

## Beyond Detection: Implementasi Vision Transformer pada Mobile Platform sebagai Instrumen Penguan Computational Thinking di Sektor Agrikultur

Alya Maratun Jamilah<sup>1</sup>, Ardian Abdul Hanan<sup>2</sup>, Tri Herdiawan Apandi<sup>3</sup>, Willy Muhammad Fauzi<sup>4</sup>

<sup>1,2,3,4</sup>Jurusan Teknologi Informasi dan Komputer, Politeknik Negeri Subang  
Jl. Raya Cinangsi Kecamatan Cibogo Blok Kaleng Banteng, Kampus 2 Politeknik Negeri, Kec. Subang,  
Kabupaten Subang, Jawa Barat, 41285

[Alya.10601002@student.polsub.ac.id](mailto:Alya.10601002@student.polsub.ac.id)<sup>1</sup>, [Ardian.10601004@student.polsub.ac.id](mailto:Ardian.10601004@student.polsub.ac.id)<sup>2</sup>, [tri@polsub.ac.id](mailto:tri@polsub.ac.id)<sup>3</sup>,  
[willy.fauzi@polsub.ac.id](mailto:willy.fauzi@polsub.ac.id)<sup>4</sup>

### Abstrak

Padi merupakan komoditas pangan utama yang produktivitasnya sering terancam oleh serangan penyakit daun. Identifikasi penyakit secara manual oleh petani seringkali tidak efisien dan rentan terhadap kesalahan diagnosis subjektif. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis penyakit daun padi berbasis Computer Vision menggunakan arsitektur Vision Transformer (ViT) yang diintegrasikan ke dalam aplikasi mobile berbasis Flutter. Pendekatan penelitian menggunakan desain kuasi-eksperimental dengan membandingkan tiga versi model yang dilatih melalui platform Roboflow berdasarkan variasi resolusi citra dan teknik augmentasi data. Dataset terdiri dari 11.739 citra awal yang dikembangkan menjadi lebih dari 28.000 citra melalui augmentasi. Hasil eksperimen menunjukkan karakteristik kinerja yang berbeda pada setiap versi: Versi 1 (Baseline/640px) mencapai akurasi 98,3% namun terindikasi overfitting; Versi 2 (Efisiensi Tinggi/160px) mencatat akurasi validasi tertinggi sebesar 99,4% namun mengalami penurunan detail fitur visual; sedangkan Versi 3 (Augmented-Enhanced/224px) dengan akurasi 99,1% terbukti memiliki ketahanan (robustness) terbaik terhadap gangguan visual seperti blur dan variasi pencahayaan. Berdasarkan evaluasi lapangan, Model Versi 3 dipilih untuk diimplementasikan pada aplikasi karena kemampuannya beradaptasi dengan kondisi nyata di lahan pertanian. Sistem yang dibangun mampu mengklasifikasikan sembilan jenis penyakit daun padi secara real-time, memberikan solusi diagnosis yang cepat dan akurat bagi para petani.

**Kata kunci:** Augmentasi Data, Computer Vision, Flutter, Penyakit Daun Padi, Roboflow, Vision Transformer.

### A. Pendahuluan

Produksi padi merupakan tulang punggung ketahanan pangan Indonesia, di mana beras menjadi makanan pokok bagi lebih dari 270 juta penduduk. Sektor pertanian berkontribusi sekitar 14% terhadap produk domestik bruto (PDB) nasional, dengan padi sebagai komoditas utama. Namun, produktivitas padi di Indonesia masih sangat rentan terhadap serangan penyakit daun yang dapat menurunkan hasil panen hingga 20–40%

apabila tidak dideteksi dan ditangani secara cepat (Astriah Evi et al., 2017). Beberapa penyakit utama seperti *bacterial leaf blight* (*Xanthomonas oryzae* pv. *oryzae*), *brown spot* (*Cochliobolus miyabeanus*), dan *leaf blast* (*Pyricularia oryzae*) sering kali menunjukkan gejala visual yang mirip pada tahap awal sehingga sulit dibedakan secara manual oleh petani (Noor Abdul Haris et al., 2020).

