



Penerapan Convolutional Neural Network Berbasis Computer Vision untuk Identifikasi Varietas Padi Berdasarkan Citra Bulir Gabah

Fachriel A. Pomalingo*, Syahrial Syahrial, Mohamad Ilyas Abas, Irawan Ibrahim

Fakultas Sain dan Ilmu Komputer, Program Studi Ilmu Komputer, Universitas Muhammadiyah Gorontalo, Gorontalo, Indonesia

Email: ^{1,*}pomalingoariel29@gmail.com, ²syahrial@umgo.ac.id, ³ilyasabas@umgo.ac.id, ⁴irawan_ibrahim@umgo.ac.id

(* : corresponding author)

Abstrak-Identifikasi varietas padi berdasarkan citra bulir gabah menghadapi permasalahan karena beberapa varietas memiliki karakteristik visual yang relatif serupa, terutama pada bentuk, ukuran, dan warna, sehingga identifikasi secara manual dapat menghasilkan penilaian yang tidak konsisten. Kondisi tersebut menunjukkan perlunya pendekatan berbasis computer vision yang mampu mempelajari karakteristik visual gabah secara otomatis. Penelitian ini bertujuan mengembangkan model Convolutional Neural Network (CNN) untuk mengidentifikasi lima varietas padi yang tersedia di Badan Perakitan dan Modernisasi Pertanian (BRMP) Gorontalo, yaitu Ciherang, Inpari 30, Inpari 42, Inpari IR Nutri Zinc, dan Mekongga. Dataset diperoleh melalui akuisisi citra langsung dengan 28 foto RAW untuk setiap varietas dan sembilan hasil cropping dari setiap foto, sehingga diperoleh 1.260 citra. Tahapan penelitian meliputi preprocessing, pengembangan Custom CNN, pengujian ukuran citra, learning rate, jumlah convolutional layer, dan pembagian data, serta perbandingan dengan MobileNetV2, ResNet50, DenseNet121, dan InceptionV3. Konfigurasi terbaik menggunakan ukuran input 224×224 piksel, learning rate 0,0005, empat convolutional layer, dan pembagian data 80:20. Model yang dikembangkan dari awal memiliki 421.957 parameter trainable dan digunakan sebagai model utama dalam pengujian. Pengujian pada 135 citra final testing menghasilkan accuracy 93,33%, precision 93,44%, recall 93,33%, dan F1-score 93,32%. Hasil tersebut menunjukkan bahwa Custom CNN mampu mengidentifikasi lima varietas padi secara efektif pada kondisi dan dataset penelitian yang digunakan.

Kata Kunci: Citra Bulir Gabah; Computer Vision; Convolutional Neural Network; Identifikasi Varietas Padi; Deep Learning

Abstract-Rice variety identification based on paddy grain images faces a challenge because several varieties have relatively similar visual characteristics, particularly in shape, size, and color, which may lead to inconsistent manual identification. This condition indicates the need for a computer vision approach that can automatically learn the visual characteristics of paddy grains. This study aimed to develop a Convolutional Neural Network (CNN) model to identify five rice varieties available at the Agricultural Research and Modernization Center of Gorontalo (BRMP Gorontalo), namely Ciherang, Inpari 30, Inpari 42, Inpari IR Nutri Zinc, and Mekongga. The dataset was obtained through direct image acquisition, consisting of 28 RAW photographs for each variety and nine cropped images from each photograph, resulting in 1,260 images. The research stages included preprocessing, Custom CNN development, evaluation of image size, learning rate, number of convolutional layers, and data splitting, as well as comparison with MobileNetV2, ResNet50, DenseNet121, and InceptionV3. The best configuration used an input size of 224×224 pixels, a learning rate of 0.0005, four convolutional layers, and an 80:20 data split. Testing on 135 final testing images achieved 93.33% accuracy, 93.44% precision, 93.33% recall, and 93.32% F1-score. These results indicate that the Custom CNN effectively identified the five rice varieties under the dataset and experimental conditions used in this study.

Keywords: Paddy Grain Image; Computer Vision; Convolutional Neural Network; Rice Variety Identification; Deep Learning

1. PENDAHULUAN

Padi merupakan komoditas pangan strategis yang memiliki hubungan langsung dengan ketahanan pangan nasional. Badan Pusat Statistik melaporkan luas panen padi Indonesia sekitar 10,20 juta hektare dengan produksi sekitar 53,63 juta ton gabah kering giling pada 2023 [1]. Besarnya skala produksi tersebut membuat pengelolaan varietas, mutu benih, dan proses identifikasi varietas menjadi bagian penting dalam pengelolaan benih karena setiap varietas memiliki karakteristik dan tujuan penggunaan yang berbeda. Dalam praktiknya, pengenalan varietas masih dapat dilakukan melalui pengamatan karakter morfologi secara langsung, sedangkan perkembangan teknologi pengolahan citra memungkinkan karakteristik visual tersebut direpresentasikan secara digital dan dianalisis menggunakan metode komputasi. Kajian mutakhir mengenai klasifikasi varietas padi menunjukkan bahwa pendekatan computer vision, machine learning, dan deep learning semakin banyak digunakan untuk mendukung identifikasi otomatis, sekaligus menghadapi tantangan berupa keterbatasan dataset, variasi kondisi akuisisi, dan kebutuhan efisiensi komputasi [2], [3].

Indonesia memiliki keragaman varietas padi yang tinggi. Pada tingkat lapangan, karakter varietas dapat diamati melalui bentuk tanaman, daun, malai, maupun gabah. Pendekatan morfologi masih banyak digunakan karena karakter tersebut dapat diamati secara langsung, tetapi identifikasi manual bergantung pada pengalaman pengamat dan dapat menjadi lebih sulit ketika karakter antar varietas memiliki kemiripan [4]. Penelitian mengenai identifikasi varietas dengan pendekatan citra telah dikembangkan untuk mengurangi subjektivitas pengamatan, termasuk pemanfaatan citra daun dan citra tanaman [5]. Pendekatan berbasis citra memberikan representasi digital yang dapat diproses secara konsisten oleh algoritma komputasi.

Tantangan menjadi lebih spesifik ketika objek yang digunakan adalah citra bulir gabah. Varietas seperti Ciherang, Mekongga, Inpari 30, Inpari 42, dan Inpari IR Nutri Zinc memiliki karakter visual yang sama-sama didominasi bentuk ramping dan memanjang dengan warna kuning sampai kuning jerami. Perbedaan ukuran,



proporsi, tingkat kecerahan, dan karakter tambahan tertentu dapat berukuran kecil secara visual. Selain itu, perubahan pencahayaan, sudut pengambilan, jarak kamera, dan latar belakang dapat memengaruhi tampilan citra. Penelitian berbasis drone dan citra tanaman juga menunjukkan pentingnya kondisi akuisisi terhadap kualitas sistem identifikasi [4]. Karena objek penelitian berupa satu bulir gabah, sistem harus dapat mempelajari pola visual yang relatif halus pada area objek yang kecil.

Perkembangan computer vision dan deep learning menyediakan pendekatan yang relevan untuk persoalan tersebut. Teknik pengolahan citra tradisional umumnya melakukan segmentasi dan ekstraksi fitur secara eksplisit, misalnya melalui thresholding, deteksi tepi, atau deskriptor bentuk dan tekstur. Pendekatan semacam ini tetap berguna untuk kondisi tertentu, namun kualitas hasilnya sangat dipengaruhi oleh pemilihan fitur dan karakteristik citra [6], [7], [8]. Konsep dasar pengolahan citra dan pembelajaran mendalam yang digunakan dalam penelitian ini juga merujuk pada kajian computer vision dan deep learning [9]. CNN memiliki karakter berbeda karena fitur visual dipelajari secara bertingkat dari data. Beberapa kajian menyebutkan bahwa CNN banyak digunakan dalam klasifikasi citra karena dapat menangkap pola lokal hingga pola yang lebih kompleks melalui lapisan konvolusi dan mekanisme pooling [10], [11].

Penelitian terdahulu menunjukkan kemampuan CNN yang kuat pada berbagai objek pertanian. Yuliany dan Rachman menerapkan CNN untuk klasifikasi hama tanaman padi dan menunjukkan bahwa pola visual dapat dipelajari dari citra secara otomatis [12]. Studi mengenai jenis beras berbasis citra juga menunjukkan penggunaan CNN dan *deep learning* untuk membedakan kelas yang memiliki karakteristik visual tertentu [13], [14]. Pada dataset varietas padi, Koklu, Cinar, dan Taspınar melaporkan performa tinggi dengan metode *deep learning* pada ribuan citra varietas beras [6]. Penelitian lain yang menggunakan CNN untuk identifikasi beras serta model yang menggabungkan CNN dengan metode deteksi juga menunjukkan bahwa pemanfaatan fitur visual dapat menghasilkan performa klasifikasi yang kompetitif [15], [16].

