

Analisis Perbandingan DenseNet121 dan DenseNet201 Berbasis Transfer Learning untuk Identifikasi Penyakit Daun Tebu

Ridhwan Fachrul Arief¹, Salamun Rohman Nudin²

D4 Manajemen Informatika, Universitas Negeri Surabaya

Jl. Ketintang, Ketintang, Kec. Gayungan, Surabaya, Jawa Timur 60231

¹ridhwan.22047@mhs.unesa.ac.id

²salamunrohman@unesa.ac.id

Abstrak - Penyakit daun tebu dapat menurunkan produktivitas tanaman dan menyebabkan kerugian pada sektor pertanian. Identifikasi penyakit secara manual masih rentan terhadap kesalahan karena kemiripan gejala antarpenyakit. Penelitian ini bertujuan membandingkan performa arsitektur DenseNet121 dan DenseNet201 berbasis transfer learning untuk klasifikasi penyakit daun tebu menggunakan Sugarcane Leaf Dataset (SLD Data yang dimanfaatkan memuat 6.748 citra yang terklasifikasi ke dalam 11 kategori penyakit. Tahapan preprocessing mencakup resize, pembagian data, augmentasi, serta normalisasi. Kedua arsitektur diuji menggunakan optimizer Adam, RMSprop, dan SGD dengan parameter pelatihan yang sama. Penilaian kinerja model menggunakan metrik accuracy, precision, recall, dan F1-score. Eksperimen menunjukkan bahwa DenseNet201 dengan optimizer Adam memberikan performa terbaik dengan accuracy 94,22%, precision 95%, recall 94%, dan F1-score 94%. Sementara itu, DenseNet121 dengan optimizer Adam menghasilkan accuracy 93,15%. Temuan menunjukkan bahwa DenseNet201 lebih unggul dibandingkan DenseNet121 dalam mengklasifikasikan penyakit daun tebu menggunakan dataset SLD.

Kata kunci: DenseNet121, DenseNet201, Transfer Learning, Klasifikasi Citra, Penyakit Daun Tebu

Abstract - Sugarcane leaf diseases can reduce crop productivity and cause losses in the agricultural sector. Manual disease identification remains prone to errors due to the similarity of symptoms among different diseases. This study aims to compare the performance of DenseNet121 and DenseNet201 architectures based on transfer learning for sugarcane leaf disease classification using the Sugarcane Leaf Dataset (SLD). The dataset utilized consists of 6,748 images categorized into 11 disease classes. The preprocessing stages included image resizing, data splitting, augmentation, and normalization. Both architectures were evaluated using Adam, RMSprop, and SGD optimizers under the same training parameters. Model performance was assessed using accuracy, precision, recall, and F1-score metrics. Experimental results showed that DenseNet201 with the Adam optimizer achieved the best performance, obtaining an accuracy of 94.22%, precision of 95%, recall of 94%, and F1-score of 94%. Meanwhile, DenseNet121 with the Adam optimizer achieved an accuracy of 93.15%. These findings indicate that DenseNet201 outperformed DenseNet121 in classifying sugarcane leaf diseases using the SLD dataset.

Kata kunci: DenseNet121, DenseNet201, Transfer Learning, Image Classification, Sugarcane Leaf Disease.

I. PENDAHULUAN

Tanaman tebu (*Saccharum officinarum* L.) merupakan salah satu komoditas penting yang mendukung kebutuhan industri gula di Indonesia [1]. Produktivitas tebu kerap terganggu oleh infeksi pada bagian daun, di antaranya karat cokelat (*brown rust*) dan bercak daun, yang menjadi salah satu kendala utama dalam budidayanya [2]. Identifikasi penyakit secara dini menjadi langkah penting untuk meminimalkan kerugian hasil panen, namun proses ini membutuhkan pendekatan yang cepat dan akurat [3]. Pendekatan kecerdasan buatan, khususnya *deep learning*, telah terbukti memiliki kapabilitas untuk mengotomatisasi proses identifikasi penyakit tanaman secara efisien [4], [5].

Convolutional Neural Network (CNN) yang dikombinasikan dengan *transfer learning* merupakan pendekatan yang umum diterapkan pada klasifikasi penyakit tanaman. Pendekatan tersebut memungkinkan model memanfaatkan representasi fitur yang telah dipelajari dari dataset berskala besar guna meningkatkan performa pada dataset yang lebih spesifik [4] [6]. Salah satu arsitektur CNN yang banyak digunakan adalah DenseNet. Arsitektur ini menerapkan mekanisme *dense connectivity* yang meningkatkan aliran informasi antar lapisan, mendorong pemanfaatan ulang fitur, dan meningkatkan efisiensi pelatihan [7]. Beberapa varian DenseNet yang umum digunakan dalam klasifikasi citra adalah DenseNet121 dan DenseNet201. Kedua arsitektur tersebut memiliki perbedaan jumlah parameter, ukuran model, dan kedalaman jaringan yang dapat memengaruhi performa model pada tugas klasifikasi citra [8][9].

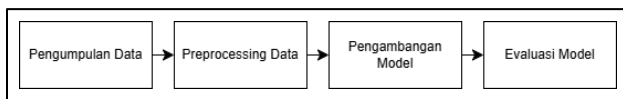
Penelitian terdahulu telah menunjukkan keunggulan DenseNet pada klasifikasi penyakit daun tebu. Devi dkk. (2024) membandingkan DenseNet169, DenseNet201, dan DenseNet264 berbasis *transfer learning* pada dataset penyakit daun tebu. Hasilnya menunjukkan bahwa DenseNet201 dan DenseNet264 menghasilkan performa lebih tinggi dibandingkan DenseNet169 [10]. Selain itu, Srinivasan dkk. (2025) menggunakan *Sugarcane Leaf Dataset (SLD)* dan melaporkan bahwa DenseNet201 termasuk salah satu arsitektur dengan performa terbaik di antara berbagai model CNN yang diuji [11]. Meskipun demikian, kedua penelitian tersebut tidak melibatkan DenseNet121 dalam proses evaluasi. Akibatnya, performa DenseNet121 relatif terhadap DenseNet201 pada

klasifikasi penyakit daun tebu menggunakan SLD masih belum diketahui.

