

KOMPARASI RESNET50, MOBILENETV2, DAN ALEXNET UNTUK DETEKSI PENYAKIT DAUN KENTANG BERBASIS ANDROID

Alya Massardi, Slamet Widodo, Azwardi

D-IV Teknologi Informatika Multimedia Digital, Jurusan Teknik Komputer,
Politeknik Negeri Sriwijaya Palembang
alyaassr@gmail.com

ABSTRAK

Kentang merupakan komoditas hortikultura yang produktivitasnya dapat menurun akibat penyakit daun. Identifikasi secara visual masih bergantung pada pengalaman pengamat dan rentan keliru karena beberapa kelas penyakit memiliki gejala yang serupa. Penelitian ini bertujuan membandingkan kinerja arsitektur ResNet50, MobileNetV2, dan AlexNet serta mengimplementasikan model terbaik pada aplikasi Android PotaLeaf. Penelitian menggunakan pendekatan kuantitatif dengan eksperimen komparatif terhadap dataset publik berisi 3.076 citra dalam tujuh kelas. Data dibersihkan secara manual menjadi 2.118 citra, diseimbangkan melalui augmentasi menjadi 3.500 citra, kemudian dibagi 70% data pelatihan, 15% validasi, dan 15% pengujian. ResNet50 dan MobileNetV2 dilatih menggunakan *transfer learning*, sedangkan AlexNet dilatih dari awal. Evaluasi dilakukan pada dataset dengan dan tanpa *data cleaning* menggunakan *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*, dilanjutkan *black box testing* serta pengujian 20 citra eksternal. ResNet50 menghasilkan performa terbaik dengan *accuracy* 92,20%, *precision* 92,40%, *recall* 92,20%, dan *F1-score* 92,20%. Data cleaning meningkatkan *accuracy* seluruh model sebesar 10,20-19,10 poin persentase. Model ResNet50 berhasil diintegrasikan ke PotaLeaf; seluruh 13 fungsi berjalan sesuai rancangan dan pengujian eksternal mencapai *accuracy* 90%.

Kata kunci: Android, data cleaning, deteksi penyakit, PotaLeaf, ResNet50

1. PENDAHULUAN

Kentang merupakan komoditas hortikultura penting, tetapi produktivitasnya dapat menurun ketika penyakit daun tidak dikenali dan ditangani secara tepat. Identifikasi lapangan yang bertumpu pada pengamatan visual sering menghadapi gejala yang saling menyerupai, variasi pencahayaan, latar belakang kompleks, dan tingkat kerusakan yang berbeda. *Convolutional Neural Network* mampu mengekstraksi warna, tekstur, bentuk, dan pola bercak secara otomatis sehingga banyak digunakan pada klasifikasi penyakit tanaman [1]. Tinjauan mutakhir di bidang pertanian kentang juga menunjukkan bahwa *deep learning* berpotensi mempercepat identifikasi, tetapi kinerjanya sangat dipengaruhi kualitas dataset, variasi kelas, dan kondisi akuisisi citra [2].

Dataset yang digunakan pada penelitian ini berasal dari citra daun kentang pada lingkungan tidak terkontrol dan mencakup tujuh kelas, yaitu Virus, Phytophthora, Pest, Nematode, Fungi, Bacteria, dan Healthy [3]. Jumlah kelas tersebut lebih representatif dibandingkan studi yang hanya membedakan early blight, late blight, dan healthy. Akan tetapi, perluasan kelas memperbesar kesulitan klasifikasi karena Fungi, Pest, dan Phytophthora dapat sama-sama menampilkan bercak, klorosis, atau kerusakan jaringan. Tantangan tersebut membuat nilai *accuracy* agregat belum cukup; *precision*, *recall*, *F1-score*, dan *confusion matrix* diperlukan untuk melihat kelas yang masih berisiko tertukar.

Penelitian sebelumnya telah mengembangkan aplikasi Android untuk klasifikasi penyakit daun kentang, tetapi jumlah kelas dan strategi evaluasinya masih beragam [4]. MobileNetV2 dengan optimasi

hyperparameter menawarkan efisiensi komputasi, namun hasilnya bergantung pada karakter dataset [5]. Model berbasis VGG16 dapat mencapai akurasi tinggi pada tiga kelas [6], sedangkan pendekatan hibrida menunjukkan bahwa kombinasi representasi fitur dan strategi pelatihan dapat memperkuat *generalization* [7]. Perbandingan langsung beberapa arsitektur pada dataset, pembagian data, augmentasi, dan protokol evaluasi yang sama masih diperlukan agar perbedaan performa tidak berasal dari kondisi eksperimen yang tidak setara.

Penelitian ini membandingkan ResNet50, MobileNetV2, dan AlexNet. ResNet50 dan MobileNetV2 memanfaatkan *transfer learning* ImageNet, sedangkan AlexNet dilatih dari awal. Pilihan tersebut memungkinkan analisis bukan hanya terhadap arsitektur, tetapi juga terhadap nilai representasi *pre-trained* ketika dataset domain-spesifik terbatas. Penelitian terdahulu telah membuktikan kelayakan CNN untuk penyakit daun kentang [8], [9] dan kombinasi fitur tekstur [10], tetapi pengaruh *data cleaning* terhadap tiga arsitektur dengan strategi pelatihan berbeda belum banyak dibahas secara komparatif.