Penelitian perbandingan arsitektur juga memperlihatkan bahwa tidak semua model memberikan respons yang sama terhadap suatu dataset. Din et al. mengembangkan RiceNet untuk klasifikasi varietas padi dan menunjukkan bahwa arsitektur CNN dapat dirancang khusus untuk karakteristik data target [17]. Bhattacharjee et al. melakukan perbandingan CNN dan pendekatan *machine learning* pada klasifikasi butir beras [18]. Azeez, Qasim, dan Mohie membahas pemrosesan citra untuk klasifikasi butir beras dan memperlihatkan bahwa representasi citra memiliki peran penting dalam proses pengenalan [19]. Sementara itu, Nugroho et al. menekankan pentingnya efisiensi sumber daya pada model CNN untuk aplikasi pertanian berbasis perangkat terbatas [20]. Hasil-hasil tersebut mendukung perlunya mempertimbangkan bukan hanya akurasi, tetapi juga kesesuaian arsitektur terhadap dataset dan konteks perangkat komputasi.

Dari sisi metodologi, terdapat pula penelitian yang membandingkan arsitektur CNN populer. Kusumawati dan Noorizki membandingkan VGG16 dan VGG19 pada klasifikasi varietas beras dan memperoleh performa tinggi pada kedua arsitektur [21]. Ridhovan dan Suharso menerapkan ResNet untuk klasifikasi penyakit daun dan menunjukkan penggunaan koneksi residual pada jaringan yang lebih dalam [22]. Xu et al. mengevaluasi pendekatan berbasis InceptionV3 pada diagnosis kerusakan motor dan menunjukkan bahwa arsitektur pralatih dapat digunakan pada tugas klasifikasi domain lain setelah penyesuaian [23]. Studi berbasis DenseNet pada citra penyakit tanaman juga memperlihatkan bahwa koneksi antarlapisan dapat digunakan untuk mempertahankan aliran informasi fitur [24]. Rajalakshmi et al. mengembangkan RiceSeedNet untuk identifikasi 13 varietas benih padi menggunakan dataset lokal dan memperoleh performa tinggi [25]. Temuan tersebut menunjukkan bahwa perbandingan arsitektur perlu dilakukan pada dataset yang sama agar pengaruh perbedaan model dapat diamati secara langsung.

Meskipun demikian, masih terdapat kesenjangan pada penelitian yang secara khusus memanfaatkan citra bulir gabah dari varietas yang tersedia pada lokasi penelitian lokal. Sebagian penelitian menggunakan dataset dari luar Indonesia atau menggunakan citra butir beras yang telah mengalami proses pascapanen, sementara objek penelitian ini adalah citra bulir gabah yang diperoleh langsung dari lokasi penelitian. Selain itu, penelitian yang membandingkan beberapa arsitektur cenderung fokus pada dataset yang sudah tersedia dan kondisi akuisisi yang relatif terstandar [17], [19]. Penelitian lokal yang secara khusus menggunakan lima varietas yang tersedia di BRMP Gorontalo serta menjaga pemisahan berdasarkan sumber foto RAW masih terbatas. Pemisahan berdasarkan sumber foto penting karena sembilan citra hasil *cropping* dari satu foto RAW memiliki hubungan yang erat dan berpotensi menghasilkan evaluasi yang terlalu optimistis apabila dipisahkan ke subset yang berbeda.

Berdasarkan kesenjangan tersebut, penelitian ini mengembangkan Custom CNN dari awal dengan arsitektur yang disesuaikan terhadap karakteristik citra bulir gabah. Model diuji secara bertahap, serta pengujian akhir terhadap data yang ditahan sejak awal. Dataset terdiri atas 28 foto RAW untuk setiap varietas, sembilan crop per foto, dan total 1.260 citra individual. Sebanyak tiga foto RAW dari setiap varietas ditahan sebagai data final testing, sedangkan 25 foto RAW per varietas digunakan untuk proses pengembangan dan validasi. Pengujian konfigurasi mencakup ukuran citra, learning rate, jumlah convolutional layer, dan rasio pembagian data training-validation. Setelah konfigurasi utama diperoleh, Custom CNN dibandingkan dengan MobileNetV2, ResNet50, DenseNet121, dan InceptionV3 pada data pengujian yang sama. Selain akurasi keseluruhan, penelitian juga



menganalisis performa setiap varietas menggunakan precision, recall, dan F1-score serta menelusuri pola kesalahan klasifikasi antarvarietas.

Pentingnya penelitian ini juga berkaitan dengan modernisasi pengujian varietas dan mutu benih padi lokal. Sistem identifikasi berbasis citra dapat menjadi pendekatan pendukung dalam proses pemeriksaan karena hasil pengamatan dapat direpresentasikan dalam bentuk digital dan diproses menggunakan model yang sama pada setiap sampel. Dalam konteks BRMP Gorontalo, pendekatan tersebut dapat menjadi dasar awal untuk pengembangan sistem identifikasi yang lebih terukur terhadap varietas yang tersedia pada lokasi penelitian.

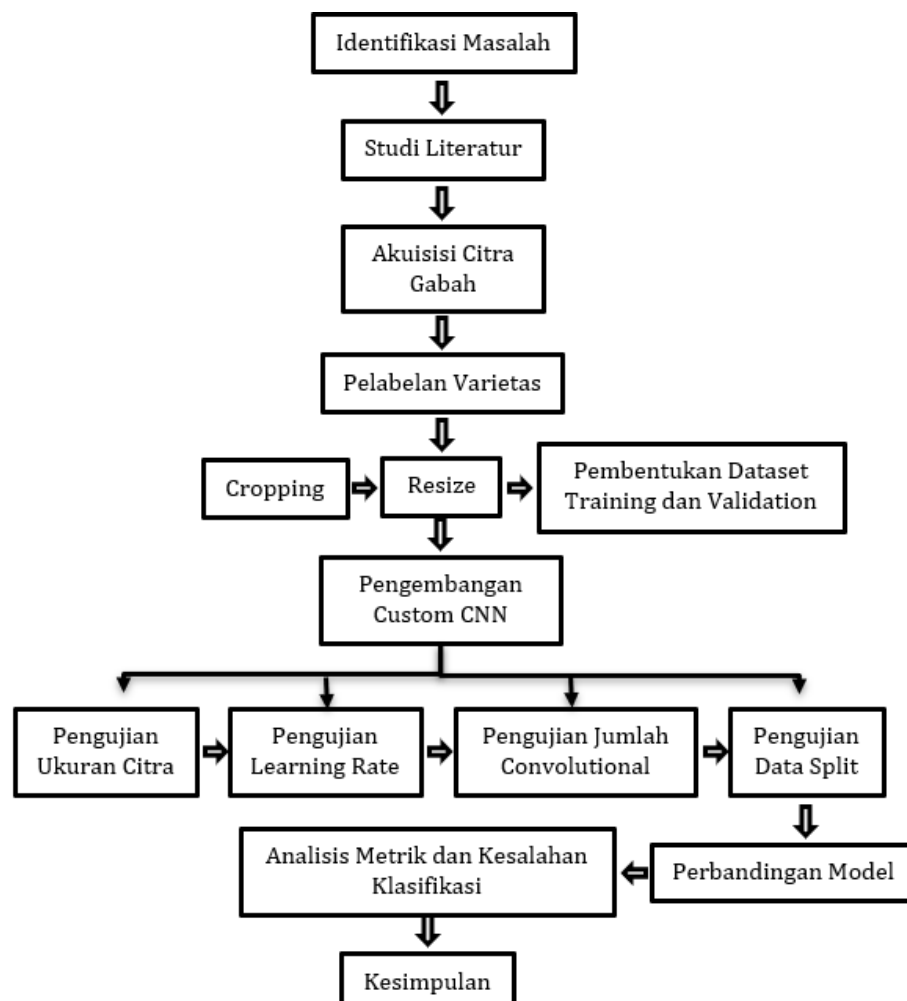
Kontribusi penelitian ini adalah tiga hal. Pertama, membangun dataset lokal yang terdiri atas lima varietas padi dari BRMP Gorontalo dengan pemisahan data berdasarkan foto RAW untuk mengurangi resiko kebocoran data. Kedua, mengembangkan Custom CNN melalui eksperimen terkontrol terhadap ukuran citra, learning rate, kedalaman konvolusi, dan rasio data. Ketiga, memberikan evaluasi komprehensif melalui pengujian data terpisah, perbandingan dengan empat arsitektur pralatih, analisis per kelas, analisis kesalahan, serta evaluasi tambahan terhadap efisiensi model.

Tujuan penelitian ini adalah mengembangkan model CNN berbasis computer vision untuk mengidentifikasi lima varietas padi berdasarkan citra bulir gabah, mengetahui performa model menggunakan accuracy, precision, recall, dan F1-score, serta membandingkan performanya dengan beberapa arsitektur CNN pralatih pada data pengujian yang sama.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui rangkaian tahapan yang mencakup identifikasi masalah, studi pustaka, akuisisi citra, pelabelan varietas, preprocessing, pembentukan dataset pengembangan dan final testing, perancangan Custom CNN, pelatihan dan validasi, pengujian konfigurasi, perbandingan model, serta analisis hasil. Alur penelitian digunakan untuk memastikan bahwa proses pengembangan model dilakukan secara bertahap dan setiap keputusan konfigurasi didasarkan pada hasil eksperimen sebelumnya.



