

© Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Karya ini berada di bawah Lisensi Creative Commons Atribusi-BerbagiSerupa 4.0 Internasional
DOI: 10.22146/jnteti.v14i3.18791

Klasifikasi Penyakit Padi Menggunakan *Convolutional Neural Network* (CNN) Berbasis Citra Daun

Moh. Heri Susanto¹, Irwan Budi Santoso¹, Suhartono¹, Ahmad Fahmi Karami¹

¹ Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Jawa Timur 65144, Indonesia

[Diserahkan: 30 Januari 2025, Direvisi: 3 April 2025, Diterima: 14 Juli 2025]

Penulis Korespondensi: Moh. Heri Susanto (email: 200605110087@student.uin-malang.ac.id)

INTISARI — Penyakit padi berdampak signifikan terhadap produktivitas pertanian, sehingga model klasifikasi sangat penting untuk membedakan penyakit daun padi secara akurat. Berbagai model klasifikasi telah diusulkan untuk mengklasifikasikan penyakit padi berbasis citra daun, tetapi masih diperlukan peningkatan kinerja lebih lanjut. Penelitian ini diusulkan menggunakan *convolutional neural network* (CNN) untuk mengklasifikasikan penyakit padi berbasis citra daun. Dalam penelitian ini, digunakan *dataset* citra daun yang memuat daun dalam kondisi hawar daun (*blight*), blas, tungro, dan sehat. Pada tahap awal dilakukan prapemrosesan data terlebih dahulu berupa pengubahan ukuran, augmentasi, dan normalisasi. Setelah prapemrosesan, dilakukan pembentukan arsitektur CNN yang bersifat *custom*, yang terdiri atas 4 lapisan konvolusi, 4 lapisan *pooling*, dan 3 *fully connected layer*. Setiap lapisan konvolusi menggunakan *kernel* berukuran 3×3 , *stride* 1, dan aktivasi ReLU, sedangkan lapisan *pooling* menggunakan *max pooling* dengan *kernel* berukuran 3×3 dan *stride* 2. Dengan penggunaan konfigurasi ukuran *batch* 32 dan optimasi Adam, mampu didapatkan hasil pengujian terbaik pada percobaan dengan menggunakan *epoch* 100 dan laju belajar 0,0002, yang menghasilkan akurasi pelatihan 0,9930, *loss* 0,0221, dan akurasi pengujian 0,9647. Hasil evaluasi model mampu mendapatkan keseimbangan antara presisi, *recall*, dan *F1-score*, yaitu 0,9647, yang menunjukkan proses klasifikasi data berjalan dengan sangat baik, tanpa adanya bias terhadap kelas tertentu. Hasil penelitian ini menunjukkan bahwa model CNN yang disederhanakan dapat memberikan kinerja klasifikasi yang kompetitif tanpa memerlukan model CNN yang kompleks atau dengan teknik tambahan lainnya. Model CNN yang diusulkan mampu bekerja lebih baik daripada model CNN yang ada, seperti Inception-ResNet-V2, VGG-16, VGG-19, dan Xception.

KATA KUNCI — Penyakit Padi, Arsitektur CNN Khusus, Citra Daun, Optimasi Adam, Evaluasi Model.

I. PENDAHULUAN

Penyakit padi menimbulkan ancaman yang signifikan terhadap produktivitas pertanian, sehingga diperlukan metode klasifikasi yang akurat untuk membedakan berbagai jenis penyakit padi. Beberapa penyakit padi utama yang memengaruhi hasil panen adalah hawar daun (*blight*), blas, dan tungro. Hawar daun adalah penyakit yang disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae* (*Xoo*) dengan ciri-ciri lesi memanjang dengan warna lesi yang bervariasi, mulai dari kuning hingga kecoklatan. Biasanya hal ini muncul pada daun yang baru berkembang dan seiring waktu lesi-lesi dapat menyatu hingga tampak seperti daun terbakar [1]. Blas, yang disebabkan oleh bakteri *Pyricularia Grisea*, menyebabkan malai dapat mengerut dan butiran padi dapat terisi setengah bahkan kosong. Gejala penyakit ini adalah daun berwarna abu-abu agak putih dan tepi pinggir berwarna coklat sedikit oranye, berbentuk belah ketupat dengan bagian tepi agak runcing [2]. Tungro, yang ditularkan oleh wereng hijau, menyebabkan tanaman menguning hingga oranye, pertumbuhan terhambat, dan berkurangnya produksi gabah [3]. Sebaliknya, tanaman padi sehat tidak menunjukkan tanda-tanda penyakit. Gambar 1 mengilustrasikan karakteristik visual jenis penyakit padi.

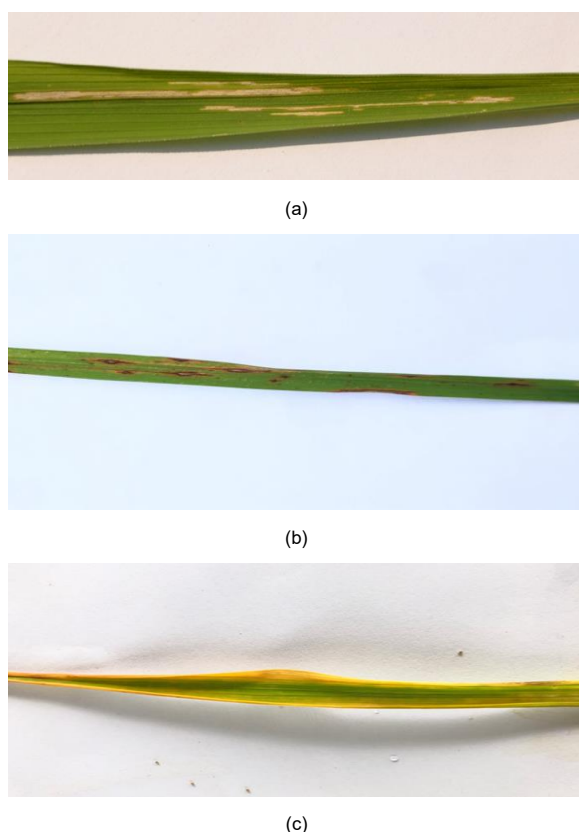
Beberapa penelitian sebelumnya telah dilakukan dengan menggunakan metode klasifikasi seperti *decision tree*, *artificial neural network* (ANN), *convolutional neural network* (CNN), *naïve Bayes*, *k-nearest neighbors* (KNN), dan *support vector machine* (SVM). Meskipun masing-masing metode penelitian memberikan kontribusi yang signifikan, di dalamnya masih terdapat keterbatasan yang bisa disebabkan oleh banyak hal, seperti data maupun keunggulan model atau arsitekturnya.

Oleh karena itu, penelitian tentang klasifikasi penyakit padi masih memiliki potensi untuk dikembangkan. Penyempurnaan lebih lanjut diperlukan untuk meningkatkan akurasi dan kinerja metode klasifikasi guna memastikan kategorisasi penyakit yang lebih andal.

Penelitian sebelumnya tentang klasifikasi penyakit padi hanya berfokus pada data citra daun sehat dan bercak coklat [4]. Metode yang diusulkan adalah *gray-level co-occurrence matrix* (GLCM) dan *intensity-level based multi-fractal dimension* (ILMFD), diikuti oleh klasifikasi dengan teknik SVM, ANN, dan *Neuro-Genetic Algorithm* (Neuro-GA) untuk dijadikan perbandingan. Hasilnya menunjukkan bahwa Neuro-GA lebih baik daripada model ANN, tetapi SVM terbukti lebih unggul dari Neuro-GA, mencapai akurasi 96,68% [4].

Penelitian lain menggunakan variasi *dataset* yang telah tersedia pada UCI Machine Learning Repository, terdiri atas *bacterial leaf blight*, *brown spots*, dan *left smut* [5]. Metode ANN digunakan dalam proses klasifikasi dengan *epoch* 100 dan mencapai akurasi 83%, presisi 84%, *recall* 84%, serta *F1-score* 83% [5]. Penelitian ini masih memiliki keterbatasan, yaitu jumlah *dataset* hanya 120, yang tergolong kecil, serta tidak adanya proses augmentasi data.