Rumusan penelitian adalah: bagaimana kinerja ketiga arsitektur pada dataset bersih dan tanpa pembersihan; pola kesalahan apa yang muncul pada tujuh kelas; serta apakah model terbaik dapat dijalankan pada aplikasi Android. Penelitian bertujuan menentukan model dengan performa paling konsisten, menjelaskan kontribusi kualitas data terhadap hasil klasifikasi, dan mengevaluasi fungsi aplikasi PotaLeaf. Kontribusi penelitian terletak pada eksperimen terkontrol dua skenario dataset, analisis

kesalahan per kelas, dan pemisahan antara kinerja klasifikasi laboratorium dengan bukti implementasi mobile.

2. TINJAUAN PUSTAKA

2.1. Arsitektur CNN dan Transfer Learning

ResNet50 menggunakan residual block dan *skip connection* untuk mempertahankan aliran gradien pada jaringan dalam. Mekanisme tersebut membantu model mempelajari fitur hierarkis tanpa penurunan performa yang lazim terjadi ketika kedalaman bertambah. MobileNetV2 menekan kebutuhan komputasi melalui *depthwise separable convolution* dan *inverted residual block*, sehingga menarik untuk perangkat bergerak. AlexNet mempunyai struktur lebih sederhana dan pada penelitian ini berfungsi sebagai pembandingan *training from scratch*. Studi AlexNet pada citra daun menunjukkan bahwa model tersebut dapat mengenali pola dasar, tetapi performanya sensitif terhadap jumlah dan kualitas data [11].

Transfer learning memindahkan representasi visual yang telah dipelajari dari dataset besar ke domain baru. Pada dataset berukuran terbatas, bobot awal dapat mengurangi kebutuhan mempelajari fitur dasar dari nol. Namun, *fine-tuning* tetap diperlukan agar fitur menyesuaikan karakter penyakit daun. Pada klasifikasi multikelas, ketidakseimbangan kelas dapat mendorong model berpihak pada kelas dominan; augmentasi dan pembobotan kelas merupakan strategi umum untuk memperbaiki kondisi tersebut [12]. Konfigurasi *batch size*, *learning rate*, *optimizer*, dan mekanisme *early stopping* juga memengaruhi konvergensi dan *generalization* [13]-[15].

2.2. Kualitas Data dan Implementasi Mobile

Data cleaning mengurangi citra buram, objek non-daun yang dominan, latar belakang mengganggu, dan sampel dengan kualitas visual rendah. Tujuannya bukan memperindah dataset, tetapi meningkatkan kesesuaian antara fitur yang dipelajari model dan gejala penyakit. Setelah pembersihan, augmentasi digunakan untuk *class balancing* dan variasi geometris. Eksperimen dilakukan pada Google Colaboratory yang menyediakan sumber daya komputasi berbasis GPU [16], menggunakan TensorFlow untuk pembangunan dan konversi model [17].

Model terpilih dikonversi menjadi *TensorFlow Lite* agar inferensi dapat dijalankan pada perangkat Android. Aplikasi mobile dinilai tidak hanya dari keluaran model, tetapi juga dari alur kamera atau galeri, penyajian *confidence*, penyimpanan riwayat, autentikasi, dan akses informasi penyakit. Pengembangan aplikasi Android memerlukan integrasi antarmuka, model, dan penyimpanan data [18]. Pengujian black box memeriksa kesesuaian fungsi dari perspektif pengguna tanpa menganalisis struktur kode internal [19].

3. METODE PENELITIAN

3.1. Desain Penelitian dan Objek Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komparatif. Eksperimen dirancang untuk membandingkan performa tiga arsitektur *Convolutional Neural Network* (CNN), yaitu ResNet50, MobileNetV2, dan AlexNet, pada dua skenario data: *dataset hasil data cleaning* dan *dataset tanpa data cleaning*. Objek penelitian berupa *dataset* sekunder dari Mendeley Data yang memuat 3.076 citra daun kentang pada lingkungan tidak terkontrol (*uncontrolled environment*). Dataset terdiri atas tujuh kelas, yaitu Virus, Phytophthora, Pest, Nematode, Fungi, Bacteria, dan Healthy. Setiap citra memiliki satu label kelas sehingga permasalahan yang diselesaikan merupakan klasifikasi multikelas.

3.2. Persiapan dan Pembagian Dataset

Persiapan data diawali dengan peninjauan seluruh citra secara manual. Citra yang mengalami *blur*, memiliki kualitas visual rendah, tidak relevan, atau didominasi objek selain daun, seperti tangan pada saat pengambilan gambar, dikeluarkan dari *dataset*. Tahap *data cleaning* ini mengurangi jumlah data dari 3.076 menjadi 2.118 citra. Selanjutnya dilakukan *Exploratory Data Analysis* melalui visualisasi distribusi kelas dan sampel citra untuk mengidentifikasi ketidakseimbangan data serta mengamati karakteristik warna, bercak, tekstur, latar belakang, pencahayaan, dan sudut pengambilan gambar pada setiap kelas.

Ketidakeimbangan kelas ditangani melalui augmentasi secara *offline* menggunakan *ImageDataGenerator*. Transformasi yang diterapkan meliputi *rotation*, *zoom*, *width shift*, *height shift*, *shear*, dan *horizontal flip* tanpa mengubah label kelas. Jumlah citra pada setiap kelas disetarakan menjadi 500 sehingga *dataset* setelah augmentasi berjumlah 3.500 citra. Citra kemudian disiapkan pada ukuran masukan 224 x 224 piksel dan dibagi menggunakan pustaka *splitfolders* dengan *seed* 42 serta rasio 70%:15%:15%. Pembagian tersebut menghasilkan 2.450 citra *training set*, 525 *validation set*, dan 525 *testing set*, atau masing-masing 350, 75, dan 75 citra per kelas. *Training set* digunakan untuk pembelajaran bobot, *validation set* untuk memantau performa selama pelatihan, dan *testing set* untuk evaluasi akhir pada data yang tidak digunakan dalam pelatihan.