Penelitian ini membandingkan performa DenseNet121 dan DenseNet201 berbasis *transfer learning* untuk klasifikasi penyakit daun tebu menggunakan SLD. Performa kedua model dievaluasi menggunakan metrik akurasi, presisi, *recall*, dan F1-score untuk mengidentifikasi arsitektur yang memberikan hasil klasifikasi terbaik. Penelitian ini diharapkan dapat memberikan referensi dalam pemilihan varian DenseNet yang sesuai untuk tugas klasifikasi penyakit daun tebu.

II. METODE PENELITIAN

Penelitian ini menerapkan pendekatan kuantitatif melalui metode eksperimen yang bersifat komparatif. Perbandingan dilakukan terhadap performa DenseNet121 dan DenseNet201 berbasis *transfer learning* pada klasifikasi penyakit daun tebu. Evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, dan F1-score. Alur penelitian ditunjukkan pada Gambar 1.



Gambar. 1 Alur Penelitian

A. Pengumpulan Data.

Data yang digunakan dalam penelitian ini adalah *Sugarcane Leaf Dataset (SLD)* yang diperoleh Mendeley Data, terdiri dari 6.748 citra daun tebu yang terbagi ke dalam 11 kelas penyakit. Distribusi data per kelas disajikan pada Tabel 1.

Tabel 1 Distribusi Dataset

No	Kategori	Jumlah Data
1	Banded Chlorosis	471
2	Brown Spot	1722
3	Brown Rust	314
4	Dried Leaves	343
5	Grassy Shoot	346
6	Healthy Leaves	430
7	Pokkah Boeng	297
8	Sett Rot	652
9	Smut	316
10	Viral Disease	663
11	Yellow Leaf	1194

B. Preprocessing Data

Tahap *preprocessing* dilakukan untuk memastikan konsistensi ukuran, distribusi piksel, dan kualitas citra sebelum digunakan dalam proses pelatihan model. Teknik *preprocessing* yang mencakup *resize*, augmentasi, dan normalisasi terbukti berkontribusi pada keseragaman data, peningkatan generalisasi model, dan percepatan konvergensi selama pelatihan [12]. Pada penelitian ini, *preprocessing*

diterapkan secara terstruktur melalui empat tahapan, yaitu *resize*, pembagian dataset, augmentasi, dan normalisasi.

1) *Resize*.

Citra dalam dataset SLD memiliki dimensi yang bervariasi antar sampel. Seluruh citra diubah ukurannya menjadi 224×224 piksel untuk memenuhi dimensi input standar arsitektur DenseNet121 dan DenseNet201 yang menggunakan bobot prelatih dari dataset *ImageNet* [11].

2) *Split Dataset*

Dataset SLD dibagi secara acak menjadi tiga subset dengan rasio 70:15:15 untuk training, validation, dan testing, sehingga diperoleh 4.723 citra pada subset training, 1.012 citra pada subset validation, dan 1.013 citra pada subset testing. Pembagian ini menghasilkan 4.723 citra untuk *training*, 1.012 citra untuk *validation*, dan 1.013 citra untuk *testing*. Pemisahan ketiga subset dilakukan secara ketat agar data yang digunakan pada tahap evaluasi dan pengujian tidak pernah dilihat model selama *training*, sehingga hasil evaluasi mencerminkan kemampuan generalisasi model yang sesungguhnya [13].

3) *Augmentasi*

Augmentasi data merupakan teknik untuk meningkatkan jumlah dan keberagaman data *train* melalui penerapan berbagai transformasi pada citra asli tanpa mengubah labelnya, sehingga model lebih mampu menggeneralisasi terhadap data baru [14], [15]. Teknik ini diterapkan secara eksklusif pada subset *training*, sedangkan subset *validation* dan *testing* tidak dikenai augmentasi guna mempertahankan objektivitas evaluasi [15]. Pada penelitian ini, setiap kelas pada data *training* ditargetkan memiliki minimal 1.000 citra melalui beberapa teknik augmentasi yang disajikan pada Tabel 2.

Tabel 2 Distribusi Dataset

No	Jenis Augmentasi	Parameter
1	Horizontal flip	True
2	Rotasi	0.05
3	Zoom	0.15
4	Contrast	0.10
5	Brightness	0.10

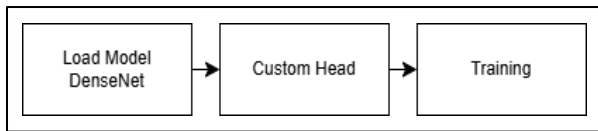
4) *Normalisasi*.

Normalisasi merupakan tahap *preprocessing* yang bertujuan menyesuaikan distribusi nilai piksel citra agar proses pelatihan model lebih stabil dan konvergen lebih cepat [16], [17]. Pada penelitian ini, normalisasi diterapkan menggunakan fungsi *preprocess_input* DenseNet201 dari Keras, yang secara otomatis menyesuaikan nilai piksel dari rentang [0–255] berdasarkan statistik dataset *ImageNet* sehingga *input* model menjadi konsisten di seluruh subset data.