Dengan *dataset* yang sama, penelitian lain juga melaporkan bahwa metode yang diusulkan, yaitu *transfer learning* dengan *pretrained* Resnet101, memperoleh akurasi validasi 100% dengan nilai *loss* 5,61% [6]. Kinerja validasi tersebut menunjukkan bahwa model mampu mempelajari pola pada *dataset* validasi dengan sangat efektif. Akan tetapi, karena penelitian ini hanya memproses hingga tahap validasi, sedangkan tahap pengujiannya tidak dilakukan, belum dapat



Gambar 1. Jenis penyakit daun padi, (a) hawar daun, (b) blas, (c) tungro.

dipastikan bahwa model dapat melakukan generalisasi dengan baik pada data yang tidak terlihat. Tahap pengujian dalam penelitian sangat penting dilakukan agar kemampuan model dapat dievaluasi.

Studi lain juga mengklasifikasikan penyakit padi dengan jenis penyakit *leaf blight*, *brown spots*, dan *leaf smut* [7]. Dengan GLCM digunakan untuk ekstraksi fitur dan KNN sebagai model klasifikasi, diperoleh akurasi pengujian 65,83% [7]. Hal ini membuktikan bahwa metode KNN juga masih belum mampu mendapatkan kinerja yang baik.

Selain itu, para ahli yang lain melakukan penelitian dengan menggunakan *decision tree* [8]. Dalam penelitian ini, fitur-fitur diekstraksi menggunakan segmentasi dan morfologi. Akurasi pengujian yang didapatkan adalah 90% [8]. Sementara itu, data yang digunakan hanya berjumlah 120 dan metode *decision tree* tidak efektif untuk menangani data yang kompleks.

Penelitian lain menggunakan model *naive Bayes* untuk mengklasifikasikan penyakit padi hawar daun, blas, dan tungro. Hasil akurasi mencapai 76% dan tingkat keyakinan prediksi 100% pada kelas hawar daun [9]. Hasil yang didapatkan menunjukkan bahwa kinerja model kurang optimal serta teknik augmentasi yang tidak dilakukan menjadikan keterbatasan dalam menghadapi keragaman data.

Selanjutnya, penelitian lain juga menggunakan variasi *dataset* yang terdiri atas tungro, blas, *leaf blight*, dan *brownspot* dengan metode CNN. Dengan jumlah keseluruhan data sebanyak 320, didapatkan hasil akurasi sebesar 91,4% [10]. *Dataset* yang relatif kecil dan tidak adanya penanganan terhadap *overfitting* seperti augmentasi/cross validation menjadi keterbatasan dalam penelitian ini. Selain itu, CNN sendiri masih memiliki banyak potensi pengembangan arsitektur, mulai dari jumlah lapisan tersembunyi, *epoch*, laju belajar, ukuran *batch*, ukuran *kernel*, metode penghitungan *loss*, metode optimasi, dan penambahan *dataset*.

Riset yang relevan, dengan data yang makin bervariasi dan jumlah data yang besar, menunjukkan adanya kinerja model yang meningkat [11]. Dengan data latih berjumlah 4.000 dan data uji berjumlah 300, didapatkan akurasi uji 95,67%. Penelitian ini menggunakan arsitektur Inception-ResNet-V2 dan metode *transfer learning* [11]. Meskipun akurasi yang didapatkan cukup tinggi, data yang digunakan memiliki karakteristik berupa latar belakang yang tidak seimbang. Kelas blas dan hawar daun memiliki latar belakang bertumpuk, sedangkan kelas *brownspot* dan sehat hanya berupa satu lembar daun padi dengan latar belakang putih polos. Hal ini memungkinkan model mengalami *overfitting* terhadap data. Dalam hal ini, model tidak fokus terhadap fitur dari daun, melainkan hanya fokus pada latar belakang.

Dengan demikian, penelitian ini mengusulkan penggunaan metode CNN untuk mengklasifikasikan penyakit padi berbasis citra daun. *Dataset* citra daun dipilih karena pada umumnya gejala penyakit padi muncul pertama kali pada daun dalam bentuk perubahan warna dan terdapat bercak sebelum menyebar ke bagian lain, seperti batang atau malai. CNN dipilih karena mampu secara efektif mempelajari dan mengekstraksi fitur-fitur penting secara manual, sehingga tidak membutuhkan proses ekstraksi fitur tambahan karena di dalamnya sudah terdapat proses ekstraksi fitur, yaitu konvolusi. Telah terbukti juga bahwa pada proses pengklasifikasian objek dalam gambar, CNN unggul metode pembelajaran mesin lainnya seperti SVM [12].

Keberhasilan CNN dalam mengklasifikasikan objek dalam gambar serta keunggulannya dibandingkan metode pembelajaran mesin lainnya menjadikannya pilihan ideal untuk mendapatkan kinerja yang tangguh dan andal dalam mendiagnosis penyakit padi. Penting untuk diklarifikasi bahwa studi ini berfokus pada klasifikasi dalam bidang *intelligent systems* tanpa membahas implementasi waktu nyata atau otomatisasi dalam lingkungan pertanian. Penelitian ini berkontribusi pada aspek mendasar tersebut, yang nantinya dapat dimanfaatkan dalam aplikasi pertanian cerdas yang lebih luas. Penelitian ini bertujuan untuk membuat arsitektur CNN yang dirancang khusus untuk klasifikasi penyakit padi, yang memastikan bahwa model tersebut mencapai kinerja kompetitif dibandingkan dengan model klasifikasi yang sudah ada.

Meskipun penelitian sebelumnya mungkin telah mencapai akurasi yang baik, akan selalu ada ruang untuk perbaikan. Kemampuan ini menjadikan CNN sebagai alat yang ampuh untuk mengklasifikasikan penyakit daun padi berdasarkan karakteristik visualnya.

Berdasarkan pembahasan di atas, penelitian ini memberikan beberapa kontribusi utama. Pertama, studi ini mengoptimalkan arsitektur CNN sederhana untuk klasifikasi penyakit padi, yang bertujuan meningkatkan akurasi dan kinerja tanpa meningkatkan kompleksitas model. Kedua, penelitian ini melakukan evaluasi mendetail terkait akurasi, presisi, *recall*, dan *F1-score* sebagai alat ukur kinerja model dalam membedakan berbagai penyakit padi. Ketiga, dilakukan analisis perbandingan kinerja antara model klasifikasi CNN yang diusulkan dan model CNN lainnya, yang dapat memberikan pemahaman tentang efektivitas pendekatan yang diusulkan.

II. METODOLOGI

A. PENGUMPULAN DATA

Pada penelitian ini, *dataset* penyakit padi berdasarkan citra daun dikumpulkan melalui Kaggle. *Dataset* ini dipilih karena

data telah diverifikasi dan divalidasi oleh Kaggle, sehingga dapat digunakan untuk analisis lebih lanjut seperti klasifikasi. *Dataset* yang digunakan terdiri atas jenis daun penyakit padi hawar daun sebanyak 220 data, blas 200 data, tungro 80 data, dan penambahan data sehat 210 data, dengan keseluruhan data sebanyak 710 data. Tabel I menunjukkan beberapa sampel yang digunakan dalam penelitian ini.

B. SKEMA DATA PELATIHAN DAN DATA PENGUJIAN

Terhadap *dataset* penyakit padi dilakukan proses pemisahan data menjadi dua kelompok utama, yaitu data pelatihan dan data pengujian. Diambil sebanyak 80% dari data untuk digunakan sebagai data dalam pelatihan model. Hal ini merupakan proses bagi model dalam melakukan pembelajaran terhadap data untuk mengenali pola atau fitur yang relevan. Proses pelatihan yang dilakukan meliputi pengubahan ukuran, augmentasi, dan normalisasi. Setelah itu, dilakukan pembentukan model CNN untuk mendapatkan bobot dan bias yang baru. Kemudian, bobot dan bias yang baru ini disimpan untuk digunakan pada pengujian.

Setelah model dilatih, 20% kumpulan data yang tersisa digunakan untuk data pengujian. Proses pengujian juga dilakukan dengan cara yang sama, yaitu pengubahan ukuran dan normalisasi. Data yang telah diproses ini kemudian dimasukkan ke dalam CNN, dengan menggunakan bobot dan bias yang dipelajari sebelumnya, untuk mengklasifikasikan data pengujian, menghasilkan hasil klasifikasi akhir berupa jenis kondisi padi.

C. PRAPEMROSESAN

Prapemrosesan citra merupakan langkah penting dalam menyiapkan data untuk model CNN, yang melibatkan teknik seperti pengubahan ukuran dan augmentasi. Karena fitur warna sangat penting untuk klasifikasi, tidak ada konversi warna yang diterapkan.

Pertama, dilakukan proses pengubahan ukuran citra untuk memastikan keseragaman seluruh data. Hal ini dilakukan karena CNN memerlukan masukan citra dengan ukuran data yang konsisten. Jika ukuran citra bervariasi, model CNN tidak dapat memprosesnya. Dalam penelitian ini, citra diubah ukurannya menjadi 64×64 piksel. Nilai piksel ini dipilih karena relatif kecil dan tidak terlalu membebani proses komputasi.