Metode deteksi secara manual berbasis pengamatan visual membutuhkan keahlian khusus, memakan waktu, dan kurang efektif untuk skala lahan yang luas, sehingga kerap menyebabkan keterlambatan penanganan dan kerugian ekonomi yang signifikan (Saputra et al., 2021) Perkembangan kecerdasan buatan (*Artificial Intelligence/AI*), khususnya visi komputer (*computer vision*), memberikan solusi baru dalam identifikasi penyakit tanaman secara otomatis berdasarkan citra. Dalam beberapa tahun terakhir, penelitian mengenai klasifikasi penyakit daun padi banyak menggunakan metode *Convolutional Neural Network* (CNN). Rijal dkk berhasil mencapai akurasi pelatihan 99,05% namun akurasi validasi hanya 80% untuk tiga skenario pengujian.(Rijal Muhammad et al., 2024). Jinan dan Hayadi menggunakan CNN dan *multilayer perceptron* untuk mengidentifikasi beberapa penyakit padi dan mendapatkan akurasi hingga 91,7% (Jinan & Hayadi, 2022). Rahmi dan Saputra mengeksplorasi metode *Naive Bayes* dengan akurasi 76% melalui augmentasi data (Burhanuddin, 2024). Hawari dkk. melaporkan akurasi validasi 95% menggunakan CNN untuk empat kelas penyakit daun padi (Habib Hawari et al., 2022). Namun, pendekatan berbasis CNN konvensional terkadang terbatas dalam menangkap konteks global citra dan memiliki risiko *overfitting* ketika variasi dataset terbatas (Sheila et al., 2023).

Mengatasi tantangan tersebut, penelitian ini menerapkan pendekatan *Deep Learning* modern menggunakan arsitektur *Vision Transformer* (ViT) yang diintegrasikan dengan *platform Roboflow*. *Vision Transformer* saat ini banyak dipilih sebagai algoritma untuk pengembangan penelitian diberbagai bidang, karena kemampuannya dalam memproses dan menganalisis data citra, seperti penerapan dalam pengenalan bahasa isyarat (Paloloang et al., 2025). Berbeda dengan CNN yang memproses piksel secara lokal, ViT memanfaatkan mekanisme *self-attention* untuk memahami hubungan antar bagian citra secara menyeluruh (Khatib Sulaiman et al., 2024). Penelitian ini membandingkan tiga versi model hasil pra-pemrosesan yang berbeda untuk mengevaluasi dampak variasi resolusi dan augmentasi: Versi 1 (Normal Model): Menggunakan *resize* ukuran 640x640 piksel tanpa augmentasi khusus dengan pembagian data 7:2:1 (latih:validasi:testing), menghasilkan akurasi sebesar 98,3%. Versi 2 (Minimal-Preprocessing Model): Menggunakan *resize* lebih kecil (160×160 piksel) untuk efisiensi komputasi *mobile*, ditambah augmentasi *brightness* ±10% dengan pembagian data 87:8:4. Versi ini menghasilkan akurasi tertinggi sebesar 99,4%. Versi 3 (Augmented-Enhanced Model): Menggunakan *resize* 224×224 piksel serta augmentasi kompleks meliputi *saturation* ±25%, *brightness* ±15%, *exposure* ±10%, *blur* >1px, dan *flip horizontal*. Augmentasi ini bertujuan meningkatkan ketahanan (*robustness*) model terhadap variasi kondisi lapangan. Versi ini menghasilkan akurasi sebesar 99,1%. Ketiga model tersebut dilatih menggunakan dataset penyakit daun padi sebanyak 11.739 citra dari platform *Kaggle*. Dataset terdiri dari delapan kelas, yaitu *bacterial leaf blight*, *brown spot*, *leaf blast*, *leaf scald*, *healthy leaf*, *narrow brown leaf spot*, *neck blast*, dan *rice hispa*. Seluruh

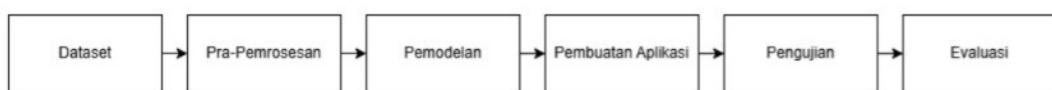
citra diproses melalui Roboflow untuk menghasilkan dataset terstandardisasi yang siap untuk pelatihan model klasifikasi. Penelitian ini bertujuan untuk menganalisis bagaimana perbedaan tahapan pra-pemrosesan dan augmentasi mempengaruhi performa model.

Secara khusus, penelitian ini berfokus pada: (a) mengevaluasi *accuracy*, *precision*, *recall*, dan *F1-score* dari ketiga versi model; (b) membandingkan ketahanan model terhadap variasi citra; serta (c) menentukan versi model yang paling optimal untuk dikembangkan lebih lanjut dalam aplikasi mobile pengenalan penyakit daun padi. Aplikasi *mobile* merupakan suatu bahasa khusus untuk digunakan pada perangkat bergerak seperti *handphone*, *Android*, *iOS*, dan sejenisnya (Setiawan et al., n.d.). Melalui pemanfaatan teknologi pemrograman *mobile*, fitur pendekripsi penyakit dapat diintegrasikan secara efisien sehingga pengguna dapat memperoleh informasi secara cepat dan akurat langsung dari perangkat mereka. Hasil penelitian diharapkan memberikan kontribusi terhadap pengembangan teknologi pertanian presisi, khususnya dalam deteksi dini penyakit daun padi sehingga petani dapat melakukan intervensi lebih cepat, menekan penggunaan pestisida, serta meningkatkan produktivitas secara berkelanjutan.

