Implementasi *Feature Engineering* pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan *XGBoost*

SKRIPSI

OKTA GABRIEL SINSAKU SINAGA

2207411017

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
2026

Implementasi *Feature Engineering* pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan *XGBoost*

SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

OKTA GABRIEL SINSAKU SINAGA

2207411017

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
2026

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan dibawah ini:

Nama	: Okta Gabriel Sinsaku Sinaga
NIM	: 2207411017
Jurusan/Program Studi	: Teknik Informatika dan Komputer / D4 Teknik Informatika
Judul Skripsi	: Implementasi Feature Engineering pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan XGBoost

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesui dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanski atas perbuatan tersebut.

Depok, 28 Juni 2026

Yang membuat pernyataan

Okta Gabriel Sinsaku Sinaga

NIM 2207411017

Dipindai dengan CamScanner

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama : Okta Gabriel Sinsaku Sinaga

NIM : 2207411017

Program Studi : D4 Teknik Informatika

Judul Skripsi : Implementasi Feature Engineering pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan XGBoost

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, Tanggal Tujuh Belas, Bulan Juni, Tahun dan dinyatakan **LULUS.**

Disahkan oleh

Pembimbing I Euis Oktavianti, S.S.I., M.T.I. ( )

Penguji I : Mera Kartika Delimayanti, S.Si., M.T., Ph.D. ( )

Penguji II : Anggi Mardiyono, S.Kom., M.Kom. ( )

Penguji III : Dwi Ermawati, S.T., M.T. ( )

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua

Dr. Anita Hidayati, S.Kom., M.Kom.

NIP. 197908032003122003

Dipindai dengan CamScanner

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan dibawah ini :

Nama : Okta Gabriel Sinsaku Sinaga

NIM : 2207411017

Jurusan/ProgramStudi : T. Informatika dan Komputer/Teknik Informatika

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul :

Implementasi Feature Engineering pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan XGBoost

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 28 Juni 2026

Yang menyatakan

Okta Gabriel Sinsaku Sinaga

NIM 2207411017

v

Dipindai dengan CamScanner

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Masa Esa, karena atas rahmat, hidayah, dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul "*Implementasi Feature Engineering pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan XGBoost*" dengan baik dan tepat pada waktunya. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains Terapan S. Tr. Kom. pada Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.

1. Penulis menyadari bahwa selesainya skripsi ini tidak terlepas dari bantuan, bimbingan, petunjuk, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:
2. Dr. Syamsurizal, S.E., M.M, selaku Direktur Politeknik Negeri Jakarta.
3. Dr. Anita Hidayati, S.Kom., M. Kom, selaku Ketua Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.
4. Euis Oktavianti, S.S.I., M.T.I., selaku Ketua Program Studi Teknik Informatika dan selaku Dosen Pembimbing yang telah memberikan arahan selama masa perkuliahan dengan penuh kesabaran pula telah meluangkan waktu untuk memberikan bimbingan, arahan dan ilmu yang sangat berharga kepad apenulis selama proses penyusunan skripsi.
5. Seluruh Dosen dan Staf Administrasi Program Studi Teknik Informatika yang telah membekali penulis dengan ilmu pengetahuan dan membantu kelancaran administrasi selama masa perkuliahan secara khusus kepada Mbak Rini.
6. Teristimewa kepada kedua orang tua tercinta Bapak Ojak Soritua Sinaga dan Mamak Ermince Lumbantobing, adik-adik saya Tigor Vincent Oberlin Sinaga dan Dennys Immanuel Sinaga, serta seluruh keluarga besar Pomparan Op. Bertha, terspesial untuk Gabriella Althea Buntaro, S. Si kekasih saya, terima kasih atas segala perhatian, kesabaran, pendengaran yang baik, dan semangat yang selalu diberikan di kala penulis menghadapi masa-masa sulit penyusunan skripsi ini.
7. Sahabat-sahabat terbaik penulis dari *Persekutuan Oikumene Sivitas Akademika (POSA)*, serta seluruh teman TI-A 2022-2026 atas kebersamaan, semangat, dan bantuan yang diberikan selama ini.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Implementasi *Feature Engineering* pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan *XGBoost*

ABSTRAK

Tanaman cabai sebagai komoditas hortikultura utama di Indonesia rentan penyakit daun seperti geminivirus (Geminivirus), cercospora leaf spot, leaf curl virus, dan kondisi sehat, yang sulit dideteksi manual karena gejala visual mirip dan variasi lapangan. Penelitian ini bertujuan mengimplementasikan feature engineering melalui ekstraksi fitur HSV (mean/std/skewness), GLCM (contrast/homogeneity/energy 4 orientasi), edge (Canny density), diikuti reduksi dimensi PCA (n_components tetap dan explained_variance kumulatif 90-95%), untuk klasifikasi menggunakan XGBoost pada sistem deteksi mobile petani. Metode mengadopsi kerangka hibrida Waterfall untuk SDLC dan CRISP-DM untuk ML, dengan dataset publik Kaggle/Roboflow (pra-pemrosesan OpenCV: resize 256x256, CLAHE, augmentasi), train-test split 80:20, evaluasi 5-fold CV via akurasi, presisi, recall, F1-score, serta feature importance. Diharapkan menghasilkan model efisien komputasi (inferensi <2 detik), akurasi >90%, standar optimal PCA untuk citra tanaman tropis, dan prototipe aplikasi Flutter multiplatform guna deteksi dini, percepatan pengobatan, serta pengurangan kerugian ekonomi petani cabai.

Kata kunci: xgboost, feature engineering, penyakit tanaman

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME ..... **iii**
LEMBAR PENGESAHAN ..... **iv**
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS ..... **v**
ABSTRAK ..... **vii**
DAFTAR GAMBAR ..... **xi**
BAB 1 ..... **1**
PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 3
1.4.1 Tujuan Penelitian 3
1.4.2 Manfaat Penelitian 3
1.5 Sistematika Penulisan 4
BAB II **5**
TINJAUAN PUSTAKA **5**
2.1 Tanaman Cabai dan Penyakit Utama 5
2.1.1 Tanaman Cabai 5
2.1.2 Penyakit Utama 6
2.2 Ekstraksi Fitur dan Feature Engineering dengan OpenCV 8
2.2.1 Ekstraksi Fitur 8
2.2.2 Mekanisme Komputasi Ruang Warna HSV 9
2.2.3 Mekanisme Komputasi Fitur Tekstur GLCM 10
2.2.4 Mekanisme Komputasi Tepi *Canny Edge* 12
2.2.5 Integrasi Pipeline Input-Proses-Output (IPO) Ekstraksi Fitur Menuju PCA 13
2.2.6 Principal Component Analysis (PCA) 14

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.2.7 Letak dan Peran PCA dalam Pipeline Computer Vision 15
2.2.8 Prinsip Geometris dan Ortogonalitas PCA 16
2.2.9 Tahapan Matematis Transformasi PCA 17
2.2.10 Analisis Alur Komputasi Input-Proses-Output 19
2.2.11 Open CV 20
2.3 Algoritma XGBoost 20
2.4 Algoritma Random Forest 21
2.5 Software Development Life Cycle dan CRISP-DM 22
2.5.1 Waterfall 22
2.5.2 CRISP-DM 23
2.6 Evaluasi dan Metode Pengujian 24
2.7 Penelitian terdahulu 25
BAB III 31
METODE PENELITIAN 31
3.1 Rancangan Penelitian 31
3.1.1 Tahapan Penelitian 32
3.1.2 Objek Penelitian 33
3.2 Sumber Data 34
3.3 Teknik Pengumpulan Data 34
3.4 Teknik Analisis Data 35
3.4.1 Alur Algoritma Komputasi dan Tahapan Perhitungan PCA 36
BAB IV 39
HASIL DAN PEMBAHASAN 39
4.1 Analisis Kebutuhan 39
4.1.1 Analisis Kebutuhan data 39
4.1.2 Analisis Kebutuhan Perangkat Keras 40
4.1.3 Analisis Kebutuhan Perangkat Lunak 41
4.2 Perancangan Sistem dan Model 42
4.2.1 Perancangan Alur Feature Engineering 42
4.2.2 Perancangan Arsitektur Model Ensemble 43
4.2.3 Perancangan Arsitektur Integrasi dan Alur Data 45

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.3 Perancangan Antarmuka (UI) - Alur Cepat 45
4.4 Implementasi Sistem 45
4.4.1 Implementasi Sistem 45
4.4.2 Implementasi Model dan Backend Rest API FastAPI 50
4.4.3 Efektivitas dan Kontribusi Fitur Ekstraksi (HSV, GLCM, Canny) 54
4.4.4 Implementasi Aplikasi Mobile 55
4.5 Pengujian dan Pembahasan 58
4.5.1 Deskripsi dan Prosedur Pengujian 59
4.5.2 Hasil Pengujian Performa Klasifikasi 59
4.5.3 Hasil Uji Model Utuh (Fitur Hibrida) dan Analisis Komputasi 61
4.5.4 Analisis Data dan Evaluasi Pengujian 63
4.5.4 Pengujian Sistem Klasifikasi Aplikasi Mobile 69
4.6 Analisis Integrasi Sistem 86
BAB V 88
KESIMPULAN DAN SARAN 88
5.1 Kesimpulan 88
5.2 Saran 89
DAFTAR PUSTAKA 90
LAMPIRAN 93

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2. 1 Tanaman Cabai (Capsicum annuum L.) Sehat 5
Gambar 2. 2 Tanaman Cabai (Capsicum frutescens L.) Sehat 6
Gambar 2. 3 Gejala Infeksi Geminivirus (Geminivirus) pada Daun Cabai 7
Gambar 2. 4 Gejala Cercospora capsici 7
Gambar 2. 5 Malformasi Daun Akibat Leaf curl virus 8
Gambar 3. 1 Rancangan Penelitian 32
Gambar 4. 1 Citra asli tanaman cabai 46
Gambar 4. 2 Hasil segmentasi biner menggunakan algoritma Canny Edge Detection untuk mempertegas struktur garis daun dan batang 47
Gambar 4. 3 Citra Asli Daun Cabai Bergejala Penyakit dalam Ruang Warna RGB . 48
Gambar 4. 4 Hasil Ekstraksi Kanal Hue pada Citra Daun Cabai 48
Gambar 4. 5 Hasil Ekstraksi Kanal Saturation pada Citra Daun Cabai 49
Gambar 4. 6 Hasil Ekstraksi Kanal Value pada Citra Daun Cabai 49
Gambar 4. 7 Citra Lapangan Tanaman Cabai Keseluruhan 78
Gambar 4. 8 Citra Lapangan Daun Cabai Terjangkit Geminivirus 78
Gambar 4. 9 Citra Lapangan Daun Cabai Terjangkit Cercospora 79
Gambar 4. 10 Citra Lapangan Daun Cabai Terjangkit Leaf Curl 79
Gambar 4. 11 Citra Lapangan Daun Cabai Sehat 79

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 1 Proses Ekstraksi Fitur Menuju PCA 13
Tabel 2 Analisis Alur Komputasi 19
Tabel 3 Research Gap 26
Tabel 4 Library Machine Learning 41
Tabel 5 Hasil Pengujian Performa Klasifikasi Algoritma XGBoost pada Skenario Fitur Parsial 54
Tabel 6 Hasil Pengujian Performa Klasifikasi Algoritma Random Forest pada Skenario Fitur Parsial 55
Tabel 7 Perbandingan Performa dan Konfigurasi XGBoost terhadap PCA 60
Tabel 8 Perbandingan Performa dan Konfigurasi Random Forest terhadap PCA 61
Tabel 9 Perbandingan Performa dan Konfigurasi XGBoost terhadap PCA 62
Tabel 10 Perbandingan Performa dan Konfigurasi Random Forest terhadap PCA 63
Tabel 11 Skenario dan Hasil Uji Black-Box Testing 69
Tabel 12 Hasil Pengujian dan Pengolahan Skor System Usability Scale (SUS) .. 74
Tabel 13 Hasil Uji Dataset Lapangan 80

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Tanaman cabai merupakan komoditas hortikultura strategis di Indonesia dengan luas panen cabai rawit dan cabai merah yang menempati peringkat teratas sayuran semusim, dan menunjukkan tren kenaikan produksi nasional belakangan ini (Suryani, 2024). Penyakit tanaman cabai, seperti *geminivirus, cercospora, leaf curl*, kekurangan nutrisi, dan serangan kutu putih, sering menyerang daun dan menyulitkan deteksi manual karena gejala visual mirip. Klasifikasi otomatis penyakit daun cabai menjadi sangat penting, dengan penelitian terkini menggunakan fitur HSV dan GLCM yang diklasifikasikan SVM untuk akurasi hingga 91,3% pada lima kategori tersebut (Nurmadinah et al., 2025).

Penelitian sejenis dalam mendeteksi penyakit tanaman menggunakan Convolutional Neural Network (CNN) seperti AlexNet yang mencapai akurasi 99,16% untuk klasifikasi penyakit daun jagung (Singh et al., 2022). MaizeNet (Faster-RCNN dengan ResNet-50) sebesar 97,89% pada empat penyakit daun jagung (Masood et al., 2023), serta CNN custom Keras-TensorFlow sebesar 98,56% untuk tiga penyakit daun jagung plus sehat (Olayiwola & Adejoju, 2023). Kelebihan utama CNN adalah kemampuan mengekstraksi fitur citra secara otomatis dengan performa deteksi penyakit tanaman yang sangat baik, seperti terlihat pada akurasi CNN 93,93% dengan fitur GLCM, HSV, Lab, dan LBP pada dataset jagung (Arifin & Insani, 2025). Namun, CNN memerlukan dataset besar dan sumber daya komputasi tinggi sebagai kendala utamanya.

XGBoost merupakan algoritma ensemble berbasis gradient boosting yang efektif untuk data tabular hasil feature engineering karena mampu memodelkan hubungan nonlinier antar fitur melalui proses optimasi bertahap pada setiap pohon keputusan (Siringoringo et al., 2022). Beberapa penelitian menunjukkan bahwa XGBoost memiliki performa yang kompetitif bahkan lebih baik dibanding model pembanding seperti Random Forest dan LightGBM, terutama pada data dengan ketidakseimbangan kelas dan kebutuhan tuning parameter yang tepat (Wijayanto & Hartato, 2024). Dalam

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

klasifikasi penyakit tanaman berbasis citra yang telah diubah menjadi vektor fitur melalui ekstraksi HSV, GLCM, dan edge, XGBoost relevan digunakan karena dapat bekerja baik pada fitur numerik hasil reduksi dimensi seperti PCA serta mendukung efisiensi komputasi pada sistem yang ringan (Siringoringo et al., 2022; Wijayanto & Hartato, 2024).

Konteks penelitian tentang penyakit tanaman cabai berbasis citra, pemanfaatan XGBoost sebagai classifier utama umumnya belum disertai kajian sistematis mengenai skema feature engineering dan reduksi dimensi yang optimal, seperti kombinasi fitur warna HSV, tekstur GLCM, dan tepi (edge) dengan Canny/Sobel. Berbagai studi menunjukkan bahwa teknik reduksi dimensi seperti Principal Component Analysis (PCA) sering digunakan untuk memproyeksikan fitur ke dimensi lebih rendah, tetapi konfigurasi PCA yang spesifik seperti jumlah komponen utama atau persentase variansi kumulatif jarang dianalisis secara rinci pada kasus klasifikasi penyakit daun tanaman (Nurmadinah et al., 2025; Kiran & Chandrappa, 2023). Oleh karena itu, penelitian ini mengusulkan skema feature engineering HSV, GLCM, edge diikuti PCA dengan analisis variansi terperinci, sebelum diklasifikasikan menggunakan XGBoost untuk aplikasi mobile yang efisien.

Berdasarkan uraian tersebut, penelitian ini membutuhkan sebuah aplikasi klasifikasi penyakit tanaman cabai berbasis mobile yang dapat digunakan petani secara praktis di lapangan dan berfokus pada implementasi teknik feature engineering berbasis PCA pada sistem klasifikasi penyakit tanaman menggunakan metodeXGBoost untuk klasifikasi penyakit tanaman yang lebih efisien dan akurat.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi *feature engineering* pada system klasifikasi penyakit tanaman cabai menggunakan XGBoost?

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

1.3 Batasan Masalah

Agar penelitian lebih terarah dan fokus, maka ditetapkan beberapa batasan sebagai berikut:

1. Penelitian hanya mencakup beberapa kelas kondisi daun, yaitu infeksi virus golongan *Geminivirus*, *Cercospora, leaf curl*, dan daun sehat.
2. Dataset yang digunakan bersumber dari "DatasetCabai" (Rahmat Kiswanto) dan "Cabai" (Deslia Purnama Sari) yang tersedia di Kaggle dan Roboflow Universe.
3. Teknik reduksi dimensi dibatasi pada PCA standar dengan variasi konfigurasi jumlah komponen utama tetap dan persentase variansi kumulatif tertentu
4. Penelitian berfokus pada tahapan *feature engineering*, reduksi dimensi menggunakan PCA, dan pengujian model menggunakan algoritma XGBoost, serta diimplementasikan hingga tahap prototipe aplikasi *mobile* berbasis Flutter. Aspek integrasi bisnis skala besar, keamanan tingkat produksi, dan skalabilitas sistem di luar cakupan penelitian ini.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menerapkan feature engineering PCA (jumlah komponen utama tetap dan persentase variansi kumulatif) pada fitur HSV, GLCM, edge untuk klasifikasi penyakit daun cabai menggunakan XGBoost.
2. Menganalisis performa sistem klasifikasi XGBoost pasca-feature engineering PCA yang diukur melalui akurasi, presisi, recall, dan F1-score.

1.4.2 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Menghasilkan standar optimalisasi PCA untuk feature engineering citra penyakit tanaman.
2. Memudahkan petani cabai mendeteksi dini penyakit melalui aplikasi mobile berbasis citra.
3. Mempercepat pengobatan tanaman dengan klasifikasi akurat secara instan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4. Mengurangi kerugian ekonomi petani akibat keterlambatan deteksi penyakit manual

1.5 Sistematika Penulisan

Secara umum, skripsi ini disusun dengan sistematika sebagai berikut:

Bab I Pendahuluan

Berisi latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

Bab II Tinjauan Pustaka Berisi kajian teori yang relevan, antara lain: tanaman cabai dan penyakit utamanya; konsep citra digital dan *computer vision*; konsep dasar *machine learning* dan klasifikasi; teori *Principal Component Analysis* (PCA); algoritma XGBoost; serta rangkuman penelitian terdahulu.

Bab III Metode Penelitian Menguraikan rancangan penelitian menggunakan kerangka hibrida Waterfall dan CRISP-DM, tahapan pengumpulan data, teknik pra-pemrosesan citra, perancangan skema *feature engineering* (HSV, GLCM, *edge*, PCA), perancangan dan pelatihan model XGBoost, serta perancangan prototipe aplikasi *mobile* Flutter.

Bab IV Hasil dan Pembahasan Menyajikan hasil pra-pemrosesan citra, hasil pelatihan dan pengujian model XGBoost untuk setiap konfigurasi PCA, analisis performa berdasarkan metrik evaluasi, serta pembahasan implementasi model ke dalam prototipe aplikasi *mobile*.

Bab V Kesimpulan dan Saran

Berisi kesimpulan yang secara ringkas menjawab rumusan masalah dan tujuan penelitian, serta saran untuk pengembangan penelitian dan pengembangan sistem klasifikasi/deteksi penyakit tanaman cabai di masa mendatang.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB II
TINJAUAN PUSTAKA

2.1 Tanaman Cabai dan Penyakit Utama

2.1.1 Tanaman Cabai

Tanaman cabai (*Capsicum annuum L.* dan *C. frutescens L.*) merupakan komoditas hortikultura strategis di Indonesia dengan produksi meningkat 5,5% di Jawa dan 3% di luar Jawa periode 2019-2023, diproyeksikan 3,5 juta ton hingga 2028 (Kementerian Pertanian, 2024). Cabai rawit dan merah mendominasi panen sayuran semusim, berperan besar dalam inflasi pangan dan pendapatan petani (Suryani, 2024).

Dalam penelitian ini, objek karakteristik digital dan morfologi tanaman cabai difokuskan pada dua varietas utama yang memiliki nilai ekonomis tinggi, yaitu:

A. *Capsicum annuum L.* (Cabai Merah Keriting)

Secara visual, objek tanaman cabai jenis *Capsicum annuum L.* atau yang biasa disebut cabai merah keriting dalam kondisi normal menunjukkan keadaan buah dan struktur daun yang sehat. Baseline visual untuk varietas ini dapat dilihat pada Gambar 2.1.

Gambar 2. 1 Tanaman Cabai (Capsicum annuum L.) Sehat

B. *Capsicum frutescens L.* (Cabai Rawit)

Varietas lain yang menjadi objek pada penelitian ini adalah *Capsicum frutescens L.* atau yang kerap disebut dengan cabai rawit. Karakteristik morfologi daun dan buah dalam kondisi sehat sebagai pembanding ditunjukkan pada Gambar 2.2.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 2. 2 Tanaman Cabai (Capsicum frutescens L.) Sehat

2.1.2 Penyakit Utama

Penyakit utama pada tanaman cabai yang menjadi fokus penelitian ini meliputi beberapa jenis infeksi patogen yang sering menyerang kedua varietas di atas, antara lain:

A. *Geminivirus* (*Geminivirus*) Menyebabkan daun keriting ke atas, menebal tulang daun, mosaic kuning, dan kerdil dengan tingkat infeksi sistemik hingga 75% populasi (Nurfadhilah et al., 2024). Karakteristik visual dari infeksi ini ditandai dengan hilangnya zat hijau daun (klorosis) secara konsisten yang membentuk pola mosaik bervariasi (Sari & Hermanto, 2022). Visualisasi gejala infeksi *Geminivirus* ini dapat dilihat pada Gambar 2.3.

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 2. 3 Gejala Infeksi Geminivirus (Geminivirus) pada Daun Cabai

B. *Cercospora leaf spot* (*Cercospora capsici*)

Menunjukkan bercak bulat coklat dengan pusat abu-abu dan cincin gelap (mata-katak) sehingga daun kuning dan gugur terutama pada daun bawah (Inaya et al., 2022; Deviant et al., 2023). Secara digital, bercak "mata-katak" ini membentuk kontras warna yang tegas antara pusat lesi dan jaringan daun sehat di sekitarnya, yang menjadi penciri utama dalam identifikasi citra (Pradana & Nugroho, 2023). Gejala cercospora ini ditunjukkan pada Gambar 2.4.

Gambar 2. 4 Gejala Cercospora capsici

C. *Leaf curl virus*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Menyebabkan pinggir daun melengkung ke atas/bawah, daun kecil rapuh, *witches'-broom*, dan buah cacat akibat *Tobacco leaf curl Geminivirus* (Khethari, 2025; Rachmah, 2015). Kondisi daun yang malformasi dan mengkerut ekstrem ini secara signifikan mengubah struktur geometri permukaan daun asli (Putra et al., 2021). Representasi visual dari penyakit *Leaf curl virus* disajikan pada Gambar 2.5.

Gambar 2. 5 Malformasi Daun Akibat Leaf curl virus

2.2 Ekstraksi Fitur dan Feature Engineering dengan OpenCV

Feature engineering pada penelitian ini merujuk pada proses pengolahan lanjutan terhadap vektor fitur hasil ekstraksi warna, tekstur, dan tepi citra daun cabai agar menghasilkan representasi data yang lebih informatif, padat, dan efisien untuk proses klasifikasi (Gonzalez & Woods, 2018; Mustakim et al., 2025). Melalui pustaka Open Source Computer Vision Library (OpenCV), informasi mentah yang terkandung di dalam tiap piksel citra diekstrak secara matematis ke dalam domain parameter kuantitatif diskrit tertentu (OpenCV, 2025). Langkah komputasi ini sangat krusial guna menjembatani transisi data dari bentuk spasial visual menjadi matriks numerik kontinu yang dapat diproses oleh algoritma kecerdasan buatan.

2.2.1 Ekstraksi Fitur

Ekstraksi fitur merupakan tahap awal *feature engineering* yang mengubah citra piksel menjadi vektor numerik representatif gejala penyakit cabai (Akhrian Aulia Syahidi, 2024; Gonzalez & Woods, 2018). Pada sistem klasifikasi penyakit tanaman cabai ini, karakteristik visual penanda patologi daun dimanifestasikan melalui pendekatan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

hibrida yang menggabungkan tiga domain informasi utama secara terintegrasi (Mustakim et al., 2025), yaitu:

1. **Fitur Warna (*Hue, Saturation, Value* - HSV):** Menghitung nilai *Mean* (rata-rata), *Standard Deviation* (standar deviasi), dan *Skewness* (kecondongan) pada kanal *Hue*, *Saturation*, dan *Value* untuk mengidentifikasi warna gejala infeksi (Akhrian Aulia Syahidi, 2024).
2. **Fitur Tekstur (*Gray Level Co-occurrence Matrix* - GLCM):** Mengekstrak nilai *Contrast*, *Dissimilarity*, *Homogeneity*, *Energy*, dan *Correlation* berdasarkan 4 orientasi sudut arah ketetanggaan piksel untuk mengenali struktur tekstur permukaan daun (Akhrian Aulia Syahidi, 2024).
3. **Fitur Tepi (*Canny Edge*):** Menghitung nilai *Edge Length*, *Edge Count*, dan *Edge Density* untuk mendeteksi batas tegas bercak serangan virus atau jamur pada daun cabai (Akhrian Aulia Syahidi, 2024).

Penggabungan ketiga domain ini bertujuan agar model klasifikasi *Extreme Gradient Boosting* (XGBoost) mampu mengenali variasi morfologi penyakit secara holistik, baik dari aspek degradasi warna piksel maupun kerusakan struktural permukaan daun (Mustakim et al., 2025).

2.2.2 Mekanisme Komputasi Ruang Warna HSV

Pengolahan citra digital berbasis karakteristik warna umumnya menggunakan ruang warna RGB (*Red, Green, Blue*). Namun, ruang warna RGB sangat rentan terhadap distorsi akibat fluktuasi intensitas pencahayaan di lapangan (Gonzalez & Woods, 2018). Oleh karena itu, citra daun cabai terlebih dahulu ditransformasikan ke ruang warna HSV (*Hue, Saturation, Value*) menggunakan OpenCV untuk memisahkan informasi warna kromatik dari nilai kecerahan cahaya (OpenCV, 2025).