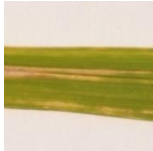
Kemudian, augmentasi digunakan untuk meningkatkan keragaman data pelatihan dan meningkatkan generalisasi model [13] dengan menerapkan rotasi yang memungkinkan model belajar objek dari posisi yang berbeda. Diterapkan juga teknik augmentasi lain, seperti *flip*, *shift*, *shear*, dan *zoom*, yang memungkinkan model mengenali objek dalam orientasi yang berbeda, memperkaya representasi data, dan mengurangi *overfitting*. Teknik-teknik ini akan membantu kemampuan model pada saat proses pengujian dan meningkatkan akurasi dengan adanya orientasi dan representasi data yang melimpah.

Terakhir, diterapkan normalisasi pada data pelatihan dan data pengujian, dengan cara menskalakan nilai piksel antara 0 dan 1 untuk memastikan konsistensi dan meningkatkan efisiensi pembelajaran model. Hal ini membantu mencegah gradien terlalu besar yang dapat memperlambat proses pelatihan.

D. ARSITEKTUR CNN USULAN

Gambar 2 merupakan arsitektur CNN yang diusulkan untuk mengklasifikasikan penyakit padi. Arsitektur ini dirancang untuk mengoptimalkan deteksi fitur pada citra berukuran $64 \times$

TABEL I
CONTOH GAMBAR DARI *DATASET* YANG DIGUNAKAN

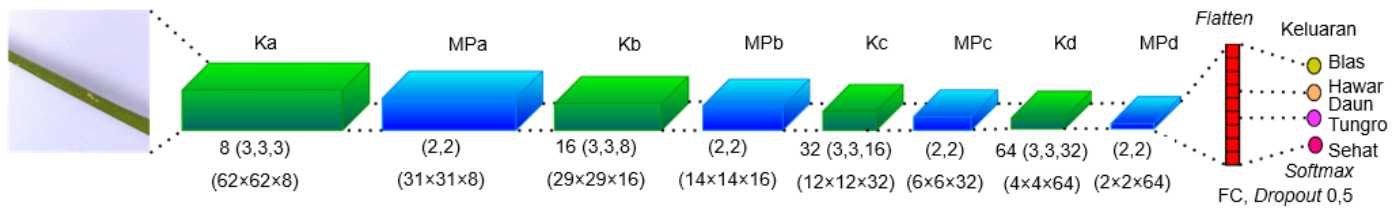
No.	Kondisi Daun Padi	Gambar
1.	Blas	
2.	Hawar daun	
3.	Tungro	
4.	Sehat	

64 piksel yang memiliki 4 lapisan tersembunyi. Lapisan tersembunyi pertama memiliki lapisan konvolusi (K_a) dengan penerapan filter 3×3 sebanyak 8 dan dengan data masukan 3 (RGB). Keluaran dari lapisan konvolusi (K_a) berukuran $62 \times 62 \times 8$. Dimensi keluaran tersebut didapatkan dari perhitungan konvolusi *no padding*. Setelah melalui proses konvolusi, data akan diproses oleh *rectified linear unit* (ReLU) untuk menangani masalah nonlinearitas. Selanjutnya, lapisan *max pooling* (M_{pa}) memproses dengan menggunakan filter 2×2 . Keluaran lapisan *max pooling* (M_{pa}) berukuran $31 \times 31 \times 8$.

Lapisan tersembunyi kedua memiliki lapisan konvolusi (K_b) yang menggunakan 16 filter berukuran 3×3 , dengan 8 adalah dimensi masukannya (diperoleh dari keluaran pada proses lapisan tersembunyi pertama). Dimensi keluaran lapisan konvolusi (K_b) adalah $29 \times 29 \times 8$, yang diperoleh dengan cara menerapkan perhitungan konvolusi *no padding*. ReLU diterapkan untuk menangani data yang nonlinear. Selanjutnya, lapisan *max pooling* memproses dengan menggunakan filter berukuran 2×2 , sehingga didapatkan keluaran berukuran $14 \times 14 \times 16$.

Lapisan tersembunyi ketiga merupakan lapisan konvolusi (K_c) dengan 32 filter berukuran 3×3 dan data masukan yang diperoleh dari keluaran lapisan tersembunyi kedua, yaitu 16. Lapisan ini juga menerapkan konvolusi *no padding*, sehingga berdasarkan aturan tersebut didapatkan keluaran konvolusi (K_c) $12 \times 12 \times 32$. Fungsi aktivasi yang digunakan juga ReLU. Kemudian, data diproses oleh lapisan *max pooling* yang memiliki filter berukuran 2×2 dan keluaran *max pooling*-nya adalah $6 \times 6 \times 32$.

Lapisan tersembunyi terakhir memiliki lapisan konvolusi (K_d) dengan filter berukuran 3×3 yang berjumlah 64 filter. Data masukannya diperoleh dari proses lapisan tersembunyi ketiga, yaitu 32. Dengan aturan konvolusi *no padding*, didapatkan keluaran $4 \times 4 \times 64$. Fungsi aktivasi yang digunakan masih sama, yaitu ReLU. Kemudian, lapisan *max pooling* memproses menggunakan filter 2×2 , sehingga mendapatkan keluaran $2 \times 2 \times 64$.



Gambar 2. Arsitektur usulan CNN.

Setelah proses ekstraksi fitur selesai, hasilnya diratakan, yang sebelumnya bersifat multidimensi menjadi satu dimensi, oleh *flatten layer*, sehingga dapat diproses oleh *fully connected layer*. Keluarannya berupa data mentah atau logit. Kemudian, teknik *dropout* digunakan untuk mengatasi *overfitting* yang bernilai 0,5. Akhirnya, pada lapisan *softmax* dilakukan proses untuk mengonversi data, yang sebelumnya bersifat data mentah, menjadi probabilitas klasifikasi akhir, yang memungkinkan model untuk menentukan jenis penyakit padi dengan memilih akurasi tertinggi berdasarkan masukan citra dengan resolusi kecil. Detail setiap langkah akan dijelaskan lebih lanjut sebagai berikut.

1) LAPISAN KONVOLUSI

Dalam arsitektur yang diusulkan, lapisan konvolusi bertugas mengekstraksi fitur dari data masukan [14] yang berupa citra daun padi. Lapisan ini bekerja dengan cara menerapkan serangkaian filter ke citra masukan untuk dilakukan proses ekstraksi fitur. Fitur-fitur ini meliputi tekstur, warna, atau bentuk corak yang muncul pada citra daun padi. Persamaan (1) merupakan proses perhitungan keluaran dalam lapisan konvolusi yang dinyatakan secara matematis [15].

$$Y(p, q) = \sum_i \sum_j X(p, q) * F(p - i, q - j) \quad (1)$$

dengan $Y(p, q)$ adalah keluaran dari operasi konvolusi pada koordinat (p, q) , $X(p, q)$ adalah masukan gambar pada koordinat (p, q) , $F(i, j)$ adalah filter atau *kernel* konvolusi pada posisi (i, j) , p dan q menunjukkan koordinat keluaran, dan i serta j menunjukkan indeks spasial pada filter. Masing-masing filter akan menghasilkan satu peta fitur setelah melakukan konvolusi dengan masukan. Jadi, jika ada delapan filter dalam satu lapisan konvolusi, akan ada delapan peta fitur yang dihasilkan.

Pada saat lapisan konvolusi bekerja, *stride* dan *padding* berperan penting dalam menentukan proses ekstraksi fitur dari citra daun padi. Pada penelitian ini, nilai *stride* yang digunakan adalah 1, memastikan bahwa filter konvolusi bergerak 1 piksel untuk setiap kali perpindahan ke posisi berikutnya. Sementara itu, *padding* tidak digunakan karena berfungsi untuk mengurangi dimensi, yaitu tinggi dan lebar masukan. Dengan tidak menggunakan *padding*, model akan mengurangi beban komputasi sambil tetap mempertahankan fitur yang relevan dengan penyakit padi. Persamaan (2) dan (3) merupakan rumus untuk menentukan dimensi keluaran konvolusi menggunakan aturan konvolusi tanpa *padding* [16] yang digunakan dalam penelitian ini.

$$W_2 = \frac{W_1 - F_1}{s} + 1 \quad (2)$$

$$H_2 = \frac{H_1 - F_2}{s} + 1 \quad (3)$$

dengan W_1 dan H_1 adalah lebar dan tinggi masukan matriks, F_1 dan F_2 adalah lebar dan tinggi filter/*kernel*, dan s merupakan *stride* atau langkah *kernel* yang diterapkan pada data masukan.