Tabel 1. Tahapan persiapan dan pembagian dataset

Tahap	Jumlah	Keterangan
Dataset awal	3.076	Tujuh kelas, distribusi tidak seimbang
Setelah cleaning	2.118	958 citra dihapus karena kualitas atau relevansi
Setelah augmentasi	3.500	500 citra pada setiap kelas
Training set	2.450	350 citra per kelas
Validation set	525	75 citra per kelas
Testing set	525	75 citra per kelas

Tabel 1 memperlihatkan bahwa *data cleaning* menghapus 958 citra atau sekitar 31,14% dataset awal. Proporsi tersebut cukup besar, sehingga pengaruh pembersihan perlu dinilai melalui eksperimen perbandingan agar peningkatan performa tidak hanya diasumsikan.

3.3. Pembangunan dan Pelatihan Model

Pelatihan model dilakukan pada *Google Colaboratory* dengan akselerasi GPU menggunakan *Python*, *TensorFlow*, dan *Keras*. Lingkungan pelatihan dikonfigurasi dengan ukuran masukan 224×224 piksel dan *mixed precision* untuk meningkatkan efisiensi komputasi. ResNet50 dan MobileNetV2 menggunakan bobot awal *ImageNet* melalui *transfer learning*, sedangkan AlexNet dibangun dengan pendekatan *training from scratch*. Preprocessing disesuaikan dengan arsitektur: ResNet50 dan MobileNetV2 menggunakan fungsi *preprocess_input*, sedangkan AlexNet menggunakan normalisasi nilai piksel melalui *rescale 1/255*.

Hyperparameter tuning dilakukan dengan menguji *batch size* 16 dan 32 serta *learning rate* 0,001 dan 0,0001. Seluruh model dikompilasi menggunakan *optimizer* Adam dan fungsi *loss* *categorical_crossentropy*. Data training memperoleh *horizontal flip* secara daring, sedangkan data validation hanya diberi *preprocessing* sesuai arsitektur. Callback *EarlyStopping* dan *ReduceLROnPlateau* diterapkan untuk mengendalikan *overfitting* dan membantu konvergensi. AlexNet dilatih dari awal selama maksimal 15 *epoch*. ResNet50 dan MobileNetV2 menjalani pelatihan awal selama lima *epoch*, kemudian 50 lapisan terakhir dibuka (*unfreeze*) dan model dilatih kembali melalui *fine-tuning* selama sepuluh *epoch* dengan *learning rate* sebesar 10% dari nilai awal. Konfigurasi terbaik menggunakan *batch size* 32 dan *learning rate* 0,0001 untuk ResNet50 dan MobileNetV2, sedangkan AlexNet menggunakan *batch size* 16 dan *learning rate* 0,0001.

3.4. Evaluasi dan Pemilihan Model

Evaluasi dilakukan secara terpisah pada skenario dataset hasil *data cleaning* dan skenario tanpa *data cleaning* untuk mengukur pengaruh kualitas data terhadap ketiga arsitektur. Performa akhir dihitung menggunakan *testing set* yang tidak terlibat dalam proses pelatihan dan validasi. Metrik yang digunakan meliputi *accuracy*, *macro precision*, *macro recall*, *macro F1-score*, dan *confusion matrix*. Kurva *training accuracy*, *validation accuracy*, *training loss*, dan *validation loss* juga dianalisis untuk menilai kestabilan pembelajaran dan indikasi *overfitting*. Model terbaik ditentukan berdasarkan nilai metrik keseluruhan, konsistensi antara performa validasi dan pengujian, serta distribusi kesalahan klasifikasi pada setiap kelas.

3.5. Implementasi dan Pengujian Sistem

Model dengan performa terbaik dikonversi ke format *TensorFlow Lite* (.tflite) dan diintegrasikan ke

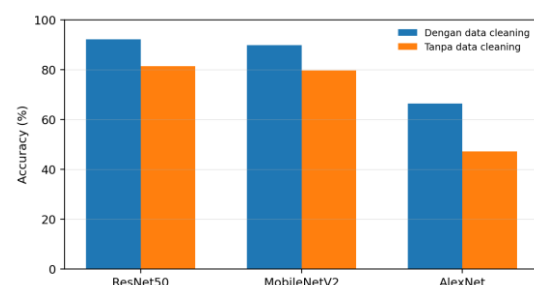
aplikasi Android PotaLeaf agar inferensi dapat dilakukan secara lokal pada perangkat (*on-device inference*). Pengguna dapat memilih citra melalui kamera atau galeri, kemudian sistem melakukan *preprocessing*, menjalankan inferensi, dan menampilkan kelas penyakit, *confidence*, deskripsi, gejala, serta rekomendasi. Aplikasi juga mencakup autentikasi, penyimpanan riwayat, profil, dan informasi aplikasi. Pengujian fungsional dilakukan melalui *black box testing* terhadap 13 fungsi aplikasi. Selain itu, model diuji menggunakan 20 citra eksternal yang berasal dari dokumentasi pribadi dan petani. Pengujian eksternal digunakan sebagai indikasi awal *generalization* di luar *testing set*, bukan sebagai estimasi performa populasi karena jumlah sampel dan cakupan kelas masih terbatas.

