

CORN PLANT DISEASE IDENTIFICATION MODEL BASED ON LEAF IMAGE USING CONVNEXT AND MAXVIT

MODEL IDENTIFIKASI PENYAKIT TANAMAN JAGUNG BERBASIS CITRA DAUN MENGGUNAKAN CONVNEXT DAN MAXVIT

Naufal Alwi Pratama^{1‡}, Arie Qur'ania², and Yusma Yanti³

^{1,2,3}Department of Computer Science, Pakuan University, Indonesia

[‡]corresponding author: 065122096@student.unpak.ac.id

Copyright © 2026 Naufal Alwi Pratama, Arie Qur'ania, and Yusma Yanti. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Fast and accurate identification of corn leaf diseases is crucial for reducing the risk of yield losses caused by delayed treatment of plant diseases. This study aims to develop and compare the performance of the ConvNeXt and MaxViT deep learning architectures in the classification of corn leaf diseases based on digital images. The research dataset consists of 7.270 corn leaf images across five classes: Common Rust, Northern Corn Leaf Blight, Gray Leaf Spot, Pest Damage, and Healthy. Of these, 2.035 images were collected as primary data from a corn field in Cilangkap, Depok, West Java, while the remaining 5.235 images were obtained as secondary data from the Kaggle Maize Diseases Dataset and the Plant Doc dataset. The research stages include data collection, data preprocessing, image augmentation, model training using a transfer learning approach, and model evaluation using the accuracy, precision, recall, and F1-score metrics with the macro average method. The study utilized several data splitting scenarios and combinations of the Adam and AdamW optimizers. The results showed that the ConvNeXt-Tiny model outperformed MaxViT-Tiny. The best model was achieved using ConvNeXt-Tiny with the AdamW optimizer and a 90:5:5 data split scenario, yielding an accuracy of 96.98% and precision, recall, and F1-score values of 97%. This result indicates that the model possesses excellent reliability in accurately detecting and distinguishing the five leaf conditions. The best model was then implemented into a web-based application using Vue.js and FastAPI, enabling practical identification of corn leaf diseases through image upload and Grad-CAM visualization to prevent delays in disease treatment and the risk of crop failure.

Keywords: Computer Vision, ConvNeXt, Corn Leaf Disease, Deep Learning, MaxViT.

Abstrak

Identifikasi penyakit daun jagung secara cepat dan akurat sangat penting untuk mengurangi risiko penurunan hasil panen akibat keterlambatan penanganan penyakit tanaman. Penelitian ini bertujuan untuk mengembangkan serta membandingkan performa arsitektur *deep learning* ConvNeXt dan MaxViT dalam klasifikasi penyakit daun jagung berbasis citra digital. Dataset penelitian terdiri atas 7.270 citra daun jagung yang terbagi ke dalam lima kelas, yaitu Common Rust, Northern Corn Leaf Blight, Gray Leaf Spot, Pest Damage, dan Healthy. Sebanyak 2.035 citra merupakan data primer yang dikumpulkan langsung dari kebun jagung di Kelurahan Cilangkap, Kota Depok, Jawa Barat, sedangkan 5.235 citra lainnya merupakan data sekunder yang bersumber dari *Kaggle Maize Diseases Dataset* dan *Plant Doc dataset*. Tahapan penelitian meliputi pengumpulan data, *preprocessing* data, augmentasi citra, pelatihan model menggunakan pendekatan *transfer learning*, serta evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* dengan metode *macro average*. Penelitian menggunakan beberapa skenario *data splitting* dan kombinasi optimizer Adam serta AdamW. Hasil penelitian menunjukkan bahwa model ConvNeXt-Tiny memberikan performa terbaik dibandingkan MaxViT-Tiny. Model terbaik diperoleh pada ConvNeXt-Tiny dengan optimizer AdamW dan skenario *data split* 90:5:5 yang menghasilkan nilai *accuracy* sebesar 96,98% serta nilai *precision*, *recall*, dan *F1-score* sebesar 97%. Hasil ini menunjukkan bahwa model memiliki keandalan yang sangat baik dalam mendeteksi dan membedakan kelima kelas kondisi daun secara tepat. Model terbaik selanjutnya diimplementasikan ke dalam aplikasi berbasis web menggunakan Vue JS dan FastAPI sehingga mampu melakukan identifikasi penyakit daun jagung secara praktis melalui proses unggah citra dan visualisasi *Grad-CAM* guna mencegah keterlambatan penanganan penyakit dan risiko gagal panen.