C. Pengembangan Model

Penelitian ini membangun dua model klasifikasi menggunakan arsitektur DenseNet121 dan DenseNet201 untuk mengidentifikasi penyakit daun tebu. Kedua arsitektur dipilih karena menerapkan mekanisme *dense connectivity* yang memungkinkan setiap lapisan menerima informasi dari seluruh

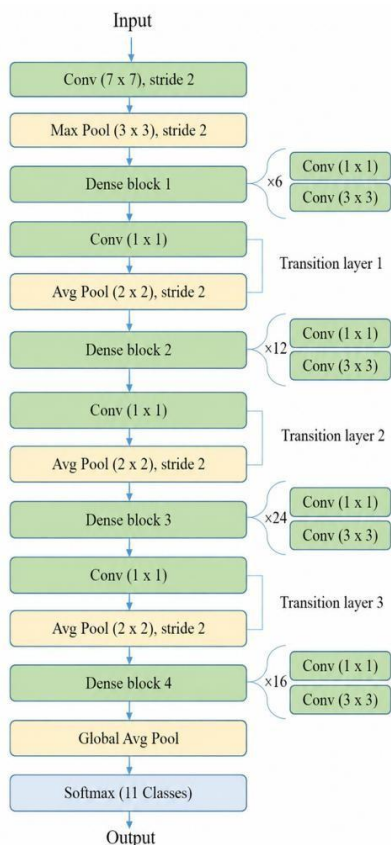
lapisan sebelumnya sehingga pemanfaatan fitur menjadi lebih efektif [7]. Model dikembangkan menggunakan pendekatan *transfer learning* dengan bobot pralatih ImageNet, kemudian lapisan klasifikasi akhir diganti dan disesuaikan dengan jumlah kelas pada dataset SLD. Alur pengembangan model ditunjukkan pada Gambar 2.



Gambar 2 Arsitektur DenseNet121

1) Arsitektur DenseNet121 dan Arsitektur DenseNet201

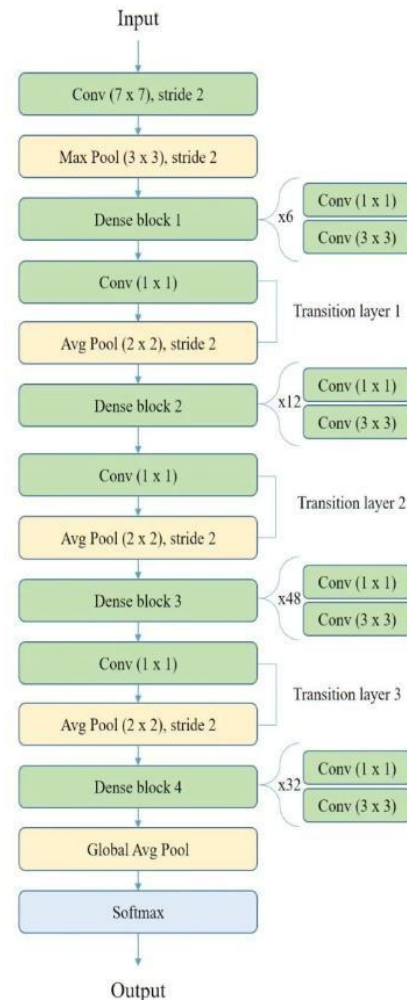
DenseNet121 dan DenseNet201 merupakan varian arsitektur *Dense Convolutional Network (DenseNet)* yang memanfaatkan koneksi padat (*dense connectivity*) antar *layer* sehingga memungkinkan penggunaan ulang fitur (*feature reuse*) dan meningkatkan aliran gradien selama proses pelatihan [18]. Pada penelitian ini, kedua model menggunakan bobot pralatih ImageNet dengan parameter *include_top=False* untuk mendukung proses *transfer learning* pada klasifikasi 11 kelas penyakit daun tebu. Arsitektur DenseNet121 yang digunakan pada penelitian ini ditunjukkan pada Gambar 3.



Gambar 3 Arsitektur DenseNet121

DenseNet121 terdiri atas empat *dense block* yang DenseNet121 terdiri atas empat *dense block* dengan konfigurasi *layer* 6, 12, 24, dan 16. Arsitektur ini menerapkan *dense*

connectivity yang memungkinkan setiap *layer* menerima keluaran dari seluruh *layer* sebelumnya sehingga mendukung *feature reuse*, meningkatkan aliran gradien, dan menghasilkan penggunaan parameter yang lebih efisien [9]. Arsitektur DenseNet201 yang digunakan pada penelitian ini ditunjukkan pada Gambar 4.



Gambar 4 Arsitektur DenseNet201

DenseNet201 memiliki struktur yang serupa dengan DenseNet121, namun dengan konfigurasi *layer* yang lebih dalam yaitu 6, 12, 48, dan 32 pada empat *dense block* [19]. Konfigurasi tersebut menghasilkan total 201 *layer* sehingga memungkinkan proses pembelajaran fitur dilakukan melalui lebih banyak lapisan dibandingkan DenseNet121 yang memiliki 121 *layer*.

2) Custom Head

Backbone DenseNet dilengkapi lapisan tambahan untuk mengadaptasi model pada klasifikasi 11 kelas. Rangkaian lapisan terdiri dari *GlobalAveragePooling2D*, *Batch Normalization*, dua lapisan *Dense* (512 dan 256 *neuron*) dengan aktivasi ReLU dan *regularisasi* L2, serta *Dropout* 0,5 dan 0,4 untuk mencegah *overfitting*. Lapisan keluaran menggunakan *Dense* 11 *neuron* dengan aktivasi *Softmax*, dengan konfigurasi lengkap disajikan pada Tabel 4.

Tabel 4. Custom Head Model

Layer	Output Shape
GlobalAveragePooling2D	(None,1920)
BatchNormalization	(None,1920)
Dense	(None,512)
Dropout (0.5)	(None,512)
Dense	(None,256)
Dropout (0.4)	(None,256)
Dense	(None,11)

3) Training Model

Training model dilakukan menggunakan subset *training* dan *validation* untuk memantau performa model pada setiap *epoch*. *Categorical Focal Loss* diterapkan untuk menangani ketidakseimbangan kelas [20] sedangkan *class weight* dihitung berdasarkan distribusi label *training* guna meningkatkan representasi kelas minoritas [21]. Selain itu, *Early Stopping* dan *ReduceLRonPlateau* digunakan untuk menjaga stabilitas pelatihan dan mengurangi risiko *overfitting*. Model dilatih menggunakan *optimizer* Adam, SGD, dan RMSprop untuk memperoleh konfigurasi yang memberikan performa terbaik.