4. HASIL DAN PEMBAHASAN

4.1. Pengaruh Persiapan Data

Pembersihan data menghasilkan dataset yang lebih konsisten secara visual, tetapi sekaligus mengurangi keragaman alami. Augmentasi kemudian menyeimbangkan seluruh kelas menjadi 500 citra. Strategi ini mengurangi bias frekuensi kelas, khususnya karena kelas Nematode hanya memiliki 68 citra pada dataset awal, sedangkan Fungi mencapai 748 citra. Keseimbangan membantu model memperoleh peluang belajar yang sama, tetapi citra augmentasi tetap berasal dari sampel dasar yang terbatas. Dengan demikian, jumlah 500 per kelas tidak identik dengan 500 kondisi biologis yang benar-benar berbeda.

Seluruh model memperoleh peningkatan *accuracy* setelah *data cleaning*. AlexNet meningkat 19,10 poin persentase, lebih besar daripada ResNet50 10,80 poin dan MobileNetV2 10,20 poin. Perbedaan ini menunjukkan bahwa *training from scratch* lebih rentan terhadap *noise* karena seluruh representasi fitur dibentuk dari dataset penelitian. Sebaliknya, bobot *pre-trained* memberi ResNet50 dan MobileNetV2 representasi awal yang lebih robust. Hasil ini mendukung kajian pemrosesan citra tanaman yang menempatkan kualitas masukan sebagai prasyarat utama klasifikasi otomatis [20].



Gambar 1. Perbandingan accuracy model pada dua skenario dataset

Dari Gambar 1 dapat dilihat bahwa urutan model tidak berubah pada kedua skenario: ResNet50 selalu tertinggi, MobileNetV2 berada di posisi kedua, dan

AlexNet terendah. Konsistensi peringkat memperkuat bahwa perbedaan bukan hanya akibat cleaning, tetapi juga strategi arsitektur dan pelatihan.

4.2. Perbandingan Performa Tiga Arsitektur

Tabel 2. Performa model pada dataset hasil data cleaning

Model	Acc.	Prec.	Rec.	F1
ResNet50	92,20	92,40	92,20	92,20
MobileNetV2	89,90	90,10	89,90	89,90
AlexNet	66,30	67,60	66,30	64,50

Tabel 2 menunjukkan keunggulan ResNet50 pada seluruh metrik. Selisih 2,30 poin *accuracy* terhadap MobileNetV2 relatif kecil, tetapi ResNet50 juga menghasilkan kesalahan lebih sedikit pada kelas sulit. Hasil tersebut menunjukkan bahwa *skip connection* dan kedalaman representasi memberi manfaat ketika gejala dibedakan melalui variasi bercak, tekstur, dan warna. MobileNetV2 tetap kompetitif karena desain ringan, sehingga secara praktis dapat menjadi alternatif ketika ukuran model, memori, atau latensi lebih penting. Penelitian ini belum membandingkan ukuran file dan waktu inferensi, sehingga istilah model terbaik dibatasi pada performa klasifikasi, bukan efisiensi deployment.

AlexNet menghasilkan *accuracy* 66,30% dan *F1-score* 64,50%. Nilai tersebut tidak berarti arsitektur selalu tidak sesuai, melainkan menunjukkan ketidaksesuaian antara *training from scratch* dan skala data yang tersedia. Model harus membentuk fitur tepi, tekstur, pola bercak, hingga representasi penyakit hanya dari 2.450 citra pelatihan. ResNet50 dan MobileNetV2 telah membawa fitur visual dari *ImageNet*, kemudian menyesuikannya melalui *fine-tuning*. Temuan ini memperlihatkan bahwa pada dataset pertanian terbatas, strategi inisialisasi bobot dapat lebih menentukan daripada usia arsitektur semata.

4.3. Analisis Kesalahan Klasifikasi

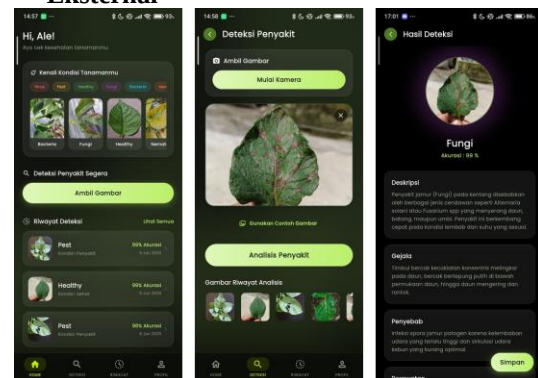
ResNet50 mencapai *recall* 1,00 pada Bacteria dan *F1-score* di atas 0,90 pada Healthy, Phytophthora, dan Virus. Kelas Fungi dan Pest memperoleh *F1-score* 0,87 dan 0,82. *Confusion matrix* menunjukkan delapan citra Fungi diprediksi sebagai Pest dan enam citra Pest diprediksi sebagai Fungi. MobileNetV2 memperlihatkan pola serupa dengan *F1-score* 0,80 pada Fungi dan 0,78 pada Pest. Konsistensi kesalahan pada dua model berbeda mengindikasikan bahwa sumber masalah terutama berada pada tumpang tindih gejala, bukan kegagalan satu arsitektur.

Fungi, Pest, dan Phytophthora dapat menampilkan bercak, klorosis, perubahan warna, atau kerusakan jaringan dengan variasi intensitas. Model yang hanya menerima satu citra daun tidak memperoleh konteks agronomis seperti pola penyebaran pada tanaman, kondisi batang, riwayat cuaca, atau informasi tanah. Akibatnya, prediksi

didasarkan pada bukti visual yang mungkin tidak unik. Pengembangan berikutnya perlu menambah citra lapangan multiskala, anotasi tingkat keparahan, dan data pendukung agar model tidak hanya mempelajari latar belakang atau pola fotografi.