2) RECTIFIED LINEAR UNIT (ReLU)

Fungsi aktivasi ReLU digunakan untuk mengatasi masalah nonlinearitas data penyakit padi berbasis citra daun secara efektif. Persamaan (4) menunjukkan formula ReLU yang digunakan, yaitu dengan mengubah nilai negatif menjadi 0 dan mempertahankan nilai positif [17].

$$\hat{y}_a(p, q) = \begin{cases} 0 & Y(p, q) < 0 \\ Y(p, q) & Y(p, q) \geq 0 \end{cases} \quad (4)$$

dengan $Y(p, q)$ adalah keluaran lapisan konvolusi dan $\hat{y}_a(p, q)$ adalah keluaran lapisan ReLU. Dengan menghilangkan nilai negatif, ReLU memastikan bahwa model hanya mempertahankan pola fitur penyakit yang relevan pada daun padi. Dengan hanya mempertahankan aktivasi yang positif, fungsi ReLU meningkatkan kemampuan jaringan untuk fokus pada fitur yang membedakan antara daun sehat dan daun yang terkena penyakit tertentu.

3) LAPISAN POOLING

Setelah diproses oleh aktivasi ReLU, selanjutnya data diproses oleh lapisan *pooling* untuk mengurangi dimensi spasial peta fitur, sehingga dapat mengurangi beban komputasi dan kompleksitas parameter dalam jaringan [18]. Sesuai dengan Gambar 2, jenis lapisan *pooling* yang digunakan adalah *max pooling*, yaitu dengan cara mengambil nilai maksimum dari setiap peta fitur. Hal ini memastikan fitur yang paling penting dan dominan pada daun yang terindikasi penyakit padi dipertahankan sambil menghilangkan informasi yang kurang signifikan. Penerapan *stride* 2, yang berarti filter bergerak dua piksel sekaligus, mengurangi ukuran peta fitur. Persamaan (5) dan (6) digunakan untuk menentukan dimensi keluaran [19].

$$W_2 = \frac{W_1 - F_1}{s} \quad (5)$$

$$H_2 = \frac{H_1 - F_2}{s} \quad (6)$$

4) FLATTEN LAYER

Setelah proses lapisan *pooling* pada lapisan tersembunyi terakhir, data akan diproses oleh *flatten layer* untuk mengubah data yang sebelumnya bersifat multidimensi menjadi satu dimensi. Ini merupakan langkah penting agar data yang berisi fitur yang telah diekstraksi dapat diproses lebih lanjut, yaitu pada *fully connected layer*.

5) FULLY CONNECTED LAYER

Fully connected layer adalah bagian terpenting dari arsitektur model pada penelitian ini. Pada tahap ini, model mulai mengidentifikasi pola penyakit tertentu berdasarkan fitur yang telah diekstraksi, yang kemudian diolah dan digabungkan menjadi keluaran mentah (logit). Setiap setelah proses pada *fully connected layer*, diterapkan ReLU dan *dropout* 50% (0,5). *Dropout* berfungsi untuk mengatasi *overfitting* [6] dengan cara menonaktifkan 50% neuronnya secara acak selama pelatihan. Hal ini mencegah ketergantungan model terhadap neuron

tertentu dan meningkatkan kemampuannya dalam menggeneralisasi ke citra daun padi yang baru terlihat, sehingga meningkatkan akurasi model.

6) SOFTMAX DAN NEGATIVE LOG-LIKELIHOOD (NLL) LOSS

Pada tahap ini, *softmax* berfungsi mengubah data mentah (logit) dari *fully connected layer* menjadi distribusi probabilitas, dengan memastikan nilai tidak negatif dan jumlah keseluruhan probabilitas ialah satu. Persamaan (7) merupakan rumus untuk menentukan keluaran *softmax* [20].

$$S(\tilde{y}_{ak}) = \frac{e^{\tilde{y}_{ak}}}{\sum_{i=1}^K e^{\tilde{y}_{ak}}} \quad (7)$$

dengan S adalah keluaran *softmax* di kelas ke- k , $\tilde{y}_{ak}$ adalah nilai keluaran *fully connected layer*, e adalah bilangan Euler $\cong 2,718$, dan K adalah jumlah keseluruhan kelas.

Setelah mendapatkan distribusi probabilitas, dilakukan perhitungan kerugian menggunakan *negative log likelihood* (NLL) *loss*. NLL *loss* merupakan perhitungan kerugian yang mirip dengan *cross entropy loss*. Perbedaannya adalah pada *cross entropy loss* dilakukan perhitungan secara terpisah dengan *softmax*, sedangkan NLL *loss* sudah mengandung *softmax* di dalamnya. NLL memberikan penalti terhadap prediksi yang salah [20], yang dilakukan dengan (8).

$$E = -\hat{y} \log \tilde{y}_{ak} \quad (8)$$

dengan $\hat{y}$ adalah label sebenarnya (bernilai 1 untuk benar dan 0 jika salah) dan $\tilde{y}_{ak}$ adalah probabilitas prediksi model untuk kelas ke- k . Di sini, keluaran NLL dapat dikatakan lebih baik prediksi probabilitasnya apabila nilai keluarannya lebih rendah. Keluaran NLL yang lebih rendah menunjukkan bahwa model berhasil mengidentifikasi pola penyakit padi pada citra daun dengan lebih baik.

7) BACKPROPAGATION DAN PERHITUNGAN GRADIEN

Nilai *loss* yang diperoleh akan digunakan pada proses *backpropagation*, dengan tujuan untuk mendapatkan bobot dan *bias* yang baru. Pada penelitian ini digunakan optimasi Adam.

Optimasi Adam adalah salah satu metode yang digunakan untuk memperbarui bobot dan *bias* [21]. Ini merupakan algoritma pengoptimalan laju belajar adaptif yang dirancang khusus untuk melatih *deep neural network* [22] dan juga merupakan gabungan dari dua optimasi, yaitu RMSprop dan momentum. Formula yang digunakan untuk memperbarui bobot ditunjukkan pada (9) sampai (12) [23].

$$m_w = \beta_1 * m_{w-1} + (1 - \beta_1) * \left(\frac{\partial E}{\partial w}\right) \quad (9)$$

$$\frac{\partial E}{\partial w} = \delta \cdot x \quad (10)$$

$$v_w = \beta_2 * v_{w-1} + (1 - \beta_2) * \left(\frac{\partial E}{\partial w}\right)^2 \quad (11)$$

$$w = w - \left(\frac{\alpha * m_w}{\sqrt{(v_w + \epsilon)}}\right) \quad (12)$$

Sementara itu, untuk mendapatkan *bias* yang baru, digunakan (13) sampai (16).

$$m_b = \beta_1 * m_{b-1} + (1 - \beta_1) * \left(\frac{\partial E}{\partial b}\right) \quad (13)$$

$$\frac{\partial E}{\partial b} = \delta \quad (14)$$

$$v_b = \beta_2 * v_{b-1} + (1 - \beta_2) * \left(\frac{\partial E}{\partial b}\right)^2 \quad (15)$$

$$b = b - \left(\frac{\alpha * m_b}{\sqrt{(v_b + \epsilon)}}\right) \quad (16)$$

Variabel m_w dan m_b adalah bagian momentum pada bobot dan *bias*, v_w dan v_b adalah bagian kecepatan pada bobot dan *bias*, β_1 adalah parameter untuk momen pertama (biasanya 0,9), β_2 adalah parameter untuk momen kedua (biasanya 0,999), $\frac{\partial E}{\partial w}$ dan $\frac{\partial E}{\partial b}$ adalah gradien *loss function* terhadap bobot dan *bias*, m_{w-1} dan m_{b-1} adalah momen bobot dan *bias* pada waktu sebelumnya, v_{w-1} dan v_{b-1} adalah kecepatan bobot dan *bias* pada waktu sebelumnya; w dan b ialah parameter bobot dan *bias* yang diperbarui; α ialah laju belajar, ϵ ialah konstanta kecil untuk mencegah pembagian dengan nol (biasanya 10^{-7}), δ ialah *loss* untuk neuron keluaran, dan x merupakan masukan yang terkait dengan bobot tersebut dalam jaringan neural.

Proses *backpropagation* akan terus berjalan hingga mencapai *epoch* yang ditentukan. Setelah itu, bobot dan *bias* yang baru disimpan dan digunakan untuk proses pengujian.