Gambar 1. Kerangka tahapan penelitian



Gambar 1 memperlihatkan hubungan antara permasalahan, data citra gabah, preprocessing, model CNN, dan klasifikasi varietas. Setelah data terkumpul, penelitian tidak langsung menggunakan seluruh citra untuk pelatihan. Data sumber terlebih dahulu dikelompokkan berdasarkan varietas dan nomor foto RAW. Tiga foto RAW pada setiap varietas ditahan sebagai *final testing*. Dengan demikian, data pengujian akhir tidak digunakan untuk memilih ukuran citra, learning rate, jumlah lapisan konvolusi, ataupun rasio training-validation.

2.2 Data dan Akuisisi Citra

Data penelitian diperoleh melalui observasi dan akuisisi citra secara langsung di BRMP Gorontalo pada periode penelitian Januari 2025 sampai September 2026. Objek penelitian terdiri atas lima varietas padi, yaitu Ciherang, Inpari 30, Inpari 42, Inpari IR Nutri Zinc, dan Mekongga. Akuisisi citra menggunakan kamera smartphone dengan bantuan tripod untuk menjaga kestabilan posisi kamera. Pengambilan citra dilakukan dengan memperhatikan pencahayaan yang cukup, posisi kamera relatif tegak lurus terhadap objek, serta jarak kamera yang dijaga agar tampilan objek relatif konsisten.

Setiap varietas memiliki 28 foto RAW dan setiap foto RAW menghasilkan sembilan citra hasil *cropping*, sehingga terdapat 252 citra per varietas dan 1.260 citra secara keseluruhan. Pembentukan sampel individual dilakukan karena unit klasifikasi penelitian adalah satu bulir gabah. Untuk mencegah kebocoran informasi antar subset, sembilan *crop* yang berasal dari satu foto RAW selalu ditempatkan pada subset yang sama. Tiga foto RAW per varietas digunakan untuk *final testing* atau 27 citra per kelas. 25 foto RAW per varietas digunakan untuk eksperimen pengembangan atau 225 citra per kelas. Komposisi dataset dan distribusi sampel per varietas diringkas pada Tabel 1. Tabel tersebut menyajikan distribusi citra berdasarkan varietas dan pembagian penggunaannya pada tahap training, validation, dan final testing.

Tabel 1. Distribusi dataset citra bulir gabah

Varietas	RAW	Training	Validation	Testing	Total
Ciherang	28	180	45	27	252
Inpari 30	28	180	45	27	252
Inpari 42	28	180	45	27	252
Inpari IR Nutri Zinc	28	180	45	27	252
Mekongga	28	180	45	27	252
Total	140	900	225	135	1.260

Tabel 1 menunjukkan bahwa setiap varietas memiliki jumlah citra yang seimbang. Keseimbangan tersebut memungkinkan perbandingan antarkelas dilakukan tanpa perbedaan jumlah sampel sebagai faktor dominan. Data final testing tidak digunakan selama proses pengembangan dan pemilihan konfigurasi model.

2.3 Preprocessing dan Pembagian Data

Preprocessing diarahkan untuk memastikan seluruh sampel memiliki bentuk input yang konsisten dan fokus pada objek. Tahap pertama adalah *cropping* dari foto RAW menjadi satu bulir gabah per citra. Tahap kedua adalah *resize* ke ukuran input yang diuji, yaitu 100×100, 150×150, 224×224, dan 320×320 piksel. Pada konfigurasi terpilih, ukuran akhir adalah 224×224 piksel dengan tiga kanal RGB. Tahap ketiga adalah normalisasi nilai piksel menggunakan pembagian 255 melalui lapisan *Rescaling* pada model.

Data pengembangan berjumlah 1.125 citra yang berasal dari 25 foto RAW per varietas. Pengujian rasio *training-validation* dilakukan pada lima skenario, yaitu 90:10, 80:20, 70:30, 60:40, dan 50:50. Karena pembagian dilakukan pada satuan foto RAW, jumlah foto sumber membuat beberapa rasio memiliki proporsi aktual yang sedikit berbeda dari target. Skenario 80:20 menggunakan 20 foto RAW untuk *training* dan 5 foto RAW untuk *validation* pada setiap varietas, menghasilkan 900 citra *training* dan 225 citra *validation*. Data 135 citra *final testing* tidak dilibatkan pada proses pemilihan konfigurasi maupun validasi model.

2.4 Arsitektur dan Pelatihan Custom CNN

Custom CNN dibangun dari awal dengan input 224×224×3. Arsitektur terdiri atas empat blok konvolusi yang masing-masing menggunakan Conv2D dengan kernel 3×3, aktivasi ReLU, dan *padding same*. Jumlah filter meningkat dari 32, 64, 128, dan 256 pada empat lapisan berturut-turut. Setiap lapisan konvolusi diikuti MaxPooling2D berukuran 2×2 untuk mengurangi dimensi spasial. Setelah blok ekstraksi fitur, GlobalAveragePooling2D mereduksi representasi fitur menjadi vektor 256 dimensi, kemudian diproses oleh Dense 128 dengan aktivasi ReLU, Dropout 0,20, dan Dense 5 dengan aktivasi Softmax sebagai classifier lima kelas.

Tabel 2. Arsitektur Custom CNN

Layer	Output shape	Parameter	Fungsi utama
Input + Rescaling	224×224×3	0	Menyeragamkan input dan normalisasi
Conv2D 32 + MaxPool	112×112×32	896	Ekstraksi fitur awal
Conv2D 64 + MaxPool	56×56×64	18.496	Ekstraksi fitur tingkat menengah



Layer	Output shape	Parameter	Fungsi utama
Conv2D 128 + MaxPool	28×28×128	73.856	Mempelajari pola visual lebih kompleks
Conv2D 256 + MaxPool	14×14×256	295.168	Ekstraksi fitur tingkat lanjut
GlobalAveragePooling2D	256	0	Peringkasan feature map
Dense 128 + Dropout 0,20	128	32.896	Representasi dan regularisasi
Dense 5 Softmax	5	645	Klasifikasi lima varietas
Total	-	421.957	Seluruh parameter trainable

Tabel 2 menunjukkan bahwa Custom CNN memiliki 421.957 parameter trainable. Jumlah tersebut lebih terarah dibandingkan jaringan pralatih yang membawa struktur dan bobot dari dataset skala besar. Seluruh parameter Custom CNN dipelajari dari dataset penelitian. Fungsi aktivasi ReLU digunakan untuk menghasilkan non-linearitas pada jaringan dan membantu pembelajaran representasi fitur yang lebih kompleks [26]. Dropout digunakan sebagai teknik regularisasi dengan menonaktifkan sebagian unit selama pelatihan untuk mengurangi ketergantungan berlebihan antarfitur [27]. Optimasi model menggunakan Adam dengan learning rate awal 0,0005. Adam merupakan metode optimasi berbasis estimasi momen gradien yang dirancang untuk memperbarui parameter secara adaptif [28]. Lapisan Softmax pada classifier akhir menghasilkan distribusi probabilitas terhadap lima kelas yang kemudian digunakan untuk menentukan kelas dengan probabilitas tertinggi.

Seluruh eksperimen dilakukan dengan maksimum 50 epoch, batch size 16, dan mekanisme ReduceLROnPlateau untuk menurunkan learning rate ketika performa validasi tidak menunjukkan peningkatan. Early stopping tidak digunakan sehingga seluruh eksperimen berjalan sampai 50 epoch. Model terbaik ditentukan berdasarkan validation accuracy.

2.5 Rancangan Eksperimen

Pengujian konfigurasi dilakukan secara berurutan. Pengujian pertama mengevaluasi ukuran citra dengan *learning rate* 0,0001. Ukuran 224×224 kemudian digunakan untuk membandingkan *learning rate* 0,0001, 0,0005, dan 0,001. Nilai terbaik digunakan pada pengujian jumlah *convolutional layer* dengan empat kedalaman jaringan. Konfigurasi hasil tahap tersebut digunakan pada pengujian pembagian data. Setelah konfigurasi utama terbentuk, Custom CNN dibandingkan dengan MobileNetV2, ResNet50, DenseNet121, dan InceptionV3. Semua model pembanding diuji pada ukuran 224×224 dan pembagian 80:20 menggunakan data *final testing* yang sama. Pendekatan berurutan ini menjaga agar satu tahap eksperimen tidak mencampurkan variabel dari tahap sebelumnya.