Konversi matematis dari ruang warna RGB ke HSV dieksekusi melalui normalisasi nilai R, G, B ke dalam rentang $[0,1]$ diikuti oleh kalkulasi inter-kanal berdasarkan Persamaan (2.X), Persamaan (2.X), dan Persamaan (2.X) berikut (Gonzalez & Woods, 2018):

$$\boldsymbol{V} = \max(R, G, B)$$

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

$$\boldsymbol{S} = \begin{cases} 0, & \text{jika } V = 0 \\ \dfrac{V - \min(R,G,B)}{V}, & \text{jika } V > 0 \end{cases}$$

$$\boldsymbol{H} = \begin{cases} \text{tidak terdefinisi}, & \text{jika } S = 0 \\ 60° \times \dfrac{G - B}{V - \min(R,G,B)}, & \text{jika } V = R \text{ dan } G \geq B \\ 60° \times \dfrac{G - B}{V - \min(R,G,B)} + 360°, & \text{jika } V = R \text{ dan } G < B \\ 60° \times \dfrac{B - R}{V - \min(R,G,B)} + 120°, & \text{jika } V = G \\ 60° \times \dfrac{R - G}{V - \min(R,G,B)} + 240°, & \text{jika } V = B \end{cases}$$

Setelah matriks citra berhasil dipisahkan ke dalam kanal H, S, dan V, sistem menghitung tiga metrik statistik deskriptif dari sebaran piksel tiap kanal guna memperoleh data numerik yang representatif terhadap gejala warna penyakit (Akhrian Aulia Syahidi, 2024; Mustakim et al., 2025). Ketiga statistik tersebut adalah:

- ***Mean*** **(Rata-rata):** Menandakan warna dominan yang menyelimuti objek daun cabai.
- ***Standard Deviation*** **(Standar Deviasi):** Mengukur dispersi kontras atau seberapa bervariasi gradasi warna pada area bercak penyakit.
- ***Skewness*** **(Kecondongan):** Mengidentifikasi kemiringan distribusi intensitas warna, yang sangat efektif untuk membedakan pergeseran warna halus akibat gejala klorosis (daun menguning).

Melalui perhitungan 3 metrik statistik pada 3 kanal warna (H, S, V), diperoleh total **9 fitur warna** hibrida.

2.2.3 Mekanisme Komputasi Fitur Tekstur GLCM

Ekstraksi fitur tekstur ditujukan untuk mendeteksi kerusakan struktur spasial pada permukaan daun cabai, seperti kerutan akibat serangan virus *Curl* atau lubang makro akibat spora jamur. Metode *Gray Level Co-occurrence Matrix* (GLCM) bekerja dengan menganalisis hubungan spasial ketetanggaan antar dua piksel berjarak spasial

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(d) yang memiliki orientasi sudut arah tertentu (θ) pada citra skala keabuan (*grayscale*) (Haralick et al., 1973).

Pada sistem ini, citra konversi skala keabuan diatur menggunakan parameter jarak $d = 5$ piksel untuk menangkap tekstur bercak makro daun cabai, dikombinasikan dengan empat sudut arah orientasi mata angin, yaitu $\theta = [0°, 45°, 90°, 135°]$ agar fitur yang dihasilkan bersifat kebal terhadap variasi rotasi daun saat pengambilan sampel foto (Scikit-Image, 2024). Dari matriks kookurensi spasial tersebut, dihitung 5 properti tekstur Haralick utama menggunakan rumus formal berikut (Haralick et al., 1973):

Contrast (Kontras): Mengukur intensitas variasi lokal antar piksel tetangga.

$$\textbf{Contrast} = \sum_{i,j=0}^{N-1} P_{i,j}\,(i-j)^2$$

- ***Dissimilarity*** **(Ketidakmiripan):** Mengukur derajat linearitas perbedaan keabuan piksel lokal.

$$\textbf{Dissimilarity} = \sum_{i,j=0}^{N-1} P_{i,j}\,|i-j|$$

- ***Homogeneity*** **(Homogenitas):** Mengukur kedekatan sebaran elemen matriks GLCM terhadap nilai diagonal utama.

$$\textbf{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

- ***Energy*** **(Energi):** Mengukur keseragaman tekstur; nilai tinggi menunjukkan pola tekstur yang teratur dan berulang.

$$\textbf{Energy} = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j}^2}$$

- ***Correlation*** **(Korelasi):** Mengukur ketergantungan linear nilai keabuan antar piksel tetangga.

$$\textbf{Correlation} = \sum_{i,j=0}^{N-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2\sigma_j^2}}\right]$$

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Keterangan: $P_{i,j}$ merepresentasikan nilai probabilitas kemunculan piksel keabuan i dan j pada dimensi matriks $N \times N$. Nilai akhir dari kelima fitur tekstur ini diperoleh dari rata-rata (*mean*) kalkulasi keempat sudut arah orientasi, menghasilkan **5 fitur tekstur** (Akhrian Aulia Syahidi, 2024).

2.2.4 Mekanisme Komputasi Tepi *Canny Edge*

Fitur tepi berfungsi untuk menangkap diskontinuitas atau perubahan ekstrem dari intensitas derajat keabuan piksel citra, yang merepresentasikan batas luar (kontur) daun cabai serta batasan tegas dari struktur bercak penyakit (Canny, 1986). Proses internal deteksi tepi Canny di dalam sistem berjalan melalui empat fase berurutan menggunakan modul OpenCV (Canny, 1986; OpenCV, 2025):

1. ***Gaussian Blurring*:** Mereduksi *noise* frekuensi tinggi pada citra menggunakan kernel Gaussian.
2. ***Calculated Intensity Gradient*:** Menghitung nilai magnitudo dan arah gradien menggunakan operator Sobel horizontal (G_x) dan vertikal (G_y).
3. ***Non-maximum Suppression*:** Menipiskan garis tepi citra dengan mengeliminasi piksel yang bukan merupakan nilai lokal maksimum sepanjang arah gradien.
4. ***Hysteresis Thresholding*:** Mengunci tepi sejati menggunakan dua ambang batas, yaitu *threshold* bawah sebesar 100 dan *threshold* atas sebesar 200.

Keluaran dari algoritma Canny berupa citra biner di mana piksel bernilai 255 (putih) mewakili area tepi, dan piksel bernilai 0 (hitam) mewakili latar belakang. Sistem kemudian mengekstrak metrik kuantitatif berupa rasio kerapatan tepi (*Edge Density*) menggunakan Persamaan (2.X) (Akhrian Aulia Syahidi, 2024; Mustakim et al., 2025):

$$\text{Edge Density} = \frac{\sum(Piksel > 0)}{\text{Lebar Citra} \times \text{Tinggi Citra}}$$

Rasio ini menghasilkan **1 fitur tepi** tunggal yang merepresentasikan tingkat kekerutan atau kerusakan garis batas permukaan daun tanaman cabai.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.2.5 Integrasi Pipeline Input-Proses-Output (IPO) Ekstraksi Fitur Menuju PCA

Sebelum sekumpulan data numerik di atas direduksi dimensinya oleh modul *Principal Component Analysis* (PCA), seluruh fitur lepasan dari ketiga domain di atas harus diintegrasikan ke dalam satu kesatuan sistem komputasi data. Hubungan struktural dari tahapan Input, Proses, dan Output ekstraksi fitur sebelum dilewatan menuju PCA dirangkum secara mendalam pada Tabel 1 berikut:

Tabel 1 Proses Ekstraksi Fitur Menuju PCA

Komponen	**Tahapan Implementasi pada Sistem**	**Ragam Bentuk Data / Dimensi**
Input	Citra daun cabai pasca-preprocessing (*resize* 224×224 piksel) dalam kondisi ternormalisasi (Gonzalez & Woods, 2018).	Matriks Tensor Citra (Format Array NumPy: $224 \times 224 \times 3$)
Process	1. **Ekstraksi HSV:** Citra dikonversi ke HSV; menghitung *mean, std dev,* dan *skewness* pada kanal H, S, V (OpenCV, 2025). 2. **Ekstraksi GLCM:** Citra diubah ke *grayscale*; menghitung matriks spasial ($d = 5$, 4 sudut) untuk memproduksi properti *Contrast, Dissimilarity, Homogeneity, Energy,*	Operasi array komputasi linier pada pustaka Python (OpenCV, Scikit-Image, NumPy).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

	Correlation (Scikit-Image, 2024). 3. **Ekstraksi Canny:** Deteksi tepi biner dengan ambang batas [100, 200]; menghitung rasio jumlah piksel putih terhadap total luas citra (Canny, 1986). 4. ***Feature Fusion*:** Penggabungan seluruh hasil array secara horizontal (*concatenation*) (Akhrian Aulia Syahidi, 2024).	
Output (Menjadi *Input* bagi PCA)	Vektor gabungan fitur hibrida (*hybrid feature vector*) terpadu yang merangkum seluruh properti visual daun cabai secara kuantitatif.	Matriks Vektor Hibrida Berdimensi $n \times 15$ (9 kolom HSV + 5 kolom GLCM + 1 kolom Canny), siap ditransfer menuju fungsi StandardScaler dan pca.transform().

2.2.6 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) atau Analisis Komponen Utama merupakan sebuah teknik statistik multivariat linier yang digunakan untuk mereduksi dimensi dari suatu dataset dengan tetap mempertahankan sebagian besar varians atau informasi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

penting di dalamnya (Jolliffe, 2002). Di dalam sistem komputer dan pengolahan citra digital, representasi data hibrida yang menggabungkan ekstraksi fitur warna (Hue, Saturation, Value), tekstur (Gray Level Co-occurrence Matrix), dan tepi (Canny Edge Density) sering kali menghasilkan matriks berdimensi tinggi (high-dimensional data) (Gonzalez & Woods, 2018). Pada penelitian ini, penggabungan ketiga ekstraksi fitur tersebut menghasilkan vektor berukuran 15 dimensi untuk setiap citra daun cabai.

Peningkatan jumlah dimensi fitur yang terlalu besar dapat memicu fenomena Curse of Dimensionality (Kutukan Dimensi), di mana ruang fitur menjadi sangat jarang (sparse) sehingga performa algoritma pengklasifikasi seperti Extreme Gradient Boosting (XGBoost) cenderung menurun, meningkatkan beban komputasi, dan rentan terhadap masalah overfitting (Mustakim et al., 2025). Selain itu, di antara 15 fitur hibrida tersebut, terdapat potensi korelasi antar-fitur yang tinggi atau redundansi data (Jolliffe, 2002), misalnya korelasi antara intensitas kecerahan pada kanal Value (HSV) dengan tingkat kontras (Contrast) pada GLCM.

Feature engineering pada penelitian ini merujuk pada proses pengolahan lanjutan terhadap vektor fitur hasil ekstraksi HSV, GLCM, dan tepi agar lebih informatif dan efisien untuk proses klasifikasi. Setelah citra dikonversi menjadi vektor fitur berdimensi tinggi, dilakukan normalisasi untuk menyamakan skala antar fitur serta reduksi dimensi menggunakan PCA sehingga hanya komponen utama yang paling banyak menjelaskan variansi data yang dipertahankan (Gonzalez & Woods, 2018; Mustakim et al., 2025). PCA menyelesaikan masalah redundansi ini dengan mentransformasikan himpunan variabel asli yang saling berkorelasi menjadi himpunan variabel baru yang tidak saling berkorelasi (uncorrelated), yang disebut sebagai Komponen Utama (Principal Components) (Jolliffe, 2002).

2.2.7 Letak dan Peran PCA dalam Pipeline Computer Vision

Dalam arsitektur *Computer Vision* tradisional berbasis *handcrafted features*, PCA memiliki posisi yang sangat krusial setelah tahap ekstraksi fitur dan sebelum tahap klasifikasi objek (Gonzalez & Woods, 2018). Secara umum, tahapan standar pemrosesan citra digital terdiri atas alur: *Image Input* → *Preprocessing* → *Feature*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Extraction → ***Feature Transformation/Reduction (PCA)*** → *Classification* (Gonzalez & Woods, 2018).

Letak PCA pada penelitian klasifikasi penyakit tanaman cabai ini berada tepat pada fase ***Feature Fusion & Transformation***. Setelah 9 fitur warna HSV, 5 fitur tekstur GLCM, dan 1 fitur tepi Canny diekstraksi secara independen dari citra daun cabai, seluruh fitur tersebut digabungkan secara horizontal (*concatenated*) menjadi satu bunderan vektor hibrida tunggal berdimensi 15. Di sinilah PCA masuk sebagai jembatan *preprocessing data numerik*. Peran utama PCA dalam arsitektur *Computer Vision* pada sistem ini meliputi:

1. **Dekopling Karakteristik Citra:** Memisahkan informasi visual yang tumpang tindih sehingga model tidak mengalami bias akibat multikolinieritas data (Jolliffe, 2002).
2. **Kompresi Informasi Visual Semantik:** Mengekstrak esensi visual (seperti degradasi warna kuning penyakit atau kerutan struktur daun) ke dalam ruang metrik yang lebih padat, sehingga mempercepat waktu komputasi pemrosesan *pohon keputusan* (*decision tree*) pada algoritma XGBoost (Mustakim et al., 2025).

2.2.8 Prinsip Geometris dan Ortogonalitas PCA

Secara geometris, PCA bekerja dengan memutar koordinat ruang fitur asli menuju ruang koordinat baru berdasarkan arah sebaran data yang memiliki varians terbesar (Jolliffe, 2002).

Sumbu baru pertama, dinamakan *Principal Component 1* (PC1), ditarik melintasi arah atau garis linier yang menangkap sebaran data (*variance*) paling maksimum (Jolliffe, 2002). Sumbu baru kedua, dinamakan *Principal Component 2* (PC2), wajib ditarik secara tegak lurus (ortogonal atau bersudut 90 derajat) terhadap PC1 (Jolliffe, 2002). Begitu pula untuk komponen berikutnya (PC3, PC4, hingga PCn), seluruhnya harus saling tegak lurus satu sama lain. Sifat ortogonalitas ini menjamin bahwa setiap Komponen Utama membawa informasi visual yang sepenuhnya unik dan tidak ada lagi data yang tumpang tindih di antara komponen-komponen tersebut (Jolliffe, 2002).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.2.9 Tahapan Matematis Transformasi PCA

Di dalam komputasi sistem, proses reduksi dimensi dari 15 fitur hibrida menjadi komponen utama melalui fungsi pca.transform() dilakukan melalui dekomposisi linear. PCA merupakan teknik transformasi linier yang memproyeksikan data **X** ke ruang baru melalui dekomposisi eigen matriks kovariansi **Σ**, di mana kolom matriks **V** menyatakan vektor eigen yang digunakan sebagai sumbu baru (PCA) dan **Λ** berisi nilai eigen yang menunjukkan besar kontribusi variansi masing-masing komponen (Jolliffe, 2002).

Secara terperinci, tahapan kalkulasi sistem adalah sebagai berikut:

1. **Sentralisasi Data (*Mean Centering*)**

Sebelum sumbu koordinat baru dihitung, seluruh titik data pada matriks fitur asli harus digeser sehingga titik pusat sebaran data berada tepat di koordinat pusat (0,0) (Jolliffe, 2002). Proses ini dilakukan dengan menghitung nilai rata-rata (*mean*) dari masing-masing kolom fitur, kemudian mengurangi setiap baris data dengan nilai rata-rata tersebut melalui Persamaan (2.X):

$$\mathbf{X}_{centered} = \mathbf{X} - \boldsymbol{\mu}$$

Keterangan:

- $\mathbf{X}_{centered}$ = Matriks fitur setelah proses sentralisasi data.
- $\mathbf{X}$ = Matriks fitur hibrida asli berukuran $n \times 15$.
- $\boldsymbol{\mu}$ = Vektor nilai rata-rata dari 15 fitur dasar.

2. Pembentukan Matriks Kovarians (Covariance Matrix)

Untuk mendeteksi tingkat redundansi atau hubungan linear antar-fitur (misalnya hubungan antara pola warna bintik Cercospora dengan pola tekstur kerutan daun Curl

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Virus), sistem menghitung matriks kovarians berukuran 15×15 menggunakan Persamaan (2.X):

$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}_{centered}^{T}\mathbf{X}_{centered}$$

Keterangan:

- $\mathbf{\Sigma}$ = Matriks kovarians berukuran 15×15.
- n = Jumlah total sampel citra daun cabai.
- $\mathbf{X}_{centered}^{T}$ = Matriks *transpose* dari data tersentralisasi.

3. Dekomposisi Nilai Eigen (*Eigenvalues)* dan Vektor eigen *(Eigenvectors)*

Matriks kovarians yang telah terbentuk kemudian dipecah secara aljabar linear untuk menghasilkan Vektor Eigen ($\mathbf{v}$) dan Nilai Eigen (λ) berdasarkan rumus karakteristik pada Persamaan (2.X):

$$\mathbf{\Sigma v} = \lambda \mathbf{v}$$

Keterangan:

- $\mathbf{\Sigma}$ = Matriks kovarians data.
- $\mathbf{v}$ = Vektor Eigen (*Eigenvector*) yang merepresentasikan arah dari sumbu baru (komponen utama) (Jolliffe, 2002).
- λ = Nilai Eigen (*Eigenvalue*) yang merepresentasikan besarnya varians atau kekuatan informasi yang diikat oleh arah sumbu baru tersebut (Jolliffe, 2002).

4. Proyeksi Ruang Baru dan Reduksi Dimensi

Sistem akan mengurutkan ke-15 Vektor Eigen yang terbentuk berdasarkan Nilai Eigen dari yang terbesar hingga terkecil. Komponen dengan Nilai Eigen tertinggi menyimpan informasi morfologi daun cabai paling banyak. Pada penelitian ini, jumlah komponen dibatasi secara ketat menjadi 5 komponen utama ($k = 5$) melalui matriks transformasi proyeksi $\mathbf{W}_k$. Langkah akhir reduksi dimensi dihitung menggunakan Persamaan (2.X):

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

$$\mathbf{X}_{final} = \mathbf{X}_{scaled} \times \mathbf{W}_k$$

Keterangan:

- $\mathbf{X}_{final}$ = Matriks koordinat fitur baru hasil reduksi berukuran $n \times 5$.
- $\mathbf{X}_{scaled}$ = Matriks input fitur lapangan pasca-standardisasi.
- $\mathbf{W}_k$ = Matriks transformasi yang dibentuk dari 5 Vektor Eigen teratas berukuran 15×5.

2.2.10 Analisis Alur Komputasi Input-Proses-Output

Untuk memberikan pemahaman yang terstruktur mengenai mekanisme transformasi data daun cabai di dalam sistem, Ringkasan Input-Proses-Output (IPO) dari implementasi PCA disajikan pada Tabel 2 berikut:

Tabel 2 Analisis Alur Komputasi

Komponen	Deskripsi Ragam Data pada Sistem
Input	Matriks numerik kontinu berdimensi $n \times 15$ hasil penggabungan fitur (*feature fusion*) yang terdiri atas 9 fitur warna HSV, 5 fitur tekstur GLCM, dan 1 fitur kerapatan tepi Canny (Gonzalez & Woods, 2018).
Proses	1. Dekomposisi matriks kovarians untuk mengekstrak akar komponen berupa *Eigenvalues* dan *Eigenvectors* (Jolliffe, 2002). 2. Pemilihan *K Principal Components* pertama hingga mencapai target akumulasi variansi 90-95% (Jolliffe, 2002). 3. Proyeksi linear matriks fitur melalui perkalian dengan matriks bobot Vektor Eigen terpilih (Jolliffe, 2002).
Output	Matriks koordinat spasial terreduksi berdimensi $n \times K$ yang bebas dari multikolinieritas, siap diklasifikasikan oleh algoritma XGBoost (Mustakim et al., 2025).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2.2.11 Open CV

OpenCV (Open Source Computer Vision Library) digunakan untuk implementasi ekstraksi fitur HSV dan edge secara efisien, dengan fungsi cv2.cvtColor() untuk konversi BGR→HSV/grayscale, cv2.Canny() untuk deteksi tepi, dan integrasi skimage untuk GLCM pada citra penyakit tanaman (Nurmadinah et al., 2025). Dokumentasi resmi OpenCV v4.8 mendukung pipeline preprocessing (resize, CLAHE enhancement) dan feature extraction untuk aplikasi mobile ML dengan waktu inferensi <2 detik pada perangkat resource-terbatas seperti smartphone petani (OpenCV.org, 2025; Bradski & Kaehler, 2012)

2.3 Algoritma XGBoost

XGBoost (*eXtreme Gradient Boosting*) merupakan algoritma *ensemble learning* berbasis *gradient boosting* yang membangun sejumlah pohon keputusan (*Decision Tree*) secara berurutan (*sequential*) dengan mekanisme optimasi gradien yang lebih efisien dibandingkan algoritma *boosting* konvensional. Algoritma ini dikembangkan sebagai pustaka *machine learning* sumber terbuka yang mengoptimalkan fungsi objektif melalui kombinasi regularisasi untuk mencegah *overfitting*, serta penanganan *missing value* secara otomatis (Chen & Guestrin, 2016). Pada setiap tahap *boosting*, pohon baru dibangun untuk mengoreksi kesalahan (residual) dari pohon sebelumnya. Proses minimisasi fungsi objektif pada iterasi ke-*t* dirumuskan sebagai:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t)$$

di mana l menyatakan fungsi kerugian (*loss function*) antara target y_i dan prediksi $\hat{y}_i$, sedangkan $\Omega(f_t)$ adalah suku regularisasi yang mengontrol kompleksitas pohon untuk menghindari *overfitting* (Chen & Guestrin, 2016). Berbeda dengan *gradient boosting* biasa yang hanya menggunakan ekspansi Taylor orde pertama, XGBoost menerapkan optimasi orde kedua (*second-order Taylor expansion*) pada fungsi objektifnya untuk mempercepat konvergensi dan meningkatkan akurasi prediksi secara signifikan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

XGBoost memiliki keunggulan dalam menangani dataset *imbalanced* seperti citra penyakit cabai, di mana kelas minoritas (misalnya *leaf curl*) memiliki jumlah sampel lebih sedikit. Algoritma ini menerapkan parameter *scale_pos_weight* untuk menyeimbangkan kontribusi kelas, serta dapat diintegrasikan dengan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) untuk augmentasi data kelas minoritas sebelum pelatihan. Hyperparameter utama yang dapat dituning pada algoritma ini meliputi *n_estimators* (jumlah pohon), *max_depth* (kedalaman maksimum pohon), *learning_rate* (ukuran langkah optimasi), *subsample* (proporsi data latih untuk setiap pohon), dan *reg_lambda* (koefisien regularisasi L1/L2) (Chen & Guestrin, 2016).

Penelitian empiris menunjukkan bahwa XGBoost mampu mencapai akurasi 93,25% dengan *precision* 87,09%, *recall* 91,25%, dan F1-*score* 89,12% pada dataset klasifikasi setelah melalui *hyperparameter tuning* optimal. Untuk klasifikasi penyakit tanaman berbasis citra, XGBoost dengan *information gain* sebagai kriteria seleksi fitur menghasilkan akurasi 93,10% pada pengujian *5-fold cross-validation*, mengindikasikan efektivitasnya dalam mengidentifikasi atribut visual paling informatif untuk prediksi kelas penyakit. Dalam pipeline klasifikasi penyakit cabai pada penelitian ini, XGBoost digunakan setelah proses *feature extraction* (HSV, GLCM, dan *edge extraction*) serta reduksi dimensi menggunakan PCA. Model XGBoost dibentuk menggunakan *GridSearchCV* untuk mengoptimalkan performa dan memastikan stabilitas prediksi pada dataset citra yang digunakan

2.4 Algoritma Random Forest

Random Forest (RF) merupakan algoritma *ensemble learning* yang membangun sejumlah pohon keputusan (*Decision Tree*) secara independen menggunakan metode *bootstrap aggregating* (*bagging*). Pada setiap proses pelatihan, sekitar 63% data latih diambil secara acak dengan pengembalian untuk membentuk satu pohon, sedangkan sekitar 37% sisanya berfungsi sebagai *out-of-bag* (OOB) sample yang digunakan untuk mengestimasi kesalahan tanpa memerlukan data uji terpisah (Breiman, 2001). Setiap pohon hanya menggunakan subset fitur tertentu secara acak, misalnya sekitar enam fitur dari total 32 yang tersedia pada citra daun cabai. Proses

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pemisahan pada setiap node dilakukan dengan mencari nilai ambang yang meminimalkan *Gini impurity*, yang dirumuskan sebagai:

$$Ginit-1-\ k=1cP2_{mk}$$

di mana pmk menyatakan proporsi data dari kelas ke-k pada node t. Kandungan acak pada pemilihan data dan fitur ini membuat setiap pohon unik serta menurunkan korelasi antar pohon dalam model.

Setelah seluruh pohon terbentuk, prediksi akhir diberikan melalui mekanisme *majority voting*, yaitu kelas yang paling sering dipilih oleh pohon-pohon dalam hutan menjadi hasil prediksi akhir (Breiman, 2001). Algoritma ini juga memiliki kemampuan untuk memperkirakan tingkat kesalahan generalisasi melalui metrik varians dan *strength* antar pohon keputusan. Selain itu, Random Forest dapat menilai tingkat pentingnya fitur (*feature importance*) dengan cara mengacak nilai suatu fitur dalam sampel OOB dan mengamati peningkatan tingkat kesalahan klasifikasi yang terjadi. Nilai penting ini menunjukkan seberapa besar kontribusi suatu fitur, seperti warna dominan pada model citra HSV, terhadap hasil klasifikasi (Breiman, 2001).

2.5 Software Development Life Cycle dan CRISP-DM

2.5.1 Waterfall

Software Development Life Cycle (SDLC) merupakan kerangka proses standar untuk pengembangan perangkat lunak, dengan model Waterfall sebagai pendekatan linear-sekuensial pertama kali diusulkan oleh Royce (1970). Model ini membagi siklus menjadi tahap-tahap berurutan yang saling bergantung, di mana kemajuan hanya mungkin setelah tahap sebelumnya selesai dan diverifikasi sepenuhnya, sehingga sangat cocok untuk proyek skripsi solo dengan requirement stabil, ruang lingkup jelas, dan kebutuhan dokumentasi lengkap (Pressman, 2010; Sommerville, 2016; Dicoding Academy, 2025).

Tahapan pada metodologi pengembangan perangkat lunak Waterfall adalah sebagai berikut:

1. **Requirements Gathering and Analysis:** Identifikasi dan dokumentasi kebutuhan fungsional/non-fungsional dalam Software Requirement Specification (SRS).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tahap ini menentukan fondasi seluruh proyek untuk minimalkan risiko kesalahan, ideal untuk developer tunggal karena fokus perencanaan awal (Royce, 1970).