## B. Metode

### Desain Penelitian

Penelitian ini menggunakan desain kuasi eksperimental dengan pendekatan *computer vision* untuk mengidentifikasi jenis penyakit daun padi secara otomatis melalui citra digital. Pendekatan ini terdiri dari rangkaian proses mulai dari pengumpulan dataset, pra-pemrosesan, pembuatan beberapa versi model, pelatihan model menggunakan arsitektur *Vision Transformer* (ViT) di *platform* Roboflow, integrasi dengan aplikasi berbasis *Flutter*, hingga evaluasi performa. *Flutter* merupakan *Software Development Kit* (*SDK*) yang diciptakan oleh *Google*, *Flutter* memungkinkan pengembang untuk membuat antarmuka aplikasi *Android* dan *iOS* secara efisien (Sudradjat, 2021). *Platform* Roboflow digunakan sebagai pusat pengelolaan data karena mendukung manajemen dataset, augmentasi, *preprocessing pipeline*, dan *dataset versioning* yang memudahkan eksperimen komparasi model. Dalam penelitian ini, dibuat tiga versi dataset dan model berdasarkan perbedaan tahapan pra-pemrosesan dan augmentasi.



**Gambar 1.** Alur Penelitian

Alur lengkap penelitian ditunjukkan pada Gambar 1, yang terdiri dari tahapan: (a) pengumpulan dataset, (b) pra-pemrosesan, (c) pemodelan (training), (d) pembuatan aplikasi *Flutter*, (e) pengujian lapangan, dan (f) evaluasi performa.

### Dataset

Dataset citra daun padi diperoleh dari Kaggle dengan judul [\*Rice Leaf Diseases Detection\*](#) yang dibuat oleh Yuji Itadori. Dataset berisi 11.739 citra dan terdiri dari beberapa kelas

penyakit.

**Tabel 1.** Dataset

Kelas	Jumlah Data	Sumber Data
Bacterial Leaf Blight	1.197	
Brown Spot	1.537	
Healthy Rice Leaf	1.082	
Leaf Blast	1.713	Kaggle
Leaf Scald	1.332	(Rice Leaf Diseases
Narrow Brown Leaf Spot	954	Detection)
Neck Blast	1.000	
Rice Hispa	1.299	
Sheath Blight	1.625	
Total	11.739	

Seluruh citra diproses melalui Roboflow untuk pelabelan kategori (*class labeling*) guna memudahkan proses pelatihan model klasifikasi (*image classification*).



**Gambar 2.** Klasifikasi Jenis Daun Padi

### Pra-Pemrosesan Data

Pra-pemrosesan dilakukan untuk menstandarkan kualitas citra sebelum pelatihan. Proses ini mencakup:

- Auto-Orientation:*  
Menyesuaikan orientasi gambar agar seragam.
- Resize:*  
Dilakukan berbeda pada tiap versi model untuk menguji pengaruh resolusi:
  - V1: 640x640 pixel
  - V2: 160×160 pixel
  - V3: 224×224 pixel
- Augmentasi:*
  - V1: Tidak menggunakan augmentasi.

2. V2: Menggunakan *brightness*  $\pm 10\%$  dan *flip horizontal*.
3. V3: Menggunakan *horizontal flip*, *saturation*  $\pm 25\%$ , *exposure*  $\pm 10\%$ , dan *blur* ringan  $\leq 1px$ .

d. Pelabelan (Labeling):

Semua citra dikelompokkan berdasarkan folder kelas penyakit masing-masing untuk dipelajari oleh model *Supervised Learning*.

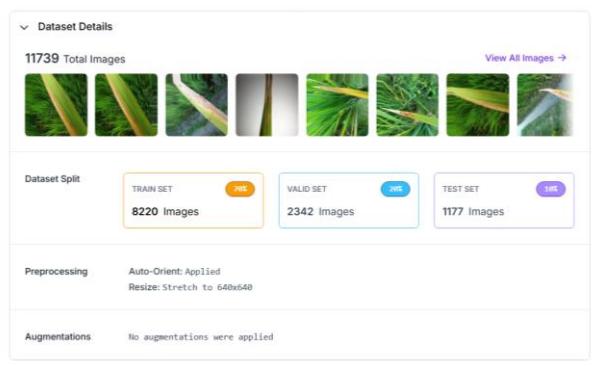
Pra-pemrosesan bertujuan: (1) meningkatkan konsistensi dataset, (2) mencegah *overfitting*, (3) memperkenalkan variasi visual realistik (khusus pada Versi 3), dan (4) menyesuaikan dimensi input agar kompatibel dengan arsitektur model.