AlexNet menghasilkan *F1-score* 0,92 pada Nematode tetapi hanya 0,26 pada Pest. Kontras tersebut menunjukkan bahwa kelas dengan gejala relatif khas, seperti penguningan lebih merata, masih dapat dipelajari oleh model sederhana. Sebaliknya, kelas dengan gejala heterogen membutuhkan representasi lebih dalam. Analisis per kelas ini penting karena *accuracy* keseluruhan dapat menutupi risiko kesalahan pada kelas tertentu. Dalam aplikasi pertanian, kesalahan kelas bukan sekadar angka; rekomendasi penanganan dapat berbeda sehingga keluaran perlu diposisikan sebagai alat bantu identifikasi awal.

4.4. Implementasi Aplikasi dan Pengujian Eksternal



Gambar 2. Halaman utama, analisis, dan hasil pada aplikasi PotaLeaf

Gambar 2 menunjukkan alur inti PotaLeaf: pengguna memilih atau mengambil citra, sistem menjalankan inferensi *TensorFlow Lite*, lalu menampilkan kelas, *confidence*, deskripsi, gejala, dan rekomendasi. Model dijalankan secara *on-device* sehingga fungsi klasifikasi tidak bergantung pada server inferensi. Seluruh 13 skenario black box, termasuk autentikasi, kamera, galeri, deteksi tujuh kelas, riwayat, profil, dan informasi aplikasi, memperoleh status sesuai.

Tabel 3. Ringkasan pengujian implementasi

Pengujian	Hasil	Interpretasi
Black box	13/13 fungsi sesuai	Alur aplikasi berjalan sesuai rancangan
Eksternal	18/20 benar	Accuracy awal 90% pada enam kelas
Kesalahan	2 citra	Healthy→Virus dan Bacteria→Healthy

Pengujian eksternal menghasilkan 18 prediksi benar dari 20 citra atau *accuracy* 90%. Nilai tersebut mendukung kemungkinan *generalization* di luar *testing set*, tetapi belum setara dengan validasi

lapangan. Sampel hanya mewakili enam kelas dan ukuran per kelas tiga sampai empat citra; kelas Virus tidak tercakup secara seimbang. Dua kesalahan pada Healthy dan Bacteria juga menunjukkan bahwa latar, pencahayaan, atau variasi gejala nyata masih dapat menggeser prediksi. Karena itu, angka 90% harus dipahami sebagai uji fungsional-eksploratif, bukan estimasi performa populasi.

4.5. Kontribusi, Implikasi, dan Keterbatasan

Kontribusi akademik penelitian terletak pada desain komparatif dua skenario data. Hasil menunjukkan tiga lapis determinan performa. Pertama, *data cleaning* meningkatkan sinyal visual yang relevan. Kedua, *transfer learning* memberi ketahanan ketika jumlah data domain-spesifik terbatas. Ketiga, karakter gejala menentukan batas diskriminasi antarkelas. Dengan demikian, peningkatan akurasi tidak cukup dicapai dengan menambah kedalaman jaringan; kualitas data, strategi pelatihan, dan definisi kelas harus dirancang secara terpadu.

Secara praktis, ResNet50 layak digunakan sebagai model PotaLeaf berdasarkan *accuracy* dan *F1-score* tertinggi. Akan tetapi, keputusan deployment jangka panjang perlu membandingkan latensi, konsumsi memori, ukuran model, dan kompatibilitas perangkat. MobileNetV2 mempunyai selisih *accuracy* kecil dan secara arsitektural lebih ringan, sehingga pengujian efisiensi dapat menghasilkan pilihan berbeda untuk *smartphone* dengan sumber daya terbatas. Aplikasi juga perlu menampilkan peringatan ketika *confidence* rendah serta menganjurkan konfirmasi tenaga pertanian pada kasus yang meragukan.

Keterbatasan penelitian mencakup penggunaan dataset sekunder, *cleaning* manual yang dapat mengandung subjektivitas, augmentasi yang tidak menambah keragaman biologis baru, tidak adanya pengujian langsung pada kebun dalam skala besar, dan sampel eksternal yang kecil. Penelitian berikutnya perlu menggunakan data lintas lokasi dan musim, membandingkan latensi tiga model pada beberapa perangkat, menerapkan validasi silang, menilai *calibration confidence*, serta menguji kemampuan membedakan tingkat keparahan. Evaluasi tersebut akan memperjelas apakah performa laboratorium tetap bertahan pada kondisi operasional.

4.6. Kesiapan Penerapan dan Tata Kelola Model

Kesiapan PotaLeaf tidak dapat ditentukan hanya dari *accuracy testing set*. Pada penggunaan nyata, distribusi citra dapat berubah akibat jenis kamera, jarak pengambilan, fase pertumbuhan, pencahayaan, latar tanah, dan tingkat keparahan penyakit. Perubahan tersebut menimbulkan domain shift: model tetap menjalankan inferensi, tetapi fitur yang diterima tidak identik dengan data pelatihan. Karena itu, aplikasi perlu merekam citra, *confidence*, koreksi pengguna, dan hasil konfirmasi ahli secara terkontrol. Data

pascapenerapan dapat digunakan untuk mengukur drift, menemukan kelas yang sering salah, dan menyusun dataset pembaruan tanpa mencampurkan data yang belum diverifikasi.

Nilai *confidence* juga harus diperlakukan secara kritis. Skor probabilitas tinggi tidak selalu berarti prediksi benar apabila model belum terkalibrasi. Antarmuka sebaiknya membedakan prediksi dengan keyakinan memadai, prediksi ambigu, dan citra yang tidak layak dianalisis. Pada kasus ambigu, pengguna perlu diminta mengambil ulang foto atau berkonsultasi dengan penyuluh. Mekanisme human-in-the-loop mencegah aplikasi menggantikan diagnosis profesional dan mengubahnya menjadi sistem pendukung keputusan. Pendekatan tersebut penting karena kesalahan antara Fungi, Pest, dan Phytophthora dapat mengarahkan tindakan penanganan yang berbeda.