8) EVALUASI

Pengukuran kinerja model terhadap data pengujian dapat dilakukan dengan cara evaluasi model. Metode yang digunakan meliputi akurasi, yang dilakukan menggunakan (17), presisi menggunakan (18), *recall* menggunakan (19), dan *F1-score* menggunakan (20). Akurasi adalah teknik untuk mengukur kekerapan model memprediksi data uji secara keseluruhan dengan benar. Makin mendekati 1 nilainya, dapat dikatakan makin bagus juga kinerja model. Sebaliknya, makin mendekati 0 nilai akurasi, dapat dikatakan bahwa model belum mencapai kinerja terbaik. Presisi adalah suatu teknik yang mengukur prediksi positif yang benar dari keseluruhan kelas positif. *Recall* adalah suatu teknik yang mengukur banyaknya data positif yang berhasil diprediksi dari keseluruhan data yang sebenarnya positif, sedangkan *F1-score* adalah suatu teknik yang mengukur rata-rata dari *recall* dan presisi.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (18)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (19)$$

$$F1 - \text{score} = 2 \frac{\text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}} \quad (20)$$

Selain itu, untuk mengetahui gambaran secara detail, digunakan *confusion matrix*. Tabel II menampilkan rangkuman terkait kinerja model klasifikasi, yaitu *true positive* (TP), *false positive* (FP), *true negative* (TN), dan *false negative* (FN).

Sebagai contoh, pada kasus prediksi hawar daun, akan dihasilkan TP apabila hawar daun benar diprediksi sebagai hawar daun. FN adalah semua elemen di baris hawar daun yang bukan TP. FP adalah elemen di kolom hawar daun yang bukan TP. Sementara itu, TN terjadi pada semua elemen yang bukan di baris dan di kolom hawar daun. Demikian pula prediksi di kelas yang lainnya.

III. HASIL DAN PEMBAHASAN

A. HASIL PENGUJIAN

Tahap awal dalam penelitian ini menggunakan beberapa skema uji coba untuk mendapatkan hasil yang terbaik. Skema uji coba ini dilakukan dengan melibatkan variasi *epoch* dan laju belajar. Variasi *epoch* digunakan untuk merujuk pada jumlah iterasi terhadap keseluruhan data, sedangkan variasi laju belajar

TABEL II
CONFUSION MATRIX

Aktual	Kelas Prediksi			
	Hawar Daun (A)	Blas (B)	Tungro (C)	Sehat (D)
Hawar daun (A)	AA	AB	AC	AD
Blas (B)	BA	BB	BC	BD
Tungro (C)	CA	CB	CC	CD
Sehat (D)	DA	DB	DC	DD

digunakan untuk menentukan kecepatan pembaruan bobot pada saat proses pelatihan. Tujuan variasi ini adalah untuk menemukan kombinasi parameter yang paling efektif dalam meningkatkan kinerja model.

Uji coba dilakukan secara bertahap, mulai A1 hingga B4. Model A1 hingga A4 digunakan sebagai uji coba dengan parameter *epoch* 30 dan variasi laju belajar 0,001, 0,0001, 0,0002, dan 0,0003. Laju belajar 0,001 dipilih karena nilai tersebut relatif rendah, sedangkan 0,0002 hingga 0,0003 dipilih sebagai pembanding. Sementara itu, model B1 hingga B4 digunakan sebagai uji coba dengan menggunakan *epoch* 100 dan variasi laju belajar 0,001, 0,0001, 0,0002, dan 0,0003. Disini, *epoch* ditingkatkan dengan tujuan untuk mengetahui pengaruh perubahan *epoch* terhadap kinerja model.

Setelah uji coba berhasil diselesaikan berdasarkan skema yang sudah dibuat, diperoleh hasil uji coba A1, dengan *epoch* 30 dan laju belajar 0,001, model menunjukkan akurasi pelatihan sebesar 0,9018. Nilai *loss* yang diperoleh cukup rendah, yaitu 0,2258. Durasi pelatihan pada uji coba ini adalah sekitar $112^m0,8^s$. Akurasi pengujian mencapai 0,9295, yang menunjukkan bahwa akurasi pengujian lebih tinggi daripada akurasi pelatihan, atau disebut *underfitting*. Ada kemungkinan model tidak mampu mempelajari fitur kompleks yang ada di data pelatihan, tetapi dapat melakukan generalisasi yang cukup baik di data pengujian.

Uji coba berikutnya adalah pada A2, yaitu dengan menggunakan *epoch* 30 dan laju belajar 0,0001. Diperoleh akurasi pelatihan 0,8305 dan nilai *loss* 0,3880, dengan durasi $128^m0,8^s$. Sementara itu, pada data uji didapatkan akurasi 0,8802. Berdasarkan hasil tersebut, penurunan laju belajar akan menyebabkan durasi yang lebih lama. Ini berarti pembelajaran dengan langkah kecil akan menyebabkan model lebih lama mencapai titik yang optimal dibandingkan dengan langkah yang lebih besar. Terlihat jelas pada akurasi pelatihan, *loss*, dan akurasi pengujian yang tidak lebih optimal dibandingkan uji coba A1 sebelumnya.

Selanjutnya, uji coba A3 dengan menggunakan *epoch* 30 dan laju belajar 0,0002 memberikan hasil yaitu akurasi pelatihan meningkat menjadi 0,9234, *loss* menurun menjadi 0,1921, durasi $112^m6,4^s$, dan akurasi pengujian meningkat menjadi 0,9436. Berdasarkan hasil tersebut, dengan laju belajar yang sedikit ditingkatkan, model dapat memperoleh kinerja yang lebih bagus. Ini menunjukkan bahwa model membutuhkan laju belajar yang lebih besar dari uji coba A2 dan lebih kecil dari uji coba A1 untuk mencapai konvergensi yang lebih cepat dan efektif tanpa menyebabkan peningkatan *loss*. Namun demikian, *underfitting* masih terjadi dalam uji coba ini.

Untuk uji coba A4, dengan menggunakan *epoch* 30 dan laju belajar 0,0003, model menunjukkan penurunan kinerja secara keseluruhan. Hasil akurasi pelatihan mencapai 0,9179, *loss* 0,2069, durasi pelatihan $112^m3,7^s$, dan akurasi pengujian 0,9084. Meskipun akurasi pelatihan relatif tinggi, akurasi

pengujian menunjukkan penurunan dibandingkan dengan uji coba A1 dan A3, yang menunjukkan adanya *overfitting*. Selama pelatihan, model bekerja sangat baik, yang ditandai dengan adanya perbandingan antara akurasi pelatihan dan akurasi pengujian yang tidak terlalu jauh.

Berikutnya adalah uji coba B1 dengan menggunakan *epoch* 100 dan laju belajar 0,001. Akurasi pelatihan mencapai 0,9850 dengan *loss* 0,0393, yang relatif lebih rendah daripada uji coba sebelumnya. Durasi pelatihan meningkat menjadi $364^m8,7^s$. Hal ini disebabkan oleh jumlah *epoch* yang ditingkatkan, sehingga model membutuhkan lebih banyak waktu dalam melakukan pembelajaran. Akurasi pengujian sebesar 0,9154 menunjukkan adanya *overfitting* ketika model terlalu banyak menghafal pada data pelatihan. Uji coba B1 memberikan wawasan bahwa peningkatan jumlah *epoch*, yang menyebabkan durasi waktu pembelajaran lebih lama, tidak secara otomatis meningkatkan kinerja model.

Pada uji coba B2 digunakan *epoch* sebanyak 100 dan laju belajar 0,0001. Dengan nilai laju belajar yang diturunkan, model menunjukkan penurunan nilai akurasi pelatihan, yaitu hanya mencapai 0,9421, dan peningkatan nilai *loss*, yaitu menjadi 0,1366. Durasi pelatihan pada uji coba ini sekitar $364^m8,7^s$. Sementara itu, nilai akurasi pengujian meningkat menjadi 0,9225. Hal ini terjadi karena masih adanya *overfitting* meskipun nilainya tidak terlalu jauh. Meskipun demikian, model telah mampu menggeneralisasi lebih baik pada data pengujian dibandingkan dengan uji coba B1.

Uji coba B3 menggunakan *epoch* sebanyak 100 dan laju belajar sebesar 0,0002. Akurasi pelatihan yang diperoleh mencapai 0,9930, yang merupakan akurasi tertinggi dibandingkan uji coba sebelumnya. Nilai *loss* menunjukkan penurunan dan sangat mendekati nilai 0, yaitu sebesar 0,0221. Akurasi pengujian juga meningkat signifikan menjadi 0,9647, yang menunjukkan kemampuan model dalam melakukan generalisasi dengan sangat baik. Durasi pelatihan tercatat $363^m10,8^s$, lebih singkat daripada uji coba B1 dan B2.

