

KLASIFIKASI PENYAKIT DAUN PADI MENGGUNAKAN MOBILENETV2 DENGAN OPTIMASI DYNAMIC RANGE QUANTIZATION PADA APLIKASI MOBILE OFFLINE

Brandon Tan, Nenden Siti Fatonah, Agung Mulyo Widodo, Popong Setiawati

Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul

Jakarta, Indonesia

brandontan10113@student.esaunggul.ac.id

ABSTRAK

Daun padi yang menunjukkan gejala penyakit perlu dikenali sejak awal karena gangguan tersebut dapat berdampak pada produktivitas tanaman. Identifikasi secara manual masih bergantung pada pengalaman pengguna dan dapat menjadi tidak konsisten ketika bercak, warna, atau tekstur daun terlihat mirip. Artikel ini menyajikan sistem klasifikasi citra daun padi berbasis MobileNetV2, kemudian model diperkecil melalui Dynamic Range Quantization agar dapat dijalankan pada aplikasi Android tanpa koneksi internet. Data penelitian diperoleh dari Kaggle dengan empat kelas, yaitu healthy, brown spot, bacterial leaf blight, dan leaf blast. Setelah proses pembersihan dan penyeimbangan, jumlah data yang digunakan adalah 11.600 citra. Model dilatih dengan transfer learning, diuji pada data terpisah, lalu dikonversi ke format TensorFlow Lite. MobileNetV2 menghasilkan akurasi 96,44%, sedangkan model TFLite DRQ menghasilkan akurasi 96,16%. Optimasi menurunkan ukuran file dari 26,301 MB menjadi 2,395 MB dan mempercepat rata-rata inferensi dari 91,15 ms/gambar menjadi 30,86 ms/gambar. Model hasil optimasi berhasil diintegrasikan ke aplikasi Android offline yang menyediakan input kamera, galeri, hasil prediksi, confidence, waktu inferensi, dan riwayat klasifikasi lokal.

Kata kunci : *penyakit daun padi, MobileNetV2, Dynamic Range Quantization, TensorFlow Lite, Android offline, klasifikasi citra*

1. PENDAHULUAN

Padi menjadi komoditas pangan yang sangat dekat dengan kebutuhan masyarakat karena hasil utamanya, yaitu beras, dikonsumsi secara luas [1]. Peran tersebut membuat kestabilan produksi padi penting untuk dijaga, terutama di wilayah Asia dan Afrika yang menjadikan beras sebagai pangan pokok [2].

Salah satu gangguan yang dapat menekan hasil produksi adalah penyakit tanaman [3].

Serangan penyakit tidak hanya muncul pada satu bagian tanaman, tetapi juga dapat terlihat pada daun yang berhubungan langsung dengan proses fotosintesis dan pertumbuhan [4], [5].

Pada daun padi, gejala yang sering dibahas dalam penelitian klasifikasi antara lain brown spot, bacterial leaf blight, dan leaf blast [6], [7].

Gejala tersebut umumnya tampak sebagai bercak, perubahan warna, atau lesi pada permukaan daun [6]. Jika gejala tidak dikenali sejak awal, dampaknya dapat berlanjut pada penurunan hasil dan kerugian petani [7].

Pemeriksaan manual masih membutuhkan pengalaman, ketelitian, dan waktu yang tidak sedikit [8].

Kondisi menjadi lebih sulit ketika beberapa penyakit memperlihatkan pola visual yang mirip [9].

Computer vision memberi peluang agar citra daun dianalisis secara otomatis, bukan hanya melalui pengamatan langsung [10].

Pada pendekatan deep learning, Convolutional Neural Network dapat mempelajari ciri visual dari data gambar tanpa proses ekstraksi fitur manual yang rumit [11].

Kemampuan tersebut telah banyak dimanfaatkan pada citra daun tanaman, sehingga CNN menjadi salah satu pendekatan yang relevan untuk membantu identifikasi penyakit di bidang pertanian [12].

Penelitian tentang penyakit padi berbasis deep learning telah berkembang dari model tunggal sampai ensemble CNN [13].

Nayak dkk. menggunakan citra smartphone dan transfer learning untuk mendeteksi penyakit serta defisiensi nutrisi padi [14].

Xu dkk. menekankan model ringan berbasis MobileNetV2 untuk klasifikasi penyakit padi [15].

Sementara itu, Ahad dkk. dan Deng dkk. membandingkan arsitektur CNN dalam konteks diagnosis penyakit padi [16], [17].

Hasil dari kajian lain juga memperlihatkan bahwa CNN dapat digunakan secara efektif pada klasifikasi penyakit daun padi [9].

Pemilihan MobileNetV2 didasarkan pada penelitian sebelumnya yang menerapkan quantization pada MobileNetV2 untuk klasifikasi penyakit tanaman pada skenario perangkat edge. Penelitian tersebut menunjukkan bahwa model MobileNetV2 terkuantisasi dapat digunakan untuk menghasilkan model yang lebih ringan dan tetap mempertahankan performa klasifikasi [18].

Beberapa arsitektur ringan lain seperti MobileNetV3 dan EfficientNet juga dapat digunakan untuk klasifikasi citra, tetapi penelitian ini tidak diarahkan sebagai studi komparasi arsitektur untuk mencari model terbaik. Oleh karena itu, MobileNetV2 dipilih karena sesuai dengan arah optimasi model ringan pada penelitian sebelumnya, sedangkan

penelitian ini difokuskan pada kelanjutan penerapan MobileNetV2 ke aplikasi Android offline.