2.6 Metrik Evaluasi

Evaluasi menggunakan *accuracy*, *precision*, *recall*, dan F1-score. Untuk klasifikasi multikelas, *accuracy* dihitung dari jumlah prediksi benar pada seluruh kelas dibandingkan jumlah sampel. K merupakan jumlah kelas dan TP_i adalah jumlah sampel kelas ke- i yang diklasifikasikan benar.

$$\text{Accuracy} = (\sum_{i=1}^K TP_i) / N \quad (1)$$

Precision mengukur proporsi prediksi suatu kelas yang benar terhadap seluruh prediksi yang masuk ke kelas tersebut, sedangkan *recall* mengukur proporsi sampel aktual suatu kelas yang berhasil dikenali. F1-score digunakan sebagai ukuran gabungan berbasis rata-rata harmonik antara *precision* dan *recall* [29]

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

Selain metrik pengujian akhir, proses pengembangan diamati melalui *training accuracy*, *validation accuracy*, *validation loss*, dan *accuracy gap*. *Accuracy gap* digunakan untuk melihat selisih antara *training accuracy* dan *validation accuracy* sehingga pemilihan konfigurasi tidak hanya mengandalkan satu angka akurasi [30]

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Data dan Pembentukan Dataset

Hasil akuisisi membentuk 1.260 citra individual dari lima varietas. Setiap varietas memiliki 252 citra yang berasal dari 28 foto RAW dan 9 hasil *cropping* per foto. Distribusi data pada Tabel 1 memperlihatkan pembagian akhir 900 citra *training*, 225 citra *validation*, dan 135 citra *final testing*. Pembagian yang seimbang pada setiap kelas memberikan jumlah dukungan yang sama ketika metrik makro dihitung. Data *final testing* diletakkan terpisah dari data pengembangan sejak awal, sehingga hasil pengujian akhir tidak digunakan untuk menentukan ukuran citra, *learning rate*, jumlah lapisan, atau pembagian *training-validation*.



Pemisahan berdasarkan foto RAW juga penting untuk mengurangi risiko keterkaitan langsung antara data pelatihan dan validasi. Setiap foto RAW memiliki 9 *crop* yang berasal dari sudut, pencahayaan, dan kondisi objek yang sama. Karena itu, memasukkan sebagian *crop* dari satu RAW ke *training* dan sebagian lainnya ke *validation* dapat membuat model dievaluasi pada citra yang sangat mirip dengan data yang dipelajari. Penelitian ini menempatkan seluruh sembilan *crop* dari satu RAW pada subset yang sama. Dengan demikian, perubahan performa pada setiap skenario *split* lebih mencerminkan perbedaan sumber citra yang digunakan untuk pembelajaran dan evaluasi.

3.2 Hasil Pengembangan Custom CNN

Custom CNN hasil pengembangan memiliki empat tahap ekstraksi fitur konvolusional. Filter 32 pada lapisan awal digunakan untuk menangkap pola dasar, kemudian meningkat menjadi 64, 128, dan 256 untuk membentuk representasi yang lebih kaya. MaxPooling2D pada setiap blok mengurangi resolusi spasial secara bertahap, dari 224×224 menjadi 14×14 pada keluaran blok terakhir. GlobalAveragePooling2D kemudian merangkum setiap feature map menjadi satu nilai sehingga jumlah fitur yang diteruskan ke Dense tidak bergantung pada ukuran spasial. Lapisan Dense 128 berfungsi sebagai representasi klasifikasi, sedangkan Dropout 0,20 memberikan regularisasi sebelum classifier lima kelas.

Jumlah parameter model sebesar 421.957 relatif terarah untuk masalah lima kelas dengan input 224×224. Seluruh parameter bersifat trainable, berbeda dari model pralatih yang membawa bobot awal dari dataset besar. Pengembangan dari awal memungkinkan model mempelajari karakteristik citra yang spesifik terhadap bulir gabah penelitian, tetapi juga membuat performa sangat bergantung pada jumlah dan keragaman data yang tersedia. Pengaruh konfigurasi tersebut kemudian diamati melalui eksperimen bertahap.

3.3 Hasil Pengujian Konfigurasi

3.3.1 Pengaruh Ukuran Citra

Ukuran citra merupakan variabel pertama yang diuji karena menentukan jumlah informasi spasial yang diterima model. Empat ukuran dibandingkan dengan *learning rate* 0,0001. Hasil pada Tabel 3 menunjukkan bahwa ukuran 224×224 menghasilkan *validation accuracy* 86,22%, lebih tinggi daripada 100×100 sebesar 82,67%, 150×150 sebesar 84,44%, dan 320×320 sebesar 84,44%. Pada saat yang sama, *validation loss* 224×224 sebesar 0,3448 menjadi yang terendah pada tahap ini. Nilai *accuracy gap* juga relatif paling kecil secara absolut, yaitu -2,56%.

Tabel 3. Hasil pengujian ukuran citra

Ukuran	Train Acc.	Val. Acc.	Val. Loss	Gap
100×100	74,11%	82,67%	0,5094	-8,56%
150×150	76,56%	84,44%	0,4705	-7,89%
224×224	83,67%	86,22%	0,3448	-2,56%
320×320	75,22%	84,44%	0,4872	-9,22%

Tabel 3 menunjukkan bahwa penambahan resolusi tidak selalu meningkatkan performa. Ukuran 100×100 menghasilkan representasi objek yang lebih ringkas sehingga sebagian detail bentuk dapat berkurang. Peningkatan menjadi 224×224 memberikan informasi visual yang lebih memadai sekaligus mempertahankan beban komputasi yang terukur. Pada 320×320, *training accuracy* justru turun menjadi 75,22% dan *accuracy gap* menjadi -9,22%. Kondisi ini menunjukkan bahwa resolusi yang lebih besar tidak secara otomatis memperbaiki pembelajaran model. Temuan tersebut sejalan dengan gagasan bahwa kualitas representasi citra perlu disesuaikan dengan kapasitas model dan karakter objek yang dipelajari [31], [32].

3.3.2 Pengaruh Learning Rate

Setelah ukuran 224×224 ditetapkan, tiga nilai *learning rate* dibandingkan. Tabel 4 menunjukkan bahwa nilai 0,0005 menghasilkan *validation accuracy* 88,00% dan *validation loss* 0,3033, lebih baik daripada 0,0001 dengan akurasi 84,89% dan loss 0,4370. Nilai 0,001 menghasilkan akurasi validasi 72,96% dan loss 0,8126. Perubahan tersebut memperlihatkan bahwa proses optimasi sensitif terhadap ukuran langkah pembaruan bobot. *Learning rate* 0,0005 memberikan titik yang lebih sesuai pada eksperimen penelitian ini sehingga dipertahankan untuk pengujian berikutnya.

Tabel 4. Hasil pengujian learning rate

Learning rate	Train Acc.	Val. Acc.	Val. Loss	Gap
0,0001	77,33%	84,89%	0,4370	-7,56%
0,0005	79,56%	88,00%	0,3033	-8,44%
0,001	76,06%	72,96%	0,8126	3,10%

Tabel 4 menunjukkan bahwa kenaikan *learning rate* dari 0,0001 menjadi 0,0005 meningkatkan akurasi validasi dan menurunkan loss. Sebaliknya, penggunaan 0,001 menyebabkan akurasi validasi menurun cukup jauh



dan *validation loss* meningkat. Pola tersebut memperlihatkan bahwa nilai *learning rate* perlu disesuaikan dengan skala dataset dan arsitektur jaringan. Pemilihan 0,0005 bukan semata-mata karena angka akurasi tertinggi, tetapi karena terjadi bersamaan dengan penurunan loss dibandingkan dua nilai lainnya.

3.3.3 Pengaruh Jumlah Convolutional Layer

Dengan ukuran 224×224 dan *learning rate* 0,0005, pengujian berikutnya membandingkan empat kedalaman jaringan. Tabel 5 memperlihatkan tren peningkatan performa ketika jaringan bertambah dari satu sampai empat lapisan. *Validation accuracy* meningkat dari 76,44% pada satu layer menjadi 88,44%, 94,67%, dan 95,56% pada dua, tiga, dan empat layer. *Validation loss* juga turun dari 0,6673 menjadi 0,2796, 0,1498, dan 0,1421. Hasil ini menunjukkan bahwa pada ruang eksperimen penelitian, kedalaman empat lapisan menyediakan kapasitas ekstraksi fitur yang paling sesuai.

Tabel 5. Hasil pengujian jumlah layer

Layer	Best Epoch	Train Acc.	Val. Acc.	Val. Loss	Gap
1	47	67,44%	76,44%	0,6673	-9,00%
2	46	88,33%	88,44%	0,2796	-0,11%
3	48	92,89%	94,67%	0,1498	-1,78%
4	48	92,78%	95,56%	0,1421	-2,78%

Tabel 5 menunjukkan bahwa penambahan lapisan tidak hanya meningkatkan jumlah parameter, tetapi juga memperluas tingkat abstraksi fitur yang dapat dipelajari. Pada satu layer, representasi yang terbentuk belum cukup untuk memisahkan kelas dengan karakter visual yang berdekatan. Peningkatan menjadi empat layer menghasilkan *validation accuracy* tertinggi dan loss terendah. Nilai *accuracy gap* -2,78% masih menunjukkan selisih yang kecil antara performa *training* dan *validation*, sehingga konfigurasi tersebut digunakan pada tahap pembagian data.

3.3.4 Pengaruh Pembagian Data

Setelah ukuran, *learning rate*, dan jumlah layer ditetapkan, lima skenario pembagian *training-validation* diuji. Tabel 6 menunjukkan bahwa skenario 90:10 dan 80:20 sama-sama menghasilkan *validation accuracy* 95,56%. Perbedaannya muncul pada *validation loss*: skenario 80:20 menghasilkan 0,1421, sedangkan 90:10 sebesar 0,1881. Skenario 70:30, 60:40, dan 50:50 masing-masing menghasilkan 89,17%, 93,11%, dan 91,48% *validation accuracy*. Berdasarkan kombinasi akurasi dan loss, skenario 80:20 digunakan sebagai konfigurasi akhir.