D. Skenario Eksperimen

Penelitian ini merancang dua skenario eksperimen untuk membandingkan performa DenseNet121 dan DenseNet201 dalam mengidentifikasi penyakit daun tebu. Pada masing-masing arsitektur, dilakukan pengujian menggunakan *optimizer* Adam, SGD, dan RMSprop dengan konfigurasi pelatihan yang identik. Dengan demikian, pengaruh arsitektur *backbone* dan *optimizer* terhadap performa model dapat dianalisis secara objektif. Detail konfigurasi eksperimen disajikan pada Tabel 5.

Tabel 5 Skenario Eksperimen

Parameter	Skenario	Skenario
Backbone	DenseNet121	DenseNet201
Pretrained	ImageNet	ImageNet
Epochs	50	50
Batch Size	8	8
Optimizer	Adam, RMSProp, SGD	Adam, RMSProp, SGD
Learning Rate	1e - 4	1e - 4
Loss Function	Categorical Focal Loss ($\gamma=2.0$, $\alpha=1.5$)	Categorical Focal Loss ($\gamma=2.0$, $\alpha=1.5$)
Class Weight	Balanced	Balanced
Early Stopping	patience=8	patience=8
ReduceLRonPlateau	factor=0.3, patience=4	factor=0.3, patience=4

E. Evaluasi Model

Model dievaluasi pada subset *testing* untuk mengukur kemampuan generalisasinya terhadap data yang belum pernah dilihat selama model dilatih. Kinerja model dianalisis menggunakan *confusion matrix* dengan metrik *accuracy*, *precision*, *recall*, dan *F1-score* [22]. Perhitungan metrik evaluasi mengacu pada Persamaan (1), Persamaan (2), Persamaan (3), dan Persamaan (4).

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$sensitivity \text{ atau } recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

III. HASIL DAN PEMBAHASAN

Bagian ini menampilkan hasil pengembangan dan evaluasi model klasifikasi penyakit daun tebu yang dibangun dengan arsitektur DenseNet121 dan DenseNet201 berbasis transfer learning. Hasil disajikan dalam empat bagian, yaitu hasil pengembangan model, evaluasi model pada subset pengujian, analisis perbandingan performa DenseNet121 dan DenseNet201, serta pembahasan keterkaitan hasil dengan penelitian terdahulu.

A. Hasil Pengembangan Model

Pengembangan model dilakukan dengan melatih DenseNet121 dan DenseNet201 menggunakan tiga *optimizer*, yaitu Adam, RMSprop, dan SGD with Momentum, sesuai skenario eksperimen pada Tabel 5. Setiap model dilatih selama maksimal 50 *epoch* dengan mekanisme *Early Stopping* dan *ReduceLRonPlateau*, sehingga jumlah *epoch* aktual yang ditempuh setiap model bervariasi tergantung pada kestabilan *loss* validasi. Ringkasan hasil akurasi pada subset *training*, *validation*, dan *testing*, beserta waktu pelatihan untuk setiap kombinasi arsitektur dan *optimizer* disajikan pada Tabel 6

Tabel 6 Hasil Pengembangan Model

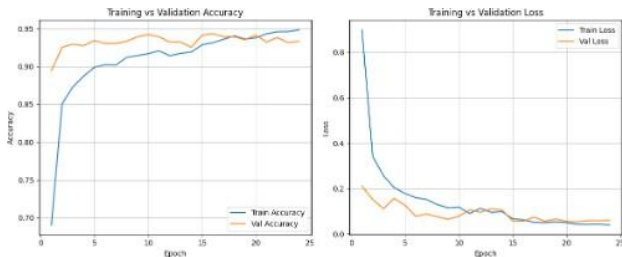
Backbone	Optimizer	Train (%)	Validat ion (%)	Test (%)	Waktu
DenseNet121	Adam	95,16	93,94	93,15	45 menit
	RMSprop	94,95	94,33	92,85	51 menit
	SGD	93,87	93,34	91,97	42 menit
DenseNet201	Adam	94,81	94,33	94,22	102 menit
	RMSprop	95,87	94,04	93,54	90 menit
	SGD	92,84	93,54	92,66	51 menit

B. Evaluasi Model

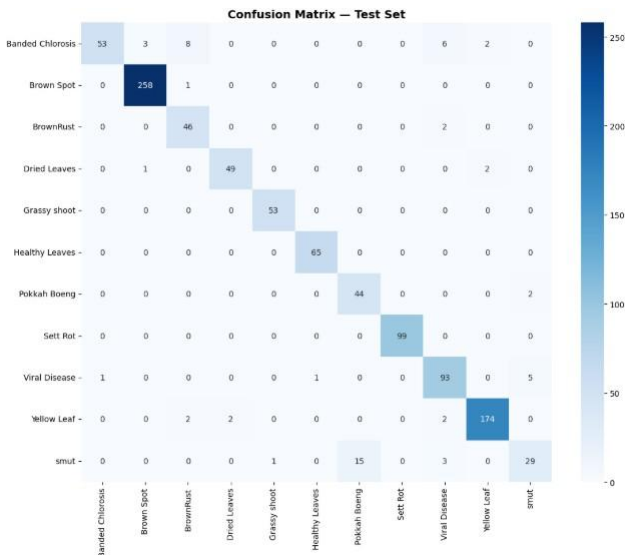
Evaluasi model dilakukan pada data pengujian untuk mengukur kemampuan klasifikasi masing-masing konfigurasi yang telah dilatih. Analisis dilakukan berdasarkan nilai akurasi pengujian, grafik akurasi dan *loss* selama pelatihan, serta *confusion matrix* untuk melihat distribusi prediksi pada setiap kelas penyakit daun tebu. Enam konfigurasi model yang terdiri atas DenseNet121 dan DenseNet201 dengan *optimizer* Adam, RMSprop, dan SGD dievaluasi untuk mengetahui performa masing-masing model.