Dynamic Range Quantization digunakan agar model MobileNetV2 lebih sesuai dengan kebutuhan aplikasi mobile offline. Pemilihan DRQ didasarkan pada penelitian sebelumnya yang membandingkan teknik quantization dan menunjukkan bahwa DRQ lebih mampu menjaga akurasi dibandingkan Post-Training Quantization ketika diterapkan pada MobileNetV2 untuk skenario perangkat edge [18].

Selain itu, DRQ tidak memerlukan pelatihan ulang, sehingga lebih praktis diterapkan pada model yang sudah selesai dilatih sebelum diintegrasikan ke aplikasi Android offline. Aspek implementasi juga dipertimbangkan karena penggunaan teknologi di area pertanian tidak selalu didukung koneksi internet yang stabil [19].

Tiga keluaran utama disusun dari kebutuhan tersebut. Keluaran pertama adalah model klasifikasi citra daun padi untuk empat kelas pada dataset penelitian. Keluaran kedua adalah hasil perbandingan performa sebelum dan sesudah optimasi DRQ. Keluaran ketiga adalah aplikasi Android yang mampu menerima gambar dan menampilkan hasil klasifikasi tanpa koneksi internet. Dengan susunan ini, penelitian tidak berhenti pada pengujian model, tetapi berlanjut sampai penerapan model pada sistem mobile.

Permasalahan utama dalam penelitian ini tidak hanya berada pada proses pengenalan penyakit daun padi dari citra, tetapi juga pada kesiapan model ketika digunakan pada perangkat Android. Kemiripan bercak, warna, dan tekstur daun dapat membuat identifikasi manual menjadi tidak konsisten, sehingga penyakit berisiko terlambat dikenali atau salah diklasifikasikan [8], [9].

Di sisi lain, model klasifikasi yang berukuran besar dan memiliki waktu inferensi tinggi dapat membuat aplikasi kurang responsif ketika dijalankan pada perangkat mobile [18].

Ketergantungan terhadap koneksi internet juga menjadi kendala karena penggunaan aplikasi di area pertanian tidak selalu didukung jaringan yang stabil [19].

Dampak dari permasalahan tersebut adalah proses identifikasi menjadi kurang praktis, hasil klasifikasi sulit digunakan secara langsung di lapangan, dan model yang akurat belum tentu layak diterapkan pada aplikasi offline. Oleh karena itu, evaluasi penelitian diarahkan pada akurasi, metrik per kelas, ukuran file, waktu inferensi, dan keberhasilan fungsi aplikasi.

Ruang lingkup penelitian dibatasi pada citra daun padi. Data tidak mencakup semua penyakit padi, tetapi hanya empat kelas yang digunakan dalam eksperimen, yaitu healthy, brown spot, bacterial leaf blight, dan leaf blast. Pembatasan ini membuat proses pelatihan dan evaluasi lebih terarah. Model dinilai berdasarkan kemampuannya membedakan pola visual pada kelas yang tersedia, bukan sebagai sistem diagnosis untuk seluruh penyakit tanaman padi.

Fokus utama penelitian berada pada keterkaitan antara performa model dan penerapannya. Akurasi tetap digunakan sebagai indikator keberhasilan, tetapi ukuran file dan waktu inferensi ikut menentukan apakah model layak digunakan pada aplikasi mobile offline. Kontribusi penelitian ini adalah penyatuan MobileNetV2, Dynamic Range Quantization, dan aplikasi Android offline dalam satu alur kerja, sehingga sistem yang dihasilkan tidak hanya akurat, tetapi juga lebih ringan, lebih cepat, dan dapat digunakan tanpa internet.

2. TINJAUAN PUSTAKA

2.1. Landasan Teori

Padi memiliki nilai penting sebagai sumber pangan, sehingga gangguan penyakit perlu diperhatikan sejak gejala awal muncul [1], [2].

Pada daun, perubahan yang dapat diamati meliputi bercak coklat, perubahan warna, lesi, atau bagian daun yang mengering [6].

Kelas yang digunakan dalam penelitian ini adalah healthy, brown spot, bacterial leaf blight, dan leaf blast. Keempat kelas tersebut dipilih karena sesuai dengan kondisi visual daun padi yang dibahas dalam penelitian terkait [6], [7].

Kelas healthy digunakan untuk daun tanpa gejala yang terlihat jelas. Brown spot ditunjukkan oleh bercak kecoklatan, bacterial leaf blight berkaitan dengan perubahan warna serta bagian daun yang mengering, dan leaf blast tampak melalui pola bercak atau lesi pada permukaan daun [6].

Karena beberapa gejala memiliki warna dan tekstur yang berdekatan, klasifikasi citra membutuhkan model yang mampu mengenali perbedaan pola secara konsisten [9].

Citra digital dapat dipahami sebagai representasi visual berbasis piksel yang dapat diproses komputer [10].

Pada klasifikasi penyakit daun, citra berfungsi sebagai masukan untuk membaca warna, tekstur, dan bentuk gejala [10].

Computer vision menyediakan dasar pengolahan gambar, sedangkan deep learning memungkinkan fitur penting dipelajari langsung dari data yang tersedia [10].

Convolutional Neural Network banyak digunakan pada klasifikasi gambar karena mampu mempelajari pola visual dari citra [11].

Proses utamanya melibatkan konvolusi untuk menangkap fitur, pooling untuk menyederhanakan dimensi, dan lapisan klasifikasi untuk menentukan label keluaran [11].

Karakter tersebut membuat CNN sesuai untuk tugas pengenalan citra, termasuk identifikasi penyakit tanaman melalui daun [11], [12].