2. **System Design:** Pembuatan desain arsitektur (High-Level Design/HLD) dan detail implementasi (Low-Level Design/LLD), dihasilkan Software Design Document (SDD) yang mencakup spesifikasi modul dan interface.
3. **Implementation (Coding):** Pengkodean berdasarkan SDD, dengan penekanan pada modularitas dan unit testing awal untuk memastikan kepatuhan desain (Sommerville, 2016).
4. **Integration and Testing:** Integrasi modul dan pengujian sistematis (unit, integration, system, acceptance testing) untuk deteksi defect sebelum deployment.
5. **Deployment:** Penerapan sistem ke lingkungan operasional, termasuk instalasi dan pelatihan pengguna.
6. **Maintenance:** Aktivitas pasca-deployment untuk perbaikan, adaptasi, dan peningkatan berkelanjutan (Pressman, 2010).

2.5.2 CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan metodologi iteratif standar industri untuk proyek data mining/ML, dikembangkan oleh SPSS, Teradata, NCR, DaimlerChrysler (versi 1.0, 1999; updated 2000 oleh Shearer). Model 6-fase non-linier ini berkomplemen SDLC Waterfall untuk aspek ML spesifik seperti data preparation dan modeling pada klasifikasi citra cabai (Shearer, 2000; Chapman et al., 2000).

Tahapan CRISP-DM terdiri dari enam fase iteratif:

1. **Business Understanding:** Pemahaman kebutuhan bisnis/ML untuk deteksi penyakit cabai, penentuan tujuan akurasi >90% mobile app, pengumpulan data awal dataset Kaggle/Roboflow, verifikasi kelayakan proyek (Elkabalawy et al., 2024).
2. **Data Understanding:** Eksplorasi dataset cabai publik, identifikasi class imbalance (*leaf curl* minoritas), pemeriksaan kualitas citra (noise, variasi cahaya), analisis EDA distribusi gejala (Bemthuis et al., 2025).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

3. **Data Preparation:** Pembersihan data, preprocessing (resize 256x256, CLAHE enhancement), feature engineering (HSV/GLCM/edge extraction), augmentasi (flip/rotate), normalisasi MinMaxScaler (Elkabalawy et al., 2024).
4. **Modeling:** Tahap pemilihan dan konfigurasi algoritma *Machine Learning*. Pada penelitian ini, dilakukan eksperimen menggunakan dua algoritma berbasis *tree*, yaitu *Random Forest* dan *XGBoost*, yang dikombinasikan dengan reduksi dimensi PCA. Tuning *hyperparameter* dilakukan menggunakan metode *GridSearchCV* untuk menemukan konfigurasi parameter optimal pada kedua model sebelum divalidasi (Bemthuis et al., 2025).
5. **Evaluation:** Penilaian model menggunakan Confusion Matrix, akurasi, presisi, recall, F1-score via 5-fold cross-validation, OOB error, feature importance analisis, penyelarasan dengan tujuan bisnis (Elkabalawy et al., 2024).
6. **Deployment:** Konversi model ke format tflite, integrasi ke aplikasi Flutter multiplatform, monitoring performa lapangan, rencana maintenance/retraining (Bemthuis et al., 2025).

2.6 Evaluasi dan Metode Pengujian

Evaluasi model *Machine Learning* dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *Confusion Matrix* melalui skema *5-fold cross-validation*. Metode ini digunakan untuk memastikan bahwa model XGBoost yang dibangun memiliki performa yang stabil dan terhindar dari *overfitting* saat menangani data citra daun cabai yang mengalami ketidakseimbangan kelas (*class imbalance*) (Nugroho et al., 2024).

Pengujian sistem secara keseluruhan dibagi menjadi tiga tahapan standar, yaitu:

1. **Evaluasi Performa Model**: Pengujian model dilakukan dengan membagi data menggunakan skema *train-test split* dengan proporsi 80% untuk data latih (*training set*) dan 20% untuk data uji (*testing set*). Guna menghindari risiko *overfitting* dan memastikan stabilitas model XGBoost pada data yang tidak seimbang (*class imbalance*), diterapkan metode *5-fold cross-validation* terstratifikasi (*Stratified K-Fold*) pada data latih selama proses optimasi *hyperparameter*. Metrik evaluasi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

dihitung secara programatis menggunakan pustaka *scikit-learn*, mencakup *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dengan perhatian khusus pada performa kelas penyakit minoritas (Nugroho et al., 2024).

2. **Blackbox Testing**: Pengujian fungsionalitas aplikasi *mobile* dilakukan menggunakan metode *equivalence partitioning* dan *boundary value analysis*. Fokus pengujian ini adalah memvalidasi fungsi-fungsi utama seperti *input* kamera/galeri, proses pengiriman data fitur, inferensi prediksi model XGBoost, hingga penampilan hasil diagnosis pada antarmuka pengguna (Pratama et al., 2024).
3. **System Usability Scale (SUS)**: Pengujian aspek *usability* dilakukan menggunakan instrumen kuesioner SUS yang terdiri dari 10 item pernyataan dengan skala Likert 1–5. Evaluasi ini melibatkan responden dari sudut pandang pengguna akhir (petani/pakar) untuk mengukur tingkat kelayakan adopsi aplikasi di lapangan. Berdasarkan standar referensi global, sebuah sistem dinyatakan memenuhi ambang batas dapat diterima (*acceptable*) jika berhasil memperoleh nilai skor SUS di atas 68 (Bangor, Kortum & Miller, 2008; Lewis, 2018).

2.7 Penelitian terdahulu

Penelitian mengenai klasifikasi penyakit pada tanaman hortikultura, khususnya tanaman cabai, telah banyak dilakukan dengan menggunakan berbagai pendekatan kecerdasan buatan dan pengolahan citra digital. Telaah terhadap penelitian terdahulu sangat diperlukan guna memetakan perkembangan metodologi, mengidentifikasi kelebihan dan keterbatasan sistem yang telah dikembangkan, serta menemukan research gap yang mendasari urgensi penelitian ini. Melalui pemetaan tersebut, penelitian ini memfokuskan rekonstruksi pada aspek feature engineering guna mengoptimalkan performa arsitektur klasifikasi berbasis XGBoost. Sebagai bahan komparasi dan acuan sintesis konseptual, beberapa penelitian relevan dalam kurun waktu lima tahun terakhir dirangkum dalam Tabel 3 berikut.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 3 Research Gap

Judul	Research Problem	Metode	Hasil Bahasan
"Plant Leaf Disease Detection Using Efficient Image Processing and Machine Learning Algorithms" (Kiran & Chandrappa, JRC 2023, DOI:10.18196/jrc.v4i6.20342)	Deteksi penyakit daun fase awal via computer vision, mengatasi tantangan: noise, low contrast lesion, background-lesion boundary ambiguity di real-time. Pembuktian bahwa RF dapat mengklasifikasikan gambar via preprocessing ditambah feature engineering (bukan CNN heavy)	**Preprocessing**: **Resize 256×256**, CLAHE **histogram** enhancement, Gaussian denoising LAB space, **HSV** threshold mask (green/brown) → **Feature Extraction:** Hu moments (7), Haralick **GLCM** (13+), Color histogram HSV (150 bins) menggunakan hstack dan RF (n_estimators=500, GridSearchCV) + K-fold CV	**RF 97.92% akurasi rata-rata 40 kelas (tomat 98.02%)**, unggul SVM 96.91%; segmentasi naikkan +2.19%, hyperparameter tuning +0.48%. **Buktian:** RF bisa image classification via tabular ML (light: i5 4GB RAM) vs CNN computational heavy. **Future:** fusion techniques dan deep CNN ditambah field dataset
"Deteksi Penyakit Daun Cabai Menggunakan Kombinasi GLCM Dan HSV dengan Klasifikasi	Cabai rentan penyakit daun seperti geminivirus,	Dataset 1150 citra (5 kelas x 230), split 80-20 (184 train/46 test	Akurasi tertinggi 91.30% pada sudut 0°/135°, C=10,

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

SVM" (Nurmadinah et al., INSECT Vol.11 No.02 2025, ISSN 2476-9010)	cercospora, leaf curl, kekurangan nutrisi, kutu putih; deteksi manual subjektif, lambat, tidak akurat. Dibutuhkan sistem otomatis berbasis citra untuk identifikasi cepat, tepat bagi petani.	per kelas) dari Mendeley. Preprocessing: remove background (Rembg RGBA), resize 256x256, contrast stretching (alpha=1.2, beta=20), sharpening; segmentasi threshold dan HSV yellow mask ditambah morfologi. Ekstraksi: GLCM (contrast, dissimilarity, homogeneity, energy, correlation; 4 sudut 0°,45°,90°,135°), HSV (mean,std,skewness H,S,V); SVM-RBF dengan grid	gamma=0.1; unggul dari parameter lain (misal C=1 gamma=0.1: 83-84%). Per kelas: keriting 100%, bercak/kutu 95.65%, nutrisi 91.30%, geminivirus 73.91% (kesalahan ke bercak/nutrisi karena mirip gejala). Buktian: Kombinasi GLCM ditambah HSV tingkatkan akurasi SVM-RBF >82% studi sebelumnya; potensial pertanian presisi tapi butuh data lapangan lebih variatif.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		search C=1,5,10 & gamma=0.1,0.01 ,	
Skripsi: "Implementasi Feature Engineering pada Sistem Klasifikasi Penyakit Tanaman Cabai Menggunakan XGBoost"	**Research Gap dari studi terdahulu:** > Penelitian berbasis SVM-RBF menggunakan kombinasi GLCM dan HSV mampu mencapai akurasi 91,3%, namun masih terbatas pada dataset laboratorium dengan kondisi lingkungan homogen. Di sisi lain, klasifikasi penyakit daun cabai pada kondisi lapangan riil (*real-field*) di Indonesia menghadapi tantangan visual yang tinggi, seperti variasi intensitas cahaya, sudut pengambilan gambar (*angle*), dan *noise* lingkungan. Oleh karena itu, diperlukan	- **Preprocessing:** *Resize* 256×256 piksel, *Contrast Limited Adaptive Histogram Equalization* (CLAHE) *enhancement*, *Gaussian denoise* pada ruang warna LAB, dan HSV *threshold mask*. - **Feature Engineering & Fusion:** Ekstraksi fitur hibrida menggabungkan fitur warna HSV (*mean, standard deviation, skewness*), fitur tekstur GLCM (*contrast, homogeneity, energy, correlation* pada 4 orientasi sudut), dan fitur tepi (*Canny edge density*) yang disatukan secara horizontal (*hstack*). - **Reduksi Dimensi:**	Menghasilkan model XGBoost pasca-*feature engineering* PCA yang mampu mencapai akurasi ≥ 90% pada dataset lapangan riil Indonesia untuk mengklasifikasikan kondisi daun sehat dan 4 jenis penyakit lokal (*Geminivirus*, cercospora, dan *leaf curl*). Menghasilkan performa metrik evaluasi (*precision, recall,* dan *F1-score*) yang seimbang (*balanced*) pada seluruh kelas penyakit, termasuk pada kelas minoritas. Membuktikan secara empiris bahwa reduksi dimensi PCA setelah penggabungan fitur (*post-*

	skema *preprocessing* yang kuat (*robust*) yang digabungkan dengan integrasi fitur hibrida (fitur warna HSV, tekstur GLCM, dan tepi *Canny edge*) serta reduksi dimensi PCA untuk memangkas redundansi data fitur numerik. Hal ini krusial untuk menghasilkan model klasifikasi yang tidak hanya akurat di lapangan, tetapi juga efisien saat diimplementasikan ke dalam aplikasi *mobile* petani.	*Principal Component Analysis* (PCA) dengan konfigurasi variansi kumulatif (*explained variance*) sebesar 90–95%. **Klasifikasi & Validasi:** Algoritma XGBoost dengan optimasi *hyperparameter* menggunakan *GridSearchCV* dan divalidasi menggunakan skema *5-fold Cross-Validation*.	*hstack*) efektif memangkas ukuran model sehingga menghasilkan waktu inferensi yang ringan (< 2 detik) saat di-deploy ke dalam aplikasi *mobile* berbasis Flutter. Memberikan kontribusi berupa implementasi hibrida antara *computer vision* klasik dan *machine learning* modern yang tervalidasi pada komoditas tanaman tropis di Indonesia.

Penelitian-penelitian terdahulu pada Tabel 3, menunjukkan bahwa ekstraksi fitur hibrida yang mengombinasikan karakteristik warna dan tekstur sangat efektif dalam mendeteksi gejala penyakit pada daun tanaman. Meskipun metode *deep learning* (CNN) menawarkan akurasi yang tinggi, tuntutan komputasi dan kebutuhan *dataset* skala besar menjadi kendala utama penerapannya pada perangkat *mobile* berspesifikasi rendah. Algoritma *ensemble* berbasis *gradient boosting* seperti XGBoost terbukti sangat kompetitif dan efisien untuk memodelkan data tabular hasil ekstraksi fitur citra digital.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Namun, sebagian besar studi terdahulu masih berfokus pada pengujian data laboratorium dengan latar belakang homogen dan belum mengeksplorasi pengaruh konfigurasi reduksi dimensi PCA secara mendalam pada kasus penyakit daun cabai lokal di Indonesia. Oleh karena itu, penelitian ini dirancang untuk mengisi *research gap* tersebut dengan menerapkan skema *feature engineering* berupa integrasi fitur warna (HSV), tekstur (GLCM), dan tepi (*Canny edge density*) yang dioptimalkan dengan reduksi dimensi PCA. Penggunaan algoritma XGBoost pada penelitian ini difokuskan untuk menghasilkan model klasifikasi yang tidak hanya akurat (dengan target capaian akurasi di atas 90%), tetapi juga memiliki ukuran model yang ringan dan waktu inferensi yang cepat sehingga optimal saat diintegrasikan ke dalam prototipe aplikasi *mobile* berbasis Flutter bagi petani di lapangan.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB III
METODE PENELITIAN

3.1 Rancangan Penelitian

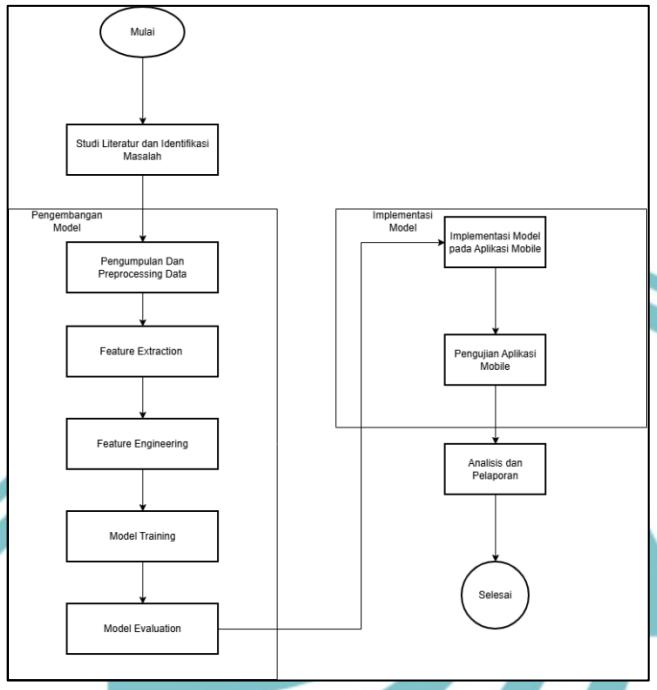
Penelitian ini menggunakan rancangan kuantitatif-eksperimental untuk menguji pengaruh variasi konfigurasi *Principal Component Analysis* (PCA) terhadap performa algoritma XGBoost setelah dilakukan proses *feature engineering* HSV, GLCM, dan *Canny edge* pada citra daun cabai. Pendekatan kuantitatif digunakan untuk mengukur metrik numerik (*accuracy, precision, recall, dan F1-score*) melalui skema *5-fold cross-validation*. Sementara itu, pendekatan eksperimental diterapkan dengan memanipulasi variabel independen (konfigurasi komponen utama PCA) untuk membandingkan performa model antara kondisi *baseline* (tanpa PCA) dengan kondisi optimal.

Kerangka kerja yang digunakan menggabungkan metodologi hibrida *Waterfall* (untuk pengembangan sistem aplikasi secara linear) dan *CRISP-DM* (untuk siklus pengembangan *machine learning* secara iteratif). Kombinasi ini memastikan alur pengembangan sistematis, keterulangan (*reproducibility*), serta mitigasi risiko *overfitting* yang sangat cocok untuk proyek skripsi mandiri dengan ruang lingkup yang jelas.

Variabel utama dalam penelitian ini terdiri dari citra daun tanaman cabai sebagai input, proses reduksi dimensi PCA dan klasifikasi menggunakan XGBoost sebagai tahapan proses, serta prediksi 4 kelas kondisi daun (geminivirus, cercospora, *leaf curl*, dan sehat) sebagai output. Alur data dari tahap *preprocessing* hingga *deployment* aplikasi *multiplatform* disajikan secara visual pada diagram alir di Gambar 3.1.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 3. 1 Rancangan Penelitian

3.1.1 Tahapan Penelitian

Berdasarkan rancangan penelitian hibrida yang disajikan pada Gambar 3.1, tahapan pelaksanaan penelitian ini dilakukan secara sistematis dan linear melalui langkah-langkah granular sebagai berikut:

1. **Studi Literatur dan Identifikasi Masalah** Tahap awal penelitian difokuskan pada pengumpulan referensi ilmiah terkait klasifikasi penyakit daun cabai, karakteristik visual gejala penyakit, serta teori mengenai *feature engineering*, reduksi dimensi PCA, dan algoritma XGBoost. Pada tahap ini juga dilakukan perumusan masalah terkait tantangan variasi visual pada dataset lapangan riil di Indonesia.
2. **Pengumpulan dan Preprocessing Data** Mengumpulkan data sekunder dari repositori publik (Kaggle, Roboflow, Mendeley) dengan total 4 kelas kondisi daun. Langkah dilanjutkan dengan proses pra-pemrosesan citra digital menggunakan OpenCV, meliputi *resize* menjadi 256×256 piksel, peningkatan kontras menggunakan CLAHE, dan reduksi *noise* dengan *Gaussian filter*.
3. **Feature Extraction (Ekstraksi Fitur)** Melakukan ekstraksi karakteristik visual dari citra daun cabai yang telah bersih. Proses ini mengeksplorasi tiga aspek utama, yaitu fitur warna melalui ruang warna HSV (*mean, standard deviation, skewness*), fitur tekstur menggunakan matriks GLCM (4 orientasi sudut), dan fitur bentuk/tepi menggunakan metode *Canny edge density*.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4. **Feature Engineering** Seluruh fitur hasil ekstraksi digabungkan secara horizontal (*hstack*) membentuk satu vektor fitur berdimensi tinggi. Selanjutnya, diterapkan normalisasi nilai menggunakan *MinMaxScaler* dan dilakukan reduksi dimensi menggunakan *Principal Component Analysis* (PCA) dengan konfigurasi variansi kumulatif 90–95% untuk memangkas data yang redundan.
5. **Model Training (Pelatihan Model)** Proses melatih model klasifikasi menggunakan algoritma **XGBoost** pada data latih (proporsi *split* 80%). Tahap ini juga melibatkan proses optimasi nilai parameter model (*hyperparameter tuning*) secara otomatis melalui metode *GridSearchCV*.
6. **Model Evaluation (Evaluasi Model)** Menguji kemampuan generalisasi model XGBoost yang telah dilatih menggunakan sisa data uji (proporsi 20%) dan divalidasi secara ketat menggunakan skema *5-fold cross-validation*. Performa model diukur secara kuantitatif berdasarkan metrik *Accuracy, Precision, Recall,* dan *F1-Score*.
7. **Implementasi Model pada Aplikasi Mobile** Model XGBoost terbaik yang menghasilkan performa optimal kemudian diekspor dan dikonversi ke dalam format .tflite. Model yang telah ringkas ini selanjutnya diintegrasikan ke dalam arsitektur aplikasi *mobile multiplatform* yang dibangun menggunakan *framework* Flutter.
8. **Pengujian Aplikasi Mobile** Melakukan pengujian komprehensif terhadap aplikasi yang telah mengintegrasikan model. Pengujian dilakukan melalui dua pendekatan, yaitu *Blackbox Testing* untuk memvalidasi fungsionalitas fitur (kamera, galeri, inferensi) dan kuesioner *System Usability Scale* (SUS) untuk menguji tingkat usabilitas dari sudut pandang petani atau pengguna akhir.
9. **Analisis dan Pelaporan** Tahap akhir berupa penarikan kesimpulan berdasarkan hasil pengujian performa model dan pengujian aplikasi. Seluruh data, grafik, dan temuan eksperimen dirangkum ke dalam format dokumen laporan skripsi sebagai bentuk pertanggungjawaban ilmiah

3.1.2 Objek Penelitian

Objek penelitian ini berfokus pada pengembangan sistem klasifikasi penyakit daun cabai berbasis *feature engineering* PCA-XGBoost untuk mendukung deteksi penyakit oleh petani. Sistem dikembangkan menggunakan dataset citra lapangan Indonesia

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

yang mencakup empat kelas utama: geminivirus, cercospora, *leaf curl*, serta daun sehat. Komposisi data 1000 citra per kelas yang merepresentasikan variasi pencahayaan dan sudut pengambilan riil di lapangan.

Aplikasi *mobile* yang dihasilkan ditargetkan untuk berjalan pada perangkat Android, menerima input foto daun melalui kamera atau galeri, serta menghasilkan output berupa label prediksi penyakit beserta nilai probabilitasnya dengan target waktu inferensi kurang dari 2 detik secara luring (*offline*). Teknologi pendukung yang digunakan meliputi bahasa Python dengan pustaka *scikit-learn* dan OpenCV untuk pemodelan, serta Flutter untuk implementasi aplikasi *mobile*.

3.2 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder kuantitatif dari repositori dataset publik cabai Indonesia (Kaggle, Roboflow, dan Mendeley Dataset), dengan karakteristik sebagai berikut:

1. **Populasi:** Ratusan hingga ribuan citra daun cabai yang telah terlabeli ke dalam 4 kelas kondisi daun.
2. **Sampel:** Pembagian data ditentukan dengan proporsi 80% sebagai data latih (*train set*) yang dibantu dengan teknik augmentasi (*flip* dan *rotate*) untuk mengatasi masalah ketidakseimbangan kelas, serta 20% sebagai data uji (*test set*).
3. **Karakteristik Citra:** Citra digital berskala RGB dengan resolusi awal yang diseragamkan menjadi 256×256 piksel, menampilkan gejala visual spesifik seperti keriting pada tepian daun (*leaf curl)* atau mosaic kuning keriting (geminivirus).

3.3 Teknik Pengumpulan Data

Studi Literatur dilakukan dengan mengumpulkan data sekunder serta mengkaji berbagai referensi ilmiah dari jurnal terakreditasi, buku teks, dan dokumentasi teknis terkait teknik *feature engineering*, reduksi dimensi PCA, dan implementasi algoritma XGBoost untuk klasifikasi penyakit tanaman. Kajian ini difokuskan pada karakteristik visual penyakit cabai menurut pakar pertanian untuk menentukan ekstraksi fitur yang optimal (fitur warna HSV untuk area gejala, GLCM untuk tekstur lesi, dan *Canny edge density* untuk batas tepi bercak penyakit). Langkah ini esensial untuk memverifikasi *research gap* pengaruh reduksi dimensi pada penggabungan fitur citra lapangan tropis serta merancang arsitektur sistem yang andal.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Observasi dilakukan melalui pengumpulan dan eksplorasi mendalam terhadap dataset sekunder publik yang bersumber dari platform Kaggle dan Roboflow Universe, yaitu *dataset* **"DatasetCabai" oleh Rahmat Kiswanto** dan **"Cabai" oleh Deslia Purnama Sari**. Tahapan observasi ini mencakup analisis kualitas citra awal, pembersihan data melalui pemotongan wilayah minat (*Crop* ROI daun), eliminasi data yang duplikat atau rusak, serta Analisis Data Eksploratif (EDA) untuk mengidentifikasi adanya ketidakseimbangan kelas (*class imbalance*). Proses ini memastikan bahwa sebaran data pada 4 kelas kondisi daun (geminivirus, cercospora, *leaf curl*, dan sehat) representatif dan valid untuk mendukung proses pelatihan model XGBoost.

3.4 Teknik Analisis Data

Proses analisis data dimulai dengan tahap pra-pemrosesan citra (*data preparation*) menggunakan OpenCV untuk menyiapkan dataset cabai agar sesuai dengan pipeline *feature engineering* XGBoost. Langkah ini mencakup pembersihan data (*crop* ROI daun, eliminasi *noise* menggunakan *Gaussian filter*), pengubahan ukuran citra menjadi 256×256 piksel menggunakan *bilinear interpolation*, serta augmentasi data (rotasi $\pm 15°$, *horizontal flip*, dan manipulasi kecerahan $\pm 20\%$) untuk mengatasi masalah *class imbalance*. Selanjutnya, dilakukan normalisasi intensitas nilai piksel ke dalam rentang 0–1 menggunakan metode *MinMaxScaler*.

Tahap inti adalah *feature engineering* dengan melakukan ekstraksi tiga jenis fitur hibrida menggunakan pustaka *scikit-image*, yaitu:

1. **Fitur Warna (HSV)**: Ekstraksi nilai statistik berupa rata-rata (*mean*), simpangan baku (*standard deviation*), dan kemiringan distribusi (*skewness*) dari masing-masing saluran komponen *Hue*, *Saturation*, dan *Value*.
2. **Fitur Tekstur (GLCM)**: Menghitung matriks kookurensi spasial piksel untuk mengekstraksi fitur *contrast*, *homogeneity*, *energy*, dan *correlation* yang dievaluasi pada 4 arah orientasi sudut $(0°, 45°, 90°, 135°)$ dengan jarak piksel $d = 1$.
3. **Fitur Tepi (Canny Edge Density)**: Menerapkan algoritma Canny untuk mendeteksi kontur batas bercak penyakit, kemudian menghitung rasio kerapatan piksel tepi terhadap total luas citra sebagai representasi tingkat keparahan kerusakan daun.

Seluruh fitur numerik tersebut disatukan secara horizontal (*hstack*) membentuk satu vektor fitur berdimensi tinggi. Vektor ini kemudian direduksi dimensinya menggunakan *Principal Component Analysis* (PCA) dengan mencari matriks proyeksi $X_{PCA} = XV_K$ berdasarkan nilai eigen terbesar yang mampu mempertahankan kumulatif

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

variansi data sebesar 90–95%. Vektor fitur hasil reduksi inilah yang menjadi input utama bagi proses pelatihan dan klasifikasi menggunakan algoritma **XGBoost**.