1) DenseNet201 + Adam

Model DenseNet201 dengan *optimizer* Adam menghasilkan akurasi pengujian sebesar 94,22% dengan waktu pelatihan 102 menit. Grafik akurasi dan *loss* pelatihan ditunjukkan pada Gambar 4, sedangkan *confusion matrix* ditunjukkan pada Gambar 5. Kurva pelatihan menunjukkan peningkatan akurasi dan penurunan *loss* yang stabil hingga akhir pelatihan. Selain itu, selisih akurasi antara data latih (94,81%) dan validasi (94,33%) hanya sebesar 0,48%, yang mengindikasikan kemampuan generalisasi model yang baik.



Gambar 1 Grafik Akurasi dan Loss Pelatihan DenseNet201 + Adam

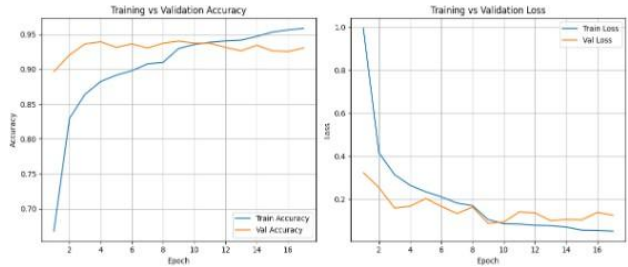


Gambar 5 Hasil Confusion Matrix Pelatihan DenseNet201 + Adam

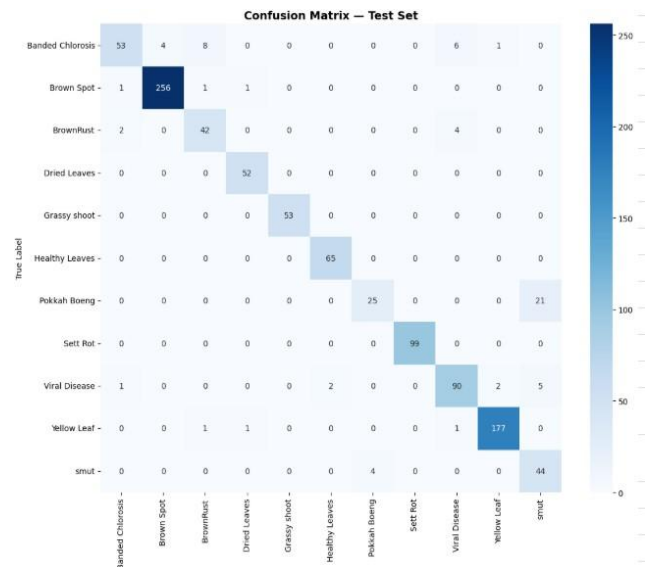
2) DenseNet201 + RMSProp

Model DenseNet201 dengan *optimizer* RMSprop menghasilkan akurasi pengujian sebesar 93,54% dengan waktu pelatihan 90 menit. Grafik akurasi dan *loss* pelatihan ditunjukkan pada Gambar 6, sedangkan *confusion matrix* ditunjukkan pada Gambar 7. Kurva pelatihan menunjukkan

proses konvergensi yang stabil hingga akhir pelatihan. Selisih akurasi antara data latih (95,87%) dan validasi (94,04%) sebesar 1,83%, yang menunjukkan kemampuan generalisasi yang baik.



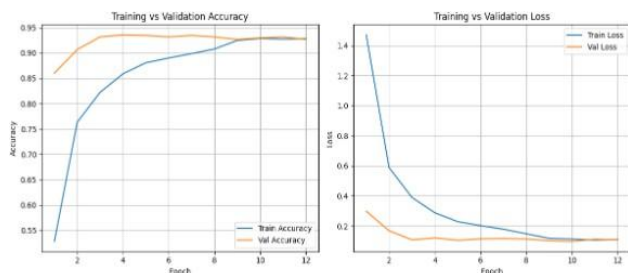
Gambar 6 Grafik Akurasi dan Loss Pelatihan DenseNet201 + RMSProp



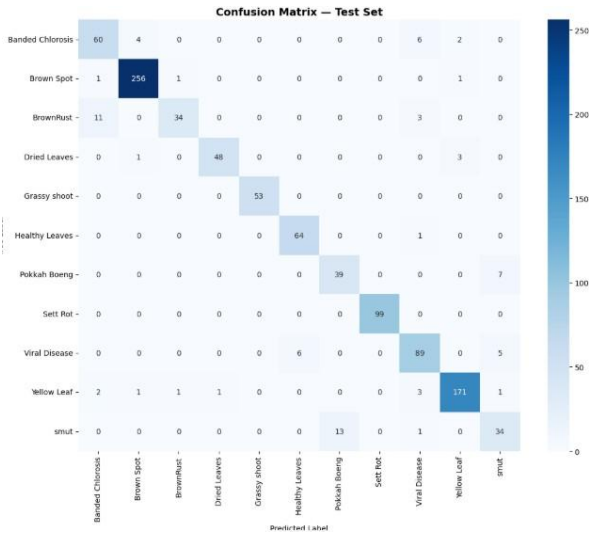
Gambar 7 Confusion Matrix Pelatihan DenseNet201 + RMSProp

3) DenseNet201 + SGD with Momentum

Model DenseNet201 dengan *optimizer* SGD menghasilkan akurasi pengujian terendah di antara konfigurasi DenseNet201, yaitu 92,66%, dengan waktu pelatihan paling singkat sebesar 51 menit. Grafik akurasi dan *loss* pelatihan serta kurva validasi ditunjukkan pada Gambar 8, sedangkan *confusion matrix* pada data pengujian ditunjukkan pada Gambar 9. Grafik pelatihan menunjukkan konvergensi yang lebih lambat dan tidak selurus konfigurasi Adam maupun RMSprop, dengan fluktuasi yang lebih terlihat pada kurva *loss* validasi.