Terakhir, uji coba B4 menggunakan *epoch* 100 dan laju belajar 0,0003. Akurasi pelatihan yang diperoleh adalah 0,9910, dengan nilai *loss* sebesar 0,0246 dan durasi sekitar $359^m23,2^s$. Sementara itu, akurasi pengujian diperoleh 0,9225, yang menunjukkan adanya *overfitting*. Kombinasi parameter ini dinilai kurang, yang menyebabkan model gagal mencapai konvergensi dan kehilangan kemampuan untuk belajar dari data pelatihan.

Tabel III menyajikan hasil uji coba keseluruhan untuk mempermudah analisis. Terlihat pada Tabel III bahwa model yang dilatih dengan *epoch* 100 cenderung menghasilkan akurasi dan *loss* yang lebih baik dibandingkan model yang dilatih dengan *epoch* 30. Hal ini terjadi karena makin banyak *epoch*, makin lama juga waktu yang dimiliki model untuk mempelajari data, sehingga model mampu mendapatkan *loss* yang lebih rendah.

Pada uji coba A1 hingga A4 dan B1 hingga B4, model mampu mencapai akurasi tertinggi dengan menggunakan laju belajar 0,0002. Ini menunjukkan bahwa laju belajar yang lebih kecil mampu membantu model belajar lebih stabil dan mendapatkan *loss* yang lebih rendah. Namun, pada laju belajar yang lebih kecil dari 0,0002, yaitu 0,0001, belum mampu dihasilkan kinerja yang lebih baik. Hal ini mungkin disebabkan oleh laju belajar yang terlalu kecil, sehingga terlalu lambat dan menghambat model dalam melakukan pembelajaran.

Jadi, kinerja terbaik diperoleh pada uji coba B3, dengan menggunakan *epoch* sebanyak 100 dan laju belajar sebesar

TABEL III
HASIL UJI COBA KESELURUHAN

Uji Coba	Akurasi Pelatihan	Loss	Durasi Pelatihan	Akurasi Pengujian
A1	0,9018	0,2258	112 ^m 0,8 ^s	0,9295
A2	0,8305	0,3880	128 ^m 7,8 ^s	0,8802
A3	0,9234	0,1921	112 ^m 6,4 ^s	0,9436
A4	0,9179	0,2069	112 ^m 3,7 ^s	0,9084
B1	0,9850	0,0393	377 ^m 24,3 ^s	0,9154
B2	0,9421	0,1366	364 ^m 8,7 ^s	0,9225
B3	0,9930	0,0221	363 ^m 10,8 ^s	0,9647
B4	0,9910	0,0246	359 ^m 23,2 ^s	0,9225

0,0002, yang menghasilkan akurasi pelatihan 0,9930, *loss* 0,0221, dan akurasi pengujian 0,9647. Variasi *epoch* dan laju belajar secara signifikan memengaruhi kinerja model, yang menunjukkan pentingnya penyetelan parameter dalam mengoptimalkan model CNN untuk klasifikasi penyakit padi.

Gambar 3(a) menampilkan grafik akurasi pada saat pelatihan di uji coba B3. Terlihat bahwa pada permulaan, akurasi pelatihan meningkat dengan cepat, yang berarti model mampu memahami pola data pelatihan dengan sangat baik. Namun, secara bersamaan model mengalami ketidakstabilan dan hal ini terjadi secara berulang. Ketidakstabilan ini menunjukkan bahwa model mengalami kesulitan pembelajaran dalam melakukan penyesuaian terhadap data yang kompleks. Tampaknya ini disebabkan oleh adanya fitur yang baru terlihat oleh model. Meskipun demikian, di akhir *epoch* model berhasil mendapatkan nilai akurasi tertinggi.

Gambar 3(b) menunjukkan grafik *loss* saat pelatihan di uji coba B3. Pada tahap awal, nilai *loss* sangat mendekati 1. Hal ini disebabkan oleh inisialisasi bobot secara acak, sehingga model belum menyesuaikan bobotnya dengan baik terhadap data. Setelah beberapa *epoch*, terlihat adanya penurunan *loss* hingga mendekati 0, yang menunjukkan bahwa model telah mampu menyesuaikan bobotnya terhadap data pelatihan. Ini berarti kesalahan prediksi telah diminimalkan.

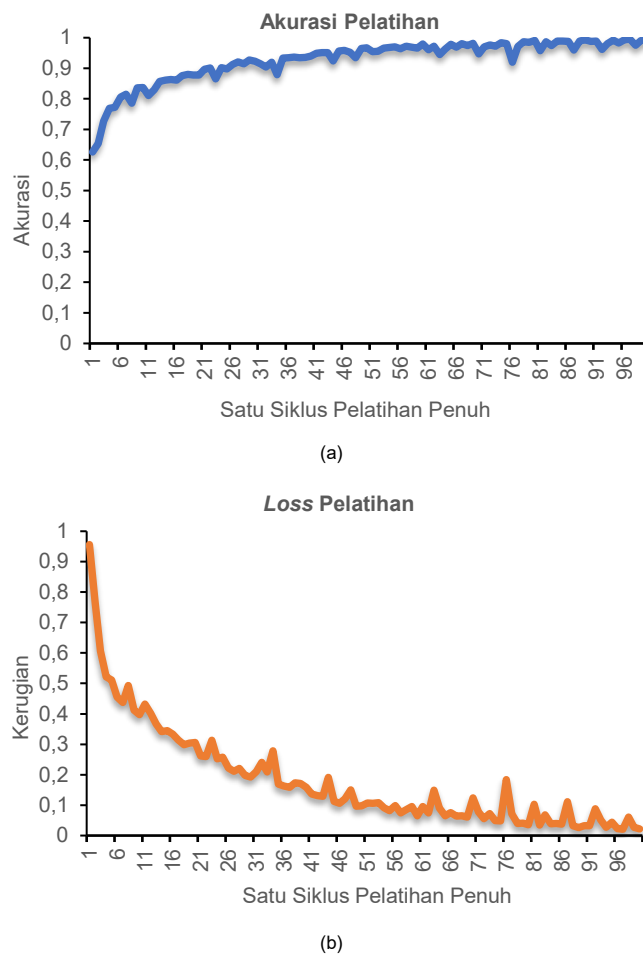
B. PEMBAHASAN

Selain akurasi, terdapat metrik evaluasi lain yang digunakan untuk mengukur kinerja model, yaitu presisi, *recall*, dan *F1-score*. Gambar 4 menampilkan metrik evaluasi dari kinerja terbaik (B3).

Dalam kasus penelitian ini, *recall* menjadi matriks yang lebih penting dibandingkan dengan presisi dan *F1-score* karena *recall* digunakan untuk mengukur banyaknya data positif yang berhasil diprediksi dari keseluruhan data yang sebenarnya positif. Hal ini berarti *recall* berfungsi untuk memastikan bahwa model dapat memprediksi semua daun yang terkena penyakit padi.

Dalam konteks pertanian, kegagalan dalam melakukan prediksi daun padi yang berpenyakit akan menyebabkan ancaman yang serius dibandingkan kesalahan dalam melakukan prediksi pada daun padi yang sehat. Oleh karena itu, *recall* menjadi prioritas utama yang harus diperhatikan untuk meminimalkan risiko daun yang terindikasi penyakit tidak terdeteksi.

Gambar 4 menunjukkan bahwa diperoleh nilai presisi, *recall*, dan *F1-score* yang sama. Ini berarti model mampu mengklasifikasikan data dengan sangat baik tanpa adanya bias terhadap kelas tertentu. Presisi tinggi berarti model jarang mengklasifikasikan daun yang sehat sebagai daun yang tidak

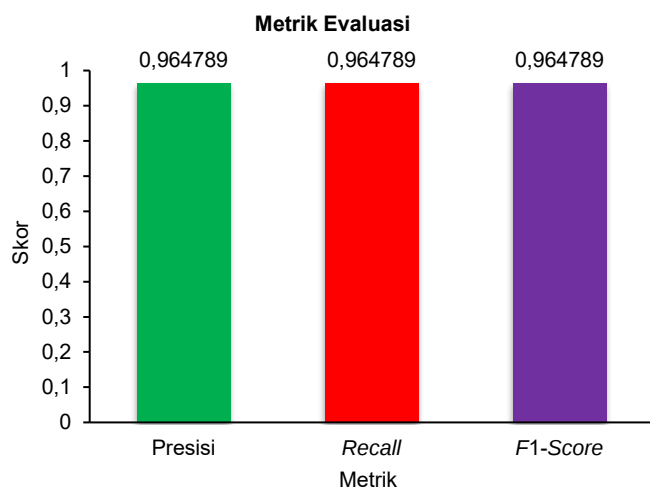


Gambar 3. Grafik yang didapatkan pada kinerja terbaik, yaitu uji coba B3, (a) akurasi pelatihan, (b) *loss* pelatihan.