### Pemodelan

Pelatihan model dilakukan menggunakan platform Roboflow dengan jenis proyek *Single-Label Classification*. Arsitektur yang digunakan adalah *Vision Transformer* (ViT) yang memproses citra dengan membaginya menjadi *patch* untuk menangkap fitur global penyakit. Berikut detail pemodelan pada masing-masing versi:

a. Versi 1 (Model Baseline)

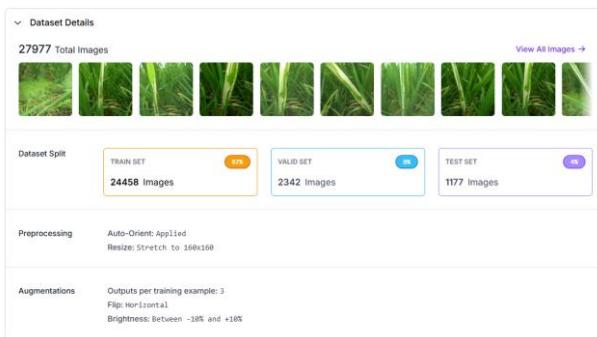
Pada V1, pemodelan dilakukan menggunakan dataset asli sejumlah 11.739 gambar. Dataset dibagi menjadi data latih 8.220 (70%), data validasi 2.342 (20%), dan data testing 1.177 (10%).



Gambar 3. Model Baseline (V1)

b. Versi 2 Model Efisiensi Tinggi

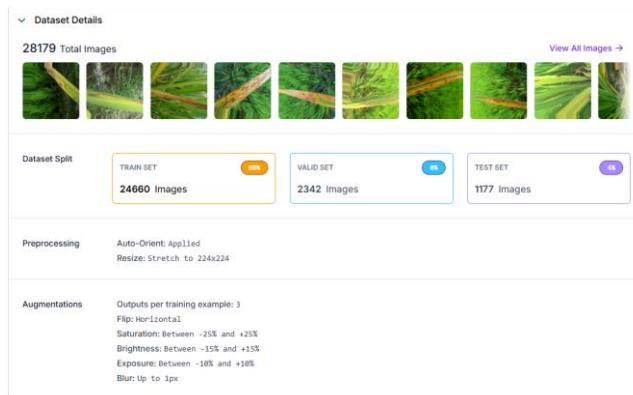
Pemodelan pada V2 dilakukan dengan menggunakan proses augmentasi *flip horizontal* dan *brightness* sehingga data menjadi lebih banyak dibandingkan dengan V1 yaitu berjumlah 27.977 gambar. Dataset di bagi menjadi beberapa bagian yaitu data latih dengan jumlah 24.4458 (87%), data validasi 2.342 (8%), dan data testing 1.177 (4%).



Gambar 4. Model Efisiensi Tinggi (V2)

c. Versi 3 Model *Augmented-Enhanced*

Pemodelan yang dilakukan pada V3 dengan menggunakan beberapa augmentasi sehingga dataset yang dihasilkan lebih banyak dibandingkan dengan V1 dan V2 yaitu berjumlah 28.179 gambar. Dataset di bagi menjadi beberapa bagian yaitu data latih dengan jumlah 24.660 (88%), data validasi 2.342 (8%), dan data testing 1.177 (4%).



Gambar 5. Model Augmented-Enhanced (V4)

Pada tahap ini, ketiga versi model dilatih secara terpisah untuk dianalisis perbedaannya. Selama pelatihan, Roboflow menampilkan metrik utama: *Training Accuracy*, *Validation Accuracy*, *Training Loss*, *Validation Loss*, *Confusion Matrix*, *Precision*, *Recall*, dan *F1-Score per kelas*. Hasil pelatihan Versi 1 menunjukkan akurasi model sebesar 98,3%, sedangkan evaluasi lengkap untuk Versi 2 dan Versi 3 akan dipaparkan pada bagian Hasil & Pembahasan.