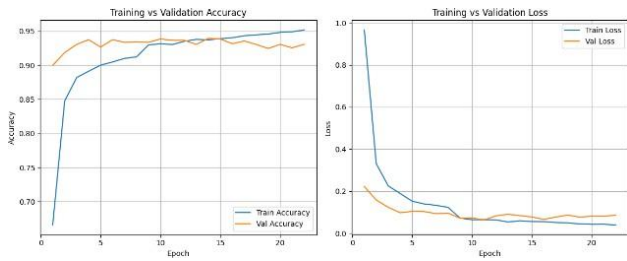
Gambar. 8 Grafik Akurasi dan Loss Pelatihan DenseNet201 + SGD with Momentum



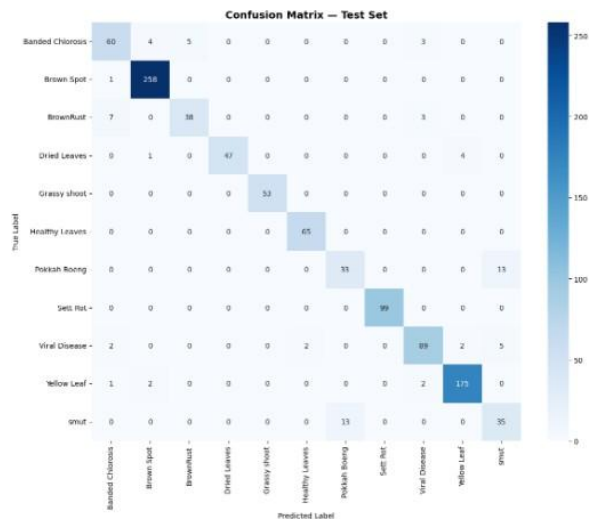
Gambar. 9 Confusion Matrix Pelatihan DenseNet201 + SGD with Momentum

4) DenseNet121 + Adam

Model DenseNet121 dengan optimizer Adam menghasilkan akurasi pengujian sebesar 93,15% dengan waktu pelatihan 45 menit. Grafik akurasi dan loss pelatihan ditunjukkan pada Gambar 10, sedangkan *confusion matrix* pada data pengujian ditunjukkan pada Gambar 11. Kurva pelatihan memperlihatkan proses konvergensi yang stabil hingga akhir pelatihan. Selain itu, akurasi data latih sebesar 95,16% dan akurasi validasi sebesar 93,94%.



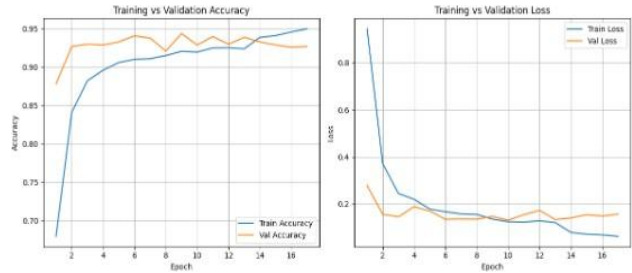
Gambar 10 Grafik Akurasi dan Loss Pelatihan DenseNet121 + Adam



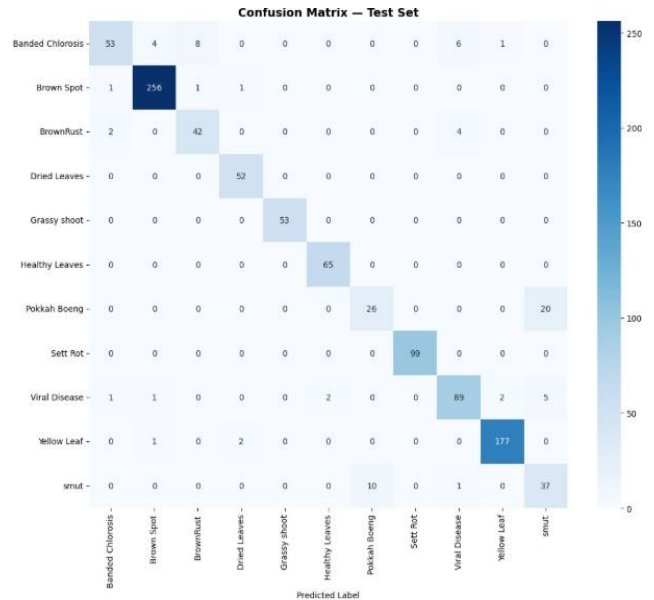
Gambar 11 Confusion Matrix Pelatihan DenseNet121 + Adam

5) DenseNet121 + RMSProp

Model DenseNet121 dengan optimizer RMSprop menghasilkan akurasi pengujian sebesar 92,85% dengan waktu pelatihan 51 menit. Grafik akurasi dan *loss* pelatihan ditunjukkan pada Gambar 12, sedangkan *confusion matrix* pada data pengujian ditunjukkan pada Gambar 13. Konfigurasi ini menempati posisi kedua pada kelompok DenseNet121 dengan selisih akurasi pengujian sebesar 0,30% dibandingkan DenseNet121+Adam. Selain itu, model memperoleh akurasi data latih sebesar 94,95% dan akurasi validasi sebesar 94,33%.



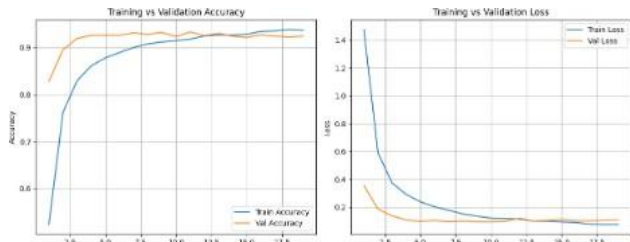
Gambar 12 Grafik Akurasi dan Loss Pelatihan DenseNet121 + RMSProp