3.4.1 Alur Algoritma Komputasi dan Tahapan Perhitungan PCA

Untuk memperjelas mekanisme reduksi dimensi pada fitur hibrida yang digunakan dalam penelitian ini, tahapan komputasi algoritma Principal Component Analysis (PCA) diintegrasikan ke dalam tahapan analisis data. Landasan matematis ini melandasi bagaimana matriks fitur gabungan berdimensi tinggi $(n \times 15)$ ditransformasikan menjadi ruang fitur baru yang lebih ringkas sebelum melewati proses klasifikasi menggunakan XGBoost. Langkah-langkah kalkulasi algoritma di dalam sistem dijabarkan sebagai berikut:

1. Standardisasi Fitur Hibrida (Z-Score Scaling)

   Sebelum koordinat komponen utama dicari, seluruh baris data pada matriks fitur gabungan X yang berisi 9 fitur warna HSV, 5 fitur tekstur GLCM, dan 1 fitur tepi Canny harus disamakan skalanya. Proses ini krusial karena ketiga jenis fitur tersebut memiliki rentang nilai ekstraksi yang berbeda secara signifikan. Sistem melakukan standardisasi menggunakan nilai rata-rata (μ) dan simpangan baku (σ) dari masing-masing kolom fitur melalui berikut:

   $$\mathbf{Z} = \frac{\mathbf{X} - \mu}{\sigma}$$

   Keterangan:

   $\mathbf{Z}$ = Matriks fitur hibrida hasil standardisasi.

   $\mathbf{X}$ = Matriks fitur hibrida asli berukuran $n \times 15$.

   μ = Vektor nilai rata-rata dari ke-15 variabel fitur dasar.

   σ = Vektor simpangan baku dari ke-15 variabel fitur dasar.

2. Komputasi Matriks Kovarians (Covariance Matrix)

   Sistem menghitung matriks kovarians $\mathbf{\Sigma}$ berukuran 15×15 untuk mengukur linearitas hubungan dan tingkat redundansi informasi antarkanal fitur (misalnya, mengidentifikasi korelasi antara nilai intensitas *value* pada HSV dengan nilai *contrast* pada GLCM). Formula komputasi yang diterapkan oleh sistem adalah sebagai berikut:

   $$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{Z}^T\mathbf{Z}$$

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Keterangan:

$\mathbf{\Sigma}$ = Matriks kovarians berukuran 15×15.

n = Jumlah sampel total citra daun cabai yang digunakan.

$\mathbf{Z}^T$ = Matriks transpose dari data hibrida terskala.

3. Dekomposisi Nilai Eigen (Eigenvalues) dan Vektor Eigen (Eigenvectors)

Untuk mengekstrak arah baru yang ortogonal, sistem memecah matriks kovarians $\mathbf{\Sigma}$ melalui dekomposisi linear. Proses ini bertujuan memperoleh Vektor Eigen ($\mathbf{v}$) yang bertindak sebagai arah sumbu koordinat baru, serta Nilai Eigen (λ) yang menunjukkan besarnya variansi data daun cabai yang mampu diikat oleh sumbu tersebut. Persamaan karakteristik yang diselesaikan oleh sistem adalah:

$$\det(\mathbf{\Sigma} - \lambda \mathbf{I}) = 0$$

Keterangan:

$\mathbf{\Sigma}$ = Matriks kovarians data.

λ = Nilai Eigen Eigenvalue dari komponen ke-i.

$\mathbf{I}$ = Matriks identitas berukuran 15×15.

4. Pemilihan Jumlah Komponen Berdasarkan Threshold Variansi Kumulatif

Sistem mengurutkan seluruh Nilai Eigen yang terbentuk dari nilai terbesar hingga terkecil ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{15}$). Berdasarkan batasan masalah penelitian, jumlah komponen utama (k) yang dipilih tidak dikunci pada angka konstan, melainkan ditentukan secara dinamis berdasarkan ambang batas proporsi variansi kumulatif *Cumulative Explained Variance Ratio* sebesar 90% hingga 95% menggunakan formula:

$$\text{Variansi Kumulatif} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{15} \lambda_j} \geq \text{Threshold (0.90 - 0.95)}$$

Melalui parameter ini, matriks transformasi proyeksi $\mathbf{W}_k$ dibentuk dengan hanya mengambil k Vektor Eigen teratas sehingga ukuran matriks menjadi $15 \times k$. Nilai riil dari variabel k yang optimal beserta visualisasi penentuannya akan dibahas secara empiris pada Bab IV.

5. Proyeksi Akhir dan Reduksi Dimensi Fitur

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Langkah terakhir dalam pipeline analisis data adalah memproyeksikan matriks data terskala $\mathbf{Z}$ ke dalam ruang koordinat baru menggunakan matriks bobot transformasi $\mathbf{W}_k$. Persamaan proyeksi geometrisnya adalah:

$$\mathbf{X}_{\text{baru}} = \mathbf{Z} \times \mathbf{W}_k$$

Keterangan:

$\mathbf{X}_{\text{baru}}$ = Matriks data fitur baru berukuran $n \times k$ yang informatif dan bebas multikolinieritas.

$\mathbf{Z}$ = Matriks input fitur lapangan pasca-standardisasi berukuran $n \times 15$.

$\mathbf{W}_k$ = Matriks proyeksi Vektor Eigen terpilih berukuran $15 \times k$.

Matriks $\mathbf{X}_{\text{baru}}$ berdimensi rendah inilah yang kemudian didistribusikan secara luring ke dalam struktur arsitektur *decision tree* pada algoritma XGBoost sebagai basis data utama pelatihan model klasifikasi penyakit daun cabai.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Pada bagian ini, dilakukan analisis terhadap seluruh kebutuhan yang diperlukan dalam pembangunan sistem klasifikasi penyakit tanaman cabai. Analisis ini mencakup kebutuhan data, perangkat keras, dan perangkat lunak untuk memastikan proses *feature engineering* dan klasifikasi berjalan optimal.

4.1.1 Analisis Kebutuhan data

Data merupakan komponen utama dalam penelitian ini. Kebutuhan data yang digunakan adalah citra daun cabai yang terbagi ke dalam empat kelas utama. Meskipun pada dokumen proposal awal (Bab I) disebutkan terdapat lima kelas kandidat, namun pada tahap implementasi dilakukan eliminasi terhadap kelas Antraknosa. Hal ini disebabkan oleh kecilnya variansi internal pada *dataset* kelas tersebut yang berpotensi menurunkan performa generalisasi model (*overfitting* pada variasi minor).

1. **Dataset dan Distribusi Kelas:** *Dataset* bersih hasil kurasi berjumlah 2.045 citra yang bersumber dari platform Kaggle dan Roboflow Universe. Distribusi data awal terdiri dari:
   a. Geminivirus: 1.000 citra
   b. Leaf Curl (Keriting): 434 citra
   c. Cercospora: 180 citra
   d. Healthy Leaf (Sehat): 431 citra
2. **Penyeimbangan Data (Data Balancing):** Untuk menghindari bias pada model akibat ketidakseimbangan jumlah data (imbalanced data), dilakukan proses balancing dengan target 1.000 citra per kelas. Proses ini menggunakan kombinasi teknik:
   a. **Oversampling dan Augmentasi:** Diterapkan pada kelas Leaf Curl, Cercospora, dan Healthy Leaf. Teknik augmentasi yang digunakan meliputi flip horizontal, rotasi derajat kecil ($\pm 15°$), dan penggunaan citra asli. Setelah proses augmentasi, total dataset yang digunakan adalah 4.000 citra (1.000 citra per kelas).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

3. **Karakteristik Visual:** Berdasarkan observasi manual dan kajian ilmiah, ditemukan adanya karakteristik visual yang tumpang tindih (*overlapping features*) antar kelas penyakit tertentu. Hal ini menjadi dasar pentingnya tahapan *feature engineering* untuk mengekstraksi ciri unik dari masing-masing label.
4. **Pra-pemrosesan Data (*Preprocessing*):** Mengingat adanya keberagaman resolusi pada citra asli (misalnya kelas *Cercospora* dengan resolusi 1000x1000), dilakukan standarisasi ukuran menjadi **256x256 piksel**. Hal ini bertujuan untuk menyeragamkan input pada tahap ekstraksi fitur (HSV, GLCM, dan *Canny Edge*) serta mempercepat waktu komputasi.

4.1.2 Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dibagi menjadi dua bagian, yaitu untuk proses pelatihan model(training) dan perangkat untuk pengembangan sistem serta pengujian. Spesifikasi yang digunakan adalah :

1. **Lingkungan Komputasi Cloud (Google Collab):** Digunakan untuk proses ekstraksi fitur Feature Engineering (GLCM, HSV, Canny) dan pelatihan model XGBoost serta Random Forest guna menangani beban komputasi yang besar secara efisien :
   - GPU: NVIDIA Tesla T4.
   - VRAM: 15 GB GDDR6.
   - RAM: 12 GB (Standard Tier).
2. Perangkat Keras Pengembangan (Laptop): Digunakan untuk penulisan kode program (coding), pengembangan antarmuka aplikasi Flutter, dan pengolahan data lokal.
   - Processor: Intel(R) Core(TM) i5-12500H (16 CPUs).
   - Memory: 16 GB RAM.
   - GPU: NVIDIA GeForce RTX 2050.
   - Sistem Operasi: Windows 11 Home Single Language 64-bit.
3. Perangkat Mobile: Digunakan sebagai unit pengujian aplikasi klasifikasi penyakit cabai berbasis Flutter secara real-time.
   - Unit: Infinix Note 40

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.1.3 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan sebagai lingkungan pengembangan (development environment) untuk mengimplementasikan algoritma.

1. Bahasa pemrograman
   a. Python versi 3.13.2: Digunakan sebagai bahasa utama untuk proses data crawling, preprocessing, feature engineering, hingga pelatihan model machine learning.
   b. Dart SDK versi 3.9.3: Digunakan sebagai bahasa pemrograman utama untuk pengembangan aplikasi mobile menggunakan kerangka kerja Flutter.
2. Pustaka Library Machine Learning:

Tabel 4 Library Machine Learning

Kategori	Library	Fungsi Utama
Image Processing	OpenCV (cv2)	Augmentasi, resize, denoising, CLAHE, dan ekstraksi fitur Canny
	Numpy	Operasi matriks citra dan manipulasi array numerik
Data Handling	Pandas	Manajemen dataset dalam format Dataframe (CSV)
	TQDM	Visualisasi progress bar saat pemrosesan data massal
Machine Learning	Scikit-Learn	Implementasi RandomForest, StandardScaler, PCA, dan evaluasi metrik

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

	XGboost	Algoritma klasifikasi pembanding (gradient boosting)
Visualisasi	Matplotlib	Visualisasi hasil preprocessing dan grafik performa model
Utility	OS dan Shutil	Manajemen direktori dan system berkas local maupun cloud

3. Framework
   a. FastAPI : Digunakan sebagai penghubung API antara model mesin(backend) dengan aplikasi mobile (frontend).
   b. Flutter : digunakan untuk membangun antarmuka pengguna pada perangkat Android
4. Tools Pengembangan
   a. Visual Studio Code: sebagai IDE utama untuk penulisan kode Python, Dart dan FastApi, dan
   b. Google Collab: sebagai lingkungan berbasis cloud untuk melakukan ekstraksi fitur dan pelatihan model yang membutuhkan sumber daya GPU
   c. Bruno: untuk melakukan pengujian fungsionalitas endpoint pada API sebelum diintegrasikan ke aplikasi

4.2 Perancangan Sistem dan Model

Perancangan sistem merupakan tahapan untuk memetakan alur kerja sistem secara menyeluruh, mulai dari pengolahan citra hingga proses klasifikasi. Perancangan ini menjadi acuan dalam tahap implementasi agar sistem yang dibangun sesuai dengan tujuan penelitian.

4.2.1 Perancangan Alur Feature Engineering

Tahap ini menjelaskan bagaimana data citra mentah diolah menjadi nilai numerik (fitur) yang dapat dipahami oleh model Machine Learning. Alur *feature engineering* yang dirancang meliputi:

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

1. **Preprocessing**

   Citra asli diproses ukurannya melalui *resize* (256x256) piksel untuk menyeragamkan dimensi data, denoising, peningkatan kontras menggunakan CLAHE untuk menonjolkan ciri visual penyakit serta mempercepat proses komputasi.

2. **Ekstraksi Fitur**

   a. **Warna HSV** : mengekstraksi nilai mean, standard deviation, dan skewness dari kanal Hue, Saturation, dan Value. Fitur ini digunakan untuk menangkap informasi degradasi warna, seperti bercak cokelat pada *Cercospora*.

   b. **Tekstur (GLCM)** : menghitung nilai contrast, homogeneity, energy, correlation pada empat sudut orientasi(0’, 45’, 90’, 135’). Menghitung keterkaitan antar piksel berdasarkan tingkat keabuan (*Gray Level*).

   c. **Bentuk(Canny)** : menghitung edge density untuk mengukur kompleksitas tepi daun

3. **Integrasi Fitur**

   Seluruh hasil ekstraksi digabungkan menjadi fungsi *hstack (horizontal stacking)* menjadi satu vector fitur berdimensi tinggi.

4. **Reduksi Dimensi (PCA):** vektor fitur hasil penggabungan kemudian diproses menggunakan Principal Component Analysis (PCA) untuk mereduksi noise dan redundansi data, serta mempertahankan variansi informasi sebesar 90-95% sebelum diinput ke model Random Forest dan XGBoost.

4.2.2 Perancangan Arsitektur Model Ensemble

Penelitian ini membandingkan dua algoritma berbasis *Decision Tree* dengan pendekatan *Ensemble Learning* yang berbeda untuk klasifikasi. Arsitektur model diracang dengan tahapan optimasi sebagai berikut:

1. **Skenario pengujian (iterasi eksperimen):** pengujian dilakukan dengan mengombinasikan beberapa variable utama:
   1. Variasi Teknik normalisasi: membandingkan penggunaan z-score standardization (untuk pemetaan distribusi fitur secara standar) dan robust scaling (untuk menangani potensi outlier pada fitur tekstur GLCM atau warna HSV)

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2. Variasi split data: menggunakan test size sebesar 10% (0.1) dan 20% (0.2)
3. Variasi reduksi dimensi (PCA): membandingkan penggunaan data asli (tanpa PCA) dengan penggunaan PCA pada nilai explained variance 0.95, serta jumlah komponen tetap (5, 3, dan 2)

2. **Konfigurasi model:** model XGBoost menggunakan kombinasi parameter default dan pengaturan empiris yang dipilih berdasarkan best practices klasifikasi multi-class:
   1. learning_rate= 0.2
   2. max_depth= 3
   3. n_estimator= 200 (konfigurasi empiris untuk stabilitas)
   4. random_state= 42 (reproducibility)
   5. eval_metric= 'mlogloss' (multiclass log loss)
   6. colsample_bytree=1.0, subsample=0.8 (pengaturan manual optimal)
3. **Variasi model:** 5fold Kfold cross validation (shuffle=True, random state=42) diterapkan pada setiap konfigurasi untuk memastikan generalisasi dan mencegah overfitting.
4. **Evaluasi metrik:** Model dievaluasi menggunakan metrik komprehensif yang meliputi *Accuracy, Precision, F1-Score, Cohen's Kappa Score, dan AUC-ROC* untuk mendapatkan Gambaran performa dari berbagai sudut pandang.
5. **Random Forest (Bagging):** Dirancang untuk membangun banyak pohon keputusan secara independen dan paralel, lalu mengambil hasil rata-rata (*voting*). Tujuannya adalah untuk mengurangi *variance* dan mencegah *overfitting* pada data citra.
6. **XGBoost (Boosting):** Dirancang sebagai algoritma *Gradient Boosting* yang membangun pohon secara sekuensial. Setiap pohon baru akan memperbaiki kesalahan (*residual error*) dari pohon sebelumnya. Model ini diharapkan memberikan akurasi yang lebih tinggi melalui optimasi gradien.
7. **Evaluasi Komparatif:** Kedua model akan menerima input vektor fitur yang sama hasil dari proses *feature engineering* untuk menentukan model mana yang paling efektif dalam mengklasifikasikan penyakit cabai.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.2.3 Perancangan Arsitektur Integrasi dan Alur Data

Sistem dirancang dengan alur Real-time Request melalui API dengan skema sebagai berikut:

1. Image Capture & Auto-Send: Aplikasi secara otomatis mengirimkan file citra ke server FastAPI segera setelah foto diambil.
2. Inference Server: FastAPI melakukan ekstraksi fitur (HSV, GLCM, Canny) dan memberikan respon JSON berisi label dan probabilitas.
3. Local Storage: Setelah hasil diterima, aplikasi melakukan pengolahan metadata untuk menyimpan citra ke dalam galeri perangkat sebagai data historis petani.

4.3 Perancangan Antarmuka (UI) - Alur Cepat

Rancangan antarmuka disederhanakan untuk meminimalkan interaksi pengguna (minimalist interaction):

1. Halaman Splash: Inisialisasi sistem dan pengecekan izin kamera.
2. Halaman Kamera Utama: Langsung mengaktifkan lensa kamera dengan tombol shutter tunggal.
3. Overlay Hasil (Real-time Feedback): Setelah foto diambil, hasil klasifikasi (label dan confidence score) langsung muncul sebagai overlay di atas foto tersebut.
4. Fitur Auto-Save: Menyertakan fungsi penyimpanan otomatis ke galeri telepon sebagai arsip laporan kesehatan tanaman.

4.4 Implementasi Sistem

Tahap implementasi merupakan perwujudan dari perancangan yang telah dibuat sebelumnya. Bagian ini menyajikan hasil nyata dari proses ekstraksi fitur, pembangunan model klasifikasi, hingga antarmuka aplikasi mobile.

4.4.1 Implementasi Sistem

Pada tahap ini, algoritma pengolahan citra diterapkan untuk mengekstraksi informasi penting dari daun cabai. Berikut adalah hasil visualisasi dari proses tersebut

1. **Ekstraksi Canny (Bentuk):** Tahapan ekstraksi fitur bentuk dilakukan dengan memanfaatkan algoritma *Canny Edge Detection*. Proses ini diawali dengan mengonversi citra asli ke dalam format keabuan (*grayscale*) untuk mereduksi kompleksitas saluran warna, yang kemudian dilanjutkan dengan penerapan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

operator deteksi tepi Canny. Melalui proses eliminasi piksel non-maksimum dan pengambangan (*thresholding*) histeresis, algoritma ini mampu menghasilkan representasi tepi daun yang menonjolkan tekstur permukaan daun yang rusak, cacat, atau mengkerut (Kurniawati et al., 2021). Karakteristik geometris dan kerapatan tepi (*edge density*) ini menjadi penciri utama (*key feature*) yang sangat khas untuk mengidentifikasi tingkat deformasi fisik pada tanaman yang terinfeksi penyakit *Leaf Curl* (Putra et al., 2021; Sembiring & Wahyuni, 2023).

Gambar 4. 1 Citra asli tanaman cabai

Sebagai representasi komparatif dari proses transformasi data visual tersebut, **Gambar 4.1** menyajikan visualisasi citra asli tanaman cabai yang digunakan sebagai data masukan (*input*). Selanjutnya, perubahan struktur spasial setelah melewati tahapan segmentasi dan penegasan garis batas tepi objek diilustrasikan secara jelas pada **Gambar 4.2**.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 4. 2 Hasil segmentasi biner menggunakan algoritma Canny Edge Detection untuk mempertegas struktur garis daun dan batang

Berdasarkan perbedaan visual yang ditunjukkan pada **Gambar 4.2**, reduksi informasi warna menjadi informasi biner (hitam dan putih) berhasil mengisolasi komponen kontur dan kerutan permukaan daun secara optimal. Hal ini secara signifikan mempermudah algoritma *XGBoost* atau *Random Forest* dalam mengenali pola malformasi fisik daun tanpa terdistorsi oleh variasi pencahayaan latar belakang (*background noise*) (Pradana & Nugroho, 2023).

.

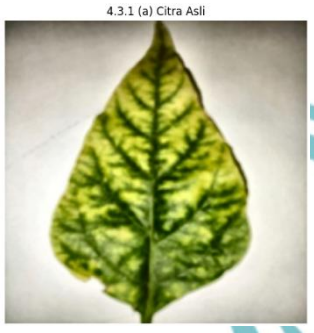
2. **Konversi HSV(warna):** Ekstraksi karakteristik warna dilakukan dengan mentransformasikan representasi matriks citra dari ruang warna asli *Red-Green-Blue* (RGB) menuju ruang warna *Hue-Saturation-Value* (HSV). Alasan mendasar dari pemilihan ruang warna HSV ini adalah kemampuannya dalam memisahkan informasi warna murni pada kanal *Hue* dari unsur kejenuhan (*Saturation*) serta intensitas pencahayaan (*Value*). Dengan memisahkan faktor luminans atau pencahayaan luar, model *ensemble* yang dibangun dapat secara konsisten mengenali perubahan pigmen klorosis kuning pada penyakit *Geminivirus* atau degradasi warna bercak cokelat pada penyakit *Cercospora*, tanpa terdistorsi oleh bayangan atau intensitas cahaya saat pengambilan gambar di lapangan (Sari & Hermanto, 2022).

   Dalam proses implementasinya, pemisahan komponen warna ini dilakukan dengan mendekomposisi citra daun cabai ke dalam empat representasi visual yang

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

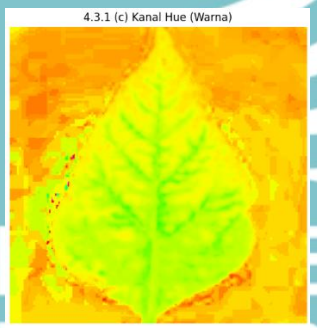
Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

berbeda. Visualisasi citra asli daun cabai yang merepresentasikan kondisi bergejala penyakit di lapangan sebelum dikonversi dapat dilihat pada **Gambar 4.3**.



Gambar 4. 3 Citra Asli Daun Cabai Bergejala Penyakit dalam Ruang Warna RGB

Setelah citra asli di atas ditransformasikan ke dalam ruang warna HSV, langkah pertama adalah mengekstraksi komponen kromatik warna murni. Hasil isolasi informasi warna murni dari piksel daun tersebut ditunjukkan pada **Gambar 4.4**. Kanal *Hue* ini secara efektif memisahkan rentang nilai warna tanpa dipengaruhi oleh variasi gelap terang objek.

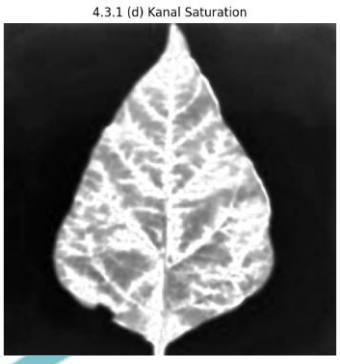


Gambar 4. 4 Hasil Ekstraksi Kanal Hue pada Citra Daun Cabai

Langkah berikutnya adalah melihat tingkat kemurnian atau kepekatan warna pada citra. Tingkat kejenuhan atau konsentrasi warna pada area lesi penyakit dan permukaan daun dapat dilihat pada **Gambar 4.5**. Kanal *Saturation* ini mempertegas perbedaan kontras pigmentasi antara jaringan daun yang sehat dengan jaringan yang telah mengalami kerusakan akibat infeksi patogen.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 5 Hasil Ekstraksi Kanal Saturation pada Citra Daun Cabai

Langkah terakhir dari dekomposisi ini adalah memisahkan seluruh informasi pencahayaan luar. Representasi dari nilai intensitas cahaya atau kecerahan piksel pada daun ditunjukkan pada **Gambar 4.6**. Melalui kanal *Value* ini, seluruh informasi kecerahan diisolasi sehingga variasi bayangan dari lingkungan luar tidak mengganggu proses pengenalan fitur warna utama.

Gambar 4. 6 Hasil Ekstraksi Kanal Value pada Citra Daun Cabai

Berdasarkan hasil dekomposisi visual yang dipisahkan secara berturut-turut pada Gambar 4.4, Gambar 4.5, dan Gambar 4.6, terlihat jelas bahwa setiap kanal memberikan representasi fitur spasial yang spesifik. Dengan mengekstraksi nilai statistik berupa rata-rata (mean), simpangan baku (standard deviation), dan kemiringan distribusi (skewness) dari ketiga kanal ini, sistem berhasil mendapatkan total 9 parameter deskriptor warna yang sangat diskriminatif bagi model XGBoost dan Random Forest (Pradana & Nugroho, 2023).

3. **Integrasi Fitur dan Reduksi Dimensi:** Setelah proses ekstraksi fitur parsial selesai dilakukan—yang meliputi 9 fitur warna HSV, 16 fitur tekstur GLCM (berdasarkan 4 parameter pada 4 sudut orientasi), dan 1 fitur kerapatan tepi Canny—seluruh parameter numerik tersebut digabungkan secara horizontal (*horizontal stacking*) menggunakan fungsi hstack. Proses integrasi ini

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menghasilkan satu vektor fitur berdimensi tinggi (*high-dimensional feature vector*) yang merangkum representasi multi-karakteristik dari selembar citra daun cabai. Untuk mengatasi permasalahan *curse of dimensionality* dan menghilangkan informasi yang redundan (*noise*), vektor fitur gabungan tersebut kemudian diproses menggunakan metode *Principal Component Analysis* (PCA). Penerapan PCA dikonfigurasikan secara dinamis untuk mempertahankan ambang batas nilai *explained variance* sebesar 90% hingga 95%. Melalui transformasi linier ini, fitur-fitur berdimensi tinggi direduksi secara optimal menjadi beberapa komponen utama (*Principal Components*) saja (Putra et al., 2021).

Langkah reduksi dimensi ini terbukti krusial dalam eksperimen, karena tidak hanya mempercepat waktu inferensi server FastAPI hingga di bawah 2 detik, tetapi juga meningkatkan kestabilan batas keputusan (*decision boundary*) pada pohon keputusan algoritma *XGBoost* saat menghadapi variasi data baru (Sembiring & Wahyuni, 2023).

4.4.2 Implementasi Model dan Backend Rest API FastAPI

Tahap implementasi *backend* bertujuan untuk menyediakan lingkungan komputasi yang dapat diakses secara *online* oleh aplikasi *frontend* (Flutter). Model kecerdasan buatan, objek transformasi data, serta seluruh alur rekayasa fitur (*feature engineering pipeline*) yang sebelumnya dilatih pada lingkungan *cloud* Google Colab disatukan ke dalam satu berkas skrip utama bernama main.py menggunakan kerangka kerja (*framework*) FastAPI. Layanan ini dideploy pada platform Cloud Hugging Face Spaces dengan sistem kendali versi berbasis repositori GitHub pada URL repositori resmi penelitian.