Kata Kunci: Computer Vision, ConvNeXt, Deep Learning, MaxViT, Penyakit Daun Jagung.

1. Pendahuluan

Jagung (*Zea mays L.*) merupakan tanaman pangan utama ketiga setelah padi dan gandum di dunia dan menempati posisi kedua setelah padi di Indonesia (Hutabarat *et al.*, 2025). Berdasarkan data terbaru dari Badan Pusat Statistik tahun 2025, dinamika sektor pertanaman jagung menunjukkan tren yang positif. Luas panen jagung di Indonesia pada tahun 2024 mencapai 2,55 juta hektare, mengalami peningkatan sebesar 2,93% dibandingkan tahun sebelumnya. Namun, di balik pertumbuhan angka produksi tersebut, sektor pertanian jagung masih dihadapkan pada risiko

kerugian yang nyata. Laporan BPS mencatat adanya luasan potensi gagal panen yang mencapai total 79 ribu hektare sepanjang tahun 2024, dengan rata-rata bulanan sebesar 6,61 ribu hektare. Salah satu faktor utama yang berkontribusi terhadap risiko gagal panen ini adalah serangan hama dan penyakit tanaman yang seringkali terlambat diidentifikasi oleh petani.

Perkembangan teknologi *Artificial Intelligence* (AI), khususnya pada bidang *Computer Vision*, menunjukkan bahwa pendekatan berbasis *Deep Learning* sangat efektif dalam klasifikasi citra tanaman. Berbagai penelitian sebelumnya telah membuktikan kemampuan arsitektur *Convolutional Neural Network* (CNN) dalam mengekstraksi fitur visual penyakit tanaman secara otomatis. Penelitian oleh Maulini et al., (2025) menggunakan arsitektur DenseNet121 dengan pendekatan *transfer learning* dan memperoleh akurasi sebesar 93,48% pada klasifikasi penyakit daun jagung. Selain itu, penelitian Tamayasa & Dewi, (2026) menunjukkan bahwa MobileNetV2 mampu mencapai akurasi hingga 96,50% dalam klasifikasi penyakit daun jagung, mengungguli model EfficientNetB3 yang hanya mencapai akurasi sebesar 93%. Hasil penelitian tersebut menunjukkan bahwa penggunaan model *deep learning* sangat relevan untuk mengatasi permasalahan di sektor pertanian, sekaligus memberikan hasil identifikasi yang cepat dan konsisten.

Meskipun berbagai model CNN telah menunjukkan performa yang baik, penelitian ini memilih untuk mengimplementasikan dan membandingkan dua arsitektur modern, yaitu ConvNeXt dan MaxViT. ConvNeXt merupakan arsitektur CNN modern yang dikembangkan dengan mengadopsi prinsip desain Vision Transformer sehingga mampu menghasilkan performa kompetitif pada tugas *computer vision* dengan tetap mempertahankan efisiensi operasi konvolusi (Liu et al., 2022). Sementara itu, MaxViT merupakan arsitektur *hybrid* berbasis CNN dan Vision Transformer yang memperkenalkan mekanisme *Multi-axis self attention* untuk menangkap pola visual secara lebih detail dan menyeluruh dengan beban komputasi yang tetap linear (Tu et al., 2022). Perbandingan kedua arsitektur tersebut diharapkan dapat memberikan gambaran mengenai efektivitas dua paradigma *deep learning* modern dalam mengidentifikasi penyakit daun jagung.

Tujuan dari penelitian ini adalah mengembangkan serta membandingkan performa model ConvNeXt dan MaxViT dalam mengidentifikasi penyakit daun jagung berbasis citra digital, sehingga diperoleh model yang mampu memberikan hasil klasifikasi secara akurat dan dapat diimplementasikan pada aplikasi berbasis web untuk mendukung deteksi dini penyakit tanaman jagung.

2. Metodologi

2.1 Bahan dan Data

Dataset citra daun jagung terdiri atas lima kelas, yaitu *Common Rust*, *Northern Corn Leaf Blight*, *Gray Leaf Spot*, *Pest Damage*, dan *Healthy*. Karakteristik visual masing-masing penyakit mengacu pada *literature review* oleh Pfordt & Paulus (2024). Ciri-ciri penyakit pada daun jagung bisa dilihat pada gambar 1.