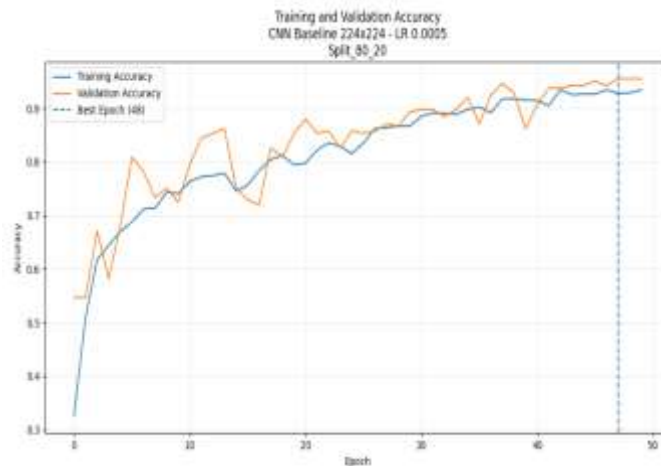
Tabel 6. Hasil pengujian pembagian data

Split	Validation accuracy	Validation loss
90:10	95,56%	0,1881
80:20	95,56%	0,1421
70:30	89,17%	0,2879
60:40	93,11%	0,2029
50:50	91,48%	0,2074

Tabel 6 menunjukkan pemilihan 80:20 juga memberikan jumlah data validasi yang lebih besar daripada 90:10 tanpa mengurangi jumlah data *training* secara drastis. Pada penelitian ini, pembagian 80:20 diwujudkan dengan 20 foto RAW untuk *training* dan 5 foto RAW untuk *validation* pada setiap varietas. Penggunaan foto sumber yang berbeda antar subset membuat validasi lebih bermakna dari pada sekadar memperbanyak *crop* dari sumber yang sama. Prinsip evaluasi berbasis sumber data ini penting pada dataset citra karena korelasi antar sampel dapat membuat metrik terlihat lebih tinggi apabila tidak dikendalikan. Dengan demikian, konfigurasi akhir yang dipilih adalah ukuran citra 224×224 piksel, *learning rate* 0,0005, empat convolutional layer, dan pembagian data 80:20.

3.4 Analisis Kurva Accuracy dan Loss Selama 50 Epoch

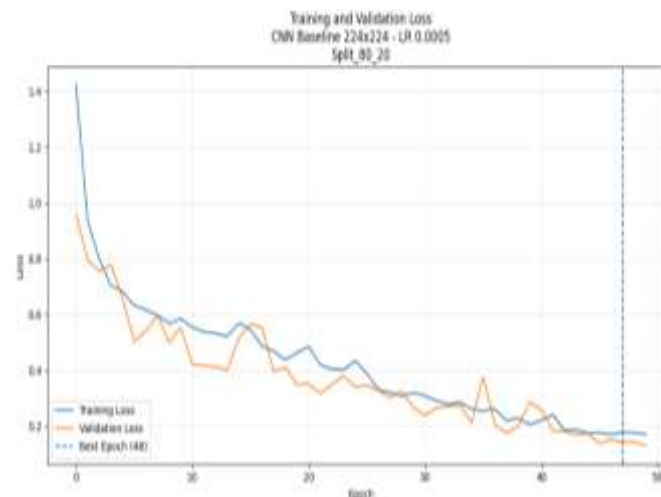
Pengamatan terhadap kurva pelatihan diperlukan untuk mengetahui pola pembelajaran model secara keseluruhan, bukan hanya berdasarkan satu nilai akurasi. Karena pelatihan Custom CNN dilakukan selama 50 epoch, kurva *training accuracy*, *validation accuracy*, *training loss*, dan *validation loss* perlu ditampilkan sebagai dua gambar terpisah. Perkembangan akurasi Custom CNN selama 50 epoch ditunjukkan pada Gambar 2 dengan membandingkan *training accuracy* dan *validation accuracy* pada setiap epoch.



Gambar 2. Kurva Training dan Validation Accuracy Custom CNN Selama 50 Epoch

Pada konfigurasi akhir, best epoch diperoleh pada epoch 48. Pada titik tersebut, training accuracy mencapai 92,78%, sedangkan validation accuracy mencapai 95,56%. Selisih keduanya sebesar -2,78 poin persentase menunjukkan bahwa akurasi validasi berada sedikit di atas akurasi pelatihan. Pola tersebut tidak menunjukkan gejala overfitting konvensional pada best checkpoint. Pada kondisi overfitting, performa training biasanya jauh lebih tinggi daripada validasi sementara performa validasi mengalami penurunan. Pada penelitian ini, validation accuracy justru sedikit lebih tinggi daripada training accuracy. Salah satu faktor yang dapat menyebabkan kondisi tersebut adalah penggunaan Dropout 0,20 selama pelatihan, sedangkan Dropout tidak aktif pada proses validasi [27]. Dengan demikian, nilai validation accuracy yang lebih tinggi tidak secara langsung dapat diartikan bahwa model mengalami underfitting.

Best epoch pada 48 dari 50 epoch menunjukkan bahwa peningkatan performa validasi telah mendekati titik optimum menjelang akhir pelatihan. Dua epoch berikutnya tidak menghasilkan validation accuracy yang lebih tinggi daripada checkpoint terbaik. Oleh karena itu, penggunaan model pada epoch 48 lebih tepat dibandingkan menggunakan bobot pada epoch terakhir secara otomatis. Perubahan nilai loss selama proses pembelajaran Custom CNN ditunjukkan pada Gambar 4 untuk memperlihatkan tingkat kesalahan prediksi model terhadap data training dan validation.



Gambar 3. Kurva Training dan Validation Loss Custom CNN Selama 50 Epoch

Kurva loss memberikan informasi tambahan terhadap kurva akurasi. Penurunan training loss menunjukkan bahwa model semakin mampu menyesuaikan bobot terhadap data pelatihan, sedangkan penurunan validation loss menunjukkan perbaikan kemampuan terhadap data yang tidak digunakan untuk pembaruan bobot. Pada konfigurasi terbaik, validation loss mencapai 0,1421 pada best epoch 48. Tidak ditemukannya gap validasi yang besar pada checkpoint terbaik memberikan indikasi bahwa model tidak mengalami ketidakseimbangan generalisasi yang ekstrem. Namun, pengamatan terhadap dua epoch setelah best epoch tetap penting karena overfitting dapat terjadi ketika training loss terus menurun sementara validation loss mulai meningkat. Oleh sebab itu, analisis kurva digunakan bersama validation accuracy, validation loss, dan accuracy gap, bukan berdasarkan satu metrik saja.



Hasil tersebut juga menunjukkan bahwa konfigurasi Custom CNN memiliki kapasitas yang cukup untuk mempelajari pola visual lima varietas tanpa memperlihatkan indikasi underfitting yang kuat. Model mampu mencapai validation accuracy 95,56%, tetapi tetap memiliki ruang untuk meningkatkan kemampuan generalisasi pada varietas yang lebih sulit dibedakan pada final testing.

3.5 Perbandingan Model

Setelah konfigurasi Custom CNN ditetapkan, empat arsitektur pralatih digunakan sebagai pembanding, yaitu MobileNetV2, ResNet50, DenseNet121, dan InceptionV3. Perbandingan dilakukan pada ukuran input 224×224 dan split 80:20. Berbeda dengan proses seleksi konfigurasi yang menggunakan data validasi, perbandingan model menggunakan 135 citra *final testing* yang tidak dilibatkan pada proses pengembangan. Hasil pada Tabel 7 menunjukkan perbedaan performa yang cukup jelas antararsitektur.

Tabel 7. Perbandingan hasil pengujian model

Model	Input	Split	Accuracy	Precision	Recall	F1-score
Custom CNN	224×224	80:20	93,33%	93,44%	93,33%	93,32%
MobileNetV2	224×224	80:20	83,70%	83,75%	83,70%	83,57%
ResNet50	224×224	80:20	90,37%	90,44%	90,37%	90,13%
DenseNet121	224×224	80:20	90,37%	90,62%	90,37%	90,43%
InceptionV3	224×224	80:20	68,89%	68,36%	68,89%	68,31%

Custom CNN memperoleh *testing accuracy* 93,33%, diikuti ResNet50 dan DenseNet121 yang sama-sama 90,37%, MobileNetV2 83,70%, dan InceptionV3 68,89%. Pada *precision*, Custom CNN mencapai 93,44%; DenseNet121 90,62%; ResNet50 90,44%; MobileNetV2 83,75%; dan InceptionV3 68,36%. Nilai *recall* dan F1-score menunjukkan pola yang serupa. Selisih akurasi Custom CNN terhadap ResNet50 dan DenseNet121 adalah 2,96 poin persentase, terhadap MobileNetV2 9,63 poin persentase, dan terhadap InceptionV3 24,44 poin persentase.