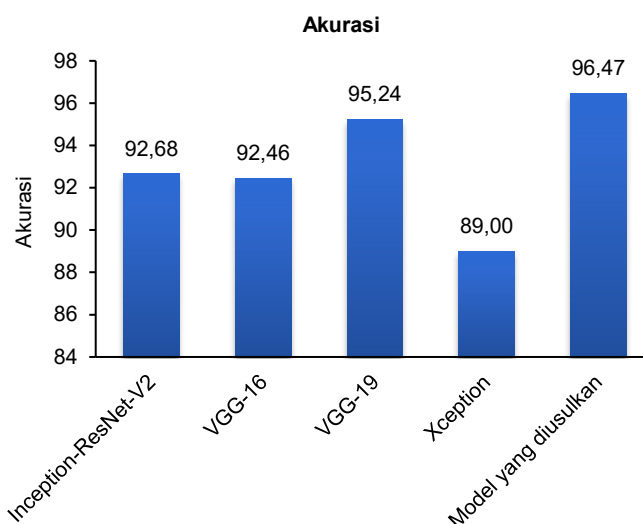
sehat. *Recall* yang tinggi berarti model sebagian besar mampu mendeteksi daun yang terinfeksi penyakit. Sementara itu, *F1-score* yang tinggi mempunyai arti model bekerja dengan sangat baik dalam menghindari kesalahan prediksi, baik daun yang sehat maupun daun yang terinfeksi penyakit. Kesamaan nilai akurasi, presisi, *recall*, dan *F1-score* ini menunjukkan bahwa model mampu memberikan kinerja yang optimal untuk semua kelas.

Selain itu, untuk mengetahui distribusi prediksi secara detail, digunakan confusion matrix seperti ditampilkan pada Gambar 5. Nilai diagonal menunjukkan klasifikasi yang benar, dengan 43 sampel hawar daun, 38 sampel blas, 14 sampel tungro, dan 42 sampel sehat diprediksi dengan benar. Namun, beberapa kesalahan terjadi, seperti 1 sampel hawar daun diklasifikasikan sebagai blas, 2 sampel blas sebagai tungro, 1 sampel tungro sebagai hawar daun, dan 1 sampel tungro sebagai blas. Intensitas warna dalam matriks secara visual membedakan antara nilai tinggi dan rendah, membantu menilai kinerja model dalam membedakan antara berbagai penyakit padi dan kondisi sehat.

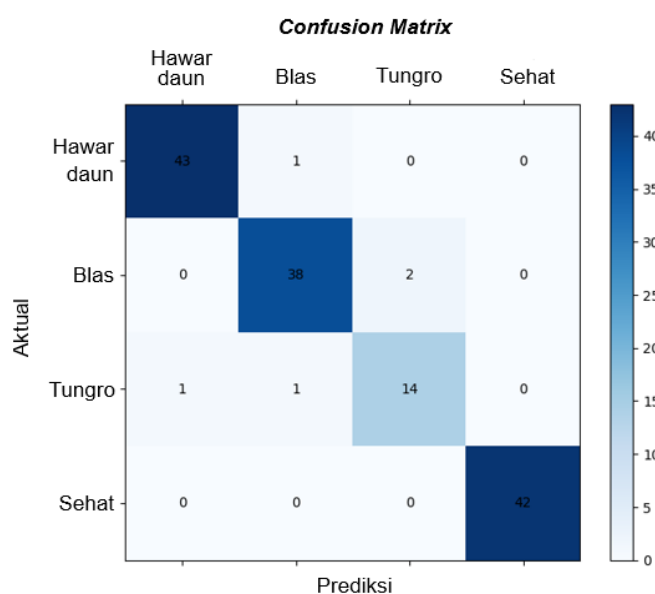
Terdapat beberapa penelitian sebelumnya yang dapat dijadikan pembandingan untuk penelitian ini, dengan menggunakan varian CNN yang ada, seperti yang dirangkum pada Gambar 6. Inception-ResNet-V2 dengan pendekatan *transfer learning* menggunakan model *pre-trained* dan melakukan penyesuaian pada lapisan terakhir. Hal ini memberikan durasi pelatihan yang singkat dan hasil akurasi pengujian mencapai 92,68% [24]. VGG-16 dengan *transfer*



Gambar 4. Hasil metrik evaluasi B3.



Gambar 6. Perbandingan dengan model CNN yang sudah ada.



Gambar 5. Hasil confusion matrix B3.

learning yang memanfaatkan bobot *pre-trained* dari ImageNet menghasilkan akurasi sebesar 92,46% [25]. Lalu, VGG-19, yang merupakan penyempurnaan versi VGG-16, memiliki kinerja lebih baik dengan akurasi 95,24% [26]. Xception dengan penggunaan teknik *depthwise separable convolution* menawarkan komputasi ringan dan akurasi yang baik dengan akurasi 89% [27]. Sementara itu, metode yang diusulkan, dengan akurasi mencapai 96,47% ini, menawarkan kinerja model yang dapat bersaing, tetapi tetap sederhana.

Secara keseluruhan, model CNN yang diusulkan menunjukkan peningkatan akurasi sekitar 1,23%. Salah satu faktor yang berkontribusi dalam peningkatan ini adalah pembentukan arsitektur yang tepat dengan karakteristik data yang digunakan. Dalam penelitian ini, arsitektur CNN *custom* yang diusulkan dirancang dengan beberapa pertimbangan, seperti jumlah lapisan tersembunyi, penggunaan *padding* atau tidak pada lapisan konvolusi, penentuan ukuran *kernel* dan jumlahnya, penentuan jenis aktivasi, penentuan teknik pada lapisan *pooling*, penentuan jenis optimasi, penentuan teknik perhitungan *loss*, penentuan metode untuk mengatasi *overfitting*, jumlah *epoch*, nilai laju belajar, dan jumlah ukuran *batch*. Dengan penggunaan arsitektur yang lebih tepat, model dapat mempelajari fitur-fitur kompleks pada data dengan lebih maksimal, mampu mengatasi masalah *overfitting/underfitting*,

dan meningkatkan akurasi. Selain itu, prapemrosesan juga berperan penting dalam meningkatkan akurasi model, sehingga kinerja yang dihasilkan lebih baik daripada model CNN lainnya pada penelitian-penelitian sebelumnya yang mungkin tidak dioptimalkan.

IV. KESIMPULAN

Pada penelitian ini model CNN yang digunakan mampu mengklasifikasikan daun yang terinfeksi dan tidak terinfeksi secara akurat. Akurasi pengujian, presisi, *recall*, dan *F1-score* memiliki nilai yang sama, yaitu 0,96. Kesalahan prediksi juga menunjukkan angka yang minimum dengan total lima kesalahan. Sementara itu, pada kelas sehat, semua data berhasil diprediksi dengan benar. Hasil tersebut menunjukkan bahwa arsitektur CNN yang digunakan dalam pendekatan ini mampu mencapai peningkatan akurasi dan kinerja yang bersaing dengan model klasifikasi CNN yang sudah ada, meskipun dengan arsitektur yang sederhana.

Karena penelitian ini hanya difokuskan pada pembangunan model CNN secara dasar, ada beberapa kemungkinan untuk dilakukan sebagai perbaikan yang lebih optimal di penelitian mendatang. Pertama, disarankan untuk mengeksplorasi integrasi teknik tambahan, seperti *transfer learning*, *fine tuning*, atau metode lainnya yang dapat meningkatkan kinerja model. Penelitian ini juga memiliki keterbatasan, yaitu hanya menggunakan arsitektur CNN yang sederhana dan tidak menggunakan teknik tambahan tersebut. Pendekatan ini dipilih karena penelitian berfokus pada kemampuan dasar CNN dalam mengenali pola visual dari citra daun padi secara langsung, sehingga hasil yang diperoleh lebih merefleksikan efektivitas dasar arsitektur CNN itu sendiri tanpa pengaruh teknik eksternal.