Pemilihan model untuk perangkat bergerak juga memerlukan evaluasi multiobjektif. ResNet50 unggul dalam metrik klasifikasi, tetapi MobileNetV2 dirancang untuk efisiensi. Model yang sedikit lebih rendah *accuracy*-nya dapat lebih bermanfaat apabila menghasilkan latensi singkat, konsumsi memori rendah, dan stabilitas pada perangkat kelas menengah. Penelitian lanjutan perlu mengukur ukuran *TensorFlow Lite*, waktu inferensi, penggunaan RAM, konsumsi energi, serta perubahan *accuracy* setelah quantization. Hasil tersebut memungkinkan keputusan model berbasis keseimbangan antara ketepatan, aksesibilitas, dan pengalaman pengguna, bukan semata-mata peringkat *accuracy*.

Tata kelola pembaruan model perlu mendokumentasikan versi dataset, kriteria *cleaning*, pembagian data, konfigurasi pelatihan, dan metrik per kelas. Setiap pembaruan harus diuji pada set independen yang tidak digunakan dalam augmentasi atau tuning. Riwayat versi memungkinkan pengembang menelusuri apakah peningkatan satu kelas justru menurunkan kelas lain. Dengan demikian, kontribusi PotaLeaf bukan hanya artefak aplikasi, tetapi rancangan siklus pembelajaran: pengumpulan bukti lapangan, validasi ahli, pembaruan data, pelatihan ulang, evaluasi, dan distribusi model secara bertanggung jawab.

4.7. Sintesis Perbandingan dengan Penelitian Terdahulu

Perbandingan dengan penelitian terdahulu perlu dilakukan secara hati-hati karena angka performa dipengaruhi oleh jumlah kelas, sumber citra, cara pembagian data, dan kondisi pengujian. Ghandi dkk. melaporkan akurasi 99,69% pada aplikasi Android dengan tiga kelas [4], sedangkan Utomo dkk. memperoleh akurasi 95% menggunakan VGG16 pada tiga kelas [6]. Nilai tersebut lebih tinggi daripada ResNet50 pada penelitian ini, tetapi tugas klasifikasinya lebih sederhana karena hanya memisahkan early blight, late blight, dan healthy. Penelitian ini menggunakan tujuh kelas dengan

beberapa gejala yang saling tumpang tindih. Oleh karena itu, perbedaan akurasi tidak dapat dimaknai sebagai keunggulan mutlak satu model tanpa mempertimbangkan kompleksitas data.

Pada penelitian Lie dkk., MobileNetV2 dengan optimasi hyperparameter menghasilkan akurasi 78,45% pada delapan kelas [5]. MobileNetV2 dalam penelitian ini mencapai 89,90% setelah data cleaning dan class balancing. Kenaikan tersebut memperlihatkan bahwa arsitektur ringan tetap mampu menghasilkan performa kuat ketika kualitas data dan prosedur pelatihan dikendalikan. Namun, hasil juga menunjukkan bahwa efisiensi komputasi tidak selalu identik dengan kemampuan diskriminasi tertinggi. Selisih 2,30 poin terhadap ResNet50 terutama terlihat pada kelas Fungi dan Pest, yaitu kelas yang membutuhkan representasi tekstur lebih rinci.

Keunggulan metodologis penelitian ini terletak pada penggunaan protokol yang sama untuk tiga arsitektur. Dataset, pembagian training-validation-testing, jumlah kelas, skenario cleaning, serta metrik evaluasi dibuat konsisten. Desain tersebut mengurangi kemungkinan bahwa perbedaan performa berasal dari variasi eksperimen. Hasilnya memperlihatkan dua temuan yang relatif stabil: model transfer learning mengungguli training from scratch, dan data cleaning memperbaiki seluruh arsitektur. Temuan tersebut konsisten dengan tinjauan CNN pada penyakit tanaman yang menempatkan kualitas data serta strategi adaptasi fitur sebagai komponen penting [1], [2].

Walaupun ResNet50 dipilih untuk PotaLeaf, pemilihan model pada implementasi nyata seharusnya menggunakan kriteria multiobjektif. Accuracy dan F1-score menggambarkan kualitas prediksi, tetapi aplikasi mobile juga dipengaruhi ukuran model, konsumsi memori, waktu inferensi, penggunaan baterai, dan kompatibilitas perangkat. MobileNetV2 dapat menjadi kandidat lebih tepat pada perangkat berspesifikasi rendah apabila selisih performa dapat diterima. Dengan demikian, kontribusi penelitian bukan sekadar menetapkan ResNet50 sebagai pemenang, melainkan menunjukkan trade-off antara kedalaman representasi dan efisiensi deployment.



Gambar 3. Alur validasi model hingga implementasi PotaLeaf

Gambar 3 menegaskan bahwa implementasi aplikasi merupakan tahap akhir dari rangkaian validasi, bukan pengganti evaluasi model. Model hanya dikonversi setelah kualitas data, stabilitas pelatihan, distribusi kesalahan, dan konsistensi testing set dinilai. Urutan ini penting agar keberhasilan integrasi Android tidak disalahartikan sebagai bukti bahwa diagnosis telah siap digunakan tanpa batasan.