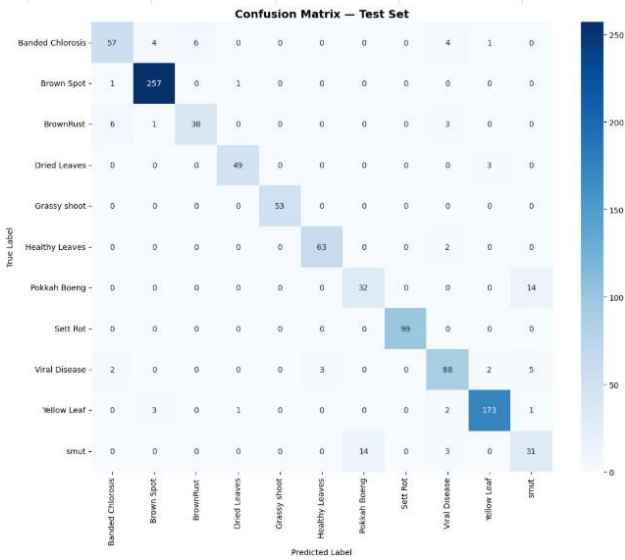
Gambar 13 Confusion Matrix Pelatihan DenseNet121 + RMSProp

6) DenseNet121 + SGD with Momentum

Model DenseNet121 dengan optimizer SGD menghasilkan akurasi pengujian sebesar 91,97% dengan waktu pelatihan 42 menit. Grafik akurasi dan loss pelatihan ditunjukkan pada Gambar 14, sedangkan *confusion matrix* pada data pengujian ditunjukkan pada Gambar 15. Konfigurasi ini menghasilkan akurasi pengujian terendah, namun memiliki waktu pelatihan paling singkat dibandingkan seluruh skenario yang diuji. Selain itu, model memperoleh akurasi data latih sebesar 93,87% dan akurasi validasi sebesar 93,34%.



Gambar 14 Grafik Akurasi dan Loss Pelatihan DenseNet121 + SGD with Momentum



Gambar 15 Confusion Matrix Pelatihan DenseNet201 + SGD with Momentum

A. Analisis Perbandingan DenseNet201 dan DenseNet121

Perbandingan kinerja model dilakukan terhadap seluruh skenario eksperimen dengan menggunakan nilai *precision*, *recall*, dan *F1-score* yang diperoleh dari data pengujian. Keenam konfigurasi tersebut merupakan kombinasi antara dua arsitektur *backbone*, yaitu DenseNet121 dan DenseNet201, dengan tiga optimizer berbeda, yaitu Adam, RMSprop, dan SGD. Analisis ini bertujuan mengidentifikasi konfigurasi terbaik dalam proses klasifikasi penyakit daun tebu menggunakan dataset SLD. Ringkasan performa setiap konfigurasi model dapat dilihat pada Tabel 7.

Tabel 7 Hasil Perbandingan Model

Backbone	Optimizer	Precision(%)	Recall (%)	F1Score (%)
DenseNet121	Adam	93	93	93
	RMSprop	93	93	93
	SGD	89	89	89
DenseNet201	Adam	95	94	94
	RMSprop	94	94	93
	SGD	91	93	93

Berdasarkan hasil yang disajikan pada Tabel 7, konfigurasi DenseNet201 dengan optimizer Adam memperoleh *precision* sebesar 95%, *recall* 94%, serta *F1-score* sebesar 94%. Pada penggunaan optimizer RMSprop, model DenseNet201 mencapai *precision* dan *recall* masing-masing sebesar 94%, dengan *F1-score* sebesar 93%. DenseNet201 dengan optimizer SGD menghasilkan nilai *precision* sebesar 91%, *recall* 93%, dan *F1-score* 93%. Untuk arsitektur DenseNet121, konfigurasi dengan optimizer Adam maupun RMSprop menunjukkan performa yang sama, yaitu *precision*, *recall*, dan *F1-score* sebesar 93%, sedangkan optimizer SGD menghasilkan nilai *precision*, *recall*, dan *F1-score* sebesar 89%. Secara umum, setiap konfigurasi DenseNet201 menunjukkan nilai *precision* yang lebih baik dibandingkan DenseNet121 ketika menggunakan optimizer yang identik. Di antara seluruh konfigurasi, optimizer SGD secara konsisten menghasilkan nilai metrik terendah pada kedua arsitektur, sedangkan optimizer Adam pada DenseNet201 menghasilkan nilai *precision* tertinggi di antara seluruh konfigurasi yang diuji.

B. Pembahasan

Hasil pengujian pada Tabel 7 menunjukkan bahwa seluruh model mampu mencapai nilai *precision*, *recall*, dan *F1-score* yang melebihi 89%. Performa tertinggi dicapai oleh DenseNet201 yang menggunakan optimizer Adam dengan nilai *precision* 95%, *recall* 94%, dan *F1-score* 94%. Sementara itu, DenseNet121 dengan optimizer Adam menghasilkan *precision*, *recall*, dan *F1-score* sebesar 93%. Temuan ini mengindikasikan bahwa DenseNet201 lebih efektif dalam melakukan klasifikasi penyakit daun tebu dibandingkan DenseNet121. Hal ini terlihat dari nilai *precision* dan *F1-score* yang lebih tinggi pada hampir seluruh skenario pengujian. Selain itu, penggunaan optimizer Adam memberikan hasil yang paling baik pada kedua arsitektur dibandingkan RMSprop maupun SGD. Pada penelitian ini, optimizer SGD menghasilkan performa terendah baik pada DenseNet121 maupun DenseNet201. Kondisi tersebut juga terlihat pada nilai akurasi pengujian yang lebih rendah dibandingkan optimizer lainnya. Meskipun demikian, waktu pelatihan SGD lebih singkat dibandingkan Adam dan RMSprop.

IV. KESIMPULAN

Penelitian ini berhasil membandingkan performa arsitektur DenseNet121 dan DenseNet201 berbasis *transfer learning* dalam klasifikasi penyakit daun tebu menggunakan *Sugarcane Leaf Dataset (SLD)*. Hasil pengujian menunjukkan bahwa DenseNet201 dengan optimizer Adam memberikan performa terbaik dengan akurasi sebesar 94,22%, *precision* 95%, *recall* 94%, dan *F1-score* 94%. Sementara itu, DenseNet121 dengan optimizer Adam menghasilkan akurasi sebesar 93,15%.