MobileNetV2 termasuk arsitektur CNN ringan yang ditujukan untuk kondisi komputasi terbatas [20].

Arsitektur ini memanfaatkan depthwise separable convolution untuk menekan operasi

komputasi dan inverted residual untuk mempertahankan representasi fitur [20].

Kajian tentang keluarga MobileNet pada perangkat kecil seperti Raspberry Pi menunjukkan bahwa arsitektur ini relevan untuk sistem yang perlu berjalan pada lingkungan dengan sumber daya terbatas [21].

Pemilihan MobileNetV2 dalam penelitian ini didasarkan pada kebutuhan model yang ringan namun tetap mampu menjaga performa klasifikasi [14], [15].

Pada aplikasi Android offline, ukuran file dan waktu inferensi menjadi faktor praktis selain akurasi [18], [19].

Dengan arsitektur yang lebih ringan, model dapat diarahkan untuk tetap efektif ketika dijalankan pada perangkat pengguna [18].

Transfer learning memungkinkan model awal yang telah dilatih sebelumnya dimanfaatkan kembali, sehingga proses pembelajaran pada data baru dapat dilakukan lebih efisien [14].

Dalam penelitian ini, MobileNetV2 menjadi dasar model untuk mempelajari pola visual penyakit daun padi. Citra disesuaikan ukurannya dan dinormalisasi sebelum pelatihan agar format input sesuai dengan kebutuhan model dan proses pembelajaran lebih stabil [22].

Dynamic Range Quantization termasuk teknik post-training quantization yang diterapkan setelah model selesai dilatih [18].

Teknik ini menurunkan presisi representasi bobot dari floating point, sementara aktivasi dikuantisasi secara dinamis ketika inferensi berlangsung [18].

Karena tidak memerlukan pelatihan ulang, DRQ menjadi pilihan praktis untuk mengoptimalkan model yang sudah tersedia [18].

Pada perangkat edge, MobileNetV2 yang dikuantisasi dapat menghasilkan ukuran model lebih kecil dan proses inferensi yang lebih efisien [18].

TensorFlow Lite digunakan untuk membawa model deep learning ke perangkat mobile yang memiliki keterbatasan sumber daya [23].

Smartphone dipilih sebagai media implementasi karena mudah digunakan dalam kegiatan lapangan [24].

Pada sisi aplikasi, antarmuka perlu dibuat jelas agar pengguna dapat memilih gambar, menjalankan klasifikasi, dan membaca hasil dengan mudah [24].

Kemampuan offline menjadi penting dalam konteks pertanian karena tidak semua area penggunaan memiliki koneksi internet yang stabil [19].

2.2. Penelitian Terdahulu

Penelitian terkait menunjukkan dua arah utama: peningkatan akurasi model dan efisiensi ketika model dijalankan pada perangkat terbatas. Ensemble CNN dapat memberikan performa tinggi, tetapi kompleksitasnya berpotensi menyulitkan implementasi mobile [13].

Di sisi lain, MobileNetV2 lebih sesuai untuk aplikasi mobile karena ringan, namun ukuran model dan waktu inferensinya tetap perlu diperhatikan [14], [15].

Atas dasar itu, penelitian ini menggunakan MobileNetV2 sebagai model utama dan menambahkan DRQ sebagai tahap optimasi.

Aspek offline menjadi pembeda penting dibanding sistem yang masih mengandalkan server atau jaringan internet [14].

Pada area pertanian, keterbatasan koneksi dapat menghambat penggunaan teknologi berbasis jaringan [19].

Dengan menjalankan prediksi langsung di perangkat, aplikasi tidak bergantung pada kualitas internet. Konsekuensinya, model harus cukup kecil, cepat, dan tetap akurat, sehingga optimasi model menjadi bagian utama dari rancangan sistem [18].

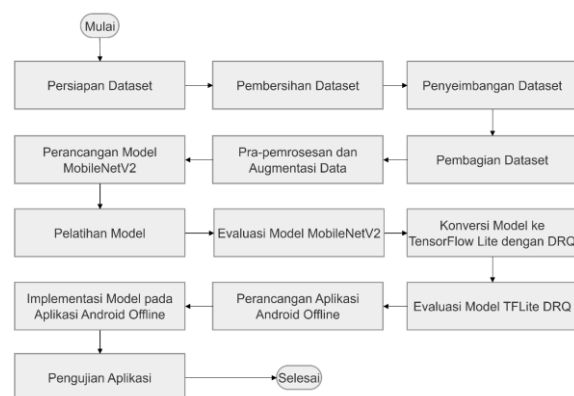
Kajian sebelumnya telah menunjukkan adanya implementasi mobile/offline untuk klasifikasi penyakit padi [14].

Di sisi lain, optimasi quantization atau DRQ pada MobileNetV2 juga telah digunakan untuk kebutuhan perangkat edge [18].

Celah yang masih dapat ditangani adalah penyusunan keduanya dalam satu alur: MobileNetV2 untuk klasifikasi penyakit daun padi, DRQ untuk efisiensi model, dan aplikasi Android offline sebagai media penggunaan. Dari celah tersebut, kontribusi penelitian diarahkan pada model yang tetap akurat, lebih ringan, lebih cepat, dan dapat berjalan tanpa koneksi internet.