Gambar 1: Penyakit pada daun jagung.

Dataset yang digunakan terdiri atas data primer dan data sekunder. Data primer diperoleh melalui pengambilan gambar langsung di kebun jagung seluas dua hektar yang berlokasi di Kelurahan Cilangkap, Kecamatan Tapos, Kota Depok, Jawa Barat. Pengambilan citra dilakukan menggunakan kamera belakang 64 MP *smartphone* POCO X6 Pro. Sementara itu, data sekunder diperoleh dari dataset publik Kaggle [Maize Diseases Dataset](#) dan *Plant Doc* dataset dari *literature review* oleh Jafar et al., (2024). Total data yang berhasil dikumpulkan sebanyak 7.270 citra daun jagung, terdiri dari 2.035 citra data primer dan 5.235 citra data sekunder. Jumlah data citra daun jagung per kelas dapat dilihat pada tabel 1.

Tabel 1: Jumlah data citra daun per kelas.

Jumlah					Total
Common Rust	Gray Leaf Spot	Northern Leaf Blight	Pest Damage	Healthy	
1.328	1.285	1.803	1.187	1.667	7.270

2.2 Metode Penelitian

Metode penelitian berfokus pada pengembangan model *deep learning* untuk klasifikasi penyakit daun jagung menggunakan arsitektur ConvNexT dan MaxViT. Analisis dilakukan menggunakan bahasa pemrograman Python dengan bantuan *package* PyTorch dan torchvision.

2.2.1. Preprocessing Data

Tahap *preprocessing* dilakukan untuk menyeragamkan format citra agar sesuai dengan spesifikasi input model. Proses ini meliputi *resizing* citra menjadi 224×224 piksel, konversi citra ke tensor, terapkan *Min-Max Scalling*, dan terakhir standardisasi per *channel* warna (RGB) menggunakan nilai *mean* [0.485, 0.456, 0.406] dan *standard deviation* [0.229, 0.224, 0.225] yang merupakan statistik standar dari dataset ImageNet. Tujuan dari standardisasi ini adalah untuk menyamakan distribusi data input dengan data yang digunakan saat model *pre-trained* dilatih, sehingga mempercepat konvergensi dan meningkatkan stabilitas performa model selama pelatihan (Iglovikov, 2025).

Pada tahap ini juga dilakukan pembagian data (*data splitting*) dengan metode *stratified split* untuk memastikan bahwa proporsi setiap label penyakit tersebar secara seimbang di setiap pembagian. Pada saat eksperimen dataset akan dibagi menjadi data latih, data validasi, dan data uji dengan rasio 70:15:15, 80:10:10, dan 90:5:5 untuk memastikan bahwa evaluasi performa model dilakukan secara objektif dan tidak terjadi *overfitting*.

2.2.2. Transformasi Data

Tahap transformasi data dilakukan menggunakan teknik augmentasi pada data latih meningkatkan variasi dan kekayaan informasi melalui teknik augmentasi data guna mencegah *overfitting*. Teknik augmentasi diterapkan secara dinamis pada data latih menggunakan pustaka *torchvision*. Teknik augmentasi yang diterapkan meliputi:

1. *Random Resized Crop*, melakukan pemotongan gambar secara acak dan mengubah ukurannya kembali menjadi 224×224 piksel dengan skala *zoom* antara 0,8 hingga 1,0.
2. *Flip (Horizontal & Vertical)*, melakukan pencerminan gambar secara horizontal maupun vertikal dengan probabilitas masing-masing sebesar 0,5 untuk memberikan variasi orientasi objek.
3. *Rotation* (-30° hingga 30° secara acak), untuk mensimulasikan perbedaan kemiringan pengambilan gambar.
4. *Translasi*, menerapkan pergeseran gambar hingga 30% baik secara horizontal maupun vertikal untuk mensimulasikan perbedaan lokasi area penyakit daun jagung di gambar.
5. *Color Jitter*, melakukan manipulasi pada aspek visual citra dengan mengubah tingkat kecerahan (*brightness*) sebesar 0,3, kontras (*contrast*) sebesar 0,3, dan saturasi (*saturation*) sebesar 0,2 untuk mengantisipasi perbedaan pencahayaan pada data riil.