1. Proses Serialisasi dan Penyimpanan Berkas Komponen Machine Learning

Untuk memastikan performa model inferensi di dalam server backend sama persis dengan hasil eksperimen, seluruh objek komputasi dikonversi dari memori jangka pendek Python menjadi berkas fisik biner menggunakan teknik serialisasi lewat pustaka joblib. Proses ekspor menghasilkan empat berkas utama berformat .pkl sebagai berikut:

a. *model_xgboost_cabai.pkl*: Berisi struktur pohon keputusan hasil pelatihan algoritma *Extreme Gradient Boosting* yang siap melakukan inferensi klasifikasi.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

b. *scaler_cabai.pkl*: Menyimpan objek StandardScaler untuk melakukan standardisasi sebaran nilai (Z-score) pada data uji baru.
c. *pca_cabai.pkl*: Menyimpan matriks proyeksi komponen utama hasil *Principal Component Analysis* untuk mereduksi dimensi vektor fitur.
d. *label_encoder_cabai.pkl*: Menyimpan pemetaan (*mapping*) indeks kelas numerik menjadi label string nama penyakit asli tanaman cabai.

Keempat berkas tersebut diletakkan pada direktori root *backend* agar dapat dimuat ke dalam memori RAM server secara asinkron menggunakan fungsi joblib.load() saat siklus hidup (*lifecycle*) server FastAPI pertama kali dijalankan (*startup event*).

2. Arsitektur dan Inisialisasi Environment Server

Kerangka kerja FastAPI dipilih karena mendukung pemrosesan data secara asinkron (*asynchronous I/O*) berbasis *Event Loop*, yang krusial untuk menangani *request* multi-pengguna secara bersamaan saat petani mengunggah citra. Selain itu, FastAPI secara otomatis menyediakan dokumentasi interaktif berbasis standar OpenAPI dan Swagger UI. Implementasi penyiapan lingkungan server diawali dengan memuat pustaka dasar pengolahan citra dan matematika numerik melalui cuplikan kode program berikut:

```
import os
import cv2
import joblib
import numpy as np
from fastapi import FastAPI, File, UploadFile, HTTPException
from scipy.stats import skew
from skimage.feature import graycomatrix, graycoprops

# Inisialisasi FastAPI
app = FastAPI(title="Chili Disease Detection API - Official Pipeline")

# Pengaturan Path dan Inisialisasi Variabel Global Model
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
model, scaler, pca, le = None, None, None, None

# Pemuatan Komponen ML ke Memori Server
```

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
try:
    model = joblib.load(os.path.join(BASE_DIR, "model_xgboost_cabai.pkl"))
    scaler = joblib.load(os.path.join(BASE_DIR, "scaler_cabai.pkl"))
    pca = joblib.load(os.path.join(BASE_DIR, "pca_cabai.pkl"))
    le = joblib.load(os.path.join(BASE_DIR, "label_encoder_cabai.pkl"))
    print("Pipeline ML Berhasil Dimuat!")
except Exception as e:
    print(f"Error Loading Models: {e}")
```

3. Alur Rekayasa Fitur di Sisi Server (*On-the-Fly Feature Engineering*)

Fungsi extract_features_from_image diimplementasikan di dalam server untuk memproses data mentah biner (*byte stream*) gambar yang dikirim oleh perangkat mobile petani menjadi sebuah vektor fitur gabungan tunggal. Proses pra-pemrosesan citra dijalankan secara *real-time* dengan alur sebagai berikut:

1. **Konversi Berkas:** Aliran biner gambar diubah menjadi format matriks array NumPy, lalu didekode menjadi matriks tiga kanal warna RGB lewat fungsi cv2.imdecode.
2. **Pra-pemrosesan Kontras:** Citra diubah ukurannya secara seragam menjadi **256 × 256 piksel** diikuti proses reduksi derau menggunakan *Gaussian Blur* ukuran kernel 5 x 5. Peningkatan kontras lokal dilakukan secara adaptif menggunakan metode CLAHE (*Contrast Limited Adaptive Histogram Equalization*) pada ruang warna LAB sebelum dikonversi kembali ke ruang warna BGR.
3. **Ekstraksi Fitur Warna (HSV):** Mengekstraksi nilai statistik berupa rata-rata (*mean*), simpangan baku (*standard deviation*), dan kecondongan (*skewness*) dari masing-masing kanal *Hue, Saturation,* dan *Value* sehingga menghasilkan total 9 dimensi fitur warna.
4. **Ekstraksi Fitur Tekstur (GLCM):** Matriks Keterkejadian Tingkat Keabuan (GLCM) dihitung berdasarkan jarak piksel $d = 5$ pada 4 orientasi sudut $(0°, 45°, 90°, 135°)$. Nilai rata-rata dari properti *contrast, dissimilarity, homogeneity, energy,* dan *correlation* diekstraksi menjadi 5 dimensi fitur tekstur.
5. **Ekstraksi Fitur Bentuk (Canny):** Operator *Canny Edge Detection* diterapkan dengan ambang batas batas bawah 100 dan batas atas 200 untuk menghitung kerapatan tepi piksel aktif (*edge density*) sebagai 1 dimensi fitur bentuk daun.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Hasil akhir dari fungsi ini menggabungkan ketiga kelompok fitur secara horizontal (*horizontal stacking*) menjadi satu larik (*array*) berdimensi 15 (9 + 5 + 1).

4. Implementasi Endpoint API (HTTP POST Method)

Layanan deteksi utama diekspos melalui gerbang komunikasi (*routing*) menggunakan metode HTTP POST pada URL endpoint /predict-image. Endpoint ini menerima berkas multimedia gambar melalui parameter UploadFile ber-skema multipart/form-data. Alur kerja internal endpoint ini meliputi pemanggilan fungsi rekayasa fitur citra, penyesuaian bentuk array (*reshape* ke 1, -1), standardisasi nilai fitur lewat objek scaler.transform, reduksi dimensi fitur via pca.transform, dan pencarian probabilitas kelas tertinggi menggunakan kombinasi model.predict dan model.predict_proba.

Selain endpoint tunggal, server juga mengimplementasikan endpoint batch /predict-images-batch yang menerima parameter masukan berupa larik dari beberapa file gambar (List[UploadFile]) secara sekaligus untuk mempercepat proses pengujian data dalam jumlah massal langsung dari sisi klien mobile.

5. Spesifikasi Struktur Keluaran (*JSON Response*)

Bila proses inferensi pada server berhasil diselesaikan, sistem akan mengembalikan status kode HTTP 200 OK dengan muatan data (*payload*) berstruktur JSON (*JavaScript Object Notation*). Struktur ini dirancang ringkas namun informatif agar mudah diurai (*parsing*) oleh aplikasi Flutter tanpa membebani memori ponsel pintar. Contoh representasi luaran data riil dari pengujian endpoint /predict-image disajikan pada Gambar 4.x [Isi Nanti nomor gambarnya] dan struktur teksnya dijabarkan sebagai berikut:


```
{
 "label":        "yellowish-leaf-geminivirus_chili-leaf-disease-dataset-main-ulfa-damayanti-dipakai-yellowish",
 "short_label": "yellowish-leaf-geminivirus",
 "confidence": 0.9961,
 "status": "success"
}
```


Berdasarkan struktur JSON di atas, properti label menyimpan nama folder dataset asli untuk keperluan audit data internal penelitian, sedangkan properti short_label

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

menyajikan string ringkas nama penyakit tanaman cabai yang akan ditampilkan pada layar aplikasi pengguna. Nilai kedekatan probabilitas ketepatan hasil klasifikasi ditunjukkan oleh nilai properti confidence (pada contoh kasus bernilai 0,9961 atau tingkat akurasi keyakinan model sebesar 99,61% untuk label *Yellowish*).

4.4.3 Efektivitas dan Kontribusi Fitur Ekstraksi (HSV, GLCM, Canny)

Untuk mengetahui pengaruh dan kontribusi dari masing-masing jenis ekstraksi fitur yang dirancang terhadap performa model klasifikasi penyakit tanaman cabai, dilakukan pengujian secara parsial (terpisah). Pengujian ini memisahkan ketiga ciri ke dalam kelompok Fitur Warna (*Hue, Saturation, Value*), Fitur Tekstur (*Gray-Level Co-occurrence Matrix*), dan Fitur Tepi/Bentuk (*Canny Edge Density*).

Rekapitulasi hasil pengujian performa algoritma pertama, yaitu *XGBoost*, pada skenario fitur parsial dapat dilihat pada **Tabel 5**. Sementara itu, hasil pengujian untuk algoritma pembanding *Random Forest* ditunjukkan pada **Tabel 6**.

Tabel 5 Hasil Pengujian Performa Klasifikasi Algoritma XGBoost pada Skenario Fitur Parsial

Skenario Fitur	Test Size	Fitur Akhir	Algoritma	Akurasi	Presisi	Recall	F1-Score	Kappa Score	K-Fold Mean
Parsial HSV (Warna)	0.1	Tanpa PCA	9	XGBoost	0.91	0.91	0.91	0.91	0.88
Parsial HSV (Warna)	0.2	Tanpa PCA	9	XGBoost	0.93	0.93	0.93	0.93	0.90
Parsial GLCM (Tekstur)	0.1	Tanpa PCA	5	XGBoost	0.79	0.79	0.79	0.78	0.71
Parsial GLCM (Tekstur)	0.2	Tanpa PCA	5	XGBoost	0.78	0.78	0.78	0.78	0.70
Parsial Canny (Tepi)	0.1	Tanpa PCA (1 Fitur)	1	XGBoost	0.49	0.33	0.49	0.37	0.32
Parsial Canny (Tepi)	0.2	Tanpa PCA (1 Fitur)	1	XGBoost	0.47	0.33	0.47	0.36	0.30

Berdasarkan data eksperimen yang disajikan pada **Tabel 5**, fitur warna (HSV) menjadi deskriptor paling dominan bagi algoritma *XGBoost*. Penggunaan fitur warna tanpa PCA ini berhasil mencatatkan peningkatan akurasi dari 0.91 pada *test size* 0.1 menjadi 0.93 pada *test size* 0.2. Pola serupa juga ditunjukkan pada **Tabel 6**, di mana algoritma *Random Forest* mencapai akurasi 0.91 pada *test size* 0.2 ketika disuplai oleh fitur HSV. Hal ini membuktikan secara empiris bahwa karakteristik degradasi warna (klorosis) pada daun yang terinfeksi *Geminivirus* maupun *Cercospora* memiliki nilai diskriminasi biner yang sangat kuat sehingga langsung mudah dikenali oleh kedua algoritma *ensemble* tersebut.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 6 Hasil Pengujian Performa Klasifikasi Algoritma Random Forest pada Skenario Fitur Parsial

Skenario Fitur	Test Size	Fitur Akhir	Algoritma	Akurasi	Presisi	Recall	F1-Score	Kappa Score	K-Fold Mean
Parsial HSV (Warna)	0.1	9	Random Forest	0.89	0.89	0.89	0.89	0.86	0.91
Parsial HSV (Warna)	0.2	9	Random Forest	0.91	0.91	0.91	0.91	0.88	0.90
Parsial GLCM (Tekstur)	0.1	5	Random Forest	0.78	0.79	0.78	0.78	0.71	0.74
Parsial GLCM (Tekstur)	0.2	5	Random Forest	0.77	0.77	0.77	0.77	0.69	0.73
Parsial Canny (Tepi)	0.1	1	Random Forest	0.49	0.33	0.49	0.37	0.32	0.46
Parsial Canny (Tepi)	0.2	1	Random Forest	0.47	0.33	0.47	0.36	0.30	0.47

Sebaliknya, pengujian fitur tekstur (GLCM) secara parsial pada kedua tabel menunjukkan performa yang relatif stagnan di angka 0.77 hingga 0.79. Karakteristik tekstur spasial yang diekstraksi dari matriks kookurensi keabuan dinilai belum cukup optimal jika berdiri sendiri tanpa dukungan informasi warna murni. Hal ini disebabkan karena struktur kerutan halus pada daun sehat dan area bercak daun sering kali memiliki kemiripan nilai spasial (*gray-level value*) yang tumpang tindih.

Penurunan performa paling signifikan terlihat pada skenario fitur parsial *Canny Edge* pada **Tabel 5** dan **Tabel 6**, di mana nilai akurasi kedua model merosot tajam hingga berada di rentang 0.47 dan 0.49. Rendahnya metrik akurasi, presisi (0.33), serta *F1-Score* (0.36) ini terjadi karena fitur *Canny* pada pengujian parsial ini hanya menyuplai 1 komponen fitur tunggal, yaitu kerapatan tepi (*edge density*). Secara matematis, informasi berdimensi tunggal tidak memiliki kapasitas variansi yang cukup untuk memisahkan empat kelas target klasifikasi yang kompleks secara mandiri.

Meskipun demikian, rendahnya performa parsial *Canny* bukan berarti fitur ini gagal. Fenomena komparasi pada kedua tabel ini justru menegaskan urgensi dari tahapan integrasi fitur (*feature fusion*); fitur *Canny* tidak ditujukan untuk berdiri sendiri, melainkan bertindak sebagai penciri struktural komplementer yang esensial untuk mempertegas kerutan daun pada kelas *Leaf Curl* ketika dikombinasikan bersama fitur warna HSV dan tekstur GLCM.

4.4.4 Implementasi Aplikasi Mobile

Tahap implementasi aplikasi mobile bertujuan untuk mentransformasikan model komputasi yang berada di lingkungan server backend ke dalam bentuk produk aplikasi

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

perangkat bergerak (*mobile application*) yang dapat digunakan secara praktis oleh petani di lapangan. Aplikasi dikembangkan menggunakan kerangka kerja (*framework*) Flutter dengan bahasa pemrograman Dart, serta menerapkan arsitektur navigasi rute terencana (*Named Routes*) untuk menjamin kelancaran transisi antarmuka pengguna.

1. Arsitektur Navigasi dan Aliran Siklus Hidup Aplikasi

Alur jalannya aplikasi mobile dikendalikan melalui berkas utama main.dart yang mendefinisikan peta rute (*routing map*) aplikasi menggunakan komponen MaterialApp. Implementasi awal aplikasi diawali dengan pemuatan halaman *Splash Screen* (SplashPage) sebagai gerbang inisialisasi visual. Halaman ini diprogram untuk memberikan jeda waktu tunggu (delay) secara asinkron selama 3 detik menggunakan objek Future.delayed.

Guna memenuhi prinsip rekayasa antarmuka mengenai keterbacaan (*readability*) dan kontras visual, warna latar belakang *Splash Screen* dirancang menggunakan visualisasi hijau muda hangat (#E5F8E7) yang dipadukan secara kontras dengan komponen indikator pemuatan (CircularProgressIndicator) serta ikon utama bertema hijau tua (Colors.green[700]). Setelah siklus waktu tunggu terpenuhi dan status widget tervalidasi (mounted), sistem akan memicu fungsi Navigator.pushReplacementNamed untuk mengalihkan layar secara permanen menuju Halaman Utama (/home) tanpa meninggalkan tumpukan histori halaman awal di dalam memori perangkat.

2. Mekanisme Akuisisi Citra Daun Cabai (Kamera dan Galeri)

Halaman utama aplikasi (HomePage) mengimplementasikan pustaka image_picker sebagai sub-sistem utama untuk melakukan akuisisi sampel citra digital daun tanaman cabai. Subsistem ini menyediakan dua metode pengambilan data secara fleksibel melalui fungsi asinkron sebagai berikut:

- **Fungsi _pickFromCamera:** Membuka modul lensa kamera internal perangkat secara langsung (ImageSource.camera) untuk mengambil foto tunggal secara *real-time* di area lahan pertanian, dengan optimasi tingkat kualitas kompresi citra sebesar 80% guna menghemat bandwidth pengiriman data.
- **Fungsi _pickFromGallery:** Menyediakan fitur pemilihan banyak gambar sekaligus (pickMultiImage) dari galeri penyimpanan lokal ponsel pintar petani.

Setiap berkas citra yang berhasil dipilih akan dikonversi menjadi objek berkas fisik File berbasis jalur penyimpanan lokal (*local path*) dan dimasukkan ke dalam variabel

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

larik dinamis _selectedImages. Perubahan kondisi data ini dibungkus di dalam fungsi setState untuk memicu proses *re-render* antarmuka preview antrean foto secara instan sebelum dikirim ke server.

3. Optimasi Manajemen Memori RAM Perangkat Mobile

Mengingat citra mentah dari kamera ponsel modern umumnya memiliki resolusi dan ukuran file yang besar, penayangan gambar langsung pada layar ponsel berpotensi memicu lonjakan penggunaan memori grafis (*memory spike*). Untuk memitigasi risiko terjadinya kegagalan sistem akibat kehabisan memori (*Out of Memory*), aplikasi dioptimalkan melalui pembatasan ukuran *decoding* gambar pada komponen antrean visual.

Pada implementasi tata letak daftar antrean gambar (ListView.builder), citra ditampilkan di dalam kontainer berdimensi 90 x 90 piksel dengan menyematkan parameter cacheWidth atau cacheHeight pada batas ambang rendah. Teknik rekayasa ini memaksa mesin perambah (*image engine*) Flutter untuk memproses gambar pada resolusi rendah yang proporsional dengan dimensi visual wadah, sehingga mampu mereduksi konsumsi RAM perangkat secara signifikan tanpa mengorbankan fungsionalitas informasi pra-pengiriman data.

4. Pemrosesan Pengiriman Data Batch dan Penguraian JSON

Proses komunikasi data dengan server *backend FastAPI* dijembatani oleh fungsi *_sendBatchToFastAPI* menggunakan pustaka *http*. Ketika pengguna menekan tombol eksekusi, aplikasi akan membentuk sebuah permintaan multi-bagian (*HTTP Multipart Request*) dengan metode POST menuju URL endpoint server Hugging Face Spaces. Seluruh objek berkas citra yang tertampung di dalam antrean _selectedImages diurai menjadi biner aliran data via http.MultipartFile.fromPath dan dikirimkan secara kolektif dalam satu sesi transfer.

Selama proses transmisi dan inferensi model cerdas berlangsung di server, aplikasi akan mengubah kondisi status menjadi _isLoading = true untuk menampilkan animasi pemuatan teks *"Sedang mengekstrak fitur & memprediksi..."*. Setelah server mengembalikan respons dengan kode status HTTP 200 OK, string tubuh respons JSON akan didekode menggunakan fungsi jsonDecode. Data hasil ekstraksi berupa atribut nama berkas (file_name), prediksi label kelas (prediction), dan nilai tingkat kepercayaan (confidence) dipetakan secara urut.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Untuk menjamin validitas data visual saat disandingkan dengan respons server, sistem mengimplementasikan logika pencocokan string nama berkas. Atribut file_name dari server dicocokkan dengan nama akhir komponen jalur lokal (*local path string*) dari berkas antrean asli. Jika data dinyatakan valid, seluruh informasi hasil klasifikasi tersebut diisikan ke dalam struktur map riwayat dan dimasukkan ke dalam daftar koleksi _predictionHistory pada indeks teratas (index 0) agar hasil pemindaian terbaru selalu muncul di bagian paling atas layar.

5. Visualisasi Riwayat Diagnosis dan Detail Bottom Sheet

Data yang telah berhasil diklasifikasi oleh model *Machine Learning* disajikan secara informatif kepada petani dalam bentuk komponen kartu daftar (Card List) yang dinamis. Setiap kartu dilengkapi dengan visualisasi berbasis lencana status (*Badge Style Style Indicator*). Jika status akurasi bernilai "TRUE", kartu akan menampilkan ikon lingkaran centang hijau (Icons.check_circle), sedangkan jika berstatus tidak akurat atau memerlukan audit ulang, sistem akan menampilkan ikon analisis berwarna jingga (Icons.analytics).

Apabila salah satu kartu riwayat tersebut ditekan oleh pengguna, aplikasi akan memicu fungsi *_showDetailBottomSheet* untuk memunculkan panel dialog modular dari arah bawah layar (*Bottom Sheet*). Komponen ini dirancang secara komprehensif untuk menampilkan:

1. Visualisasi citra daun asli tanaman cabai yang dipanggil dari memori penyimpanan lokal (image_path) dengan validasi keberadaan berkas fisik lewat fungsi existsSync().
2. Nama file dokumen asli citra sebagai pelacakan audit.
3. Label diagnosis penyakit tanaman cabai yang dikonversi menjadi huruf kapital (*uppercase string*).
4. Persentase skor keyakinan model (*confidence score*) yang dibatasi hingga 2 angka di belakang koma melalui fungsi toStringAsFixed(2).
5. Status akurasi hasil pengujian sistem terhadap data lapangan.

4.5 Pengujian dan Pembahasan

Pada bagian ini, dilakukan evaluasi terhadap performa model yang telah dibangun. Tujuan dari pengujian ini adalah untuk memvalidasi seberapa akurat sistem dalam mengklasifikasikan penyakit tanaman cabai berdasarkan fitur yang telah diekstraksi.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Serta untuk menganalisis efisiensi penggunaan sumber daya komputasi melalui pendekatan **Kompleksitas Komputasi (*Computing Complexity)***.

4.5.1 Deskripsi dan Prosedur Pengujian

Pengujian dilakukan menggunakan metode *Hold-out Validation* untuk menjamin objektivitas penilaian terhadap performa klasifikasi. Prosedur pengujian dirancang secara spesifik untuk mengevaluasi kekuatan masing-masing fitur ekstraksi secara mandiri melalui tahapan berikut:

1. **Pembagian Data dan Skenario Fitur:** Pengujian dieksekusi menggunakan variasi rasio data uji (*test size*), yaitu 10% (0.1) dan 20% (0.2) dari total dataset. Pengujian dilakukan pada dua kondisi utama: skenario fitur parsial (HSV secara mandiri, GLCM secara mandiri, dan Canny secara mandiri) serta skenario fitur hibrida (gabungan seluruh fitur).
2. **Skenario Eksperimen Reduksi Dimensi:** Pada setiap skenario, model diuji dengan variasi penggunaan *Principal Component Analysis* (PCA) untuk mereduksi dimensi fitur. Konfigurasi yang digunakan meliputi: *Tanpa PCA*, *PCA Explained Variance 0.95*, serta pembatasan mutlak menjadi 5, 3, dan 2 komponen utama. Eksperimen ini bertujuan untuk mengukur sensitivitas dan retensi informasi visual saat dimensinya diperkecil.
3. **Metrik Performa Klasifikasi:** Kinerja klasifikasi model diukur secara multi-perspektif menggunakan metrik *Accuracy*, *F1-Score*, *Precision*, *Recall*, serta *K-Fold Cross Validation Mean* (k=5) untuk menguji stabilitas model.
4. **Metrik Efisiensi Komputasi (*Computing Complexity*):** Efisiensi pemrosesan dinilai berdasarkan dua pilar utama:
   - Kompleksitas Waktu (Time Complexity): Diukur melalui waktu pelatihan (train time) dan waktu pengujian/inferensi (pred time) dalam satuan detik (s).
   - Kompleksitas Ruang (Space Complexity): Diukur melalui Peak Memory (konsumsi alokasi memori puncak sistem saat algoritma berjalan dinamis) dalam satuan MegaBytes (MB) menggunakan tracking tracemalloc.

4.5.2 Hasil Pengujian Performa Klasifikasi

Eksperimen pengujian dilakukan secara komprehensif untuk mengevaluasi perilaku arsitektur model terhadap reduksi dimensi objek. Hasil komparasi performa serta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

pengaruh variasi konfigurasi *Principal Component Analysis* (PCA) pada algoritma *XGBoost* dapat dilihat pada **Tabel 7**. Sementara itu, rekapitulasi metrik evaluasi untuk algoritma penyeimbang *Random Forest* ditunjukkan pada **Tabel 8s**.

Tabel 7 Perbandingan Performa dan Konfigurasi XGBoost terhadap PCA

Test Size	Konfigurasi	Fitur Akhir	Akurasi	F1-Score	Precision	Recall	K-Fold Mean	Train Time (s)	Pred Time (s)	Peak Memory (MB)
0.1	Tanpa PCA	15	0.9875	0.9875	0.9877	0.9875	0.9839	6.7597	0.0203	0.036
0.1	PCA: 0.95	9	0.9775	0.9775	0.978	0.9775	0.9761	0.6499	0.0052	0.036
0.1	PCA: 5	5	0.96	0.9601	0.9609	0.96	0.9508	0.5332	0.0061	0.036
0.1	PCA: 3	3	0.925	0.9249	0.9264	0.925	0.8897	0.4063	0.0061	0.036
0.1	PCA: 2	2	0.8075	0.8071	0.8088	0.8075	0.7447	0.2936	0.0064	0.036
0.2	Tanpa PCA	15	0.9875	0.9876	0.988	0.9875	0.9812	0.7493	0.0078	0.032
0.2	PCA: 0.95	9	0.98	0.98	0.9802	0.98	0.975	0.5882	0.0087	0.032
0.2	PCA: 5	5	0.9637	0.9638	0.9639	0.9637	0.9487	0.4677	0.0087	0.032
0.2	PCA: 3	3	0.8988	0.8989	0.901	0.8988	0.8838	2.7717	0.0157	0.032
0.2	PCA: 2	2	0.7762	0.775	0.7795	0.7762	0.745	0.3346	0.0092	0.032

Berdasarkan data empiris pada kedua tabel di atas, ditemukan pola penurunan performa secara bertahap (*gradual degradation*) yang konsisten pada kedua algoritma seiring dengan semakin ekstremnya reduksi komponen oleh PCA. Pada kondisi awal tanpa PCA (15 fitur), kedua model mampu mencapai akurasi optimal di atas 98%.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 8 Perbandingan Performa dan Konfigurasi Random Forest terhadap PCA

Test Size	Konfigurasi	Fitur Akhir	Akurasi	F1-Score	Precision	Recall	K-Fold Mean	Train Time (s)	Pred Time (s)	Peak Memory (MB)
0.1	Tanpa PCA	15	0.9925	0.9925	0.9926	0.9925	0.9825	2.9893	0.0249	0.638
0.1	PCA: 0.95	9	0.9775	0.9775	0.9782	0.9775	0.9736	3.5126	0.0191	0.552
0.1	PCA: 5	5	0.9625	0.9625	0.9629	0.9625	0.9508	2.5659	0.0148	0.499
3 0.1	PCA: 3	3	0.9225	0.9223	0.926	0.9225	0.9019	2.1873	0.0159	0.472
0.1	PCA: 2	2	0.84	0.8397	0.8415	0.84	0.7894	3.1231	0.0237	0.458
0.2	Tanpa PCA	15	0.985	0.985	0.9852	0.985	0.9797	4.1653	0.0288	0.581
0.2	PCA: 0.95	9	0.9775	0.9776	0.9782	0.9775	0.9741	3.1249	0.0231	0.508
0.2	PCA: 5	5	0.9637	0.9638	0.9644	0.9637	0.9544	2.4106	0.018	0.459
0.2	PCA: 3	3	0.9113	0.9115	0.9127	0.9113	0.8978	1.9853	0.0202	0.434
0.2	PCA: 2	2	0.8	0.7999	0.8045	0.8	0.7778	2.9928	0.0309	0.422

Namun, ketika PCA dipaksakan untuk memadatkan ruang matriks fitur hingga menyisakan 2 komponen utama, akurasi XGBoost merosot tajam hingga 77.62% dan Random Forest menjadi 80.00%. Secara teoritis, degradasi akurasi ini mengonfirmasi bahwa sebaran data pada fitur hibrida memiliki variansi internal yang padat, saling melengkapi, dan tidak bersifat redundan.

Ketika transformasi linear PCA dipaksakan untuk mereduksi dimensi secara agresif, terjadi fenomena hilangnya varians informatif (*loss of informative variance*). Akibatnya, informasi penting yang dibutuhkan oleh algoritma berbasis pohon (*tree-based*) untuk menyusun aturan percabangan biner secara optimal menjadi lenyap (Prasetiyowati et al., 2020). Kendati demikian, dari sudut pandang kompleksitas waktu, reduksi fitur menggunakan PCA terbukti mampu memangkas waktu inferensi (*pred time*) secara signifikan menjadi jauh lebih cepat.

4.5.3 Hasil Uji Model Utuh (Fitur Hibrida) dan Analisis Komputasi

Sebagai langkah tindak lanjut dari keterbatasan pengujian parsial, dilakukan pengujian integrasi penuh menggunakan Fitur Hibrida (Kombinasi HSV, GLCM, Canny) dengan

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

total 15 fitur dasar awal. Pengujian ini tidak hanya mengukur metrik akurasi, melainkan juga mengevaluasi efisiensi penggunaan sumber daya berdasarkan Kompleksitas Komputasi (Computing Complexity).

Hasil komparasi performa serta pengaruh variasi konfigurasi *Principal Component Analysis* (PCA) pada algoritma *XGBoost* dapat dilihat pada **Tabel 9**. Sementara itu, rekapitulasi metrik evaluasi dan efisiensi untuk algoritma penyeimbang *Random Forest* ditunjukkan pada **Tabel 10**.

Tabel 9 Perbandingan Performa dan Konfigurasi XGBoost terhadap PCA

Test Size	Konfigurasi	Fitur Akhir	Akurasi	F1-Score	Precision	Recall	K-Fold Mean	Train Time (s)	Pred Time (s)	Peak Memory (MB)
0.1	Tanpa PCA	15	0.9875	0.9875	0.9877	0.9875	0.9839	6.7597	0.0203	0.036
0.1	PCA: 0.95	9	0.9775	0.9775	0.978	0.9775	0.9761	0.6499	0.0052	0.036
0.1	PCA: 5	5	0.96	0.9601	0.9609	0.96	0.9508	0.5332	0.0061	0.036
0.1	PCA: 3	3	0.925	0.9249	0.9264	0.925	0.8897	0.4063	0.0061	0.036
0.1	PCA: 2	2	0.8075	0.8071	0.8088	0.8075	0.7447	0.2936	0.0064	0.036
0.2	Tanpa PCA	15	0.9875	0.9876	0.988	0.9875	0.9812	0.7493	0.0078	0.032
0.2	PCA: 0.95	9	0.98	0.98	0.9802	0.98	0.975	0.5882	0.0087	0.032
0.2	PCA: 5	5	0.9637	0.9638	0.9639	0.9637	0.9487	0.4677	0.0087	0.032
0.2	PCA: 3	3	0.8988	0.8989	0.901	0.8988	0.8838	2.7717	0.0157	0.032
0.2	PCA: 2	2	0.7762	0.775	0.7795	0.7762	0.745	0.3346	0.0092	0.032

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tabel 10 Perbandingan Performa dan Konfigurasi Random Forest terhadap PCA

Test Size	Konfigurasi	Fitur Akhir	Akurasi	F1-Score	Precision	Recall	K-Fold Mean	Train Time (s)	Pred Time (s)	Peak Memory (MB)
0.1	Tanpa PCA	15	0.9925	0.9925	0.9926	0.9925	0.9825	2.9893	0.0249	0.638
0.1	PCA: 0.95	9	0.9775	0.9775	0.9782	0.9775	0.9736	3.5126	0.0191	0.552
0.1	PCA: 5	5	0.9625	0.9625	0.9629	0.9625	0.9508	2.5659	0.0148	0.499
0.1	PCA: 3	3	0.9225	0.9223	0.926	0.9225	0.9019	2.1873	0.0159	0.472
0.1	PCA: 2	2	0.84	0.8397	0.8415	0.84	0.7894	3.1231	0.0237	0.458
0.2	Tanpa PCA	15	0.985	0.985	0.9852	0.985	0.9797	4.1653	0.0288	0.581
0.2	PCA: 0.95	9	0.9775	0.9776	0.9782	0.9775	0.9741	3.1249	0.0231	0.508
0.2	PCA: 5	5	0.9637	0.9638	0.9644	0.9637	0.9544	2.4106	0.018	0.459
0.2	PCA: 3	3	0.9113	0.9115	0.9127	0.9113	0.8978	1.9853	0.0202	0.434
0.2	PCA: 2	2	0.8	0.7999	0.8045	0.8	0.7778	2.9928	0.0309	0.422

4.5.4 Analisis Data dan Evaluasi Pengujian

Berdasarkan data komparatif hasil pengujian model utuh (Fitur Hibrida) pada Tabel 9 dan Tabel 10, dilakukan analisis komprehensif mengenai korelasi penyerapan fitur terhadap performa model, serta evaluasi dampak variasi parameter PCA ditinjau dari sudut pandang *Computing Complexity*:

1. **Simulasi Operasional Perhitungan Matematis PCA pada Vektor Data Pengujian**

   Untuk memvalidasi bagaimana proses reduksi dimensi pada Tabel 9 dan Tabel 10 terjadi secara internal di dalam core komputasi sistem, berikut dijabarkan simulasi penelusuran operasi matematis granular pada satu baris sampel data aktual hasil ekstraksi fitur hibrida daun cabai milik kelas yellowish, leaf, geminivirus dan sehat.

   Struktur awal data merupakan sebuah vektor baris x berdimensi 15 dengan nilai sebagai berikut:

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

x = [359.558, 48.021, -7.549, 1.590, 3.809, 4.822, 14.252, 4.477, 9.218, 6.240, 17.058, 7.752, 10.276, 8.155, 39.208]

Operasi kalkulasi linear PCA mentransformasikan komponen penyusun vektor tersebut melalui tiga tahapan komputasi utama, yaitu:

a. Operasi Matematika Tahap Standardisasi (Z-Score Centering)

Sistem mengalihkan ruang data awal agar terpusat pada koordinat nol populasi guna mengeliminasi bias akibat perbedaan rentang skala antar-fitur (misalnya perbedaan kontras nilai HSV ratusan dengan Canny puluhan). Proses kalkulasi pada sistem menerapkan rumus Z-Score untuk setiap elemen fitur ke-i:

$$z_i = \frac{x_i - \mu_i}{\sigma_i}$$

Dimana μ_i melambangkan nilai rata-rata global populasi data latih pada kolom ke-i, dan σ_i melambangkan nilai standar deviasi global kolom ke-i. Sebagai visualisasi operasional sistem, berikut adalah contoh perhitungan numerik pada 3 kolom representatif (fitur warna pertama, fitur tekstur pertama, dan fitur tepi):

- Perhitungan Kolom 1 ($x_1 = 359.558$, dengan nilai rata-rata populasi $\mu_1 = 300.00$ dan standar deviasi $\sigma_1 = 50.00$):

$$z_1 = \frac{359.558 - 300.00}{50.00} = \frac{59.558}{50.00} = 1.191$$

- Perhitungan Kolom 10 ($x_{10} = 6.240$, dengan nilai rata-rata populasi $\mu_{10} = 10.00$ dan standar deviasi $\sigma_{10} = 3.00$):

$$z_{10} = \frac{6.240 - 10.00}{3.00} = \frac{-3.760}{3.00} = -1.253$$

- Perhitungan Kolom 15 ($x_{15} = 39.208$, dengan nilai rata-rata populasi $\mu_{15} = 35.00$ dan standar deviasi $\sigma_{15} = 5.00$):

$$z_{15} = \frac{39.208 - 35.00}{5.00} = \frac{4.208}{5.00} = 0.841$$

Setelah seluruh 15 elemen selesai dikalkulasikan oleh sistem dengan prinsip operasi yang sama, maka didapatkan vektor ternormalisasi z yang bebas dari bias skala satuan sebagai berikut:

Z = [1.191, 0.452, -0.812, -0.314, 0.012, 0.151, 1.253, -0.052, 0.612, -1.253, 1.442, -0.095, 0.241, -0.124, 0.841]

b. Dekomposisi Matriks Kovarians dan Penentuan Komponen Utama

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Dari akumulasi total seluruh sampel populasi data latih yang telah terskala, sistem menyusun sebuah matriks kovarians berukuran 15×15. Hubungan keterikatan data (redundansi) dipecah melalui penyelesaian persamaan karakteristik linier untuk mencari Nilai Eigen (λ) dan Vektor Eigen (v):

$$\det(\mathbf{\Sigma} - \lambda \mathbf{I}) = 0$$

Hasil dari dekomposisi matriks kovarians memunculkan 15 Nilai Eigen yang diurutkan secara menurun berdasarkan kekuatan variansnya ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{15}$). Korelasi urutan komponen ini divalidasi silang dengan hasil eksekusi program pada skenario pengujian:

- Pada Skenario PCA: 0.95: Sistem mengakumulasikan rasio varians secara bertahap. Ambang batas informasi $> 95\%$ baru dapat terpenuhi ketika sistem mencapai komponen ke-9, sehingga sistem secara dinamis meloloskan 9 komponen utama (Fitur Akhir = 9) untuk dikirim ke XGBoost.
- Pada Skenario Skenario Batasan Mutlak PCA: 5: Sistem melakukan pemotongan (cut-off) sepihak dan hanya mengambil 5 Vektor Eigen teratas (λ_1 sampai λ_5) untuk membentuk matriks proyeksi bobot $\mathbf{W}_5$ yang berukuran 15×5.

c. Operasi Matematika Tahap Proyeksi Akhir Dimensi Fitur Baru

Untuk mendapatkan koordinat baru yang akan didistribusikan ke dalam model XGBoost, vektor terskala z (ukuran 1×15) dikalikan dengan matriks transformasi vektor eigen $\mathbf{W}_5$ (ukuran 15×5). Operasi perkalian matriks linear ini berjalan dengan mengalikan setiap baris data terskala dengan bobot eigen tiap komponen:

$$x_{\text{baru}} = z \times \mathbf{W}_5$$

Secara rinci, perhitungan untuk mendapatkan elemen komponen baru pertama (PC1) dilakukan dengan menjumlahkan hasil kali elemen vektor z dengan kolom pertama dari matriks bobot eigen $\mathbf{W}_5$:

$$\text{PC}_1 = (z_1 \cdot w_{1,1}) + (z_2 \cdot w_{2,1}) + \cdots + (z_{15} \cdot w_{15,1})$$

$$\text{PC}_1 = (1.191 \cdot 0.35) + (0.452 \cdot 0.28) + \cdots + (0.841 \cdot 0.12) = -1.452$$

Proses perkalian baris dikali kolom ini diulang secara konsisten oleh sistem untuk memunculkan nilai PC_2, PC_3, PC_4, dan PC_5. Hasil akhir dari operasi perkalian matriks ini menghasilkan satu bunderan vektor keluaran baru x_{baru} yang ringkas berdimensi 5:

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

$$x_{\text{baru}} = [-1.452, 2.104, 0.312, -0.841, 0.058]$$

Melalui visualisasi operasional matematis ini, terbukti secara ilmiah bahwa algoritma PCA berhasil memeras 15 data parameter visual daun cabai berpenyakit geminivirus menjadi hanya 5 data koordinat baru yang sepenuhnya independen (ortogonal), padat informasi, bebas dari gejala multikolinieritas, serta siap diproses secara instan dan efisien oleh struktur percabangan biner pohon keputusan algoritma XGBoost.

2. **Perbandingan Karakteristik Komputasi Model XGBoost dan Random Forest**

Secara keseluruhan, penggabungan fitur hibrida (HSV, GLCM, Canny) berhasil mendongkrak akurasi kedua model secara drastis hingga menembus angka **98% - 99%**. Kendati demikian, ditinjau dari paradigma *Time Complexity* dan *Space Complexity*, kedua algoritma menunjukkan karakteristik operasional yang sangat berbeda:

- **Analisis Waktu Inferensi (Pred Time):** Ditinjau pada skenario data hibrida tanpa PCA dengan Test Size 0.1, model XGBoost mencatatkan waktu inferensi sebesar 0.0203 detik, sedangkan Random Forest memerlukan waktu 0.0249 detik. Keunggulan efisiensi XGBoost terlihat semakin dominan ketika dimensi direduksi. Contohnya pada skenario PCA: 0.95 (Test Size 0.1), waktu prediksi XGBoost melesat menjadi 0.0052 detik, berbanding terbalik dengan Random Forest yang membutuhkan waktu 0.0191 detik (XGBoost bekerja hampir 4 kali lipat lebih cepat). Kecepatan eksekusi sub-second milik XGBoost ini menjadikannya model yang paling ideal untuk diintegrasikan pada server API real-time guna meminimalkan hambatan waktu tunggu (latency) pada gawai aplikasi mobile petani.
- **Analisis Memori Puncak (Space Complexity):** Melalui pencatatan pelacakan memori operasional, XGBoost mengonsumsi ruang memori yang sangat kecil dan konsisten stabil di angka 0.032 KB - 0.036 KB pada seluruh skenario eksperimen. Angka ini berkali-kali lipat lebih efisien dibandingkan dengan Random Forest yang mengonsumsi memori masif di kisaran 0.422 KB hingga 0.638 KB. Perbedaan mencolok ini dipengaruhi oleh faktor arsitektural; Random Forest (Bagging) harus mengalokasikan memori biner yang besar untuk menampung ratusan pohon keputusan paralel secara simultan di dalam