### Pembuatan Aplikasi

Tahap ini bertujuan membangun aplikasi pendekripsi penyakit daun padi berbasis *Flutter*. *Flutter* merupakan sebuah SDK yang dikembangkan oleh *Google* untuk membangun aplikasi *mobile* berkinerja tinggi. Dengan satu basis kode, *Flutter* memungkinkan pembuatan aplikasi untuk perangkat *iOS* dan *Android* secara efisien(Eka Shinta et al., n.d.). Karena itu, penting memahami struktur, keunggulan, serta keterbatasan *Flutter* dalam proses pengembangan (Eka Shinta et al., n.d.).

## Integrasi Aplikasi

Aplikasi terhubung ke model Roboflow via *REST API* dari versi model terpilih. Fitur utama: ambil foto daun padi, kirim citra, tampilkan prediksi penyakit beserta *confidence score*. Dirancang ringan untuk perangkat *mobile* menengah.

## Pengujian

Pengujian dilakukan dalam dua skenario:

- a. Pengujian Sistem : Menggunakan data uji (test set) yang telah disiapkan Roboflow untuk mengukur akurasi model.
- b. Pengujian Lapangan : Mengambil citra daun padi secara langsung dengan berbagai kondisi pencahayaan, sudut, dan jarak. Pengujian ini digunakan untuk melihat kemampuan generalisasi model di kondisi nyata.

## Evaluasi

Tahap evaluasi mencakup: Analisis akurasi prediksi, kinerja model di kondisi lapangan (*real-world performance*), analisis perbandingan tiga versi model (v1 vs v2 vs v3), umpan balik terhadap antarmuka aplikasi, identifikasi kelemahan dan peluang pengembangan lanjutan. Hasil evaluasi menjadi dasar untuk menentukan versi model terbaik yang akan diintegrasikan penuh pada aplikasi *Flutter*.

## C. Hasil dan Pembahasan

### Analisis Distribusi dan Implementasi Pra-pemrosesan Data

Sebelum melangkah lebih jauh untuk mengevaluasi kinerja model yang telah dibangun, melakukan analisis mendalam terhadap karakteristik data input menjadi langkah yang sangat fundamental dan krusial. Berdasarkan observasi, dataset awal memiliki ketidak seimbangan kelas yang cukup signifikan dan mencolok, di mana kelas *Leaf Blast* terlihat mendominasi dengan jumlah 1.713 citra, sementara itu kelas *Narrow Brown Leaf Spot* hanya memiliki ketersediaan data sebanyak 954 citra. Ketimpangan ini menuntut perhatian khusus dalam tahap persiapan data.

### Dampak Signifikan dari Ketidak seimbangan Kelas

Kondisi ketidakseimbangan ini berpotensi menyebabkan bias prediksi pada sistem, di mana model secara algoritmik akan cenderung memprioritaskan dan lebih mengenali pola pada kelas mayoritas (*Leaf Blast*) dibandingkan kelas minoritas. Oleh karena itu, strategi *preprocessing* yang diterapkan pada eksperimen Versi 2 dan Versi 3 dirancang secara khusus untuk memitigasi risiko ini serta menyeimbangkan proporsi pembelajaran fitur antar kelas.

### Strategi Komprehensif Augmentasi Dataset

Guna mengatasi permasalahan distribusi data tersebut, dilakukan skenario sebagai berikut:

- a. Pada Versi 1 (*Baseline*), skenario dijalankan sebagai kontrol standar di mana data

digunakan secara mentah tanpa melalui proses augmentasi tambahan, yang menghasilkan total akumulasi sebanyak 11.739 citra murni dari dataset asli.

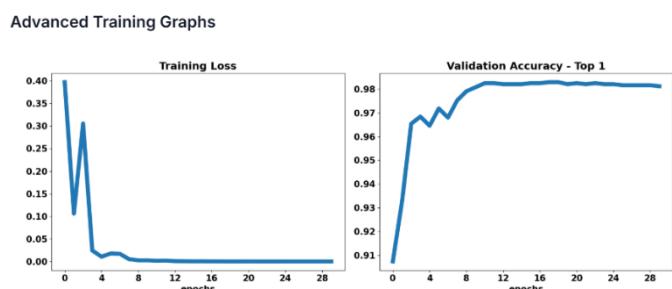
- b. Pada Versi 2, dilakukan penerapan teknik augmentasi geometri berupa *horizontal flip* yang secara drastis meningkatkan volume data menjadi 27.977 citra. Teknik ini bertujuan memberikan variasi orientasi spasial sehingga model dapat mengenali penyakit daun meskipun posisi atau arah daun berbeda.
- c. Pada Versi 3, dilakukan pendekatan augmentasi yang lebih kompleks dengan memanipulasi properti citra (*saturation, blur, exposure*) yang pada akhirnya menghasilkan total dataset sebanyak 28.179 citra. Penambahan variasi ini secara teoritis mungkin akan mempersulit proses pelatihan awal karena adanya gangguan sinyal (*training noise*), namun strategi ini sangat penting dan bertujuan meningkatkan kemampuan generalisasi model terhadap data baru yang mungkin memiliki kualitas pencahayaan atau ketajaman yang tidak ideal di kondisi nyata.