Perbedaan hasil tersebut tidak dapat ditafsirkan hanya dari kedalaman atau jumlah parameter model. MobileNetV2 dirancang untuk efisiensi melalui *depthwise separable convolution*, sedangkan ResNet menggunakan koneksi residual dan DenseNet menggunakan koneksi padat antarlapisan [22], [33], [24]. InceptionV3 juga memakai strategi faktorisasi konvolusi untuk mengurangi beban komputasi [23]. Arsitektur tersebut membawa struktur representasi yang terbentuk pada dataset pralatih dan kemudian disesuaikan dengan data penelitian. Custom CNN, sebaliknya, mempelajari seluruh parameter dari data lokal sejak awal. Perbedaan tersebut menunjukkan bahwa model yang lebih kompleks tidak secara otomatis menghasilkan performa pengujian yang lebih tinggi pada dataset tertentu.

3.6 Evaluasi Inference Time dan Ukuran Model

Untuk mengevaluasi efisiensi model, selain performa klasifikasi dilakukan pengukuran waktu inferensi (*inference time*) dan ukuran berkas model. Pengujian menggunakan model dalam format TensorFlow Lite pada lingkungan perangkat keras dan perangkat lunak yang sama untuk seluruh arsitektur, dengan ukuran masukan 224 × 224 piksel dan batch size 1. Seluruh model diuji pada CPU menggunakan satu thread dan sepuluh kali pengulangan pengukuran. Waktu pemuatan model, pembacaan dan decoding berkas citra, serta proses resize tidak dimasukkan ke dalam pengukuran. Interval pengukuran mencakup preprocessing khusus arsitektur apabila dilakukan di luar model, penyesuaian tipe data masukan, pemasukan tensor, eksekusi interpreter TensorFlow Lite, dan pengambilan keluaran prediksi. Adapun preprocessing yang tertanam di dalam model turut dihitung sebagai bagian dari eksekusi interpreter. Ukuran model dihitung berdasarkan ukuran berkas TensorFlow Lite hasil konversi dalam satuan megabita (MB).

Tabel 8. Perbandingan inference time dan ukuran model

Model	Accuracy	Inference Time/Citra (ms)	Ukuran Model (.tflite) (MB)
Custom CNN	93,33%	22,6940	1,6942
MobileNetV2	83,70%	9,9549	9,5530
ResNet50	90,37%	104,6356	95,0357
DenseNet121	90,37%	81,1967	28,4656
InceptionV3	68,89%	74,6820	88,2238

Berdasarkan Tabel 8, Custom CNN memperoleh akurasi tertinggi sebesar 93,33% dengan ukuran model terkecil, yaitu 1,6942 MB, dan waktu inferensi rata-rata sebesar 22,6940 ms per citra. MobileNetV2 menghasilkan waktu inferensi terendah sebesar 9,9549 ms per citra, tetapi akurasinya sebesar 83,70%, lebih rendah 9,63 poin persentase dibandingkan Custom CNN. ResNet50 dan DenseNet121 masing-masing memperoleh akurasi sebesar 90,37%. Namun, DenseNet121 memiliki ukuran model lebih kecil dan waktu inferensi lebih rendah dibandingkan ResNet50. Sementara itu, InceptionV3 memperoleh akurasi sebesar 68,89% dengan ukuran model 88,2238 MB

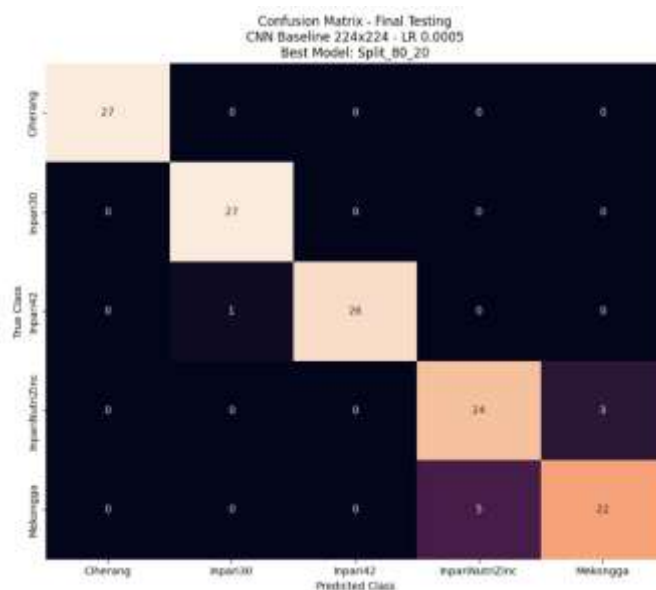


dan waktu inferensi sebesar 74,6820 ms per citra. Hasil ini menunjukkan bahwa Custom CNN memberikan keseimbangan yang baik antara akurasi klasifikasi dan kebutuhan penyimpanan, sedangkan MobileNetV2 lebih unggul dalam kecepatan inferensi pada lingkungan pengujian yang digunakan.

Custom CNN memiliki 421.957 parameter. Jumlah parameter menggambarkan kompleksitas struktur model, tetapi tidak secara langsung menentukan waktu inferensi maupun ukuran berkas setelah konversi. Oleh karena itu, evaluasi efisiensi dalam penelitian ini didasarkan pada hasil pengukuran waktu inferensi dan ukuran model TensorFlow Lite secara aktual. Kedua indikator tersebut memberikan gambaran mengenai kebutuhan waktu komputasi dan penyimpanan model dalam lingkungan pengujian, meskipun belum sepenuhnya merepresentasikan kinerja pada seluruh perangkat penerapan.

3.7 Confusion Matrix dan Analisis Kesalahan

Hasil klasifikasi pada data final testing dianalisis menggunakan confusion matrix untuk mengetahui pola prediksi benar dan salah pada setiap varietas. Pola prediksi lima varietas pada 135 citra final testing ditunjukkan pada Gambar 4.



Gambar 4. Confusion matrix final testing Custom CNN

Gambar 4 memperlihatkan pola prediksi pada 135 citra *final testing*. 27 citra Ciherang seluruhnya diprediksi sebagai Ciherang, dan seluruh 27 citra Inpari 30 juga diprediksi benar. Pada Inpari 42 terdapat satu citra yang diprediksi sebagai Inpari 30 sehingga 26 dari 27 citra benar. Inpari IR Nutri Zinc menghasilkan 24 prediksi benar dan tiga citra diprediksi sebagai Mekongga. Sebaliknya, dari 27 citra Mekongga, 22 diprediksi benar dan lima diprediksi sebagai Inpari IR Nutri Zinc. Total terdapat 126 prediksi benar dan 9 prediksi salah. Pola tersebut menunjukkan bahwa kesalahan paling dominan terjadi pada pasangan Mekongga dan Inpari IR Nutri Zinc. Kesalahan terjadi dua arah, tetapi jumlah kesalahan Mekongga menjadi Inpari IR Nutri Zinc lebih besar. Kondisi tersebut menunjukkan bahwa fitur visual pada sebagian citra kedua kelas belum menghasilkan batas pemisah yang cukup kuat bagi model.

3.8 Visualisasi Citra Gabah yang Salah Diklasifikasikan

Untuk memberikan gambaran mengenai kesalahan klasifikasi yang terjadi pada pengujian akhir, dilakukan visualisasi terhadap sembilan citra gabah yang salah diprediksi oleh model Custom CNN. Citra yang ditampilkan diambil dari dataset final testing, dengan tetap mempertahankan label sebenarnya dan label hasil prediksi model. Visualisasi ini digunakan untuk mengamati karakteristik citra pada kasus kesalahan klasifikasi.



Gambar 5. Visualisasi sembilan citra gabah yang salah diklasifikasikan oleh model Custom CNN

Gambar 5 menampilkan sembilan citra gabah yang salah diklasifikasikan, yang terdiri atas satu citra Inpari42 yang diprediksi sebagai Inpari30, tiga citra InpariNutriZinc yang diprediksi sebagai Mekongga, serta lima citra Mekongga yang diprediksi sebagai InpariNutriZinc. Dengan demikian, sebagian besar kesalahan klasifikasi melibatkan pasangan varietas Mekongga dan InpariNutriZinc. Pola ini menunjukkan bahwa kedua kelas tersebut lebih sering tertukar dibandingkan pasangan varietas lainnya pada dataset pengujian yang digunakan.

3.9 Analisis Per Kelas

Analisis per kelas diperlukan karena akurasi keseluruhan dapat menyembunyikan perbedaan kemampuan model pada varietas tertentu. Tabel 8 menunjukkan bahwa Ciherang dan Inpari 30 memperoleh *recall* 100%, sedangkan Inpari 42 96,30%, Inpari IR Nutri Zinc 88,89%, dan Mekongga 81,48%. Nilai *precision* Inpari IR Nutri Zinc sebesar 82,76% menunjukkan bahwa sebagian citra yang diprediksi sebagai kelas tersebut berasal dari Mekongga. Pada Mekongga, *precision* 88,00% dan *recall* 81,48% menunjukkan model lebih sering kehilangan sampel aktual Mekongga daripada menambahkan banyak prediksi Mekongga dari kelas lain.