4.8. Kesiapan Penerapan Lapangan dan Tata Kelola Model

Penggunaan PotaLeaf di lapangan menghadapi domain shift, yaitu perbedaan antara citra pelatihan dan citra operasional. Citra kebun dapat memiliki bayangan tajam, daun tumpang tindih, kerusakan mekanis, tanah sebagai latar, atau kombinasi beberapa penyakit. Kondisi tersebut belum seluruhnya direpresentasikan oleh dataset sekunder. Karena itu, aplikasi perlu menyediakan panduan pengambilan citra, misalnya satu daun dominan, pencahayaan cukup, jarak kamera konsisten, dan fokus yang jelas. Protokol sederhana dapat menurunkan variasi yang tidak relevan tanpa membebani pengguna.

Nilai confidence juga tidak boleh diperlakukan sebagai kepastian diagnosis. Model dapat menghasilkan confidence tinggi pada citra yang berada di luar distribusi pelatihan. Aplikasi sebaiknya menerapkan ambang keputusan dan kategori “perlu pemeriksaan lanjutan” ketika confidence rendah atau dua kelas memiliki probabilitas berdekatan. Pendekatan human-in-the-loop memungkinkan petani menggunakan hasil sebagai indikasi awal, kemudian meminta konfirmasi penyuluh atau ahli tanaman sebelum menentukan tindakan pengendalian. Posisi ini lebih aman daripada menyajikan hasil sebagai diagnosis final.

Tabel 4. Risiko implementasi dan strategi mitigasi

Risiko	Dampak	Mitigasi
Domain shift	Akurasi menurun pada kondisi kebun	Tambah data lintas lokasi, musim, dan perangkat
Confidence tidak terkalibrasi	Prediksi salah terlihat meyakinkan	Gunakan ambang, top-2, dan opsi pemeriksaan lanjut
Kelas tidak terwakili	Model memaksa prediksi ke tujuh kelas	Sediakan kelas unknown dan mekanisme pelaporan
Perubahan distribusi data	Performa menurun setelah penggunaan	Pantau kesalahan dan lakukan retraining berkala
Saran penanganan terlalu umum	Tindakan pengguna tidak sesuai	Validasi rekomendasi bersama pakar pertanian

Tabel 4 memperlihatkan bahwa risiko implementasi tidak hanya berasal dari kesalahan kode, tetapi juga dari ketidaksesuaian data dan interpretasi pengguna. Black box testing membuktikan fungsi antarmuka berjalan sesuai rancangan, sedangkan pengujian eksternal memberi indikasi generalization awal. Keduanya belum menilai perubahan performa setelah aplikasi digunakan berulang kali di lingkungan berbeda. Oleh sebab itu, sistem perlu mencatat prediksi, confidence, waktu, dan umpan balik pengguna secara terkontrol agar kasus salah klasifikasi dapat dianalisis tanpa mengabaikan privasi.

Tata kelola pembaruan model perlu menetapkan versi model, tanggal pelatihan, komposisi dataset,

hasil evaluasi, dan perangkat uji. Setiap penambahan data baru harus melalui verifikasi label agar kesalahan pengguna tidak langsung masuk ke dataset pelatihan. Model baru sebaiknya dibandingkan dengan versi produksi menggunakan testing set tetap dan data lapangan terbaru. Mekanisme tersebut mencegah penurunan performa tersembunyi dan memastikan bahwa peningkatan pada satu kelas tidak mengorbankan kelas lain.

Kesiapan operasional juga bergantung pada komunikasi hasil. Halaman hasil PotaLeaf telah menampilkan deskripsi, gejala, dan rekomendasi, tetapi bahasa perlu dibedakan antara informasi edukatif dan instruksi tindakan. Rekomendasi sebaiknya menyertakan batasan, anjuran pemeriksaan kondisi lain, dan rujukan kepada tenaga pertanian. Strategi ini penting karena penyakit yang tampak serupa dapat membutuhkan penanganan berbeda, sedangkan penggunaan pestisida yang tidak tepat dapat meningkatkan biaya dan risiko lingkungan.

Dengan mempertimbangkan aspek tersebut, PotaLeaf dapat dinilai siap sebagai prototipe alat bantu identifikasi, bukan pengganti diagnosis profesional. Kekuatan sistem berada pada proses inferensi on-device, cakupan tujuh kelas, dan performa ResNet50 yang konsisten. Kesiapan untuk penggunaan luas memerlukan validasi prospektif di kebun, pengukuran efisiensi pada beberapa smartphone, pengujian usability oleh petani, serta mekanisme pemeliharaan model. Tahapan lanjutan tersebut akan mengubah prototipe penelitian menjadi sistem pendukung keputusan yang lebih dapat dipercaya.

5. KESIMPULAN DAN SARAN

ResNet50 merupakan model dengan performa klasifikasi terbaik pada dataset hasil *data cleaning* dengan *accuracy* 92,20%, *precision* 92,40%, *recall* 92,20%, dan *F1-score* 92,20%, diikuti MobileNetV2 89,90% dan AlexNet 66,30%. Pembersihan data meningkatkan *accuracy* seluruh model sebesar 10,20-19,10 poin persentase, sedangkan kesalahan dominan tetap terjadi pada Fungi, Pest, dan Phytophthora karena kemiripan visual. Model ResNet50 berhasil dikonversi ke *TensorFlow Lite* dan diintegrasikan pada PotaLeaf; seluruh fungsi aplikasi berjalan sesuai rancangan dan pengujian 20 citra eksternal menghasilkan 90% *accuracy*. Temuan menegaskan bahwa kualitas data dan *transfer learning* merupakan faktor utama, tetapi hasil eksternal belum cukup untuk menyatakan kesiapan diagnosis lapangan. Penelitian berikutnya disarankan memperluas data lintas kondisi kebun, menguji efisiensi komputasi dan calibration *confidence*, membandingkan ResNet50 dengan model ringan, serta melibatkan pakar pertanian dalam validasi rekomendasi.