3. METODE PENELITIAN

Tahapan penelitian dimulai dari penyiapan dataset, pembersihan data, penyeimbangan kelas, dan pembagian data. Setelah itu, citra dipra-pemroses dan data latih diberi augmentasi. Model MobileNetV2 kemudian dirancang, dilatih, dan dievaluasi. Model terbaik dikonversi ke TensorFlow Lite menggunakan DRQ, diuji kembali, lalu dimasukkan ke aplikasi Android offline. Dengan alur tersebut, hasil penelitian tidak hanya berupa nilai evaluasi model, tetapi juga implementasi yang dapat dijalankan pada perangkat mobile.



Gambar 1. Alur penelitian

Gambar 1 merangkum urutan kerja penelitian, mulai dari pengolahan dataset, pelatihan MobileNetV2, optimasi DRQ, sampai penerapan model pada aplikasi Android offline.

3.1. Dataset dan Pembagian Data

Dataset penelitian diambil dari Kaggle dan berisi empat kelas citra daun padi: healthy, brown spot, bacterial leaf blight, dan leaf blast. Pemeriksaan awal dilakukan untuk memastikan file yang digunakan benar-benar dapat diproses sebagai citra. Setelah data dibersihkan, jumlah citra pada setiap kelas dibuat seimbang. Hasil akhir tahap ini adalah 11.600 citra, terdiri dari 2.900 citra untuk masing-masing kelas.

Sebelum dibagi menjadi data latih, validasi, dan uji, dataset terlebih dahulu disaring. File non-gambar tidak disertakan, sedangkan citra yang gagal dibaca dikeluarkan sebagai data rusak. Pemeriksaan duplikasi dilakukan dengan hashing agar gambar identik tidak masuk berulang. Langkah ini menjaga data pelatihan dan evaluasi tetap representatif.

Penyeimbangan kelas menggunakan undersampling. Jumlah citra pada setiap kelas disamakan agar model tidak lebih banyak mempelajari kelas tertentu. Setelah proses ini, setiap kelas berisi 2.900 citra. Distribusi yang seimbang membantu pembagian data latih, validasi, dan uji memiliki komposisi kelas yang setara, sehingga hasil evaluasi lebih mudah dibandingkan.

Dataset dibagi dengan rasio 70:15:15. Dari rasio tersebut diperoleh 8.116 citra latih, 1.740 citra validasi, dan 1.744 citra uji. Pembagian dilakukan secara acak pada tiap kelas untuk menjaga variasi data. Data uji dipisahkan sejak awal, sehingga pengukuran akhir dilakukan pada citra yang tidak digunakan saat pelatihan maupun validasi.

Tabel 1. Pembagian dataset

Kelas	Total	Latih	Validasi	Uji
Healthy	2.900	2.029	435	436
Brown Spot	2.900	2.029	435	436
Bacterial Leaf Blight	2.900	2.029	435	436
Leaf Blast	2.900	2.029	435	436
Total	11.600	8.116	1.740	1.744

Tabel 1 menunjukkan pembagian dataset ke data latih, validasi, dan uji dengan jumlah kelas yang seimbang. Data latih digunakan untuk pembelajaran model, data validasi untuk memantau performa selama pelatihan, dan data uji untuk evaluasi akhir. Seluruh citra diubah menjadi 224 x 224 piksel dan dinormalisasi sebelum diproses oleh MobileNetV2.

3.2. Pra-pemrosesan dan Augmentasi Data

Augmentasi diterapkan hanya pada data latih. Transformasi yang digunakan adalah RandomFlip dan RandomRotation agar model melihat variasi orientasi daun selama pembelajaran. Data validasi dan data uji dibiarkan tanpa augmentasi karena keduanya dipakai

untuk memantau dan mengukur performa pada citra asli. Dataset juga disusun dalam batch dan prefetching untuk membantu efisiensi proses pelatihan.

Variasi augmentasi dibuat tetap wajar untuk citra daun. RandomFlip membantu model menghadapi orientasi daun yang berbeda, sedangkan RandomRotation memberi variasi sudut pengambilan gambar. Transformasi yang terlalu ekstrem tidak digunakan karena dapat mengubah ciri penyakit. Dengan cara ini, data latih menjadi lebih beragam tanpa mengaburkan label kelas.

3.3. Perancangan dan Pelatihan MobileNetV2

Tabel 2. Konfigurasi pelatihan model

Parameter	Nilai
Model	MobileNetV2
Input gambar	224 x 224 piksel
Jumlah kelas	4
Optimizer	Adam
Loss function	Sparse Categorical Crossentropy
Batch size	32
Maksimal epoch	30
Callback	EarlyStopping dan ModelCheckpoint

Tabel 2 memuat konfigurasi pelatihan yang digunakan. MobileNetV2 berperan sebagai ekstraktor fitur, optimizer yang dipakai adalah Adam, dan fungsi loss yang digunakan adalah Sparse Categorical Crossentropy. Lapisan akhir disesuaikan dengan empat kelas keluaran. EarlyStopping menghentikan pelatihan ketika performa validasi tidak membaik, sedangkan ModelCheckpoint menyimpan bobot terbaik selama proses pelatihan.

Pelatihan diarahkan agar model mempelajari pola umum citra daun padi dan tetap mampu bekerja pada data baru. Data latih digunakan untuk memperbarui bobot, sementara data validasi memantau perubahan loss dan accuracy pada setiap epoch. Bobot yang dipakai untuk evaluasi akhir berasal dari performa validasi terbaik, bukan otomatis dari epoch terakhir.

