasaran. Melalui deteksi dini, sistem memungkinkan penanganan yang lebih tepat sesuai jenis penyakit, sehingga dapat mengurangi biaya produksi serta meminimalkan risiko resistensi patogen akibat penggunaan pestisida yang berlebihan atau tidak tepat. Secara khusus, sistem aplikasi ini memiliki beberapa keunggulan sebagai berikut:

- Efisiensi Memori: Keuntungan menghemat memori HP karena tidak perlu menyimpan foto terlebih dahulu untuk mengklasifikasikan karena real time bekerja dengan langsung memproses gambar yang tertangkap langsung oleh kamera.
- Aksesibilitas: Implementasi pada platform Android yang merupakan sistem operasi mobile paling banyak digunakan di Indonesia (87% pangsa pasar) memastikan aksesibilitas yang luas bagi petani.
- Kemudahan Penggunaan: Antarmuka aplikasi yang intuitif dengan mode kamera *real-time* memungkinkan petani mengidentifikasi penyakit tanpa perlu keahlian khusus dalam pengolahan gambar.
- Kemandirian Teknologi: Kemampuan operasi offline memungkinkan penggunaan di daerah rural dengan infrastruktur teknologi terbatas, yang merupakan karakteristik umum dari banyak area pertanian di Indonesia.
- Efisiensi Waktu: Dibandingkan dengan metode identifikasi manual atau konsultasi dengan ahli pertanian, sistem ini menawarkan penghematan waktu signifikan dalam diagnosis penyakit tanaman, yang berdampak langsung pada kecepatan respons penanganan.

4. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan aplikasi Android untuk deteksi dini penyakit daun tomat secara *real-time* menggunakan arsitektur CNN MobileNetV2V2 dengan akurasi pelatihan model sebesar 95% dan akurasi pengujian *real-time* di lapangan sebesar 88,33% untuk klasifikasi enam kategori kondisi daun tomat yaitu Bercak Bakteri, Busuk Daun Awal, Busuk Daun Akhir, Embun Tepung, Keriting Kuning, dan Sehat. Keunggulan aplikasi yang dikembangkan mencakup kemampuan deteksi langsung melalui kamera perangkat tanpa proses unggah citra, dapat beroperasi secara *offline*, antarmuka yang sederhana dan mudah digunakan, serta klasifikasi berbasis fitur visual seperti warna, bentuk, dan pola bercak.

Penggunaan arsitektur MobileNetV2V2 terbukti optimal untuk implementasi di perangkat mobile karena ukurannya yang kecil, jumlah parameter yang rendah, serta kecepatan inferensi yang tinggi, memungkinkan klasifikasi dilakukan secara efisien dan *real-time*. Analisis *confusion matrix* menunjukkan bahwa model berhasil melakukan klasifikasi sempurna untuk kelas Keriting Kuning dan Sehat, serta menghasilkan akurasi tinggi untuk Embun Tepung. Namun, tantangan masih ditemukan pada kelas dengan karakteristik visual yang mirip, seperti Bercak Bakteri, Busuk Daun Awal, dan Busuk Daun Akhir. Secara keseluruhan, sistem ini memberikan kontribusi nyata dalam membantu petani tomat mengenali penyakit lebih awal, mengambil tindakan penanganan dengan cepat dan tepat, serta mendukung efisiensi penggunaan pestisida dan pengurangan risiko kerugian produksi.

Berdasarkan kesimpulan di atas, rekomendasi pengembangan untuk penelitian selanjutnya mencakup perluasan cakupan dataset dengan memasukkan lebih banyak varietas tomat serta penerapan deteksi infeksi ganda (*multi-label classification*) guna meningkatkan akurasi klasifikasi pada kondisi nyata. Untuk memperkuat kinerja model dalam menghadapi variasi kondisi lingkungan di lapangan seperti pencahayaan yang berubah, sudut pengambilan gambar, dan gangguan visual lainnya dapat diterapkan teknik augmentasi lanjutan dan pendekatan *domain adaptation*. Selain itu, pengembangan fitur tambahan seperti rekomendasi dosis pestisida serta pelacakan riwayat penyakit dinilai penting untuk memperkaya fungsionalitas aplikasi dan meningkatkan relevansinya dalam mendukung pengambilan keputusan petani di lapangan.