DAFTAR PUSTAKA

[1] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture*,

vol. 11, no. 8, pp. 1–18, 2021, doi: 10.3390/agriculture11080707.

- [2] B. Gülmez, "A comprehensive review of convolutional neural networks based disease detection strategies in potato agriculture," *Potato Research*, vol. 68, no. 2, 2025, doi: 10.1007/s11540-024-09786-1.
- [3] N. H. Shabrina et al., "A novel dataset of potato leaf disease in uncontrolled environment," *Data in Brief*, vol. 52, p. 109955, 2024, doi: 10.1016/j.dib.2023.109955.
- [4] S. Ghandi, I. Ma'ruf Nugroho, and Y. Raymond Ramadhan, "Penerapan metode Convolutional Neural Network dalam aplikasi pendeteksi penyakit daun tanaman kentang berbasis Android," *JATI*, vol. 8, no. 5, pp. 8701–8708, 2024, doi: 10.36040/jati.v8i5.10769.
- [5] J. Lie, A. Rahman, and D. Udjilawa, "Klasifikasi penyakit daun kentang berbasis CNN MobileNetV2 dengan optimasi Randomized Search," *TIN: Terapan Informatika Nusantara*, vol. 6, no. 9, pp. 1598–1607, 2026, doi: 10.47065/tin.v6i9.9108.
- [6] P. T. Utomo, B. Santoso, and S. Kacung, "Sistem deteksi penyakit pada daun tanaman kentang menggunakan metode CNN arsitektur VGG-Net," vol. 7, pp. 777–784, 2025.
- [7] J. H. Sinamenye, A. Chatterjee, and R. Shrestha, "Potato plant disease detection: leveraging hybrid deep learning models," *BMC Plant Biology*, vol. 25, no. 1, 2025, doi: 10.1186/s12870-025-06679-4.
- [8] A. M. Lesmana, R. P. Fadhillah, and C. Rozikin, "Identifikasi penyakit pada citra daun kentang menggunakan Convolutional Neural Network," *Jurnal Sains dan Informatika*, vol. 8, no. 1, pp. 21–30, 2022, doi: 10.34128/jsi.v8i1.377.
- [9] A. J. Rozaqi, A. Sunyoto, and M. Rudyanto Arief, "Deteksi penyakit pada daun kentang menggunakan pengolahan citra dengan metode Convolutional Neural Network," *Creative Information Technology Journal*, vol. 8, no. 1, pp. 22–31, 2021.
- [10] A. Rosid, A. Ghoful, and F. Santoso, "Klasifikasi penyakit tanaman kentang berdasarkan citra daun dan batang dengan metode Convolutional Neural Network dan Gray Level Co-Occurrence Matrix," *G-Tech: Jurnal Teknologi Terapan*, 2024.
- [11] D. Irfansyah, M. Mustikasari, and A. Suroso, "Arsitektur Convolutional Neural Network AlexNet untuk klasifikasi hama pada citra daun tanaman kopi," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 6, no. 2, pp. 87–92, 2021, doi: 10.30591/jpit.v6i2.2802.
- [12] M. N. Razali, N. Arbaiy, P. C. Lin, and S. Ismail, "Optimizing multiclass classification using convolutional neural networks with class weights and early stopping for imbalanced datasets," *Electronics*, 2025.

-
- [13] J. A. Ilemobayo et al., “Hyperparameter tuning in machine learning: a comprehensive review,” *Journal of Engineering Research and Reports*, vol. 26, no. 6, pp. 388–395, 2024, doi: 10.9734/jerr/2024/v26i61188.
- [14] F. G. Lalamentik, O. A. Lantang, and F. D. Kambey, “Implementation of parameter tuning for optimizing CNN model performance,” *Jurnal Teknik Informatika*, vol. 20, no. 1, pp. 77–86, 2025.
- [15] M. I. Nurzula and C. Fatichah, “Optimasi hyperparameter pada Convolutional Neural Network untuk klasifikasi jenis penyakit kanker kulit menggunakan Bayesian Optimization,” *ILKOMNIKA*, vol. 7, no. 1, pp. 146–157, 2025.
- [16] R. Gelar Guntara, “Pemanfaatan Google Colab untuk aplikasi pendeteksian masker wajah menggunakan algoritma deep learning YOLOv7,” *Jurnal Teknologi dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, 2023, doi: 10.47233/jteksis.v5i1.750.
- [17] H. Budianto, T. Khalimi, R. Ismaya, E. Kurniadi, and E. Dharmawan, “Implementation of TensorFlow-based deep learning in a learning application,” *Journal of Physics: Conference Series*, vol. 1933, no. 1, 2021, doi: 10.1088/1742-6596/1933/1/012007.
- [18] Erdisna, S. R. Ningsih, F. Suryana, R. Hidayat, and D. M. Putra, “Development of mobile learning applications as learning media using Android Studio,” *Jurnal Teknologi Informasi dan Pendidikan*, vol. 17, no. 2, pp. 412–426, 2024, doi: 10.24036/jtip.v17i2.923.
- [19] A. R. Rambe and H. Prihantoro, “Pengujian otomatis aplikasi mobile dengan teknik black-box menggunakan Appium,” *Jurnal Automata*, vol. 3, no. 2, pp. 1–6, 2022.
- [20] S. Julyani and K. Khairullah, “Pemanfaatan pengolahan citra untuk deteksi dan identifikasi hama pada tanaman secara otomatis,” *Ranah Research*, vol. 7, no. 5, pp. 3518–3525, 2025, doi: 10.38035/rrij.v7i5.1661.