Tabel 9. Metrik klasifikasi per kelas

Varietas	Precision	Recall	F1-score	Support
Ciherang	1,0000	1,0000	1,0000	27
Inpari 30	0,9643	1,0000	0,9818	27
Inpari 42	1,0000	0,9630	0,9811	27
Inpari IR Nutri Zinc	0,8276	0,8889	0,8571	27
Mekongga	0,8800	0,8148	0,8462	27
Macro average	0,9344	0,9333	0,9332	135
Weighted average	0,9344	0,9333	0,9332	135

Pola pada Tabel 9 berkaitan dengan hasil confusion matrix, tetapi interpretasinya tetap dibatasi pada dataset penelitian. Kesalahan antara Inpari IR Nutri Zinc dan Mekongga terjadi pada dua arah, dengan jumlah



kesalahan lebih banyak dari Mekongga menuju Inpari IR Nutri Zinc. Hal tersebut dapat terjadi ketika perbedaan bentuk dan warna visual yang dipelajari model tidak cukup tegas pada sebagian citra. Karakteristik morfologi dalam skripsi menunjukkan bahwa Inpari IR Nutri Zinc dan Inpari 42 sama-sama memiliki bentuk ramping dan warna kuning jerami, sedangkan Ciherang dan Inpari 30 juga memiliki bentuk panjang, ramping, dan warna kuning bersih.

3.10 Pembahasan dan Keterkaitan dengan Penelitian Terdahulu

Nilai *testing accuracy* 93,33% pada penelitian ini berada pada rentang performa yang relevan untuk sistem klasifikasi citra varietas, tetapi perbandingan angka tidak dapat dilakukan secara langsung tanpa mempertimbangkan dataset dan objek yang digunakan. Koklu et al. menggunakan dataset varietas beras dalam jumlah besar dan melaporkan performa CNN yang sangat tinggi [6]. Penelitian Din et al. juga menggunakan arsitektur CNN yang dikembangkan untuk klasifikasi varietas padi [17]. Perbedaan utama penelitian ini adalah objek berupa bulir gabah dari lima varietas yang diperoleh langsung dari BRMP Gorontalo, bukan dataset beras dari lingkungan lain. Dengan demikian, nilai 93,33% harus dibaca sebagai performa pada distribusi data dan kondisi eksperimen penelitian ini.

Penelitian Marnelius et al. dan Bhattacharjee et al. menunjukkan bahwa klasifikasi berbasis citra pada butir beras dapat dilakukan melalui model *deep learning* [14], [18]. Namun, citra beras dan citra gabah memiliki karakter visual yang berbeda karena gabah masih memiliki sekam. Penelitian ini memfokuskan pengenalan langsung pada satu bulir gabah, sehingga model harus belajar pola bentuk keseluruhan, proporsi, permukaan, dan warna sekam. Pilihan preprocessing yang sederhana berupa *cropping*, *resize*, dan normalisasi juga menjaga agar informasi visual tetap berasal dari objek penelitian tanpa menambahkan transformasi yang tidak digunakan pada pengujian akhir.

Dari sisi efisiensi dan implementasi, model Custom CNN menggunakan 421.957 parameter dan dibangun dari awal. Pilihan ini relevan dengan konteks penelitian yang menggunakan perangkat kerja dengan sumber daya terbatas dan pelatihan dilakukan melalui Google Colab. Literatur tentang CNN yang dioptimalkan untuk perangkat terbatas menunjukkan bahwa kompleksitas model merupakan aspek yang perlu diperhatikan bersama performa [20]. Sementara itu, MobileNetV2, ResNet50, DenseNet121, dan InceptionV3 menyediakan pembandingan arsitektur dengan desain yang berbeda [22], [24], [33]. Hasil pengujian menunjukkan bahwa Custom CNN pada dataset penelitian menghasilkan *testing* yang lebih tinggi dari pada keempat model pembandingan, tetapi temuan tersebut tidak berarti bahwa arsitektur Custom CNN akan selalu lebih unggul pada dataset lain.

Pengaruh konfigurasi juga menunjukkan pentingnya *experimental design*. Ukuran 224×224 memberikan validasi terbaik pada pengujian resolusi; *learning rate* 0,0005 meningkatkan validasi dibandingkan dua nilai lain; empat lapisan konvolusi memberikan validasi tertinggi; dan split 80:20 menghasilkan loss validasi terendah di antara dua skenario dengan akurasi tertinggi. Hasil ini mendukung penggunaan eksperimen terkontrol seperti yang disarankan dalam studi klasifikasi citra, karena perubahan kinerja perlu dikaitkan dengan variabel yang benar-benar diubah [31], [32]. Pemilihan satu konfigurasi berdasarkan tahapan tersebut juga mengurangi keputusan berbasis satu percobaan tunggal.

Model mencapai checkpoint terbaik pada epoch ke-48 dengan training accuracy 92,78% dan validation accuracy 95,56%. Pada final testing, Custom CNN memperoleh akurasi 93,33% dengan 126 dari 135 citra terklasifikasi benar. Kesalahan terbanyak melibatkan Mekongga dan Inpari IR Nutri Zinc, dengan lima citra Mekongga diprediksi sebagai Inpari IR Nutri Zinc dan tiga citra sebaliknya. Evaluasi per kelas melalui precision, recall, dan F1-score diperlukan untuk mengidentifikasi pola kesalahan yang tidak terlihat dari akurasi total [30]. Pada pengujian efisiensi TensorFlow Lite, Custom CNN memiliki ukuran model terkecil, yaitu 1,6942 MB, dengan waktu inferensi 22,6940 ms/citra. MobileNetV2 lebih cepat dengan waktu 9,9549 ms/citra, tetapi akurasinya lebih rendah, yaitu 83,70%. Sementara itu, ResNet50 mencapai validation accuracy 96,89%, tetapi akurasi pengujian akhirnya hanya 90,37%. Hasil ini menunjukkan bahwa akurasi validasi yang tinggi tidak selalu menghasilkan generalisasi terbaik, sedangkan Custom CNN menawarkan keseimbangan antara akurasi klasifikasi dan ukuran model.

Keterbatasan penelitian terutama terletak pada jumlah varietas dan sumber citra yang terbatas pada varietas yang tersedia di BRMP Gorontalo. Meskipun pemisahan berdasarkan RAW membantu mengurangi kebocoran antara subset, jumlah foto sumber tetap 28 per varietas. Selain itu, kondisi akuisisi belum mencakup seluruh variasi lingkungan yang mungkin ditemui pada penerapan yang lebih luas. Pengembangan berikutnya dapat menambah foto, lokasi, kondisi pencahayaan, dan perangkat yang berbeda serta mengevaluasi strategi augmentasi atau metode pra-proses lain. Penelitian tentang pemrosesan citra juga menunjukkan bahwa pemilihan representasi objek dapat memengaruhi kualitas klasifikasi [7], [34], [35].

Secara keseluruhan, hasil penelitian menunjukkan bahwa keberhasilan klasifikasi tidak hanya ditentukan oleh jumlah parameter atau nama arsitektur, tetapi oleh kecocokan antara data, preprocessing, konfigurasi optimasi, kedalaman model, dan prosedur evaluasi. Pendekatan Custom CNN dari awal memberikan hasil pengujian 93,33% pada dataset ini. Hasil tersebut diperoleh setelah konfigurasi ditetapkan melalui eksperimen dan diuji kembali pada 135 citra yang tidak digunakan untuk memilih model. Prosedur tersebut memberikan dasar



evaluasi yang lebih terpisah antara proses pengembangan dan proses pengujian akhir, sehingga angka performa dapat dipahami sesuai ruang lingkup dataset .

4. KESIMPULAN

Penelitian ini berhasil merancang dan mengembangkan model *Convolutional Neural Network* berbasis *computer vision* untuk mengidentifikasi lima varietas padi berdasarkan citra bulir gabah, yaitu Ciherang, Inpari 30, Inpari 42, Inpari IR Nutri Zinc, dan Mekongga. Dataset penelitian terdiri atas 1.260 citra individual yang dibentuk dari 28 foto RAW per varietas dan 9 hasil *cropping* per foto. Pemisahan berdasarkan sumber foto diterapkan agar citra dari satu RAW tidak tersebar ke subset yang berbeda. Hasil pengujian bertahap menghasilkan konfigurasi 224×224 piksel, *learning rate* 0,0005, empat *convolutional layer*, dan pembagian data 80:20 sebagai konfigurasi akhir. Custom CNN memiliki 421.957 parameter trainable dan diuji pada 135 citra *final testing* yang tidak digunakan selama proses pemilihan konfigurasi. Model mengklasifikasikan 126 citra dengan benar dan memperoleh *accuracy* 93,33%, *precision* 93,44%, *recall* 93,33%, serta F1-score 93,32%. Pada tingkat kelas, Ciherang dan Inpari 30 memperoleh akurasi 100%, Inpari 42 sebesar 96,30%, Inpari IR Nutri Zinc 88,89%, dan Mekongga 81,48%. Perbandingan dengan MobileNetV2, ResNet50, DenseNet121, dan InceptionV3 pada data pengujian yang sama menunjukkan Custom CNN memperoleh nilai metrik keseluruhan yang lebih tinggi pada ruang eksperimen penelitian ini. Keterbatasan penelitian masih terdapat pada jumlah varietas, jumlah sumber foto, dan variasi kondisi akuisisi. Pengembangan selanjutnya dapat memperluas dataset, menambah variasi sumber citra, dan menguji strategi preprocessing serta arsitektur lain untuk meningkatkan pengenalan pada kelas yang masih menghasilkan kesalahan, terutama Mekongga dan Inpari IR Nutri Zinc.