2.2.3. Pelatihan Model

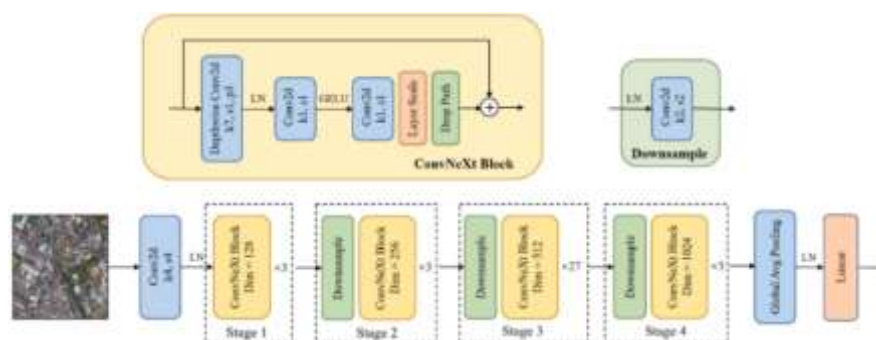
Tahap pelatihan dilakukan menggunakan dua arsitektur *deep learning* yang dilatih secara terpisah, yaitu ConvNeXt dan MaxViT. Kedua

model dilatih menggunakan pendekatan *transfer learning*. Model akan melakukan ekstraksi fitur pada citra berukuran 224×224 secara mendalam dari citra daun jagung untuk menemukan pola unik yang membedakan setiap jenis penyakit.

ConvNeXt

ConvNeXt merupakan arsitektur *Convolutional Neural Network* (CNN) modern yang dikembangkan dengan mengadopsi berbagai prinsip desain dari Vision Transformer (ViT) sehingga mampu menghasilkan performa kompetitif dibandingkan model transformer modern. Arsitektur ini memodernisasi struktur ResNet melalui penggunaan *depthwise convolution*, *inverted bottleneck*, kernel berukuran besar 7×7 , serta penggantian *Batch Normalization* menjadi *Layer Normalization* untuk meningkatkan stabilitas pelatihan model (Liu et al., 2022). Arsitektur ConvNeXt yang digunakan dalam penelitian ini adalah varian ConvNeXt-Tiny dengan bobot *pre-trained* ImageNet-1K yang tersedia melalui pustaka torchvision.

ConvNeXt memproses citra secara hierarkis melalui beberapa stage yang secara bertahap mengurangi resolusi spasial dan meningkatkan jumlah kanal fitur. Pada tahap akhir, fitur citra dirangkum menggunakan *global average pooling* sebelum dilakukan klasifikasi melalui *fully connected layer*. Pendekatan ini memungkinkan ConvNeXt mempertahankan efisiensi operasi konvolusi sekaligus memiliki kemampuan ekstraksi fitur visual yang kuat pada tugas klasifikasi citra. Diagram arsitektur ConvNeXt dapat dilihat pada Gambar 2. Dalam proses pelatihan dengan metode *transfer learning*, seluruh lapisan backbone ConvNeXt-Tiny di-*freeze* sehingga bobot hasil *pre-training* pada ImageNet-1K tetap dipertahankan. Hanya lapisan *classifier* (*head*) yang diaktifkan dan dilatih ulang untuk belajar mengklasifikasi penyakit daun jagung.



Gambar 2: Arsitektur ConvNeXt.

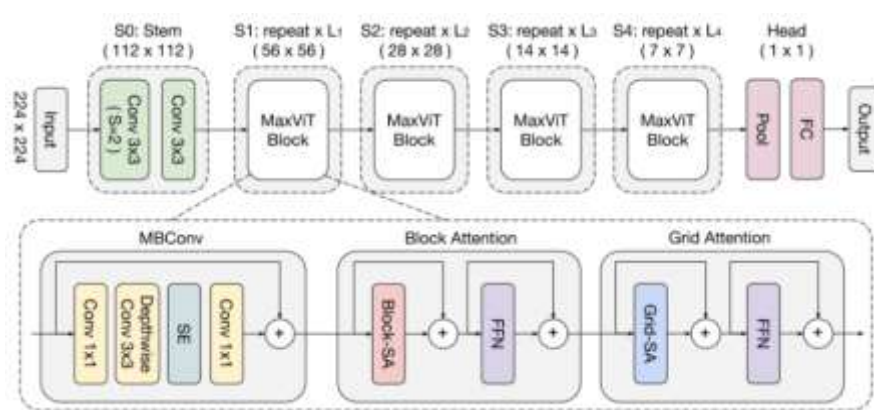
MaxViT

MaxViT (*Multi-Axis Vision Transformer*) merupakan arsitektur *Vision Transformer* hierarkis yang dirancang untuk mengatasi tingginya