Berdasarkan hasil tersebut, DenseNet201 terbukti lebih unggul dibandingkan DenseNet121 dalam mengidentifikasi penyakit daun tebu. Selain itu, optimizer Adam memberikan hasil yang lebih baik dibandingkan RMSprop dan SGD pada kedua arsitektur yang diuji. Dengan demikian, kombinasi DenseNet201 dan Adam dapat direkomendasikan sebagai model yang efektif untuk klasifikasi penyakit daun tebu.

REFERENSI

- [1] J. F. Sinuraya, E. A. Suryana, L. R. Shaffitri, S. R. Suharyono, and H. R. Hermawan, "Kinerja Industri dan Dinamika Kebijakan Komoditas Gula Kristal Putih Nasional," *Indonesian Sugar Research Journal*, vol. 4, no. 2, pp. 68–79, Dec. 2024, doi: 10.54256/isrj.v4i2.129.
- [2] R. Viswanathan, "Emerging diseases of sugarcane and new disease management strategies," *Journal of Sugarcane Research*, vol. 12, no. 1, pp. 1–15, Jun. 2022, doi: 10.37580/jsr.2022.1.12.1-15.
- [3] S. D. Daphal and S. M. Koli, "Enhanced deep learning technique for sugarcane leaf disease classification and mobile application integration," *Heliyon*, vol. 10, no. 8, Apr. 2024, doi: 10.1016/j.heliyon.2024.e29438.
- [4] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," Aug. 01, 2021, *MDPI AG*. doi: 10.3390/agriculture11080707.
- [5] D. Tejaswi, T. Sri Vaishnavi, B. Nandini, P. Nuka Raju, and D. Jayanth Babu, "Plant disease detection using deep learning," *International Journal of Science and Research Archive*, vol. 12, no. 1, pp. 2476–2488, Jun. 2024, doi: 10.30574/ijrsra.2024.12.1.1043.
- [6] M. I. Hossen, M. Awrangjeb, S. Pan, and A. Al Mamun, "Transfer learning in agriculture: a review," *Artif. Intell. Rev.*, vol. 58, no. 4, Apr. 2025, doi: 10.1007/s10462-024-11081-x.
- [7] M. K. A. Mazumder, M. M. Kabir, A. Rahman, M. Abdullah-Al-Jubair, and M. F. Mridha, "DenseNet201Plus: Cost-effective transfer-learning architecture for rapid leaf disease identification with attention mechanisms," *Heliyon*, vol. 10, no. 15, Aug. 2024, doi: 10.1016/j.heliyon.2024.e35625.
- [8] A. Ismail, W. Hamdy, A. H. Ibrahim, and W. A. Awad, "Classification of rice plant diseases using efficient DenseNet121," *Sci. Rep.*, vol. 16, no. 1, Dec. 2026, doi: 10.1038/s41598-026-38078-6.
- [9] A. D. Saputra, D. Hindarto, and H. Santoso, "Disease Classification on Rice Leaves using DenseNet121, DenseNet169, DenseNet201," *Sinkron*, vol. 8, no. 1, pp. 48–55, Jan. 2023, doi: 10.33395/sinkron.v8i1.11906.
- [10] B. S. Devi, K. S. Chatrapati, and N. Sandhya, "Enhanced Sugarcane Disease Detection Using DenseNet201 and DenseNet264 with Transfer Learning and Fine-tuning," *Frontiers in Health Informatics*, vol. 13, no. 3, 2024, [Online]. Available: www.healthinformaticsjournal.com
- [11] S. Srinivasan, S. M. Prabin, S. K. Mathivanan, H. Rajadurai, S. Kulandaivelu, and M. A. Shah, "Sugarcane leaf disease classification using deep neural network approach," *BMC Plant Biol.*, vol. 25, no. 1, Dec. 2025, doi: 10.1186/s12870-025-06289-0.
- [12] A. Y. Ashurov *et al.*, "Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections," *Front. Plant Sci.*, vol. 15, 2024, doi: 10.3389/fpls.2024.1505857.
- [13] K. M. Kahloot and P. Ekler, "Algorithmic Splitting: A Method for Dataset Preparation," *IEEE Access*, vol. 9, pp. 125229–125237, 2021, doi: 10.1109/ACCESS.2021.3110745.
- [14] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, May 2023, doi: 10.1016/j.patcog.2023.109347.
- [15] C. Hastings Blow, L. Qian, C. Gibson, P. Obiomon, and X. Dong, "Data augmentation via diffusion model to enhance AI fairness," *Front. Artif. Intell.*, vol. 8, 2025, doi: 10.3389/frai.2025.1530397.
- [16] K. Anirudh, D. Srujan, A. Sai, and G. S. O. V., "Comparative deep learning approaches for bean leaf disease recognition," *Front. Plant Sci.*, vol. 17, Apr. 2026, doi: 10.3389/fpls.2026.1703449.
- [17] Z. Salman, A. Muhammad, and D. Han, "Plant disease classification in the wild using vision transformers and mixture of experts," *Front. Plant Sci.*, vol. 16, 2025, doi: 10.3389/fpls.2025.1522985.
- [18] T. Zhou, X. Ye, H. Lu, X. Zheng, S. Qiu, and Y. Liu, "Dense Convolutional Network and Its Application in Medical Image Analysis," 2022, *Hindawi Limited*. doi: 10.1155/2022/2384830.
- [19] J. Kumari, S. K. Behera, P. K. Sathy, and A. Nanthaamornphong, "Enhanced Parkinson's Disease Diagnosis via MRI Analysis: Integrating Deep Features From DenseNet201 With Neural Network Techniques," *Applied Computational Intelligence and Soft Computing*, vol. 2024, no. 1, 2024, doi: 10.1155/ACIS/5582371.
- [20] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo, "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation," Nov. 2021, [Online]. Available: <http://arxiv.org/abs/2102.04525>
- [21] K. Ahadian *et al.*, "Maize disease classification using transfer learning and convolutional neural network with weighted loss," *Heliyon*, vol. 10, no. 21, Nov. 2024, doi: 10.1016/j.heliyon.2024.e39569.
- [22] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," 2023, *Frontiers Media SA*. doi: 10.3389/fpls.2023.1158933.