DAFTAR PUSTAKA

- [1] R. Khan, N. Ud Din, A. Zaman, and B. Huang, "Automated Tomato Leaf Disease Detection Using Image Processing: An SVM-Based Approach with GLCM and SIFT Features," *J. Eng. (United Kingdom)*, vol. 2024, 2024, doi: 10.1155/2024/9918296.
- [2] M. E. Astuti and T. Achamar, "Pemanfaatan Buah Tomat Selain Sebagai Konsumsi Rumah Tangga dalam Kehidupan Sehari-hari," *J. Hulonthalo Serv. Soc.*, vol. 1, no. 1, pp. 22–27, 2022, doi: <https://doi.org/10.47918/jhss.v1i1.651>.
- [3] S. Syahara and Y. Vera, "Penyuluhan pemanfaatan buah Tomat sebagai produk kosmetik antioksidan alami di desa Manunggang Julu," *J. Educ. Dev.*, vol. 8, no. 1, pp. 21–22, 2020, doi:

- <https://doi.org/10.37081/ed.v8i1.1488>.
- [4] Badan Pusat Statistik Indonesia, “Produksi Tanaman Sayuran, 2021-2023.” <https://www.bps.go.id/id/statistics-table/2/NjEjMg==/produksi-tanaman-sayuran.html> (accessed Jan. 10, 2024).
- [5] M. Mustafa, R. W. Mwangi, Z. Szalai, N. Kappel, and L. Csambalik, “Sustainable responses to open field tomato (*Solanum lycopersicum* L.) stress impacts,” *J. Agric. Food Res.*, vol. 21, no. April 2024, 2025, doi: 10.1016/j.jafr.2025.101825.
- [6] E. Setia Budi, A. Nofrialdi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, “RESOLUSI: Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network,” *Media Online*, vol. 4, no. 5, p. 509, 2024, doi: 10.30865/resolusi.v4i5.1892.
- [7] J. Lu, L. Tan, and H. Jiang, “Review on convolutional neural network (CNN) applied to plant leaf disease classification,” *Agric.*, vol. 11, no. 8, pp. 1–18, 2021, doi: 10.3390/agriculture11080707.
- [8] Nining Putri Ningsih, Emi Suryadi, Lalu Darmawan Bakti, and Bahtiar Imran, “Klasifikasi Penyakit Early Blight Dan Late Blight Pada Tanaman Tomat Berdasarkan Citra Daun Menggunakan Metode Cnn Berbasis Website,” *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 1, no. 3, pp. 27–35, 2022, doi: 10.69916/jkbt.v1i3.10.
- [9] A. J. Bastari and A. Cherid, “Klasifikasi Penyakit Tanaman Tomat Menggunakan Convolutional Neural Network dan Implementasi Model H5 Pada Aplikasi Desktop,” *Simkom*, vol. 8, no. 2, pp. 199–207, 2023, doi: 10.51717/simkom.v8i2.194.
- [10] A. Maysela and N. Rohma, “Diagnosa Penyakit Tanaman Tomat pada Citra Daun Menggunakan Metode Convolutional Neural Network (CNN),” *JIMU J. Ilm. Multi Disiplin*, vol. 02, no. 03, pp. 555–567, 2024, [Online]. Available: <https://ojs.smkmerahputih.com/index.php/jimu/article/view/407>
- [11] F. Alghifari and D. Juardi, “Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes,” *J. Ilm. Inform.*, vol. 9, no. 02, pp. 75–81, 2021, doi: 10.33884/jif.v9i02.3755.
- [12] T. Gori, A. Sunyoto, and H. Al Fatta, “Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 1, pp. 215–224, 2024, doi: 10.25126/jtiik.20241118074.
- [13] Y. Prasetyo *et al.*, “Analisis Perbandingan Optimizer SGD dan Adam pada Model CNN untuk Klasifikasi Jamur Edible dan Poisonous,” vol. 9, no. 1, pp. 1348–1354, 2025, doi: <https://doi.org/10.36040/jati.v9i1.12659>.
- [14] M. W. Sardjono, V. Ramadhan, V. Ramadhan, M. Cahyanti, and E. R. Swedia, “Klasifikasi Bentuk Bingkai (Frame) Kacamata Menggunakan CNN dengan Arsitektur Inception V3 dan Augmented Reality Berbasis Android,” *J. Syst. Comput. Eng.*, vol. 5, no. 2, pp. 204–218, 2024, doi: 10.61628/jsce.v5i2.1292.
- [15] Y. Gulzar, “Fruit Image Classification Model Based on MobileNetV2V2 with Deep Transfer Learning Technique,” *Sustain.*, vol. 15, no. 3, 2023, doi: 10.3390/su15031906.
- [16] D. Dhimas, P. Putra, G. Kurnia Anaga, and W. T. Fitriyana, “Implementasi Algoritma Convolutional Neural Network Arsitektur MobileNetV2v2 Untuk Klasifikasi Ekspresi Wajah Pada Dataset FER,” *Pros. Semin. Nas. Teknol. Dan Sains*, vol. 3, no. 1, pp. 291–297, 2024, doi: <https://doi.org/10.29407/4t4bh918>.
- [17] Keras, “Keras Applications.” <https://keras.io/api/applications/> (accessed May 16, 2025).
- [18] A. Dharmaputra, M. Cahyanti, M. R. D. Septian, and E. R. Swedia, “Aplikasi Face Mask Detection Menggunakan Neural Network MobileNetV2v2 Berbasis Android,” *Sebatik*, vol. 25, no. 2, pp. 382–389, 2021, doi: 10.46984/sebatik.v25i2.1503.
- [19] H. F. Dalimunthe and P. Simanjuntak, “Aplikasi Pengenalan Perangkat Keras Komputer Berbasis Android Menggunakan Augmented Reality,” *Comput. Sci. Ind. Eng.*, vol. 9, no. 2, pp. 183–194, 2023, doi: 10.33884/comasiejournal.v9i2.7624.
- [20] C. Setianingsih, “Pengembangan Sistem Backend Menggunakan . Net Pada Aplikasi Konsultasi Makanan Diet Menggunakan Deep Learning,” vol. 11, no. 4, pp. 3085–3090, 2024.
- [21] Unity Technology, “Introduction to Barracuda,” 2023. <https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html> (accessed Mar. 23, 2025).
- [22] C. Mills, “Getting Started With Deep Learning in Unity,” *Christian Mills*, 2022.