kompleksitas komputasi pada mekanisme *self-attention* konvensional. Model ini memperkenalkan konsep *multi-axis attention* yang menggabungkan *blocked local attention* dan *dilated global attention* sehingga mampu menangkap informasi spasial lokal maupun global dengan kompleksitas linear (Tu et al., 2022). Arsitektur MaxViT yang digunakan dalam penelitian ini adalah varian MaxViT-Tiny dengan bobot *pre-trained* ImageNet-1K yang tersedia melalui pustaka torchvision.

Inti dari arsitektur ini adalah *MaxViT-Block* yang mengintegrasikan lapisan CNN bernama MBConv dengan mekanisme attention. *Blocked local attention* digunakan untuk menangkap detail lokal pada area kecil gambar, sedangkan *dilated global attention* memungkinkan model memahami hubungan global antar bagian citra. Kombinasi kedua mekanisme tersebut membuat MaxViT efektif dalam mengekstraksi fitur visual yang kompleks pada tugas klasifikasi citra.

Dalam proses klasifikasi citra, MaxViT memproses data secara hierarkis melalui beberapa stage dengan resolusi fitur yang semakin kecil dan jumlah kanal yang meningkat secara bertahap. Pada tahap akhir, fitur citra dirangkum menggunakan *global average pooling* sebelum diteruskan ke lapisan klasifikasi untuk menghasilkan prediksi kelas. Diagram arsitektur MaxViT dapat dilihat pada Gambar 3. Selama *training*, parameter pada lapisan *stem* dan Stage 1-4 di-*freeze* dengan tujuan untuk mempertahankan bobot fitur visual yang telah dipelajari dari dataset ImageNet-1K. Pembelajaran secara aktif hanya terjadi pada bagian *head classifier*, di mana model belajar menyesuaikan parameter internalnya agar dapat memetakan fitur-fitur visual penyakit daun jagung secara akurat berdasarkan fungsi *loss* yang dihitung selama pelatihan.



Gambar 3: Arsitektur MaxViT.

Model dilatih untuk mempelajari pola visual penyakit daun jagung melalui proses optimasi *hyperparameter* menggunakan *optimizer*, *learning rate*, *batch size*, dan sejumlah *epoch* tertentu. Tahap pelatihan model akan menggunakan *cross-entropy loss function* dan

skenario *data splitting* dari tahap *preprocessing data* serta beberapa *hyperparameter* lainnya yang menghasilkan kombinasi yang dilihat pada tabel 2.

Tabel 2: Skenario Pelatihan Data.

Kombinasi	Arsitektur	Optimiz er	Data Split	Learning Rate	Batch Size	Epoch s
Model 1	ConvNeXt	Adam	70:15:15	0.001	32	30
Model 2	ConvNeXt	Adam	80:10:10	0.001	32	30
Model 3	ConvNeXt	Adam	90:5:5	0.001	32	30
Model 4	ConvNeXt	AdamW	70:15:15	0.001	32	30
Model 5	ConvNeXt	AdamW	80:10:10	0.001	32	30
Model 6	ConvNeXt	AdamW	90:5:5	0.001	32	30
Model 7	MaxViT	Adam	70:15:15	0.001	32	30
Model 8	MaxViT	Adam	80:10:10	0.001	32	30
Model 9	MaxViT	Adam	90:5:5	0.001	32	30
Model 10	MaxViT	AdamW	70:15:15	0.001	32	30
Model 11	MaxViT	AdamW	80:10:10	0.001	32	30
Model 12	MaxViT	AdamW	90:5:5	0.001	32	30

2.2.4. Evaluasi Model

Evaluasi bertujuan untuk mengukur kemampuan model dalam mengklasifikasikan penyakit daun jagung berdasarkan data pengujian yang belum pernah dilihat sebelumnya. Proses evaluasi yang digunakan dalam penelitian ini akan menggunakan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang didapatkan dari *Confusion Matrix*. *Confusion Matrix* adalah sebuah matriks performa model yang menunjukkan jumlah data aktual yang diprediksi benar dan jumlah data aktual yang diprediksi salah untuk setiap kelasnya (Austin et al., 2024).