3.4. Evaluasi dan Optimasi DRQ

Pengujian model dilakukan menggunakan data uji yang tidak dilibatkan dalam pelatihan dan validasi. Metrik evaluasi meliputi accuracy, precision, recall, F1-score, dan confusion matrix. Selain itu, ukuran file model dan waktu inferensi juga diukur. Dua aspek tersebut dibutuhkan karena model akan berjalan di aplikasi Android offline, sehingga efisiensi komputasi menjadi bagian dari keberhasilan sistem.

Confusion matrix dipakai untuk membaca sebaran prediksi benar dan salah pada tiap kelas. Precision menunjukkan seberapa tepat prediksi pada suatu kelas, recall menunjukkan seberapa banyak data kelas tersebut berhasil ditemukan, dan F1-score menggambarkan keseimbangan keduanya. Kombinasi metrik ini digunakan agar performa per kelas tetap terlihat, tidak hanya bergantung pada akurasi keseluruhan.

Setelah MobileNetV2 selesai dilatih, model diuji untuk memperoleh akurasi, precision, recall, F1-score, ukuran file, dan waktu inferensi. Model terbaik kemudian dikonversi ke TensorFlow Lite dengan Dynamic Range Quantization. Model TFLite DRQ diuji kembali pada data yang sama agar perubahan performa, ukuran, dan kecepatan inferensi dapat dibandingkan secara langsung.

Konversi dilakukan melalui TensorFlow Lite Converter. Bobot terbaik dari proses pelatihan dimuat, lalu opsi DRQ diaktifkan saat konversi. File keluaran berbentuk .tflite dan tetap menerima input 224 x 224 piksel dengan tiga channel RGB serta empat kelas keluaran. File ini ditempatkan pada folder assets proyek Android agar aplikasi dapat memanggil model secara lokal.

3.5. Implementasi Aplikasi Android Offline

Di dalam aplikasi Android, gambar dari kamera atau galeri melalui tahap pemrosesan sebelum masuk ke model. Citra disesuaikan ke ukuran input, dikonversi ke RGB, dan dinormalisasi sesuai format model. Setelah inferensi selesai, probabilitas keluaran dipetakan ke label healthy, brown spot, bacterial leaf blight, dan leaf blast. Label dengan probabilitas tertinggi ditampilkan sebagai hasil prediksi.

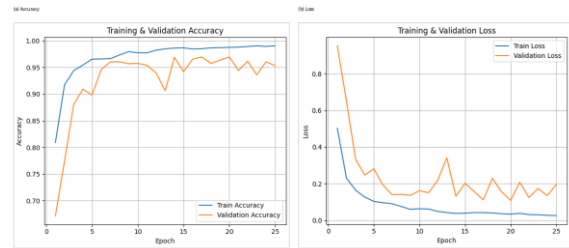
Aplikasi dikembangkan dengan Android Studio, Kotlin, Jetpack Compose, TensorFlow Lite, dan Room Database. Model TFLite DRQ disertakan di dalam aplikasi sehingga inferensi berjalan pada perangkat pengguna. Alurnya meliputi pengambilan gambar dari kamera atau galeri, pratinjau citra, klasifikasi, tampilan label, confidence, waktu inferensi, serta penyimpanan riwayat lokal. Desain ini dibuat agar pengguna tetap memperoleh hasil klasifikasi meskipun tidak tersedia koneksi internet.

Riwayat klasifikasi digunakan untuk menyimpan hasil pengujian yang sudah dilakukan. Informasi yang disimpan mencakup citra atau data hasil klasifikasi, label prediksi, confidence, dan waktu inferensi. Fitur ini memungkinkan pengguna melihat kembali hasil sebelumnya tanpa menjalankan klasifikasi ulang.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pelatihan Model MobileNetV2

Pelatihan MobileNetV2 berhenti pada epoch ke-25 dari batas maksimal 30 epoch karena EarlyStopping aktif. Bobot terbaik tercatat pada epoch ke-20 dengan validation loss 0,1086 dan validation accuracy 0,9695 atau 96,95%. Kurva akurasi memperlihatkan kenaikan performa pada data latih dan validasi, sementara kurva loss menunjukkan penurunan kesalahan selama proses pelatihan. Pola grafik memperlihatkan pelatihan yang relatif stabil. Jarak antara training accuracy dan validation accuracy masih terkendali, sehingga tidak terlihat overfitting yang dominan pada data validasi. Nilai validation loss terendah pada bobot terbaik menjadi dasar penggunaan model tersebut untuk evaluasi data uji.

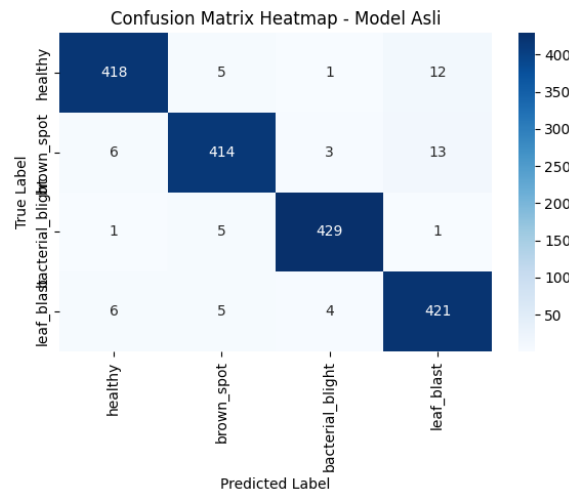


Gambar 2. Grafik training dan validation accuracy serta loss

Gambar 2 memperlihatkan akurasi yang meningkat dan loss yang menurun selama pelatihan, sehingga proses pembelajaran model dapat dibaca stabil.