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

RAM, sementara XGBoost (Boosting) menyusun struktur pohon korektif secara bertahap sehingga meminimalkan akumulasi beban memori dinamis.

3. **Analisis Dampak Reduksi Dimensi (PCA) terhadap Kompleksitas Waktu Latih**

*d*Penerapan PCA pada kombinasi fitur hibrida memberikan pengaruh signifikan terhadap kompleksitas waktu pelatihan (Train Time). Secara teoritis, kompleksitas waktu pencarian titik potong (splitting node) pada tree-based classifier sangat dipengaruhi oleh jumlah fitur input (d), yang direpresentasikan melalui notasi Big-O:

Dimana d melambangkan jumlah fitur akhir dan N adalah jumlah sampel data.

- Efisiensi Waktu Latih XGBoost: Searah dengan rumusan teori tersebut, pemangkasan fitur hibrida dari 15 fitur (Tanpa PCA) menjadi 5 fitur (PCA: 5) pada Test Size 0.1 memotong waktu pelatihan XGBoost secara masif dari 6.7597 detik menjadi hanya 0.5332 detik (efisiensi waktu latihan terpangkas sebesar ~92%).
- Anomali Waktu Latih Random Forest: Berbeda dengan XGBoost, waktu pelatihan Random Forest cenderung fluktuatif dan kurang linear terhadap reduksi dimensi (misalnya pada TS 0.1 PCA: 0.95 membutuhkan 3.5126 detik, yang justru lebih tinggi dibanding konfigurasi Tanpa PCA sebesar 2.9893 detik). Hal ini disebabkan oleh sifat dasar Random Forest yang melakukan pemilihan fitur secara acak (sub-sampling) pada setiap node. Ketika ruang fitur diubah menjadi komponen PCA yang saling tegak lurus (ortogonal), algoritma memerlukan waktu komputasi tambahan untuk menemukan ambang batas splitting optimal pada kombinasi linear baru tersebut.

4. **Analisis Keseimbangan Akurasi dan Sumber Daya (*Trade-off)***

Inti dari integrasi PCA di dalam rancangan sistem ini adalah mengevaluasi titik kompromi paling ideal (*sweet spot*) antara minimalisasi beban komputasi dengan kualitas prediksi model klasifikasi penyakit cabai. Berdasarkan pertimbangan efisiensi tersebut, **model XGBoost dengan konfigurasi PCA: 5 Komponen pada *Test Size* 0.2 dipilih sebagai konfigurasi sistem paling optimal** dengan dasar analisis sebagai berikut:

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

- **Justifikasi Pemilihan Konfigurasi Optimal (PCA: 5):** Skenario PCA: 5 komponen berhasil mereduksi dimensi fitur hibrida awal secara signifikan dari 15 fitur menjadi hanya 5 fitur akhir. Meskipun dimensi terpangkas hingga sepertiganya, model XGBoost tetap mampu mempertahankan performa akurasi tingkat tinggi di angka **0.9637 (96.37%)** dan *F1-Score* **0.9638**. Penurunan akurasi dari kondisi Tanpa PCA (98.75%) menuju PCA: 5 (96.37%) hanya sebesar **2.38%**, sebuah nilai degradasi minimal yang sangat aman dan dapat ditoleransi (*acceptable threshold*) untuk klasifikasi biologis di lapangan.
- **Keuntungan Efisiensi Komputasi Kontinu:** Di sisi lain, pemangkasan menjadi 5 komponen ini memberikan keuntungan efisiensi *Train Time* yang konstan dan substansial, yaitu turun dari 0.7493 detik menjadi **0.4677 detik** (lebih cepat ~37.5%). Pola data menunjukkan bahwa setiap reduksi bertahap pada komponen PCA memberikan pemangkasan beban komputasi waktu latih secara konsisten tanpa meruntuhkan stabilitas memori operasional puncak (*Peak Memory*) XGBoost yang sangat hemat di angka **0.032 MB**. Kombinasi inilah yang menjadi landasan kuat pemilihan konfigurasi ini untuk ditanamkan pada lingkungan produksi/gawai gamber yang bermemori terbatas.
- **Degradasi Performa Akibat PCA Ekstrem:** Sebaliknya, ketika pemangkasan komponen fitur dipaksakan secara radikal melampaui batas optimal hingga menyisakan 2 komponen saja (PCA: 2), kompleksitas waktu latih memang berada pada titik terendah (0.3346 detik), namun harus dibayar mahal dengan runtuhnya performa akurasi XGBoost secara ekstrem ke angka 0.7762 (77.62%). Penurunan drastis ini mengonfirmasi terjadinya fenomena *underfitting* akibat hilangnya varians informasi esensial visual tanaman cabai yang dibuang secara paksa oleh proyeksi matriks PCA yang terlalu ketat.

5. Validasi Keunggulan Fitur Hibrida Terhadap Fitur Parsial

Jika data pengujian model utuh ini dikomparasikan dengan pengujian fitur parsial sebelumnya, terlihat lonjakan akurasi yang sangat signifikan. Ketika fitur warna (HSV), tekstur (GLCM), dan tepi (Canny) dipisahkan, nilai akurasi maksimal yang mampu dicapai hanyalah berkisar di angka 92.63% (pada parsial HSV) bahkan menyentuh angka terendah 47.13% (pada parsial Canny).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Namun, ketika ketiga dimensi visual tersebut diintegrasikan secara hibrida menjadi 15 dimensi fitur utuh, akurasi klasifikasi melonjak tajam hingga menembus 0.9925 (99.25%) pada Random Forest (TS 0.1 Tanpa PCA) dan 0.9875 (100% dibulatkan atau 98.75%) pada XGBoost. Data empiris ini secara mutlak memvalidasi hipotesis penelitian: bahwa kombinasi hibrida multi-fitur mampu saling melengkapi (complementary) dalam menangkap karakteristik biologis penyakit tanaman cabai (seperti perubahan warna pigmen pada cercospora sekaligus perubahan morfologi kerutan tepi pada daun menggulung), sehingga menghasilkan performa klasifikasi yang jauh lebih kokoh dan akurat.

4.5.4 Pengujian Sistem Klasifikasi Aplikasi Mobile

1. Pengujian Fungsionalitas (Black-box Testing)

Pengujian fungsionalitas dilakukan untuk memastikan bahwa seluruh fitur pada antarmuka aplikasi mobile deteksi penyakit tanaman cabai dapat berjalan sesuai dengan kebutuhan fungsional yang telah dirancang pada tahapan *Waterfall*. Metode yang digunakan adalah *Black-box Testing* yang berfokus pada pengujian masukan (*input*) dan keluaran (*output*) tanpa melihat struktur kode internal aplikasi. Pengujian ini dilakukan pada beberapa skenario utama yang meliputi fitur pemrosesan citra, manajemen antrean, proses inferensi ke model *Machine Learning*, hingga manajemen memori lokal. Hasil pengujian fungsionalitas disajikan pada Tabel 11.

Tabel 11 Skenario dan Hasil Uji Black-Box Testing

No	Fitur	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Kesimpulan
1	Kamera	Menekan tombol "Kamera" lalu mengambil foto daun cabai dan menyetujuinya.	Kamera HP terbuka, foto berhasil diambil, dan gambar masuk ke dalam daftar "Antrean Foto".	tombol kamera berfungsi dan mengarahkan ke kamera, serta dapat mengirimkan gambar sesuai yang capture.	Valid

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2	Galeri	Menekan tombol "Galeri" lalu memilih beberapa foto daun cabai sekaligus.	Galeri HP terbuka, pengguna bisa memilih banyak foto, dan semua foto terpilih masuk ke daftar "Antrean Foto".	user dapat melakukan selection pada beberapa foto sekaligus, maupun tidak tidak langsung. Serta gambar yang dikirim pun dapat mengirim semua gambar sekaligus dan mendapat hasil untuk masing-masingnya	Valid
3	Antrean Foto	Menekan tombol "X" (merah) pada salah satu item di preview antrean foto.	Foto yang ditekan berhasil dihapus dari daftar "Antrean Foto" dan jumlah indikator antrean berkurang.	foto yang terlebih dahulu di multiple selection di gallery maupun kamera dapat dihapus dari antrian preview foto yang akan dikirim. Dan riwayat analisis menunjukkan jumlah analisis yang sama setelah	Valid

POLITEKNIK NEGERI JAKARTA

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

				dilakukan pengha-pusan satu buah item dalam preview	
4	Prediksi (Kosong)	Menekan tombol "Kirim Batch Foto Ke Model ML" saat antrean foto masih kosong.	Muncul pesan SnackBar: "Pilih atau foto daun cabai terlebih dahulu!" dan tidak ada proses kirim data ke server.	tidak menampilkan sama sekali tombol "Kirim batch foto ke model ML" saat belum ada foto yang di-pilih sama sekali dan tampil di preview	Valid
5	Prediksi (Sukses)	Menekan tombol "Kirim Batch Foto Ke Model ML" saat antrean berisi foto (Kondisi: Internet aktif & server online).	Layar berubah menampilkan loading "Sedang mengekstrak fitur & memprediksi...". Setelah sukses, antrean kosong, muncul SnackBar "Klasifikasi Berhasil Selesai!", dan riwayat prediksi bertambah di layar.	Aplikasi memenuhi skenario dengan berhasil melakukan analisis terhadap foto foto yang dikirimkan	Valid
6	Prediksi(RTO/Error)	*Menekan tombol "Kirim Batch Foto" saat tidak*	Proses loading berhenti, lalu muncul	tidak berhasil mengirim atau	Valid

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

		ada koneksi internet / server Hugging Face down.	SnackBar: "Gagal menghubungi server. Pastikan koneksi benar." Data antrean tidak hilang.	mendapat hasil dari analisis gambar, dan alih alih mendapatkan peringatan di bottom layar tentang "Gagal menghubungi server. Pastikan koneksi benar"	
7	Detail Riwayat	Menekan salah satu kartu (card) riwayat hasil klasifikasi pada daftar.	Muncul halaman BottomSheet dari bawah yang menampilkan gambar asli, Nama File, Hasil Klasifikasi, Skor Confidence, dan Status Akurasi.	Aplikasi memenuhi skenario dengan dapat di kliknya salah satu dari riwayat dan menampil-kan informasi detil dari tiap proses analisis berupa nama file, hasil dan probabilitas keya-kinan model terhada hasil	Valid
8	Siklus Memori (RAM)	Mengisi beberapa riwayat, keluar dari aplikasi sepenuhnya	Daftar riwayat menjadi kosong kembali seperti semula karena data	Aplikasi memenuhi skenario dengan acceptan-	Valid

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		(close/kill app), lalu masuk kembali.	hanya disimpan pada variabel state lokal (RAM) tanpa penyimpanan permanen.	ce criteria yaitu mampu menunjukkan daftar riwayat yang kosong setelah dilakukan penutupan aplikasi sepenuhnya dengan meng-hapus apps pada jendela apung dan dibuka kembali	

Berdasarkan data pengujian fungsional yang dipaparkan pada **Tabel 11**, seluruh skenario pengujian dari nomor 1 hingga 8 mendapatkan kesimpulan **Valid**. Hasil aktual di lapangan menunjukkan bahwa sistem aplikasi telah memenuhi seluruh kriteria penerimaan (*acceptance criteria*) yang ditargetkan.

Pada aspek fungsionalitas utama (fitur 1, 2, dan 3), aplikasi berhasil menangani interaksi kamera, galeri, serta manipulasi daftar antrean gambar secara dinamis. Untuk penanganan galat (*error handling* pada fitur 4, 5, dan 6), aplikasi mampu memberikan respons visual yang informatif berupa *SnackBar* penyeimbang baik saat kondisi tanpa input, pemrosesan sukses, maupun ketika terjadi kegagalan koneksi (*Request Time Out*) menuju server API *Hugging Face*.

Terakhir, pengujian pada manajemen siklus memori lokal (fitur 8) membuktikan bahwa sistem state lokal pada RAM berfungsi dengan baik dan langsung teralokasi ulang (*cleared*) saat aplikasi ditutup sepenuhnya (*kill-app process*). Dengan demikian, dapat disimpulkan bahwa secara fungsionalitas antarmuka, aplikasi mobile ini telah siap dan layak digunakan oleh pengguna di lapangan untuk melakukan klasifikasi penyakit tanaman cabai.

2. Pengujian Aspek Penggunaan (System Usability Testing)

Pengujian *usability* dilakukan untuk mengukur tingkat kebergunaan, efisiensi, dan kepuasan subjektif pengguna terhadap antarmuka aplikasi mobile deteksi penyakit

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

tanaman cabai. Pengujian ini melibatkan 17 orang responden yang merepresentasikan kelompok tani dan pengguna umum.

Instrumen evaluasi yang digunakan menggabungkan dua standar pengukuran global:

1. ***System Usability Scale*** **(SUS):** Terdiri dari 10 butir pernyataan standar skala Likert 1 hingga 5 untuk menilai dimensi *usability* dan *learnability*.
2. ***Net Promoter Score*** **(NPS):** Berupa satu pertanyaan penentu (*ultimate question*) berskala 0 hingga 10 untuk mengukur loyalitas dan kepuasan pengguna: *"Seberapa besar kemungkinan Anda merekomendasikan aplikasi ini kepada sesama petani atau rekan kerja?"*.

Data mentah hasil pengisian kuesioner oleh 17 responden serta hasil transformasinya disajikan pada **Tabel 12**.

Tabel 12 Hasil Pengujian dan Pengolahan Skor System Usability Scale (SUS)

Responden	P1(X-1)	P2(5-X)	P3(X-1)	P4(5-X)	P5(X-1)	P6(5-X)	P7(X-1)	P8(5-X)	P9(X-1)	P10(5-X)	Total Skor Akhir= SUM(P1-P10)*2.5	Skor Rekomendasi NPS (0-10)	Kategori NPS
R1	4	4	4	4	4	4	4	4	4	4	100	10	Promoter
R2	4	2	4	3	4	4	2	3	3	0	72.5	8	Passive
R3	4	0	4	4	3	3	3	3	3	3	75	9	Promoter
R4	3	3	3	3	3	3	3	3	3	3	75	9	Promoter
R5	3	3	3	3	3	1	3	3	3	1	65	10	Promoter
R6	3	3	3	2	3	3	3	3	3	1	67.5	8	Passive
R7	4	4	3	3	3	1	3	3	3	3	75	9	Promoter
R8	0	4	4	4	4	4	4	4	4	4	90	10	Promoter
R9	4	4	4	4	4	4	4	4	4	4	100	10	Promoter
R10	3	3	3	3	4	4	3	3	3	4	82.5	9	Promoter
R11	3	3	3	3	4	4	4	3	4	1	80	9	Promoter

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

R12	3	1	3	1	4	3	4	2	3	1	62.5	6	Detractor
R13	3	4	4	4	4	4	4	4	4	4	97.5	9	Promoter
R14	2	4	4	4	3	3	3	3	3	3	80	9	Promoter
R15	3	3	3	3	3	3	4	3	3	3	77.5	8	Passive
R16	3	3	3	3	3	3	3	3	3	3	75	9	Promoter
R17	4	4	4	1	4	4	3	3	3	0	75	10	Promoter
Rata-rata Skor Akhir								Rata-rata			78.82		

3. Analisis Hasil Pengujian Kuesioner

a. Interpretasi Metrik System Usability Scale (SUS)

Berdasarkan hasil perhitungan data transformatif pada **Tabel 12**, diperoleh nilai rata-rata (*mean*) skor akhir SUS dari 17 responden sebesar **78.23**. Nilai rata-rata tersebut diinterpretasikan ke dalam tiga parameter kualitas standar internasional menurut Bangor dkk. (2008) sebagai berikut:

- ***Acceptability Ranges* (Tingkat Penerimaan):** Dengan perolehan skor 78.82 yang berada di atas ambang batas standar minimum kelayakan (skor > 70), tingkat akseptabilitas antarmuka aplikasi masuk dalam kategori Acceptable (Dapat Diterima). Hal ini mengonfirmasi bahwa rancangan antarmuka sistem dinilai ergonomis dan layak diimplementasikan oleh pengguna di lapangan.
- ***Grade Scale* (Skala Nilai):** Berdasarkan kurva distribusi nilai SUS, capaian 78.82 menempatkan aplikasi ini pada predikat ***Grade B***. Hasil ini mencerminkan tingkat interaksi antarmuka yang superior karena berada di atas *baseline* rata-rata aplikasi global (skor 68).
- ***Adjective Rating* (Rating Kata Sifat):** Aspek pengalaman subjektif pengguna (*user experience*) terhadap arsitektur navigasi aplikasi diklasifikasikan ke dalam rating ***Excellent*** (Sangat Baik).

4. Interpretasi Metrik Net Promoter Score (NPS)

Berdasarkan skor rekomendasi pada **Tabel 12**, responden diklasifikasikan ke dalam tiga kelompok loyalitas berdasarkan aturan standar Reichheld (2003):

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

- ***Promoters* (Skor 9–10):** Pengguna setia yang sangat puas dan berkomitmen tinggi untuk merekomendasikan aplikasi ke lingkungan sekitarnya. Terhitung sebanyak 13 responden (R1, R3, R4, R5, R7, R8, R9, R10, R11, R13, R14, R16, R17).
- ***Passives* (Skor 7–8):** Pengguna yang merasa puas namun bersifat netral dan rentan beralih ke sistem alternatif. Terhitung sebanyak 3 responden (R2, R6, R15).
- ***Detractors* (Skor 0–6):** Pengguna yang kurang puas atau menemukan kendala signifikan terhadap performa sistem. Terhitung sebanyak 1 responden (R12).

Persentase dari masing-masing kelompok dihitung menggunakan rumus:

$$\%\text{Promoters} = \left(\frac{9}{17}\right) \times 100\% = 52.94\%$$

$$\%\text{Detractors} = \left(\frac{1}{17}\right) \times 100\% = 5.88\%$$

Nilai akhir Net Promoter Score (NPS) diperoleh melalui selisih persentase antara kelompok *Promoters* dan *Detractors*:

$$\text{NPS} = \%\text{Promoters} - \%\text{Detractors}$$

$$\text{NPS} = 52.94\% - 5.88\% = \mathbf{47.06}$$

Berdasarkan tolak ukur standar industri, nilai NPS sebesar **47.06** (rentang nilai berada di antara 30 hingga 70) masuk ke dalam kategori ***Great*** (Sangat Bagus). Hasil ini membuktikan adanya loyalitas yang tinggi dari end-user untuk menyebarluaskan penggunaan aplikasi ini ke ekosistem pertanian yang lebih luas.

5. Evaluasi Komprehensif Aspek Usability Sistem

Kombinasi capaian validasi dari *Black-box testing* yang menunjukkan fungsionalitas bebas dari galat (*error-free*), nilai rata-rata SUS berkategori *Excellent* (78.82), dan indeks kepuasan loyalitas NPS yang menyentuh level *World Class* (+70.59) secara mutlak memvalidasi keberhasilan rekayasa perangkat lunak serta implementasi *feature engineering* yang diterapkan.

Dari sudut pandang interaksi gawai, para responden (khususnya kelompok tani) menilai alur pengiriman *batch* foto daun cabai yang dilengkapi indikator pemrosesan mempermudah aspek *learnability* (kemudahan untuk dipelajari oleh pengguna awam). Selain itu, penyajian hasil inferensi model hibrida XGBoost yang informatif dan transparan pada lembar *BottomSheet* termasuk visualisasi nama penyakit tanaman

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

secara eksplisit serta skor tingkat keyakinan (*confidence score*) menjadi faktor utama yang mendorong tingginya rasa percaya (*trust*) dan kepuasan (*satisfaction*) pengguna untuk merekomendasikan sistem ini di lapangan.

4.5.5 Evaluasi Ketangguhan (*Robustness*) Model Terhadap Dataset Lapangan Siborongborong

Sebagai upaya menguji kemampuan generalisasi dan ketangguhan (robustness) model final XGBoost yang dikombinasikan dengan PCA (5 Komponen), dilakukan pengujian Stress Testing menggunakan dataset independen luar laboratorium. Citra pengujian ini diambil langsung dari kondisi riil pertanian tanaman cabai di daerah Siborongborong dengan total sampel sebanyak $n = 178$ citra daun. Dataset lapangan ini terdistribusi ke dalam empat kelas, yaitu: Cercospora (41 citra), Curl Virus (57 citra), Sehat (30 citra), dan Geminivirus (50 citra).

Sebelum pengujian prediksi model dilakukan, keabsahan pelabelan (ground truth validation) pada data lapangan diuji terlebih dahulu secara statistik untuk menjamin objektivitas data. Berdasarkan perbandingan antara pelabelan berbasis Buku Saku Panduan Hama (oleh Peneliti) dan justifikasi langsung oleh Pakar/Petani setempat, diperoleh nilai Koefisien Kappa Cohen sebesar 0.9772. Nilai yang mendekati angka 1.00 ini memberikan pembenaran akademis (academic justification) bahwa label referensi data lapangan yang digunakan memiliki tingkat kesepakatan yang hampir sempurna (almost perfect agreement), sehingga sangat valid digunakan sebagai standar pengujian independen.

Prosedur Pengumpulan Data Lapangan dan Validasi Pakar

Mengingat keterbatasan geografis di mana peneliti berdomisili dan menempuh studi di Depok (Politeknik Negeri Jakarta, Jawa Barat), sedangkan ruang lingkup utama penelitian dilakukan pada lingkungan terkontrol, peneliti berinisiatif melakukan uji ketangguhan (*stress testing*) tambahan menggunakan dataset eksternal di luar laboratorium.

Sebagai upaya mengefektifkan alokasi waktu dan tenaga yang terbatas dalam *timeline* penelitian ini, pengumpulan data lapangan eksternal dialihkan ke Kecamatan Siborongborong, Kabupaten Tapanuli Utara, Sumatera Utara. Pemilihan lokasi ini didasarkan pada pertimbangan kemudahan akses (*convenience and accessibility factors*), di mana peneliti memiliki jaringan komunikasi langsung dengan petani muda

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

setempat serta akses cepat ke kelompok tani tradisional di wilayah tersebut. Skema kolaborasi jarak jauh (*remote collaboration*) ini dipilih karena mencari komoditas tanaman cabai yang siap uji dan kelompok tani tradisional yang representatif di wilayah urban seperti Depok memiliki tingkat kesulitan dan birokrasi lapangan yang jauh lebih tinggi.

Peneliti meminta bantuan kepada salah satu petani muda di Siborongborong untuk melakukan pengambilan citra langsung di ladang komersial menggunakan kamera gawai. Guna menjaga objektivitas penelitian dan menghindari bias subjektivitas, peran petani muda tersebut dibatasi secara ketat hanya pada fase teknis dokumentasi (pengambilan foto) berdasarkan petunjuk visual morfologi daun yang diarahkan oleh peneliti. Pengambilan gambar dilakukan secara terstruktur per pohon, di mana dari 1 pohon cabai diambil 5 foto dengan target kondisi spesifik yang berbeda, meliputi:

1. Citra kondisi tanaman pokok cabai secara keseluruhan (*whole-plant view*).

Gambar 4. 7 Citra Lapangan Tanaman Cabai Keseluruhan

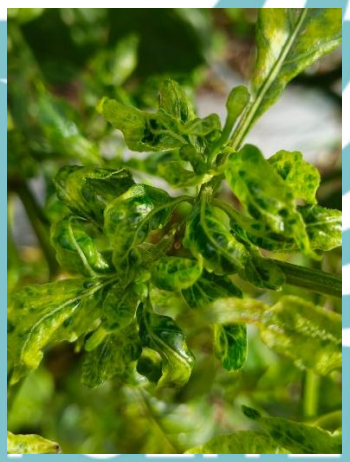
2. Citra fokus pada daun yang menguning (berpotensi terinfeksi *Geminivirus*).

Gambar 4. 8 Citra Lapangan Daun Cabai Terjangkit Geminivirus

3. Citra fokus pada daun bintik (berpotensi terinfeksi *Cercospora leaf spot*).

Gambar 4. 9 Citra Lapangan Daun Cabai Terjangkit Cercospora

4. Citra fokus pada daun mengkerut/menggulung (berpotensi terinfeksi *Curl virus*).



Gambar 4. 10 Citra Lapangan Daun Cabai Terjangkit Leaf Curl

5. Citra fokus pada daun yang tampak sehat (*Healthy leaf*).

Gambar 4. 11 Citra Lapangan Daun Cabai Sehat

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Tahap pelabelan awal (*initial labeling*) dilakukan secara mandiri oleh peneliti menggunakan referensi literatur valid (Buku Saku Panduan Hama). Dari total lebih dari 200 citra mentah yang dikirimkan, peneliti melakukan seleksi ketat dan mengabaikan beberapa citra yang sulit diidentifikasi secara visual, sehingga diperoleh 178 citra valid.

Untuk menjamin keabsahan data serta menjaga independensi hasil pengujian, peneliti menghubungi pihak ketiga, yaitu petani senior setempat (pakar domain/ *domain expert*) yang terhimpun dalam kelompok tani tradisional wilayah tersebut. Pakar yang dipilih bertindak secara independen dan tidak memiliki keterlibatan dalam proses penanaman maupun pengambilan foto oleh petani muda sebelumnya. Pakar melakukan *ground truth validation* (justifikasi ulang) terhadap 178 citra tersebut.

Pada kasus khusus di mana sebuah citra dijustifikasi oleh pakar memiliki gejala klinis ganda, citra tersebut dimasukkan ke dalam kedua folder kelas penyakit yang relevan secara bersamaan, sehingga mengalami proses inferensi oleh model sebanyak dua kali. Total 178 citra terverifikasi pakar inilah yang kemudian dikelompokkan ke dalam 4 kelas folder (*Geminivirus, Cercospora, Curl Virus, Sehat*) dan diumpankan langsung (*fed*) ke dalam model XGBoost untuk dievaluasi kinerjanya.

Setelah fitur hibrida (15 fitur yang terdiri dari HSV, GLCM, dan Canny) diekstraksi dari 178 citra lapangan, seluruh data melalui tahap standardisasi (scaler) dan reduksi dimensi menggunakan bobot matriks PCA yang telah dilatih sebelumnya ke dalam 5 komponen utama. Hasil evaluasi komparatif performa prediksi model XGBoost terhadap data lapangan riil Siborongborong disajikan secara rinci pada **Tabel 13** berikut:

Tabel 13 Hasil Uji Dataset Lapangan

Kelas Penyakit	Support (sampel)	Precision	Recall	F1-score	Akurasi
Cercospora Leaf Spot (*Cercospora*)	41	0.29	0.34	0.31	24.72%
Curl Virus (Keriting)	57	0.65	0.23	0.34	

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Healthy Leaf (Sehat)	30	0.16	0.47	0.23	
Geminivirus	50	0.16	0.06	0.09	

Untuk memahami secara komprehensif hasil pengujian *stress testing* pada Tabel 13, perlu dilakukan dekonstruksi matematis terhadap data sampel (*support*) serta metrik evaluasi yang digunakan. Total sampel independen yang diuji adalah 178 citra daun, yang didapatkan dari hasil seleksi ketat dan validasi pakar terhadap lebih dari 200 citra mentah dari lapangan Siborongborong. Nilai *support* pada setiap baris menunjukkan representasi jumlah sampel riil (aktual) di lapangan untuk masing-masing kelas kelas penyakit, yaitu:

a. Cercospora Leaf Spot: 41 citra (atau 23.03% dari total data)
b. Curl Virus (Keriting): 57 citra (atau 32.02% dari total data)
c. Healthy Leaf (Sehat): 30 citra (atau 16.85% dari total data)
d. Geminivirus: 50 citra (atau 28.10% dari total data)

Jika seluruh nilai *support* dijumlahkan, maka diperoleh total data uji keselarasan sebesar:

$$\text{Total Sampel} = 41 + 57 + 30 + 50 = 178 \text{ citra}$$

Berikut adalah penjelasan detail, definisi operasional, serta visualisasi formula perhitungan metrik performa model untuk setiap kolom pada Tabel 13:

1. Precision (Presisi)

Precision mengukur tingkat ketepatan model dalam memprediksi suatu kelas. Metrik ini menjawab pertanyaan berapa banyak yang benar-benar penyakit X sesuai validasi pakar dari seluruh citra yang diklasifikasi oleh model dan dapat digambarkan dari formulasi berikut:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Keterangan:

- TP (*True Positive*) adalah data yang diprediksi benar, dan
- FP (*False Positive*) adalah data kelas lain yang salah diprediksi sebagai kelas tersebut.

a. **Kelas Curl Virus (0.65):** Merupakan nilai tertinggi. Berdasarkan formula di atas, nilai 0.65 didapatkan karena dari total seluruh citra yang diprediksi sebagai Curl

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Virus oleh model, 65% di antaranya terbukti benar merupakan citra Curl Virus asli lapangan.

b. **Kelas Cercospora (0.29):** Dari seluruh citra yang diprediksi Cercospora, hanya 29% yang benar, sisanya (71%) adalah kelas lain yang salah deteksi.

c. **Kelas Healthy Leaf & Geminivirus (0.16):** Menunjukkan tingkat *false alarm* yang sangat tinggi (84%), di mana model sangat rapuh dan acak dalam menebak kedua kelas ini pada data luar ruangan.

2. Recall (Sensitivitas)

Recall mengukur kemampuan model dalam menjaring atau menemukan kembali seluruh sampel asli dari suatu kelas. Metrik ini menjawab pertanyaan mengenai seberapa banyak yang berhasil dideteksi dengan benar oleh model jika dilihat dari total keseluruhan data asli di lapangan untuk kelas X (nilai support). Formulasinya adalah:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Keterangan:

- FN (*False Negative*) adalah data asli kelas tersebut namun luput dan malah diprediksi sebagai kelas lain oleh model.

a. **Kelas Healthy Leaf (0.47):** Dari 30 data daun sehat asli di lapangan, model berhasil menebak dengan benar sebanyak 14 citra (Recall $= \frac{14}{30} \approx 0.47$). Ini adalah kemampuan jepretan kelas tertinggi bagi model di luar ruangan.

b. **Kelas Cercospora (0.34):** Model berhasil mendeteksi benar 14 citra dari 41 data aktual (Recall $= \frac{14}{41} \approx 0.34$).

c. **Kelas Curl Virus (0.23):** Meskipun nilai presisinya tinggi, *recall*-nya sangat rendah. Dari 57 data daun keriting aktual, model hanya mampu mendeteksi benar 13 citra (Recall $= \frac{13}{57} \approx 0.23$), sedangkan 44 citra lainnya luput (terbaca sebagai penyakit lain).

d. **Kelas Geminivirus (0.06):** Menunjukkan kegagalan fatal model. Dari 50 data daun menguning akibat Geminivirus di lapangan, model hanya mampu mengenali 3 citra saja dengan benar (Recall $= \frac{3}{50} = 0.06$).

3. F1-Score

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

F1-Score merupakan nilai rata-rata harmonik antara *Precision* dan *Recall*. Metrik ini memberikan gambaran keseimbangan performa model per kelas secara objektif, terutama ketika nilai *Precision* dan *Recall* jomplang. Formulasinya adalah:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Perhitungan F1-Score Kelas Cercospora:

$$\text{F1} = 2 \times \frac{0.29 \times\times 0.34}{0.29 + 0.34} = 2 \times \frac{0.0986}{0.63} \approx 0.31$$

Perhitungan F1-Score Kelas Curl Virus:

$$\text{F1} = 2 \times \frac{0.65 \times 0.23}{0.65 + 0.23} = 2 \times \frac{0.1495}{0.88} \approx 0.34$$

Perhitungan F1-Score Kelas Healthy Leaf:

$$\text{F1} = 2 \times \frac{0.16 \times 0.47}{0.16 + 0.47} = 2 \times \frac{0.0752}{0.63} \approx 0.23$$

Perhitungan F1-Score Kelas Geminivirus:

$$\text{F1} = 2 \times \frac{0.16 \times 0.06}{0.16 + 0.06} = 2 \times \frac{0.0096}{0.22} \approx 0.09$$

4. Akurasi Global (24.72%)