Tabel 3: Confusion Matrix.

Confusion Matrix		Actual Value	
		1 (Positive)	0 (Negative)
Predicted	1 (Positive)	True Positive (TP)	False Positive (FP)
Values	0 (Negative)	False Negative (FN)	True Negative (TN)

Keempat nilai dari *confusio* nilai dari *confusion matrix* di tabel 3 akan digunakan untuk menghitung nilai *Accuracy*, *Precision*, *Recall*, dan *F1-score* dari model melalui rumus pada persamaan 1 sampai 4.

a. *Accuracy* menunjukkan seberapa besar proporsi prediksi yang

benar dibandingkan dengan keseluruhan data yang diuji.

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

- b. *Precision* menilai sejauh mana model mampu memprediksi kelas positif secara tepat.

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

- c. *Recall* menggambarkan kemampuan model dalam menemukan seluruh sampel positif yang sebenarnya ada.

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

- d. *F1-score* digunakan untuk mengetahui keseimbangan antara ketepatan dan kelengkapan prediksi yang dilakukan oleh model.

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

Pada kasus *multi-class classification*, nilai *precision*, *recall*, dan *F1-Score* dihitung secara terpisah untuk masing-masing kelas. Namun, daripada memiliki beberapa nilai per kelas, akan lebih baik dirata-ratakan guna mendapatkan satu angka tunggal untuk menggambarkan kinerja keseluruhan. Ada beberapa *averaging method* yang bisa digunakan, tetapi dalam penelitian ini akan digunakan metode *macro average* karena memperlakukan setiap kelas secara setara (Leung, 2022). Rumus perhitungannya disajikan pada persamaan 5 sampai 7.

$$Precision_{macro\ avg} = \frac{1}{n} \sum_{i=1}^n Precision_{class\ i} \quad (5)$$

$$Recall_{macro\ avg} = \frac{1}{n} \sum_{i=1}^n Recall_{class\ i} \quad (6)$$

$$F1\ Score_{macro\ avg} = \frac{1}{n} \sum_{i=1}^n F1\ Score_{class\ i} \quad (7)$$

3. Hasil dan Pembahasan

3.1 Evaluasi Model

Tahap evaluasi model dilakukan menggunakan data uji yang belum pernah digunakan pada proses pelatihan maupun validasi. Perhitungan *precision*, *recall*, dan *F1-score* yang ditampilkan menggunakan metode *macro average* sehingga setiap kelas memiliki kontribusi yang sama dalam perhitungan nilai evaluasi tanpa dipengaruhi oleh jumlah data pada masing-masing kelas. Hasil evaluasi model untuk setiap skenario pelatihan ditampilkan pada Tabel 4.

Tabel 4: Evaluasi Model.

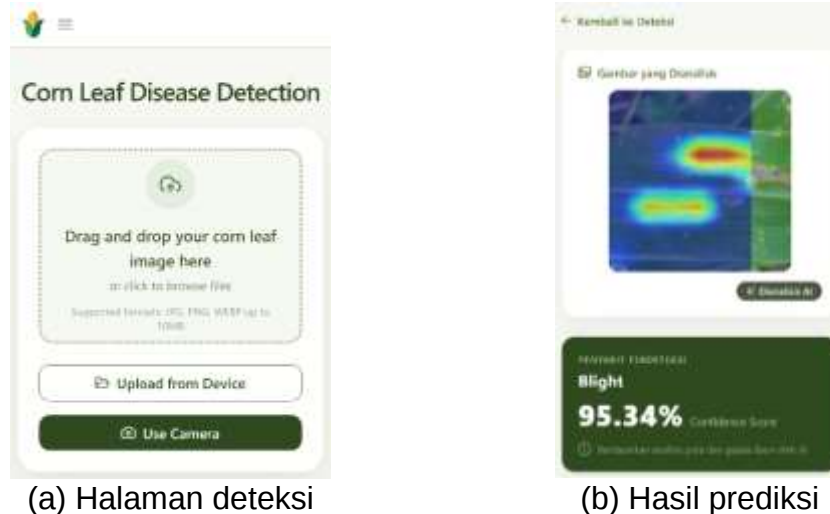
Model	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>
Model 1	95%	95%	95%	95,51%
Model 2	96%	95%	96%	95,60%
Model 3	97%	97%	97%	96,70%
Model 4	96%	96%	96%	95,88%
Model 5	97%	97%	97%	96,84%
Model 6	97%	97%	97%	96,98%
Model 7	94%	94%	94%	93,58%
Model 8	95%	95%	95%	94,91%
Model 9	94%	93%	93%	93,13%
Model 10	94%	94%	94%	93,49%
Model 11	95%	94%	95%	94,77%
Model 12	93%	93%	93%	92,86%