4.2. Evaluasi Model MobileNetV2

MobileNetV2 diuji pada 1.744 citra. Hasilnya menunjukkan akurasi 96,44%, dengan mayoritas prediksi benar berada pada diagonal confusion matrix. Sebagian kesalahan masih terjadi pada kelas tertentu, terutama karena kemiripan bercak, warna, dan tekstur daun. Precision, recall, dan F1-score pada tiap kelas berada pada nilai tinggi. Artinya, performa model tidak hanya terlihat dari akurasi keseluruhan, tetapi juga dari kemampuan mengenali masing-masing kelas. Kelas bacterial leaf blight menghasilkan metrik yang kuat, sedangkan kesalahan tersisa terutama berkaitan dengan kemiripan gejala visual pada citra daun.



Gambar 3. Confusion matrix MobileNetV2

Gambar 3 menunjukkan bahwa prediksi MobileNetV2 terkonsentrasi pada diagonal utama, sehingga jumlah kesalahan klasifikasi relatif kecil.

Tabel 3. Metrik evaluasi MobileNetV2

Kelas	Precision	Recall	F1-score	Support
Healthy	0,9698	0,9587	0,9642	436
Brown Spot	0,9650	0,9495	0,9572	436
Bacterial Leaf Blight	0,9817	0,9839	0,9828	436
Leaf Blast	0,9418	0,9656	0,9536	436

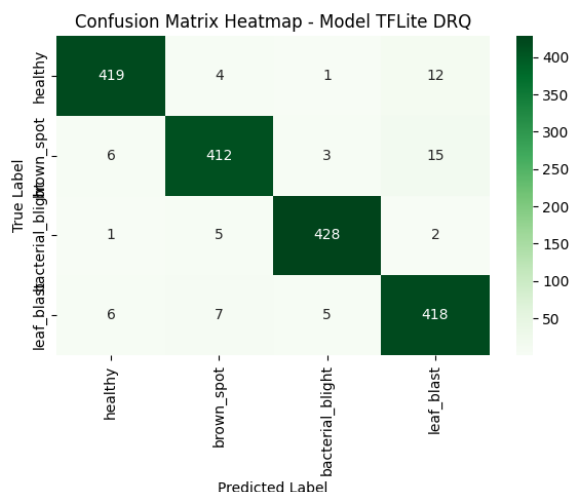
Kelas	Precision	Recall	F1-score	Support
Accuracy			0,9644	1.744
Macro Avg	0,9646	0,9644	0,9645	1.744
Weighted Avg	0,9646	0,9644	0,9645	1.744

Tabel 3 memperlihatkan bahwa precision, recall, dan F1-score MobileNetV2 berada pada tingkat tinggi untuk seluruh kelas pengujian.

4.3. Evaluasi Model TFLite DRQ

Model yang telah dikonversi ke TensorFlow Lite dengan Dynamic Range Quantization menghasilkan akurasi 96,16%. Selisih terhadap model awal hanya 0,28%, sehingga proses optimasi masih mempertahankan performa klasifikasi dalam rentang yang baik.

Confusion matrix TFLite DRQ masih memiliki pola yang dekat dengan model MobileNetV2 awal. Perubahan kecil pada beberapa kelas tidak mengubah pola utama, yaitu sebagian besar citra uji tetap masuk ke kelas yang benar. Kondisi ini menunjukkan bahwa model hasil optimasi masih layak digunakan pada aplikasi Android.



Gambar 4. Confusion matrix TFLite DRQ

Gambar 4 menunjukkan bahwa pola prediksi TFLite DRQ tetap stabil setelah optimasi dilakukan.

Tabel 4. Metrik evaluasi TFLite DRQ

Kelas	Precision	Recall	F1-score	Support
Healthy	0,9699	0,9610	0,9654	436
Brown Spot	0,9626	0,9450	0,9537	436
Bacterial Blight	0,9794	0,9817	0,9805	436
Leaf Blast	0,9351	0,9587	0,9468	436
Accuracy			0,9616	1.744
Macro Avg	0,9618	0,9616	0,9616	1.744
Weighted Avg	0,9618	0,9616	0,9616	1.744

Tabel 4 menunjukkan bahwa metrik TFLite DRQ pada tiap kelas masih mendekati performa MobileNetV2 asli.

4.4. Perbandingan Model MobileNetV2 dan TFLite DRQ

Tabel 5. Ringkasan perbandingan model

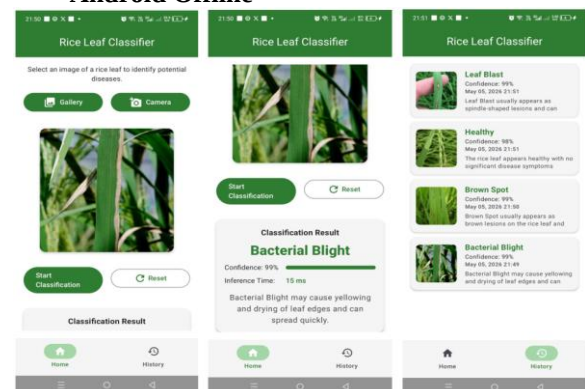
Aspek	MobileNetV2	TFLite DRQ	Perubahan
Akurasi	96,44%	96,16%	Turun 0,28%
Ukuran model	26,301 MB	2,395 MB	Turun 90,89%
Rata-rata inferensi	91,15 ms/gambar	30,86 ms/gambar	Lebih cepat 60,29 ms

Tabel 5 merangkum dampak DRQ terhadap akurasi, ukuran file, dan waktu inferensi. Ukuran model turun secara besar, inferensi menjadi lebih cepat, sedangkan akurasi hanya berkurang sedikit. Kombinasi tersebut sesuai dengan kebutuhan aplikasi mobile offline karena model perlu ringan, responsif, dan tetap memiliki performa klasifikasi yang baik.