REFERENCES

- [1] BPS, "Luas Panen dan Produksi Padi Di Indonesia 2023 (Angka Sementara)," Badan Pusat Statistik.
- [2] F. H. Hibatullah, F. Raidasari, A. P. Triana, V. K. L. Siagian, and T. Simarmata, "Revealing Food Fulfillment Threads And Innovative Technology For Enhancing Rice Productivity And Ensuring The Food Security In Indonesia," *Int. J. Food, Agric. Nat. Resour.*, vol. 5, no. 3, pp. 45–51, 2024.
- [3] M. H. Ghifary and E. I. Sela, "Klasifikasi Varietas Benih Padi Berdasarkan Morfologi dengan Algoritma Random Forest," *Kesatria J. Penerapan Sist. Inf. (Komputer dan Manajemen)*, vol. 5, no. 2, pp. 615–622, 2024.
- [4] A. Ardiyansah, "Sistem Identifikasi Varietas Padi Berbasis Citra Drone untuk Mendukung Proses Sertifikasi Benih," 2022.
- [5] S. Mohapatra, C. Marandi, A. Sahoo, S. Mohanty, and K. Tudu, "Rice Leaf Disease Detection And Classification Using A Deep Neural Network," in *International Conference on Computing, Communication and Learning*, Springer, 2022, pp. 231–243.
- [6] M. Koklu, I. Cinar, and Y. S. Taspinar, "Classification Of Rice Varieties With Deep Learning Methods," *Comput. Electron. Agric.*, vol. 187, p. 106285, 2021.
- [7] Yudha Riwanto, Inggrid Yanuar Risca Pratiwi, and Muhammad Ainul Fikri, "Segmentasi Citra Menggunakan Pendekatan Trial and Error dan Metode Otsu untuk Identifikasi Objek," *J. Inform. Polinema*, vol. 11, no. 3, pp. 371–378, 2025, doi: 10.33795/jip.v11i3.7046.
- [8] F. Marpaung, F. Aulia, and R. C. Nabila, "Computer vision dan pengolahan citra digital," 2022, *Pustaka Aksara*.
- [9] W. S. Gan, "Deep Learning and Artificial Neural Networks," in *Signal Processing and Image Processing for Acoustical Imaging*, Springer, 2020, pp. 69–79.
- [10] J. Jumadi, Y. Yupianti, and D. Sartika, "Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering," *JST (Jurnal Sains dan Teknol.*, vol. 10, no. 2, pp. 148–156, 2021, doi: 10.23887/jstundiksha.v10i2.33636.
- [11] P. Shruti and R. Rekha, "A Review Of Convolutional Neural Networks, Its Variants And Applications," in *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, IEEE, 2023, pp. 31–36.
- [12] S. Yuliany and A. N. Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *J. Buana Inform.*, vol. 13, no. 1, pp. 54–65, 2022.
- [13] C. D. Marnelius, K. Usman, N. Kumalasari, and C. Pratiwi, "Klasifikasi Jenis Beras Berbasis Citra Dengan Menggunakan Deep Learning," *vol*, vol. 10, pp. 4211–4216, 2023.
- [14] S. A.-A. Shuvo *et al.*, "Image-Based Classification Of Bangladeshi Rice Varieties Using Deep Convolutional Neural Networks," *Agric. Prod. Process. Storage*, vol. 2, no. 1, p. 26, 2026.
- [15] L. J. . Alfurqan, A. R. ., Pratama, D. Y. ., Rahmadiansah, A. ., Sawitri, D. ., & Mawarani, "Rice Identification Using Convolutional Neural Network with YOLOv7 algorithm and VGG16," *IPTEK J. Eng.*, vol. Vol. 10 No, 2025, [Online]. Available: <https://journal.its.ac.id/index.php/joe/article/view/8258>
- [16] M. Z. Altim, A. YUDHISTIRA, and R. A. SYAMSU, "Pengklasifikasi Beras Menggunakan Metode Cnn (Convolutional Neural Network)," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 7, no. 1, pp. 151–155, 2022.
- [17] N. M. U. Din *et al.*, "Ricenet: A Deep Convolutional Neural Network Approach For Classification Of Rice Varieties," *Expert*



Syst. Appl., vol. 235, p. 121214, 2024.

- [18] A. Bhattacharjee, K. R. Singh, T. S. Singh, S. Datta, U. R. Singh, and G. Thingbaijam, "Intelligent Systems And Applications In Engineering A Comparative Study On Rice Grain Classification Using Convolutional Neural Network And Other Machine Learning Techniques," pp. 0–1, 2024.
- [19] Z. N. Azeez, A. A. Qasim, and N. M. Mohie, "Classification of Rice Grains by Image Processing: Klasifikasi Butir Beras dengan Pemrosesan Citra," *Acad. Open*, vol. 9, no. 2 SE-Food Science, p. 10.21070/acopen.9.2024.10303, Oct. 2024, doi: 10.21070/acopen.9.2024.10303.
- [20] H. Nugroho, J. X. Chew, S. Eswaran, and F. S. Tay, "Resource-optimized cnns for real-time rice disease detection with ARM cortex-M microprocessors," *Plant Methods*, vol. 20, no. 1, p. 159, 2024, doi: 10.1186/s13007-024-01280-6.
- [21] W. I. Kusumawati and A. Z. Noorizki, "Perbandingan Performa Algoritma VGG16 Dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras," *J. Comput. Electron. Telecommun.*, vol. 4, no. 2, 2023.
- [22] A. Ridhovan and A. Suharso, "Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum," *JUPI (Jurnal Ilm. Penelit. Dan Pembelajaran Inform.)*, vol. 7, no. 1, pp. 58–65, 2022.
- [23] L. Xu, S. S. Teoh, and H. Ibrahim, "OPEN A deep learning approach for electric motor fault diagnosis based on modified InceptionV3," *Sci. Rep.*, pp. 1–15, 2024, doi: 10.1038/s41598-024-63086-9.
- [24] P. Sitompul, H. Okprana, and A. Prasetyo, "Identification of Rice Plant Diseases Through Leaf Image Using DenseNet 201: Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201," *JOMLAI J. Mach. Learn. Artif. Intell.*, vol. 1, no. 2, pp. 143–150, 2022.
- [25] R. Rajalakshmi, S. Faizal, S. Sivasankaran, and R. Geetha, "RiceSeedNet : Rice seed variety identification using deep neural network," *J. Agric. Food Res.*, vol. 16, no. September 2023, p. 101062, 2024, doi: 10.1016/j.jafr.2024.101062.
- [26] X. Glorot and A. Bordes, "Deep Sparse Rectifier Neural Networks," vol. 15, pp. 315–323, 2011.
- [27] G. Hinton, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," vol. 15, pp. 1929–1958, 2014.
- [28] D. P. Kingma, "Adam: A method for stochastic optimization," pp. 1–13, 2015, doi: 10.48550/arXiv.1412.6980.
- [29] J. R. Terven, D. M. Cordova-esparza, A. Ramirez-pedraza, E. A. Chavez-urbiola, and J. A. Romero-gonzalez, "Loss Functions And Metrics In Deep Learning Published," 2025, [Online]. Available: <https://doi.org/10.1007/s10462-025-11198-7>
- [30] B. P. Pratiwi, A. S. Handayani, and S. Sarjana, "Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi Wsn Menggunakan Confusion Matrix," *J. Inform. Upgris*, vol. 6, no. 2, 2020.
- [31] A. A. Abhinowo, R. R. Isnanto, and D. Eridani, "Pemilihan Model Terbaik Algoritma Convolutional Neural Network untuk Klasifikasi Jenis Bencana Alam," *J. Tek. Komput.*, vol. 1, no. 4, pp. 199–208, 2023.
- [32] A. Fahrezantara, S. Rizal, and N. K. C. Pratiwi, "Pemanfaatan Convolutional Neural Network Dalam Klasifikasi Penyakit Tanaman Singkong Menggunakan Arsitektur Densenet," *EProceedings Eng.*, vol. 9, 2023.
- [33] D. Sinha and M. El-Sharkawy, "Ultra-thin Mobilenet," in *2020 10th annual computing and communication workshop and conference (CCWC)*, IEEE, 2020, pp. 234–240.
- [34] J. C. Canales, J. C. Canales, F. García-Lamont, A. Yee-Rendon, J. S. R. Castilla, and L. R. Mazahua, "Optimal Segmentation Of Image Datasets By Genetic Algorithms Using Color Spaces," *Expert Syst. Appl.*, vol. 238, p. 121950, 2024.
- [35] K. D. Toennies, "Image Classification: A Computer Vision Task," in *An Introduction to Image Classification: From Designed Models to End-to-End Learning*, Springer, 2024, pp. 1–17.