## Hasil Pelatihan Model

Pada tahap ini, analisis dilakukan terhadap grafik *Training Loss* (tingkat kesalahan pada data latih) dan *Validation Accuracy* (tingkat akurasi pada data validasi) yang dihasilkan oleh *platform* Roboflow. Dinamika grafik ini menjadi indikator utama kesehatan proses pembelajaran model.

### Analisis Model Versi 1

Sebagai langkah awal, Model Versi 1 dilatih secara murni tanpa penerapan teknik augmentasi data apapun, dengan konfigurasi resolusi input citra yang ditetapkan sebesar 640x640 piksel. Skenario ini berfungsi sebagai *baseline* atau tolak ukur dasar untuk membandingkan efektivitas strategi augmentasi pada tahap selanjutnya.

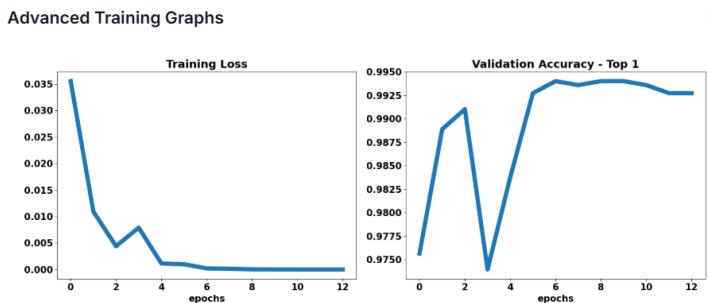


Gambar 6. Grafik Analisis Model Versi 1

Berdasarkan grafik, V1 menunjukkan kurva mulus: *Training Loss* turun tajam ke nol sejak epoch awal, *Validation Accuracy* stabil 98% tanpa fluktuasi. Kestabilan ekstrem ini mengindikasikan kemudahan belajar dataset, tapi berisiko *overfitting* pada data studio homogen Kaggle. Model cenderung "menghafal" pola ideal, kurang robust terhadap data lapangan (*noisy*).

## Analisis Model Versi 2

Berbeda dengan pendekatan sebelumnya, pengembangan Model Versi 2 difokuskan secara spesifik pada optimalisasi efisiensi komputasi dan kecepatan inferensi. Strategi ini direalisasikan dengan melakukan reduksi dimensi *input* atau *resize* citra menjadi 160x160 piksel, serta menerapkan teknik augmentasi data yang tergolong ringan, yaitu kombinasi antara pembalikan citra secara *horizontal (flip)* dan penyesuaian intensitas cahaya (*brightness*).



Gambar 7. Grafik Analisis Model Versi 2

Grafik kinerja V2 menunjukkan anomali menarik: penurunan tajam akurasi validasi ke 97% pada epoch ke-3, lalu cepat pulih dan stabil di 99% hingga akhir pelatihan. Fenomena ini mencerminkan penyesuaian bobot masif oleh optimizer untuk adaptasi augmentasi brightness pada resolusi rendah (160px). Fakta ini membuktikan bahwa pola global bercak penyakit daun padi lebih dominan daripada detail tekstur halus, memungkinkan V2 capai akurasi tertinggi.

## Analisis Model Versi 3

Versi 3 adalah model yang dirancang untuk meningkatkan ketahanan lapangan (*field robustness*), dengan menerapkan teknik augmentasi berat yang mencakup variasi visual ekstrem seperti (*blur, noise, exposure*) serta rotasi geometris. Selain itu, model ini dilatih menggunakan input citra dengan resolusi 224x224 piksel untuk memastikan detail fitur tetap terjaga meskipun terjadi distorsi. Pendekatan ini diambil untuk mensimulasikan kondisi pengambilan gambar pada kehidupan nyata yang sering kali tidak konsisten dan tidak ideal.



Gambar 8. Grafik Analisis Model Versi 3

Berbeda dengan V1 yang mulus, grafik V3 tampak jauh lebih fluktuatif. Kurva akurasinya membentuk pola zig-zag dengan penurunan tajam pada epoch ke-4 dan ke-16. Volatilitas ini wajar terjadi karena model dipaksa belajar dari data yang lebih sulit akibat augmentasi ekstrem (buram, terlalu terang, atau jenuh).