Penurunan akurasi sebesar 0,28% setelah quantization dapat terjadi karena proses DRQ mengubah representasi bobot model dari presisi floating point ke presisi yang lebih rendah. Perubahan presisi tersebut dapat menimbulkan pembulatan nilai bobot, sehingga sebagian kecil informasi numerik yang sebelumnya digunakan model dalam membedakan pola citra dapat berubah. Meskipun demikian, penurunan akurasi yang terjadi masih kecil karena pola prediksi pada confusion matrix TFLite DRQ tetap mendekati model MobileNetV2 asli dan metrik evaluasi tiap kelas masih berada pada tingkat tinggi.

Pada sisi implementasi, ukuran model yang lebih kecil memudahkan penyimpanan model di dalam aplikasi. Waktu inferensi yang lebih singkat juga membuat proses klasifikasi lebih responsif ketika pengguna memilih atau mengambil gambar. Dengan demikian, manfaat DRQ tidak hanya terlihat pada file model, tetapi juga pada pengalaman penggunaan aplikasi.

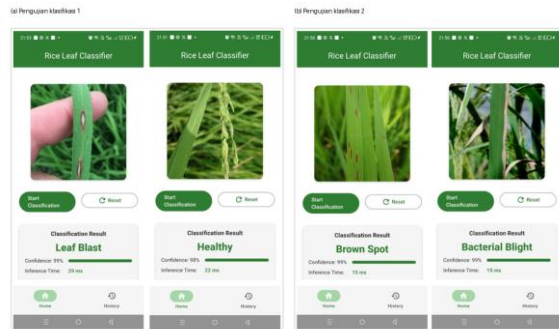
4.5. Implementasi dan Pengujian Aplikasi Android Offline



Gambar 5. Pengujian fungsi utama aplikasi

Model TFLite DRQ dimasukkan ke aplikasi Android offline. Aplikasi dapat menerima gambar dari kamera atau galeri, menampilkan pratinjau, menjalankan klasifikasi, menunjukkan label prediksi, confidence, waktu inferensi, dan menyimpan riwayat pada perangkat.

Gambar 5 memperlihatkan alur utama aplikasi, yaitu memilih atau mengambil citra, memprosesnya, lalu menampilkan hasil klasifikasi kepada pengguna.



Gambar 6. Pengujian klasifikasi 1 dan 2

Gambar 6 memperlihatkan pengujian klasifikasi pada kondisi offline. Hasil prediksi tampil langsung pada halaman utama, sedangkan riwayat klasifikasi tersimpan di perangkat. Hasil ini menunjukkan bahwa model yang telah dioptimasi dapat digunakan dalam alur aplikasi mobile, tidak hanya berhenti sebagai file model.

Pada pengujian aplikasi, label prediksi, confidence, dan waktu inferensi berhasil ditampilkan. Informasi tersebut membantu pengguna memahami hasil klasifikasi, termasuk tingkat keyakinan model. Pengujian kamera, galeri, dan riwayat menunjukkan bahwa model TFLite DRQ telah terhubung dengan alur aplikasi Android secara utuh.

5. KESIMPULAN DAN SARAN

Sistem klasifikasi penyakit daun padi berbasis MobileNetV2 berhasil dibuat dan dijalankan pada aplikasi Android offline. Pada empat kelas citra, yaitu healthy, brown spot, bacterial leaf blight, dan leaf blast, model MobileNetV2 memperoleh akurasi 96,44%. Setelah dikonversi ke TensorFlow Lite dengan Dynamic Range Quantization, akurasi menjadi 96,16% atau turun 0,28%. Optimasi tersebut menurunkan ukuran model dari 26,301 MB menjadi 2,395 MB, setara penurunan sekitar 90,89%, serta mempercepat rata-rata inferensi dari 91,15 ms/gambar menjadi 30,86 ms/gambar. Model hasil optimasi juga berhasil digunakan pada aplikasi dengan fitur kamera, galeri, pratinjau gambar, prediksi, confidence, waktu inferensi, dan riwayat lokal.

Berdasarkan keterbatasan pengujian dan hasil yang diperoleh, penelitian selanjutnya disarankan untuk menambah jumlah kelas penyakit dan variasi citra agar kemampuan model dapat diuji pada kondisi daun yang lebih beragam. Pengujian juga perlu dilakukan pada beberapa perangkat Android dengan

spesifikasi berbeda untuk melihat kestabilan waktu inferensi dan penggunaan aplikasi pada perangkat yang lebih bervariasi. Selain itu, aplikasi perlu diuji pada kondisi lapangan serta dapat dikembangkan dengan perbandingan arsitektur ringan lain seperti MobileNetV3 atau EfficientNet dan penambahan rekomendasi penanganan penyakit agar hasil klasifikasi lebih bermanfaat bagi pengguna.