Selain itu, penelitian ini menggunakan kumpulan data yang terbatas, yang diperoleh dari sumber yang tersedia untuk umum. Memperluas kumpulan data dengan gambar yang beragam dan berkualitas tinggi dapat membantu kemampuan generalisasi model. Lebih jauh, mengeksplorasi pendekatan gabungan antara metode CNN dengan metode klasifikasi lainnya dapat memberikan ekstraksi fitur dan kinerja lebih baik.

KONFLIK KEPENTINGAN

Penulis menegaskan bahwa tidak ada potensi konflik kepentingan dalam penelitian ini.

KONTRIBUSI PENULIS

Konseptualisasi, Moh. Heri Susanto dan Irwan Budi Santoso; metodologi, Moh. Heri Susanto; perangkat lunak, Moh. Heri Susanto; validasi, Moh. Heri Susanto, Irwan Budi Santoso, Suhartono, dan Ahmad Fahmi Karami; analisis formal, Moh. Heri Susanto, Irwan Budi Santoso, dan Suhartono; investigasi, Moh. Heri Susanto; kurasi data, Moh. Heri Susanto; penulisan—penyusunan draf asli, Moh. Heri Susanto, dan Irwan Budi Santoso; penulisan—peninjauan dan penyuntingan, Suhartono dan Ahmad Fahmi Karami.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang atas dukungan dan bimbingan yang diberikan selama penelitian ini.

REFERENSI

- [1] P.M. Giopany, "Penyakit hawar daun bakteri pada tanaman padi." Distan. Tanggal akses: 15-Mar-2025. [Online]. Tersedia: https://distan.bulelengkab.go.id/informasi/detail/artikel/55_penyakit-hawar-daun-bakteri-pada-tanaman-padi?
- [2] V. Silvana, "Mengenal penyakit blas pada tanaman padi." Distan. Tanggal akses: 15-Mar-2025. [Online]. Tersedia: https://distan.bulelengkab.go.id/informasi/detail/artikel/57_mengenal-penyakit-blas-pada-tanaman-padi?
- [3] S. Sipi dan Subiadi, "Penyakit tungro dan keracunan Fe pada tanaman padi." Tanggal akses: 15-Mar-2025. [Online]. Tersedia: <https://repository.pertanian.go.id/server/api/core/bitstreams/6eb9a1de-0c26-4d49-9534-203a469b8f24/content>
- [4] S. Chaudhary dan U. Kumar, "Automated detection and classification of rice crop diseases using advanced image processing and machine learning techniques," *Trait. Signal*, vol. 41, no. 2, hal. 739–752, Apr. 2024, doi: 10.18280/ts.410216.
- [5] N. Sunandar dan J. Sutopo, "Utilization of artificial neural network in rice plant disease classification using leaf image," *Int. J. Res. Sci. Eng.*, vol. 4, no. 2, hal. 1–10, Feb./Mar. 2024, doi: 10.55529/ijrise.42.1.10.
- [6] U.N. Oktaviana, R. Hendrawan, A.D.K. Annas, dan G.W. Wicaksono, "Klasifikasi penyakit padi berdasarkan citra daun menggunakan model terlatih Resnet101," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 5, no. 6, hal. 1216–1222, Des. 2021, doi: 10.29207/resti.v5i6.3607.
- [7] R.A. Saputra dkk., "Rice leaf disease image classifications using KNN based on GLCM feature extraction," dalam *Int. Conf. Adv. Inf. Sci. Dev. (ICAISD)*, 2020, hal. 1–6, doi: 10.1088/1742-6596/1641/1/012080.
- [8] A.D.P. Alwy, M.S.N. Wahid, B.N.N. Ag, dan M.M. Fakhri, "Klasifikasi penyakit padi dengan ekstraksi fitur LBP dan GLCM," *J. Deep Learn. Comput. Vis. Digit. Image Process.*, vol. 1, no. 1, hal. 1–10, Mar. 2023, doi: 10.61255/decoding.v1i1.51.
- [9] Rahmi dan R.A. Saputra, "Klasifikasi Penyakit padi melalui citra daun menggunakan metode naive Bayes," *J. Inform. Tek. Elektro Terap.*, vol. 12, no. 2, hal. 885–892, Apr. 2024, doi: 10.23960/jitet.v12i2.4012.
- [10] A.K. Abasi dkk., "Enhancing rice leaf disease classification: A customized convolutional neural network approach," *Sustainability*, vol. 15, no. 20, Okt. 2023, Art. no 15039, doi: 10.3390/su152015039.
- [11] Krishnamoorthy N. dkk., "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environ. Res.*, vol. 198, hal. 1–8, Jul. 2021, doi: 10.1016/j.envres.2021.111275.
- [12] I.W.S.E. Putra, A.Y. Wijaya, dan R. Soelaiman, "Klasifikasi citra menggunakan convolutional neural network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, hal. 65–69, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [13] R. Jain dkk., "Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning," *Measurement*, vol. 165, hal. 1–10, Des. 2020, doi: 10.1016/j.measurement.2020.108046.
- [14] K. Azmi, S. Defit, dan Sumijan, "Implementasi convolutional neural network (CNN) untuk klasifikasi batik tanah liat Sumatera Barat," *J. Unitek*, vol. 16, no. 1, hal. 28–40, Jan.-Jun. 2023.
- [15] A. Batool dan Y.-C. Byun, "A lightweight multi-path convolutional neural network architecture using optimal features selection for multiclass classification of brain tumor using magnetic resonance images," *Results Eng.*, vol. 25, hal. 1–15, Mar. 2025, doi: 10.1016/j.rineng.2025.104327.
- [16] S.S. Bhat, A. Ananth, dan P.S. Venugopala, "Design and evolution of deep convolutional neural networks in image classification - A review," *Int. J. Integr. Eng.*, vol. 15, no. 1, hal. 213–225, Mei 2023, doi: 10.30880/ijie.2023.15.01.019.
- [17] I.B. Santoso dan I.K.E. Pumama, "Epileptic EEG signal classification using convolutional neural networks based on optimum window length and FFT's length," dalam *ICCCM '20, Proc. 8th Int. Conf. Comput. Commun. Manag.*, 2020, hal. 87–91, doi: 10.1145/3411174.3411179.
- [18] M.A. Rahman, M.R. Islam, M.A.H. Rafath, dan S. Mhejabin, "CNN based COVID-19 detection from image processing," *J. ICT Res. Appl.*, vol. 17, no. 1, hal. 99–113, Apr. 2023, doi: 10.5614/itbj.ict.res.appl.2023.17.1.7.
- [19] A. Ghosh dkk., "Fundamental concepts of convolutional neural network," dalam *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, Cham, Swiss: Springer, 2019, hal. 519–567.
- [20] S.B. Shrestha, L. Zhu, dan P. Sun, "Spikemax: Spike-based loss methods for classification," 2020, *arXiv: 2205.09845*.
- [21] C.R. Rahman dkk., "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosyst. Eng.*, vol. 194, hal. 112–120, Jun. 2020, doi: 10.1016/j.biosystemseng.2020.03.020.
- [22] Wardianto, Farikhin, dan D.M.K. Nugraheni, "Analisis sentimen berbasis aspek ulasan pelanggan restoran menggunakan LSTM dengan ADAM optimizer," *JOINTECS (J. Inf. Technol. Comput. Sci.)*, vol. 8, no. 2, hal. 67–76, Mei 2023, doi: 10.31328/jointecs.v8i2.4737.
- [23] A. Bhattacharjee, A.A. Popov, A. Sarshar, dan A. Sandu, "Improving the adaptive moment estimation (ADAM) stochastic optimizer through an implicit-explicit (IMEX) time-stepping approach," 2024, *arXiv: 2403.13704*.
- [24] M.A. Islam dkk., "An automated convolutional neural network based approach for paddy leaf disease detection," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 12, no. 1, hal. 280–288, Jan. 2021, doi: 10.14569/IJACSA.2021.0120134.
- [25] S. Ghosal dan K. Sarkar, "Rice leaf diseases classification using CNN with transfer learning," dalam *IEEE Calcutta Conf. (CALCON)*, 2020, hal. 230–236, doi: 10.1109/CALCON49167.2020.9106423.
- [26] A.A.J.V. Priyanka dan I.M.S. Kumara, "Classification of rice plant diseases using the convolutional neural network method," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 12, no. 2, hal. 123–129, Agu. 2021, doi: 10.24843/lkjiti.2021.v12.i02.p06.
- [27] A.R. Muslikh, D.R.I.M. Setiadi, dan A.A. Ojugo, "Rice disease recognition using transfer learning Xception convolutional neural network," *J. Tek. Inform. (Jutif)*, vol. 4, no. 6, hal. 1535–1540, Des. 2023, doi: 10.52436/1.jutif.2023.4.6.1529.