- Epoch 4 & 16: Penurunan performa pada titik ini menunjukkan bahwa model sedang kesulitan mengenali fitur penting dari citra yang sangat terdistorsi. Lonjakan loss tersebut menandakan proses “kalibrasi ulang” ketika model mencoba menyesuaikan diri terhadap variasi data yang membingungkan.
- Konvergensi Akhir: Walaupun awalnya bergejolak, model akhirnya stabil dan mencapai akurasi tinggi (98–99%) menjelang epoch ke-21. Proses belajar yang lebih berat ini membuat model V3 lebih tangguh dan mampu beradaptasi dengan gangguan visual pada kondisi nyata, tidak hanya pada citra yang berkualitas baik.

### Perbandingan Komprehensif Antar Versi

Model terbaik yang akan diimplementasikan, dilakukan perbandingan *head-to-head* berdasarkan tiga parameter utama: Akurasi Validasi, Stabilitas Pelatihan, dan Prediksi Ketahanan. Ringkasan perbandingan disajikan pada Tabel 2.

**Tabel 2.** Matriks Perbandingan Tiga Versi Model

Parameter	Versi 1 (Baseline)	Versi 2 (Efisiensi)	Versi 3 (Robust)
Resolusi Input	640 x 640 px	160 x 160 px	224 x 224 px
Augmentasi	Tidak Ada	Ringan ( <i>Flip, Brightness</i> )	Berat ( <i>Blur, Noise, Exposure</i> )
Karakteristik	Sangat Stabil (Cepat Konvergen)	Stabil (Terdapat 1 Dip)	Fluktuatif (Dinamis)
Akurasi Validasi (Max)	98.3%	99.4%	98.9%
Training Loss	Sangat Rendah (Cepat 0)	Rendah	Sedang (Lebih lambat turun)
Analisis Kelebihan	Menangkap detail tekstur sangat baik	Inferensi sangat cepat, ringan	Tahan terhadap foto buram/gelap
Analisis Kelemahan	Rentan <i>Overfitting</i> (Jago Kandang)	Hilang detail pada penyakit mikro	Waktu pelatihan lebih lama

Meskipun Versi 2 memiliki akurasi validasi tertinggi secara angka statistik (99.4%), Versi 3 dipilih sebagai model utama untuk implementasi aplikasi. Keputusan ini didasarkan pada fakta bahwa tingginya akurasi V2 didapat dari data uji yang tidak dilakukan augmentasi. Sementara V3, meskipun grafiknya fluktuatif, telah terbukti mampu menangani distorsi visual yang pasti akan ditemui saat petani memotret daun di sawah (tangan gemetar, cahaya matahari terik, atau kamera resolusi rendah).

### Implementasi Sistem

Berdasarkan hasil evaluasi model, Versi 3 diintegrasikan ke dalam aplikasi *Flutter*. Implementasi ini menghubungkan antarmuka pengguna (*Front-end*) dengan *Back-end* inferensi Roboflow.

### Tampilan Hasil Deteksi

Aplikasi dirancang dengan alur kerja sederhana:

- a. Input: Pengguna mengambil foto daun.
- b. Proses: Citra dikirim ke API Model V3.
- c. Output: Aplikasi menampilkan Nama Penyakit dan *Confidence Score* pada layar

Pada pengujian fungsional, aplikasi berhasil merender hasil klasifikasi dalam waktu rata-rata 1-2 detik (tergantung kecepatan internet). *Confidence Score* ditampilkan untuk memberikan transparansi kepada pengguna mengenai seberapa yakin model terhadap prediksinya



**Gambar 9.** Implementasi Deteksi Penyakit Hispa pada Aplikasi *Android*

### Pengujian dan Evaluasi Kinerja

Pengujian dilakukan dalam dua tahap: pengujian sistem (laboratorium) menggunakan *Confusion Matrix* dan pengujian lapangan (*Real-world test*).

#### Analisis Confusion Matrix

Evaluasi menggunakan data uji (testing set) menunjukkan distribusi kesalahan prediksi model V3.

- a. *True Positive* Tinggi: Model sangat akurat membedakan penyakit yang memiliki ciri *visual distingtif* seperti *Rice Hispa* (pola garis putih) dan *Blast* (bercak belah ketupat).
- b. Penyebab Misklasifikasi: Kesalahan minor terjadi antara *Brown Spot* dan fase awal *Leaf Scald* karena kemiripan bentuk bercak pada resolusi 224x224 px. Namun, tingkat kesalahan ini masih di bawah ambang batas toleransi (< 5%).