DAFTAR PUSTAKA

- [1] J. Sudan *et al.*, "Explicating genetic architecture governing nutritional quality in pigmented rice," *PeerJ*, vol. 9, 2023, doi: 10.7717/peerj.15901.
- [2] A. N. M. R. Bin Rahman and J. Zhang, "Trends in rice research: 2030 and beyond," *Food Energy Secur.*, vol. 12, no. 2, Mar. 2023, doi: 10.1002/fes3.390.
- [3] P. Mohan Kumar, Ramavath Laxmi Bhargavi, Shaik Imran, and Mrs. J. Sowmya, "Rice Plant Disease Detection Using Efficient Net V2," *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 3, no. 01, pp. 31–51, Jan. 2025, doi: 10.47392/irjaeh.2025.0006.
- [4] S. Conde, S. Catarino, S. Ferreira, M. P. Temudo, and F. Monteiro, "Rice Pests and Diseases Around the World: Literature-Based Assessment with Emphasis on Africa and Asia," *Agriculture (Switzerland)*, vol. 15, no. 7, Apr. 2025, doi: 10.3390/agriculture15070667.
- [5] Z. Liu *et al.*, "Effects of Bacterial Leaf Blight on Photosynthesis and Chlorophyll a Fluorescence in Susceptible and Resistant Rice Cultivars," *Int. J. Agric. Biol.*, vol. 25, no. 2, pp. 338–344, 2021, doi: 10.17957/IJAB/15.1673.
- [6] J. Arcila-Diaz, D. Altamirano-Chavez, L. Arcila-Diaz, and C. Valdivia, "Real-time Identification of Rice Leaf Diseases using Convolutional Neural Networks," *International Journal of Computing*, vol. 23, no. 4, pp. 709–714, Jan. 2024, doi: 10.47839/ijc.23.4.3773.
- [7] K. Hu, X. Zheng, X. Su, L. Wu, Y. Liu, and Z. Deng, "Identification of rice leaf disease based on DepMulti-Net," *Front. Plant Sci.*, vol. 16, 2025, doi: 10.3389/fpls.2025.1522487.
- [8] H. Liu, Y. Cui, J. Wang, and H. Yu, "Analysis and Research on Rice Disease Identification Method Based on Deep Learning," *Sustainability (Switzerland)*, vol. 15, no. 12, Jun. 2023, doi: 10.3390/su15129321.
- [9] S. Akter, R. I. Sumon, H. Ali, and H. C. Kim, "Utilizing Convolutional Neural Networks for the Effective Classification of Rice Leaf Diseases Through a Deep Learning Approach," *Electronics (Switzerland)*, vol. 13, no. 20, Oct. 2024, doi: 10.3390/electronics13204095.
- [10] Z. Luo, "The Application and Development of Image Recognition Technology in the Field of Computer Vision," in *Highlights in Science, Engineering and Technology CSIC*, 2024.

- [11] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," Mar. 01, 2023, *MDPI*. doi: 10.3390/computation11030052.
- [12] B. Tugrul, E. Elfatimi, and R. Eryigit, "Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review," *Agriculture (Switzerland)*, vol. 12, no. 8, Aug. 2022, doi: 10.3390/agriculture12081192.
- [13] P. Pai *et al.*, "Deep learning-based automatic diagnosis of rice leaf diseases using ensemble CNN models," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-13079-z.
- [14] A. Nayak, S. Chakraborty, and D. K. Swain, "Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection," *Smart Agricultural Technology*, vol. 4, Aug. 2023, doi: 10.1016/j.atech.2023.100195.
- [15] Y. Xu, D. Li, C. Li, Z. Yuan, and Z. Dai, "LiSA-MobileNetV2: an extremely lightweight deep learning model with Swish activation and attention mechanism for accurate rice disease classification," *Front. Plant Sci.*, vol. 16, 2025, doi: 10.3389/fpls.2025.1619365.
- [16] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, Sep. 2023, doi: 10.1016/j.aiia.2023.07.001.
- [17] R. Deng *et al.*, "Automatic Diagnosis of Rice Diseases Using Deep Learning," *Front. Plant Sci.*, vol. 12, Aug. 2021, doi: 10.3389/fpls.2021.701038.
- [18] Dr. S. S. Pranav Dhole and I. Nandwanshi, "Crop Disease Classification for Edge Devices: A Quantized MobileNetV2 Approach," *IJIREICE*, vol. 13, no. 10, Oct. 2025, doi: 10.17148/ijireeice.2025.131006.
- [19] E. Mamabolo *et al.*, "Application of precision agriculture technologies for crop protection and soil health," Dec. 01, 2025, *Elsevier B.V.* doi: 10.1016/j.atech.2025.101270.
- [20] K. Ding, Z. Niu, J. Hui, X. Zhou, and F. T. S. Chan, "A Weld Surface Defect Recognition Method Based on Improved MobileNetV2 Algorithm," *Mathematics*, vol. 10, no. 19, Oct. 2022, doi: 10.3390/math10193678.
- [21] W. Glegoła, A. Karpus, and A. Przybyłek, "MobileNet family tailored for Raspberry Pi," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 2249–2258. doi: 10.1016/j.procs.2021.08.238.
- [22] H. Hamda and M. E. Wibowo, "Enhancing Image Classification Performance Using Multi CNN Feature Fusion Method," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 19, no. 3, Jul. 2025, doi: 10.22146/ijccs.98531.
- [23] D. J. Shin and J. J. Kim, "A Deep Learning Framework Performance Evaluation to Use YOLO in Nvidia Jetson Platform," *Applied Sciences (Switzerland)*, vol. 12, no. 8, Apr. 2022, doi: 10.3390/app12083734.
- [24] W. M. Ayada, M. A. Ezz, and E. Hammad, "Design Quality Criteria for Smartphone Applications Interface and its Impact on User Experience and Usability," *Wesam Ayada & Maram Hammad 339 International Design Journal*, vol. 13, no. 4, pp. 339–354, Jul. 2023, doi: 10.21608/idj.2023.305364