Berdasarkan hasil evaluasi pada Tabel 4, performa terbaik pada arsitektur ConvNeXt diperoleh oleh Model 6 dengan kombinasi *data split* 90:5:5 dan optimizer AdamW. Model tersebut menghasilkan nilai *precision*, *recall*, dan *F1-score* sebesar 97% serta *accuracy* sebesar 96,98%. Sementara itu, performa terbaik pada arsitektur MaxViT diperoleh oleh Model 8 dengan kombinasi *data split* 80:10:10 dan optimizer Adam. Model tersebut menghasilkan nilai *precision*, *recall*, dan *F1-score* sebesar 95% dengan *accuracy* sebesar 94,91%. Hasil evaluasi menunjukkan bahwa arsitektur ConvNeXt memiliki performa klasifikasi yang lebih baik dibandingkan MaxViT pada dataset penyakit daun jagung yang digunakan dalam penelitian ini.

3.2 Implementasi Sistem

Sistem identifikasi penyakit daun jagung dikembangkan dalam bentuk aplikasi berbasis web yang dapat diakses melalui perangkat desktop maupun mobile. Aplikasi menggunakan Vue JS sebagai *frontend* untuk antarmuka pengguna, sedangkan FastAPI digunakan sebagai *backend* untuk proses inferensi model *deep learning*.

Pada proses inferensi, pengguna dapat mengunggah citra daun jagung melalui halaman deteksi. Citra yang diterima kemudian dikirim ke *backend server* untuk dilakukan *preprocessing* dan prediksi menggunakan model terbaik hasil pelatihan. Setelah proses inferensi selesai, sistem mengembalikan hasil berupa label penyakit, *confidence score*, serta visualisasi *Grad-CAM* yang menunjukkan area daun yang menjadi perhatian utama model dalam melakukan klasifikasi. Tampilan halaman deteksi dan hasil prediksi dapat dilihat pada gambar 4.



(a) Halaman deteksi (b) Hasil prediksi
Gambar 4: Tampilan aplikasi web deteksi penyakit daun jagung.

4. Simpulan dan Saran

Hasil evaluasi menunjukkan bahwa model ConvNeXt-Tiny memiliki performa yang lebih baik dibandingkan MaxViT-Tiny pada dataset yang digunakan dalam penelitian ini. Performa terbaik diperoleh oleh model ConvNeXt-Tiny dengan optimizer AdamW dan skenario *data split* 90:5:5 yang menghasilkan nilai *accuracy* sebesar 96,98% serta nilai *precision*, *recall*, dan *F1-score* sebesar 97%. Sedangkan performa terbaik model MaxViT-Tiny hanya sebesar 94,91% dengan nilai *precision*, *recall*, dan *F1-score* sebesar 95% pada skenario *data split* 80:10:10 dan optimizer Adam.

Model terbaik diintegrasikan ke dalam aplikasi berbasis web menggunakan Vue JS dan FastAPI sehingga pengguna dapat melakukan identifikasi penyakit daun jagung melalui proses unggah citra. Sistem mampu menampilkan hasil prediksi berupa label penyakit, *confidence score*, dan visualisasi *Grad-CAM* untuk membantu interpretasi hasil klasifikasi model.