- <https://christianjmill.com/posts/deep-learning-unity-intro/> (accessed Apr. 09, 2024).
- [23] Koki Ibukuro, "ONNX Runtime on Unity," *Medium*, 2024. <https://medium.com/%40asus4/onnx-runtime-on-unity-a40b3416529f> (accessed Apr. 09, 2024).
- [24] Ronny Westphal, *Developing a Neural Network Model for Semantic Segmentation*. 100 44 Flemingsberg, Swedia: KTH Skolan för kemi, bioteknologi och hälsa, 2023.
- [25] L. Ayu Amizhora and T. Sutabri, "Rancangan Prototype Aplikasi Laundrytime Menggunakan Metode Ucd," *J. Salome Multidisipliner Keilmuan*, vol. 1, no. 2, pp. 91–99, 2023.
- [26] R. Nurhidayat and K. E. Dewi, "Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek," *Komputa J. Ilm. Komput. dan Inform.*, vol. 12, no. 1, pp. 91–100, 2023, doi: 10.34010/komputa.v12i1.9458.
- [27] W. I. Rahayu, C. Prianto, and E. A. Novia, "Perbandingan Algoritma K-Means dan Naive Bayes untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan pada PT. Pertamina (Persero)," *J. Tek. Inform.*, vol. 13, no. 2, pp. 1–8, 2021.
- [28] R. Agustina, R. Magdalena, and N. K. C. Pratiwi, "Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 2, p. 446, 2022, doi: 10.26760/elkomika.v10i2.446.
- [29] K. Setiyanto and M. Bolang, "Analisis perbandingan hasil klasifikasi jenis penyakit tanaman tomat menggunakan arsitektur MobileNetV2, densenet121, dan xception 1,2," vol. 3, no. 3, pp. 56–69, 2024, doi: <https://doi.org/10.56127/jts.v3i3.1898>.
- [30] A. R. Sujiwanto, R. R. B. P. Prawirodirjo, and P. Palupiningsih, "Analisis Perbandingan Performa Model Klasifikasi Kesehatan Daun Tomat menggunakan arsitektur VGG, MobileNetV2, dan Inception V3," *J. Ilmu Komput. dan Agri-Informatika*, vol. 10, no. 1, pp. 98–110, 2023, doi: 10.29244/jika.10.1.98-110.