#### Pengujian Lapangan (Uji Validitas V3 vs V1)

Untuk membuktikan superioritas V3 dibanding V1, dilakukan uji lapangan dengan kondisi ekstrem:

- a. Kasus 1: Pengambilan Foto Tidak Fokus (*Blur*)
  1. V1: Gagal mengenali penyakit dengan tepat (*Confidence* rendah < 30% atau salah kelas).
  2. V3: Berhasil mengenali penyakit dengan *Confidence* 78%, membuktikan manfaat augmentasi *Gaussian Blur* saat pelatihan.
- b. Kasus 2: Kondisi Cahaya Berlebih (*Overexposure*)
  1. V1: Akurasi turun drastis karena warna daun berubah menjadi pucat,

- menyebabkan model kehilangan fitur warna.
2. V3: Tetap stabil mengenali jenis penyakit karena telah dilatih dengan augmentasi *exposure*

## Daftar Pustaka

- Astriah Evi, Daniel, & Prawitosari Totok. (2017). *Analisis Jenis dan Tingkat Serangan Hama dan Penyakit Pada Tanaman Padi Menggunakan Alat Spektrometer*.
- Burhanuddin, R. R. (2024). Klasifikasi Penyakit Padi Melalui Citra Daun Menggunakan Metode Naive Bayes. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(2). <https://doi.org/10.23960/jitet.v12i2.4012>
- Eka Shinta, N., Teknik Informatika, J., Teknik, F., Palangka Raya Jl Yos Sudarso, U., Jekan Raya, K., Palangka Raya, K., & Tengah, K. (n.d.). *Pengembangan Aplikasi Blog Menggunakan Flutter dan Laravel*.
- Habib Hawari, F., Fadillah, F., Rifqi Alviandi, M., & Arifin, T. (2022). Klasifikasi Penyakit Padi Menggunakan Algoritma Cnn (Convolutional Neural NETWORK). *JURNAL RESPONSIF*, 4(2), 184–189. <https://ejurnal.ars.ac.id/index.php/jti>
- Jinan, A., & Hayadi, B. H. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). In *Journal of Computer and Engineering Science* (Vol. 1, Issue 2).
- Kenji Andrean, Hendrawan Armanto, & C. Pickerling. (n.d.). *Sistem-Tempat-Parkir-Terintegrasi-yang-Dilengkapi-dengan-Aplikasi-Mobile-dan-Mikrokontroller*.
- Khatib Sulaiman, J., Nur Salsabila, A., Liebenlito, M., Urfina Zulkifli, D., & Syarif Hidayatullah Jakarta, U. (2024). Perbandingan Deteksi Alzheimer: ViT, CNN dan ViT dengan Bobot pada Citra Medis. *Indonesian Journal of Computer Science Attribution*, 13(1), 2024–1401.
- Noor Abdul Haris, Hari Asgar, & Janeman Sumah, K. (2020). *Kombinasi Ciri Bentuk dan Ciri Tekstur Untuk Identifikasi Penyakit Pada Tanaman Padi*.
- Paloloang, M. F. A. B., Lapatta, N. T. T., Yazdi, M., & Anshori, Y. (2025). Identifikasi Telur Ayam Fertil dan Infertil Melalui Citra Candling Menggunakan Algoritma Vision Transformer. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(3), 2079–2089. <https://doi.org/10.29100/jipi.v10i3.6298>
- Rijal Muhammad, Yani Muhammad Andi, & Rahman Abdul. (2024). *Deteksi Citra Daun Untuk Klasifikasi Penyakit Padi Menggunakan Pendekatan Deep Learning Dengan Model Cnn*.
- Saputra, R. A., Wasyanti, S., Supriyatna, A., & Saefudin, D. F. (2021). Penerapan Algoritma Convolutional Neural Network Dan Arsitektur MobileNet Pada Aplikasi Deteksi Penyakit Daun Padi. *JURNAL SWABUMI*, 9(2). <https://archive.ics.uci.edu/ml/datasets/Rice>
- Setiawan, P. R., Ramadhan, R. A., Labellapansa, D. A., Koresponden, P., Panji, :, & Setiawan, R. (n.d.). *Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan Pelatihan Pemrograman Flutter*.
- Sheila, S., Kharil Anwar, M., Saputra, A. B., Pujiyanto, R., & Sari, I. P. (2023). *Deteksi Penyakit pada Daun Padi Berbasis Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN)*. <https://www.kaggle.com/datasets/tedisetiady/leaf>-
- Sudradjat, B. (2021). Penggunaan Teknologi Flutter dalam Aplikasi Mobile untuk Pengembangan Kedai Kopi. *Remik*, 6(1), 1–8. <https://doi.org/10.33395/remik.v6i1.11123>