Pengembangan dari penelitian ini dapat menggunakan dataset yang lebih besar dengan kondisi pencahayaan yang lebih beragam, lalu menerapkan teknik *quantization* dan *pruning* untuk menghasilkan model yang lebih ringan dan efisien, sehingga dapat dijalankan langsung pada perangkat *smartphone* melalui aplikasi *mobile* tanpa bergantung pada *server*. Selain itu, performa MaxViT-Tiny yang lebih rendah dari ConvNeXt-Tiny menunjukkan bahwa arsitektur hybrid berbasis transformer masih perlu dikaji lebih lanjut. Penggunaan varian yang lebih besar seperti MaxViT-Small atau MaxViT-Base berpotensi meningkatkan performa.

Daftar Pustaka

Austin, Y. S., Irfano, H., Christopher, J. Y., Sukma, L. C., Putra, O. P., Ardhanto, R. I., & Yudistira, N. (2024). Klasifikasi Penyakit Alzheimer dari Scan MRI Otak Menggunakan ConvNeXt. *Jurnal Teknologi*

Informasi Dan Ilmu Komputer, 11(6).
<https://doi.org/10.25126/jtiik.2024118117>

Badan Pusat Statistik. (2025). *Luas Panen dan Produksi Jagung di Indonesia 2024*.
<https://www.bps.go.id/id/publication/2025/08/15/2bb49a0af17ec89ac16971de/luas-panen-dan-produksi-jagung-di-indonesia-2024.html>

Biswal, K., Roy, N. S., & Chatterjee, T. (2026, February 8). *Maize Diseases*. Kaggle.
<https://www.kaggle.com/datasets/kaustavbiswal/maize-diseases?select=v1.1>

Gopalan, K., Srinivasan, S., Pragya, P., Singh, M., Mathivanan, S. K., & Moorthy, U. (2025). Corn leaf disease diagnosis: enhancing accuracy with resnet152 and grad-cam for explainable AI. *BMC Plant Biology*, 25(1). <https://doi.org/10.1186/s12870-025-06386-0>

Hutabarat, B. T. K., Ramadhan, P. S., & Sari, K. (2025). Sistem Pakar Mendiagnosa Penyakit Tanaman Jagung Menggunakan Metode Dempster Shafer. *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD)*, 4(1), 151–161. <https://doi.org/10.53513/jursi.v4i1.7763>

Iglovikov, V. (2025, August 23). *Input Normalization: What We Know, What We Don't, and Why It Works Anyway*. Alumentations.
<https://alumentations.ai/blog/2025/03-the-mystery-of-input-normalization/>

Jafar, A., Bibi, N., Naqvi, R. A., Sadeghi-Niaraki, A., & Jeong, D. (2024). Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. In *Frontiers in Plant Science* (Vol. 15). <https://doi.org/10.3389/fpls.2024.1356260>

Leung, K. (2022, January 4). *Micro, Macro & Weighted Averages of F1 Score, Clearly Explained*. Towards Data Science.
<https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f/>

Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2022-June*, 11966–11976. <https://doi.org/10.1109/CVPR52688.2022.01167>

Maulini, R., Sahlinal, D., Meilantika, D., Rofianto, D., Pujiana, T., Produksi Tanaman Pangan, T., Tanaman Pangan, B., Negeri Lampung, P., & Lampung, B. (2025). Deteksi Dini Penyakit Tanaman Jagung Berbasis Transfer Learning dengan Arsitektur DenseNet121. *Jurnal Sistem Komputer Dan Informatika (JSON) Hal*, 7(2), 559–567.
<https://ejurnal.stmik-budidarma.ac.id/index.php/JSON/article/view/9094>

- Pfordt, A., & Paulus, S. (2024). A review on detection and differentiation of maize diseases and pests by imaging sensors. *Journal of Plant Diseases and Protection*, 132(1), 40. <https://doi.org/10.1007/s41348-024-01019-4>
- Tamayasa, K. A., & Dewi, L. J. E. (2026). Analisis Perbandingan Model Arsitektur MobileNetV2 dan EfficientNetB3 dalam Klasifikasi Penyakit Daun Jagung. *Jurnal Informatika Dan Teknik Elektro Terapan*, 14(1). <https://doi.org/10.23960/jitet.v14i1.8624>
- Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., & Li, Y. (2022). MaxViT: Multi-axis Vision Transformer. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13684 LNCS, 459–479. https://doi.org/10.1007/978-3-031-20053-3_27