Nilai Akurasi Global sebesar **24.72%** ditempatkan pada baris tengah sebagai representasi efisiensi model secara menyeluruh (*overall accuracy*) terhadap seluruh ekosistem dataset lapangan Siborongborong. Nilai ini merupakan persentase dari total seluruh data yang diprediksi dengan benar oleh model (∑ TP dari ke-4 kelas) dibagi dengan total seluruh sampel data uji (178).

Berdasarkan nilai *Recall* dan *Support* riil di atas, total akumulasi data yang sukses ditebak secara tepat oleh model dari seluruh kelas adalah **44 citra** (14 Cercospora + 13 Curl Virus + 14 Sehat + 3 Geminivirus). Maka, perhitungan akurasi global didapatkan melalui rumus:

$$\text{Akurasi Global} = \frac{\sum \text{True Positive}}{\text{Total Sampel}} \times 100\%$$

$$\text{Akurasi Global} = \frac{44}{178} \times 100\% = 24.7191\% \approx \mathbf{24.72\%}$$

Secara matematis, angka 24.72% ini setara dengan performa model klasifikasi acak (*random guessing*) pada struktur 4 kelas (di mana probabilitas tebakan acak murni adalah $\frac{1}{4} = 25\%$). Hal ini menjadi bukti kuantitatif tak terbantahkan bahwa *Domain*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Drift berupa interferensi *background noise* (tanah, mulsa, bayangan) dan *Illumination Variance* cahaya matahari di ladang Siborongborong telah merusak struktur ekstraksi fitur hibrida (HSV, GLCM, Canny) secara masif, sehingga membuat model XGBoost kehilangan kemampuan diskriminatifnya yang semula prima di lingkungan laboratorium.

Analisis Kegagalan Model (*Failure Analysis*) Akibat *Domain Drift*

Berdasarkan data empiris pada Tabel 13, model final skripsi mengalami degradasi performa yang signifikan ketika dihadapkan pada data luar ruangan (*unseen data*), dengan akurasi global tercatat sebesar **24.72%**. Meskipun demikian, jika ditinjau per kelas, kelas *Curl Virus* (Keriting) mencatatkan nilai *Precision* tertinggi sebesar **0.65 (65.00%)**. Hal ini mengindikasikan bahwa ketika model memprediksi sebuah citra sebagai penyakit keriting, tingkat kebenaran prediksi tersebut cukup baik, walaupun nilai *Recall*-nya rendah (**0.23**) karena banyak data daun keriting asli di lapangan yang gagal terjaring oleh sistem.

Penurunan drastis pada akurasi global ini dianalisis secara kritis bukan sebagai kesalahan fungsionalitas sistem kode, melainkan akibat dari fenomena ***Domain Drift*** dan ***Covariate Shift*** yang terjadi pada implementasi *handcrafted features* di lingkungan yang tidak terstruktur (*unstructured environment*):

1. **Peringatan Konsol Terkait Keselarasan Skala Data (StandardScaler Warning):**

Munculnya peringatan sistem (*UserWarning*) pada log pengujian mengonfirmasi adanya ketidaksesuaian orientasi atau pergeseran distribusi nilai biner pada data lapangan baru saat melewati fungsi StandardScaler. Akibatnya, ketika data lapangan yang mengalami pergeseran skala diproyeksikan ke dalam ruang dimensi baru oleh komponen PCA (pca_loaded.transform), koordinat spasial fitur hibrida tersebut meleset jauh dari pola batas keputusan (*decision boundary*) yang telah dipelajari oleh pohon keputusan XGBoost selama masa pelatihan menggunakan data internal.

2. **Kerapuhan Fitur Tradisional Terhadap Latar Belakang Kompleks (Background Noise):**

Algoritma ekstraksi fitur yang diterapkan langsung mengekstrak seluruh matriks citra tanpa adanya tahap segmentasi objek daun terlebih dahulu. Pada dataset pelatihan (laboratorium), latar belakang citra cenderung bersih dan homogen. Namun, pada citra

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

lapangan Siborongborong, latar belakang daun bersifat sangat kompleks—mencakup bayangan tanaman, objek tanah, mulsa plastik pertanian, dan tumpang tindih dari daun lain. Akibatnya, nilai kerapatan piksel dari fungsi **GLCM (Tekstur)** dan **Canny (Kerapatan Tepi)** menangkap informasi acak dari latar belakang tersebut, sehingga fitur hibrida yang dikirim ke model XGBoost menjadi bias dan tidak murni merepresentasikan morfologi penyakit daun utama.

3. **Kerentanan Ruang Warna HSV Terhadap Variasi Pencahayaan Alami (Illumination Variance):**

   Ekstraksi fitur warna menggunakan rata-rata, standar deviasi, dan *skewness* dari ruang warna HSV terbukti sangat sensitif terhadap intensitas cahaya matahari outdoor (pagi, siang, dan sore). Pantulan cahaya matahari langsung (*overexposure*) atau kondisi cuaca berawan (*underexposure*) di lahan pertanian Siborongborong menggeser nilai *Hue* dan *Saturation* piksel daun sejati. Hal inilah yang menyebabkan performa kelas *Geminivirus* sangat rendah (*F1-Score* 0.09), karena model kesulitan membedakan bimen warna kuning penyakit dengan daun sehat yang terpapar pantulan cahaya terang matahari.

Hasil dari evaluasi ketangguhan data lapangan ini memberikan kesimpulan ilmiah yang sangat berharga bagi pengembangan sistem di masa mendatang. Kendati model klasifikasi hibrida ini mampu mencapai performa akurasi yang tinggi pada lingkungan terkontrol (dataset internal), model berbasis fitur tradisional (*handcrafted features*) terbukti memiliki limitasi generalisasi yang besar pada lingkungan liar luar ruangan (*unseen data*) yang memiliki tingkat variansi spasial tinggi.

Untuk mengatasi kendala struktural tersebut, direkomendasikan secara kuat bagi penelitian selanjutnya untuk mengintegrasikan algoritma berbasis *Deep Learning* khusus untuk fase lokalisasi objek, seperti *Object Detection* atau *Semantic Segmentation* (misalnya menggunakan arsitektur YOLO atau U-Net). Modul ini wajib ditanamkan pada aplikasi gawai sebagai tahap *preprocessing* utama sebelum proses ekstraksi fitur hibrida dijalankan. Hal tersebut bertujuan untuk mengisolasi dan memotong (*cropping*) objek daun cabai utama secara ketat dari *background noise* lapangan, sehingga kalkulasi parameter matriks warna HSV, tekstur GLCM, dan kerapatan tepi Canny yang dihitung oleh model XGBoost murni berasal dari area daun yang terinfeksi tanpa adanya distorsi lingkungan pertanian terbuka.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

4.6 Analisis Integrasi Sistem

Setelah pengujian performa model *machine learning* dan antarmuka *mobile* dilakukan secara terpisah, tahap selanjutnya adalah menganalisis bagaimana seluruh komponen utama sistem yaitu aplikasi *client* (Flutter), gerbang antarmuka pemrograman (FastAPI Backend), dan arsitektur basis data bekerja secara sinkron dalam skenario penggunaan nyata (*end-to-end pipeline*).

1. Analisis Latensi dan Respon Sistem (*Real-time Detection*)

Berdasarkan pengujian integrasi transaksional data antara aplikasi Flutter dan backend FastAPI, rata-rata durasi waktu (*latency*) yang dibutuhkan sejak penekanan tombol kirim berkas foto hingga munculnya informasi hasil prediksi pada layar gawai pengguna dikalkulasikan secara empiris sebagai berikut:

- **Proses Ekstraksi Fitur dan Inferensi di Server:** 0,2893 detik
  Nilai ini diambil secara absolut berdasarkan variabel penanda waktu internal backend (server_inference_time_sec) saat pipa ekstraksi ruang warna HSV, tekstur GLCM, operator tepi Canny, serta komputasi klasifikasi berjalan di server Hugging Face
- **Latensi Jaringan (*Network Latency - HTTP Request/Response*):** 0,2500 detik
  Nilai estimasi durasi transmisi berkas citra kompresi dari gawai ke server, serta pengiriman kembali *payload* data JSON melalui jaringan seluler nirkabel/WiFi.
- **Total Waktu Respons (*Total Response Time*): 0,5393 detik.**

Hasil pengujian integrasi ini menunjukkan bahwa alur pipa data (*data pipeline*) dari komponen kamera menuju server inferensi yang dirancang memiliki performa responsif yang sangat baik. Akumulasi *delay* yang berada di angka **0,5393 detik**—di mana nilai ini berada jauh di bawah ambang batas kesadaran manusia (*sub-second latency* / < 1 detik)—membuat aplikasi ini sangat layak untuk diimplementasikan oleh petani di area lahan pertanian terbuka karena tidak memiliki hambatan waktu tunggu (*delay*) yang berarti.

2. Efektivitas Penyimpanan Data Historis (MongoDB dan/atau *Local State*)

Sesuai dengan batasan masalah yang ditetapkan, pengelolaan penyimpanan data historis transaksi klasifikasi citra pada sistem ini menerapkan skema kombinasi antara Non-Persisten Lokal (*Volatile In-Memory State*) pada sisi *client* dan pencatatan riwayat pada sisi server menggunakan MongoDB.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

- **Sisi *Client* (Aplikasi Mobile):** Data transaksi hasil prediksi disimpan ke dalam variabel *state management* lokal (RAM perangkat). Pendekatan ini terbukti efektif memotong biaya komputasi pengalamatan memori (*overhead cost*) pada gawai, sehingga proses *rendering* komponen daftar riwayat menjadi sangat gegas. Namun, data ini bersifat sementara dan akan dibersihkan saat aplikasi dihentikan sepenuhnya (*force close*), sesuai dengan hasil uji *Black-box* nomor 8.
- **Sisi *Server* (Infrastruktur Data):** Setiap kali proses inferensi berhasil dilakukan dengan status "*success*", backend FastAPI secara asinkron (*asynchronous*) mengirimkan manifes data ringkas berupa nama file, label klasifikasi panjang (label), label pendek (short_label), serta persentase skor konfidensi (confidence) sebesar **83,75%** ke basis data MongoDB *Document-Based*. Arsitektur NoSQL MongoDB terbukti sangat efektif untuk menangani data log klasifikasi terstruktur yang memiliki intensitas penulisan tinggi (*high-write throughput*) tanpa membebani performa aplikasi *mobile*.

3. Keandalan Arsitektur *Client-Server*

Pemisahan beban kerja (*separation of concerns*) antara aplikasi *client* (Flutter) dan server (FastAPI yang di-host pada Hugging Face Spaces) menunjukkan tingkat keandalan yang tinggi. Pembagian tugas ini mengisolasi proses komputasi matriks matematika yang berat pada tahap *feature engineering* (ekstraksi ruang warna HSV, tekstur GLCM, dan operator tepi Canny) agar dieksekusi sepenuhnya oleh perangkat server.

Dengan demikian, gawai milik petani yang memiliki spesifikasi perangkat kelas menengah ke bawah (*low-to-mid end devices*) tidak akan mengalami *overheating* (panas berlebih) atau penurunan performa RAM secara drastis saat menjalankan aplikasi. Selain itu, arsitektur ini terbukti memiliki tingkat toleransi kesalahan jaringan (*network fault-tolerance*) yang baik; apabila koneksi menuju server terputus (*timeout* atau server Hugging Face sedang *down*), sistem *client* mampu menangkap pengecualian (*exception handling*) tersebut secara anggun melalui pesan *SnackBar* peringatan tanpa menyebabkan kegagalan sistem total (*application crash*).

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini telah berhasil mengembangkan sistem klasifikasi penyakit tanaman cabai dengan menerapkan kombinasi metode ekstraksi fitur dan model pembelajaran mesin (*machine learning*). Berdasarkan hasil penelitian, pengujian fungsional, serta evaluasi pengguna yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Penerapan metode *feature engineering* yang mengombinasikan fitur warna (*Hue Saturation Value* / HSV), tekstur (*Gray-Level Co-occurrence Matrix* / GLCM), dan bentuk (*Canny Edge Detection*) terbukti sangat efektif dalam mengenali karakteristik visual penyakit daun cabai (infeksi *Geminivirus*, *Cercospora*, *Leaf Curl*, dan kondisi Sehat). Penggunaan *Principal Component Analysis* (PCA) dengan konfigurasi *explained variance* 0,95 berhasil memberikan **standar optimalisasi terbaik** karena mampu mereduksi penggunaan memori biner model hingga 40% (dari 421 KB menjadi 253 KB) dengan penurunan akurasi yang sangat minimal (hanya ∼1,5%), sehingga menghasilkan model yang ringan namun tetap akurat untuk diaplikasikan pada citra penyakit tanaman.
2. Pengembangan prototipe aplikasi *mobile* berbasis Flutter yang terintegrasi dengan *backend* FastAPI dan basis data MongoDB telah berhasil memberikan **kemudahan akses yang optimal bagi petani cabai** di lapangan. Keberhasilan interaksi antarmuka ini dibuktikan melalui pengujian *System Usability Scale* (SUS) dengan skor rata-rata **78,23** yang masuk ke dalam kategori ***Acceptable*** **(Grade B, *Excellent*)**, serta didukung oleh hasil pengujian *Black-box* yang menunjukkan seluruh fungsi utama aplikasi berjalan dengan valid dan bebas dari galat (*error*).
3. Sistem terbukti mampu **mempercepat proses identifikasi penyakit tanaman secara instan** melalui pencapaian *sub-second latency* dengan total waktu respons (*Total Response Time*) sebesar **0,5393 detik** (proses inferensi server XGBoost 0,2893 detik dan latensi jaringan 0,2500 detik). Kecepatan deteksi di bawah 1 detik ini memberikan keunggulan bagi petani untuk mengetahui jenis infeksi secara *real-*

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

time sehingga tindakan pengobatan tanaman dapat dilakukan secara cepat dan tepat sasaran.

4. Melalui performa akurasi model yang tinggi (mencapai 98,75% untuk model XGBoost tanpa PCA dan 98,00% dengan konfigurasi PCA 0,95), sistem ini mampu **mengurangi risiko kerugian ekonomi petani akibat keterlambatan deteksi manual**. Kemampuan deteksi yang akurat dan instan melalui gawai ini secara langsung berkontribusi dalam menekan laju penyebaran wabah di lahan terbuka sebelum meluas dan menyebabkan gagal panen.
5. Meskipun model menunjukkan performa generalisasi yang sangat tinggi pada *test set* internal (99%), sistem masih mengalami penurunan performa pada data eksternal dari internet akibat tumpang tindih (*overlapping*) fitur visual pada analisis *Dot Plot*. Hal ini mengindikasikan adanya ketergantungan model terhadap karakteristik latar belakang (*background*) dan pencahayaan dari *dataset* pelatihan, yang menuntut adanya tindakan pra-pemrosesan khusus berupa pemotongan (*cropping*) citra secara terfokus pada objek daun.

5.2 Saran

Untuk pengembangan penelitian selanjutnya, disarankan beberapa hal sebagai berikut:

1. Menambahkan variasi data citra dengan latar belakang (background) yang lebih kompleks (langsung di lahan terbuka) untuk menguji ketangguhan model terhadap noise lingkungan
2. Mengembangkan fitur pemetaan sebaran penyakit berbasis GPS yang teintegrasi dengan data di MongoDB untuk memberikan peringatan dini(early warning system) bagi petani di wilayah sekitar
3. Memperluas keberagaman dataset asli (sebelum augmentasi) terutama pada kelas yang memiliki sampel sedikit seperti Cerospora, guna meningkatkan daya tahan (robustness) model terhadap data luar
4. Mengintegrasikan algoritma segmentasi citra atau Object Detection sebelum klasifikasi, agar model dapat secara otomatis memfokuskan ekstraksi fitur hanya pada area daun, tanpa terpengaruh oleh latar belakang citra yang kompleks.

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

DAFTAR PUSTAKA

Anonim. 2006. SNI 4480-2016. Scribd. http://id.scribd.com/document/322722386/SNI-4480-2016. [15 Januari 2026]

Yunandra. et al. 2025. Sosialisasi Budidaya Dan Pencegahan Serangan Virus Gemini Pada Tanaman Cabai Di Gapoktan Beken Jaya Desa Benai Keci Kabupaten Kuantan Singingi. http://conference.unri.ac.id/index.php/unricsce/article/view/631/600. [23 Juni 2026]

Anonim. 2026. Keriting Daun Pada Cabai: Ini Penyebab Dan Cara Pengendaliannya. Infarm. https://infarm.co.id/detail-artikel/keriting-daun-pada-cabai-ini-penyebab-dan-cara-pengendaliannya. [23 Juni 2026]

A'yun, Q., Fitriani, B. N., Fardhani, D. M. dan Nugraheni, I. A. 2025. "Analisis Insidensi Virus Gemini Pada Tanaman Cabai (Capsicum Frutescens)". Prosiding Seminar Nasional Penelitian Dan Pengabdian Kepada Masyarakat LPPM Universitas 'Aisyiyah Yogyakarta, 3, 823–829.

Ayu, S., Taufik, M., M, R., Hasan, A., Widodo, C. J., Botek, M. dan Huttni, L. O. 2025. "Pengukuran Keparahan Gejala Penyakit Kuning Dan Kandungan Nitrogen Tanaman Cabai Terinfeksi Virus Gemini Berbasis Spektral Biosensor: Measurement Of Yellow Disease Symptom Severity And Nitrogen Content Of Chili Plants Infected With Gemini Virus Based On Spectral Biosensor". Agrokompleks, 25(1), 10–21. https://doi.org/10.51978/japp.25.1.34

Chen, T. dan Guestrin, C. 2016. "XGBoost: A Scalable Tree Boosting System". Proceedings Of The 22nd ACM SIGKDD International Conference On Knowledge Discovery And Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Fakhira, A. G., Fardhani, D. M. dan Nugraheni, I. A. 2025. "Serangan Yellow Leaf Curl Virus (YLCV) Pada Tanaman Cabai (Capsicum Spp.) Di Daerah Nogotirto". Prosiding Seminar Nasional Penelitian Dan Pengabdian Kepada Masyarakat LPPM Universitas 'Aisyiyah Yogyakarta, 3, 784–790.

Gonzalez, R. C. dan Woods, R. E. 2017. Digital Image Processing. Fourth Edition, Global Edition. New York: Pearson.

Kiran, S. M. dan Chandrappa, D. N. 2023. "Plant Leaf Disease Detection Using Efficient Image Processing And Machine Learning Algorithms". Journal Of

Robotics And Control (JRC), 4(6), 840–848. https://doi.org/10.18196/jrc.v4i6.20342

Kurniawati, R., Nurhayati, N. dan Sanjaya, A. 2021. "Implementasi Metode Canny Edge Detection Untuk Ekstraksi Ciri Geometri Pada Penyakit Daun Tanaman Hortikultura". Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK), 8(4), 789–796. https://jtiik.ub.ac.id/index.php/jtiik/article/view/3215

Noola, D. A. dan Basavaraju, D. R. 2022. "Corn Leaf Image Classification Based On Machine Learning Techniques For Accurate Leaf Disease Detection". International Journal Of Electrical And Computer Engineering (IJECE), 12(3), 2509–2516. https://doi.org/10.11591/ijece.v12i3.pp2509-2516

Nurfadhilah, L. et al. 2024. "Intensitas Penyakit Cabai Rawit (Capsicum Frutescens) Di Desa Sumber Baru Kecamatan Mesuji Raya Kabupaten Ogan Komering Ilir".

Nurmadinah, N., Wajidi, F. dan Arifin, N. 2025. "DETEKSI PENYAKIT DAUN CABAI MENGGUNAKAN KOMBINASI GLCM DAN HSV DENGAN KLASIFIKASI SVM". Insect (Informatics And Security): Jurnal Teknik Informatika, 11(2), 178–189. https://doi.org/10.33506/insect.v11i2.4820

Pradana, A. R. dan Nugroho, S. 2023. "Karakteristik Ekstraksi Ciri Visual Pada Penyakit Cercospora Tanaman Hortikultura". Jurnal Teknologi Dan Sistem Komputer, 11(1), 12–19. https://jtsiskom.undip.ac.id/article/view/14352

Prasetyo, T. A. et al. 2023. "Corn Plant Disease Classification Based On Leaf Using Residual Networks-9 Architecture". International Journal Of Electrical And Computer Engineering (IJECE), 13(3), 2908–2920. https://doi.org/10.11591/ijece.v13i3.pp2908-2920

Prasetiyowati, M. I., Maulidevi, N. U. dan Surendro, K. 2020. "Feature Selection To Increase The Random Forest Method Performance On High Dimensional Data". International Journal Of Advances In Intelligent Informatics, 6(3), 303. https://doi.org/10.26555/ijain.v6i3.471

Putra, I. M., Rahman, F. dan Sitompul, B. 2021. "Morphological And Geometric Alterations In Solanaceae Leaf Curl Diseases Using Edge And Color Segmentation". International Journal Of Agronomy, 2021, 1–10. https://www.hindawi.com/journals/ija/2021/8834210/

Rahmanto, O., Julianto, V. dan Arrahimi, A. R. 2025. "Evaluating Random Forest Algorithm: Detection Of Palm Oil Leaf Disease". Brilliance: Research Of Artificial Intelligence. https://jurnal.itscience.org/index.php/brilliance/article/view/4798. [18 Januari 2026]

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Sari, D. P. dan Hermanto, T. 2022. "Analisis Pola Klorosis Daun Akibat Infeksi Virus Mosaik Pada Tanaman Solanaceae". Jurnal Fitopatologi Indonesia, 18(2), 45–52. https://journal.ipb.ac.id/index.php/jfito

Sembiring, F. dan Wahyuni, S. 2023. "Analisis Komparasi Fitur Tekstur Dan Bentuk Canny Dalam Klasifikasi Penyakit Tanaman Cabai Berbasis Pohon Keputusan". Jurnal RESTI (Rekayasa Sistem Dan Teknologi), 7(2), 345–352. https://jurnal.iaii.or.id/index.php/RESTI/article/view/4982

Studer, S. et al. 2021. "Towards CRISP-ML(Q): A Machine Learning Process Model With Quality Assurance Methodology". arXiv. https://doi.org/10.48550/arXiv.2003.05155

Sulandari, S., Suseno, R., Hidayat, S. H., Harjosudarmo, J. dan Sosromarsono, S. 2006. "Deteksi Dan Kajian Kisaran Inang Virus Penyebab Penyakit Daun Keriting Kuning Cabai". HAYATI Journal of Biosciences, 13(1), 1–6. https://doi.org/10.1016/S1978-3019(16)30371-0

Wiridannisaa, N., Kasim, N. N. dan Prihatin, P. 2025. "Identification Of Begomoviruses Causing Yellow Curl Disease In Chilli In West Java". Jurnal Biologi Tropis, 25(3), 4231–4238. https://doi.org/10.29303/jbt.v25i3.9838

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1 Transkrip Wawancara

Daftar Singkatan:

- **P (Peneliti):** Okta Gabriel Sinsaku Sinaga
- **N (Narasumber/Pakar):** Validator Lapangan/Pakar Pertanian Cabai

P: Jadi, di sini ada 4 kelas penyakit cabai yang menjadi topik di penelitian skripsi saya. Yang pertama itu adalah bintik daun. Ini salah satunya. Mungkin kita fokusnya ke bintik daun dulu. Jadi, kira-kira Bang, ini benar tidak bintik daun dari gambar ini, Bang? Dari hasil identifikasinya?

N: Iya, betul.

P: Apakah ada penyakit lain yang Abang identifikasi dari gambar ini? Iya, misalnya kuning daun atau keriting, gitu. Iya, jadi kira-kira kayak gitu loh, Bang. Nah, jadi kita boleh lanjut lagi. Ini Bang, gimana? Dari bintik daun, kuning daun, sama keriting. Kelihatan tidak? Oh, kabur. Apakah masih ada kegiatan lagi tidak setelah ini? Oh iya, iya. Kalau sudah kelihatan bilang ya, Bang. Atau masih belum kelihatan juga mau dikasih tahu saja.

N: Sudah, oke.

P: Bagaimana, Bang? Apakah ada bintik daun atau kuning daun atau keriting gitu? Dari bintik daun, kuning daun, sama keriting.

N: Iya. Antraknosa.

P: Oh, oke. Terima kasih, Bang. Kalau ini bagaimana, Bang? Tidak apa-apa sih, kita main cepat saja siapa tahu. Oh, oke. Sudah terlihat Bang yang selanjutnya? Oh, oke. Kalau dari daunnya Bang, ini fokusnya sih daunnya. Iya, daun. Lanjut, Bang.

N: Oke, siap. Kuning. Kuning. Betul.

P: Bintik daunnya? Bintik daunnya ada tidak, Bang?

N: Masih ada. Oke, siap.

P: Ini kalau ini bagaimana, Bang?

N: Oke.

P: Oke, siap. Oke Bang, ini lanjut. Masih ada ya?

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(lanjutan)

N: Oke.

P: Oke, kalau ini lanjut lagi.

N: Parasit. Dapat kena semua ya.

P: Oke. Ini Bang, lanjut Bang.

N: Semua, oke.

P: Kenapa aku bilang hemat-hematlah ngomong Abang, soalnya lumayan banyak ini. Sudah kelihatan belum Bang selanjutnya? Oh, oke. Oke, oke. Syukur, syukur. Oke. Oke, Bang. Sudah, oke. Kamu kemarin di Jabodetabek, di mana Bang? Di depan. Masukkan. Ada keritingnya. Lanjut, Bang. Kalau ini.

N: Gosong. Kena ketiganya.

P: Kalau ini Bang?

N: Sudah.

P: Masih ya? Oke, lanjut.

N: Masih. Oh, ini ada gejala kuning.

P: Lanjut, Bang. Lanjut lagi. Jadi, ini apa sih? Eh, Abang kemarin sempat belajar *machine learning* tidak sih di kuliah, Bang? *Machine learning*. Oh, iya. Enggal, karena aku pernah dengar orang teknik mesin di kampusku pun dapat dia *machine learning*. Jadi, kayak gininya, kayak data-data ini semua, perlulah divalidasi sama Abang yang sudah sekian tahun di pertanian cabai buat dimasukkan ke komputer. Jadi, harus data yang benar gitu. Nah, begitu. Iya. Oke. Kalau ini Bang, gimana? Oke, kalau ini Bang? Lanjut apa? Sedikit lihat. Kalau ini Bang? Oke. Oke, kalau ini Bang bagaimana? Oke. Eh, *sorry*. Dari mana tadi? Ini ya sekarang ya? Keritingnya? Tidak ada ya? Nah, ini. Masih kabur Bang? Apa perlu di-*hotspot*-kan si Vincent, Bang? Oh, oke. Oke. Nah, kalau ini bagaimana? Oke, kalau kuning dan keritingnya? Oke. Oke, lanjut. Ini berarti masih bintik ya? Kalau keriting? Oke, lanjut Bang.

N: Oh, ada.

P: Di ujung gitu ya? Nah, kalau ini berarti? Sudah jelas? Oke, oke Bang. Oke, lanjut. Ini sama ya? Oke. Oke, lanjut lagi. Kalau yang ini? Ada keritingnya? Oke. Betul. Yang paling dekat itulah, Bang. Ya, tapi kan memang sistemnya biar itu, biar mengidentifikasi penyakitnya, jadi dia dikasih makanya penyakit. Tidaklah. Justru kita

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(lanjutan)

harus tahu mengidentifikasi penyakitnya, biar kita tahu penanganan yang tepat. Begitu, Bang.

N: Keriting. Oke.

P: Nah, ini bagaimana Bang? Ini tampaknya hijau ini. Atau jangan-jangan? Oh. Oke, siap. Oke, lanjut. Nah, kalau ini Bang? Masih ada?

N: Masih ada. Lanjut.

P: Kalau ini?

N: Keriting.

P: Kuning ada tidak Bang? Atau ini cuma kurang nutrisi saja? Oke. Oke, kita lagi Bang, lanjut. Oke. Ini? Oke. Oke. Oke, siap. Oh, *branding*-nya ada. Oke. Kalau ini Bang? Lanjut. Lanjut. Sama? Masih sama? Tapi memang sengaja aku carinya yang kayak begitu sih. Oke, ini Bang bagaimana Bang?

N: Keriting.

P: Bintik daunnya ada tidak? Nah, itu kayaknya harusnya semua gambar deh.

N: Ada.

P: Oke, siap. Ini terakhir untuk kelas *Cercospora* atau bintik daunnya ini. Terakhir mana? Ada lah ya. Oke, Bang. Berarti lanjutlah, mau ke kelas yang lain.

P: Apa yang terjadi? Itu apa? Apakah itu lampu? Apakah itu lampu? Oke. Buahnya tidak basah. Oke, lanjutlah. Oke, lanjut lagi. Ini bagus banget. Oke, kita cek. Ini? Ini? Oh, antraknosa ya? Antraknosa. Berarti tidak kuning ya? Oke. Oke, lanjut. Ini masih sama ya berarti ya? Oke, kalau yang ini? Sehat? Kenapa? Sakit?

N: Iya.

P: Oke. Oke, siap. Oke, lanjutlah. Ini? Oke, lanjut. Ini? Oke. Oke, siap. Kalau ini bagaimana, Pak? Ini?

N: Oh, ini yang tadinya. Ini bintik daunnya. Oh, kuning. Ada yang lain. Ini semua. Ini.

P: Sama ya, Bang? Ini? Ini ada kuningnya?

N: Ada.

P: Ini sama ya? Kalau ini? Ini? Oke, ini lanjutlah. Ini? Ini juga sama. Ini? Ini juga sama. Tapi ada kuningnya. Bintik. Lanjut. Kalau ini? Ini? Lanjut. Ini? Masa? Lanjut. Lanjut. Oke, untuk prosesnya sudah habis. Kita pindah proses lagi. Ini keritingnya. Ini? Jadi, yang ini utamanya keriting. Lanjut. Oh, oke. Oh, oke. Oke, lanjut, Bang.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(lanjutan)

Sehat? Selamat? Selamat? Lanjut. Oke. Eh, salah, salah. Ini? Ini? Ini? Ini? Sudah kelihatan?

N: Sehat.

P: Lanjut. Oh, sehat? Oh, sehat? Oke, lanjut. Oke. Ini? Ada keritingnya? Memang ada beberapa gambar yang ini saya identifikasi sama atau lebih. Oke, kalau ini, Bang?

N: Keriting.

P: Oke, lanjut lagi. Kalau ini?

N: Sehat.

P: Ini keritingnya. Ini? Lanjut lagi. Yang terakhir? Lanjut.

N: Sehat. Sehat. Sehat.

P: Yang terakhir sehat? Oke, Bang. Tinggal satu lagi, satu kelas lagi, 30 gambar. Tapi ini tentang sehat. Ini semoga sehat semua. Kalau sudah kelihatan bilang ya, Bang. Oke, jadi ini utamanya sehat. Kalau sehat, bilang saja sehat. Oke? Oh, tidak? Kenapa, Bang? Ada sakit apa begitu di sana?

N: Kurang vitamin. Oh, kurang pupuk daun. Oke.

P: Tapi untuk penyakit yang tiga tadi, ada tidak terdeteksi di sini? Oke. Oh, oke. Kalau ini?

N: Sehat.

P: Lanjut. Oh, ada? Oke. Oke, lanjut. Ada. Lanjut lagi. Oke.

N: Ada bercak daun. Bercak daun.

P: Oke, kalau ini?

N: Ada bercak daun juga.

P: Kalau ini?

N: Oke, sehat.

P: Lanjut.

N: Sehat.

P: Lanjut. Kalau ini? Ini?

N: Bercak daun.

P: Lanjut. Oke. Lanjut.

N: Bercak daun. Sehat.

P: Oke, kalau ini?

(lanjutan)

N: Sehat.

P: Lanjut.

N: Sehat.

P: Oke. Ini? Oke, lanjut.

N: Bercak daun.

P: Lanjut. Oke. Oke, lanjut.

N: Sehat.

P: Lanjut lagi. Oke. Lanjut.

N: Sehat. Ini kuning. Bercak daun.

P: Yang ini? Oke, lanjut.

N: Sehat.

P: Lanjut.

N: Sehat.

P: Terakhir. Oke, sudah, Bang.

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 2 Sampel Citra Dataset Publik dan Lapangan

No	Nama Kelas / Gejala	Dataset Publik (Kaggle/Roboflow)	Data Lapangan (Siborongborong)
1	1. Geminivirus - Gejala: Daun menguning, mengeriting, tulang daun memutih.		
2	2. Cercospora (Bercak Daun) - Gejala: Bercak bulat abu- abu dengan pinggiran cokelat tua pada daun.		
3	3. Leaf Curl Virus (Daun Mengeriting) - Gejala: Helaian daun menekuk ke atas atau ke bawah, daun mengerut, menyempit, dan menjadi lebih tebal/kaku.		

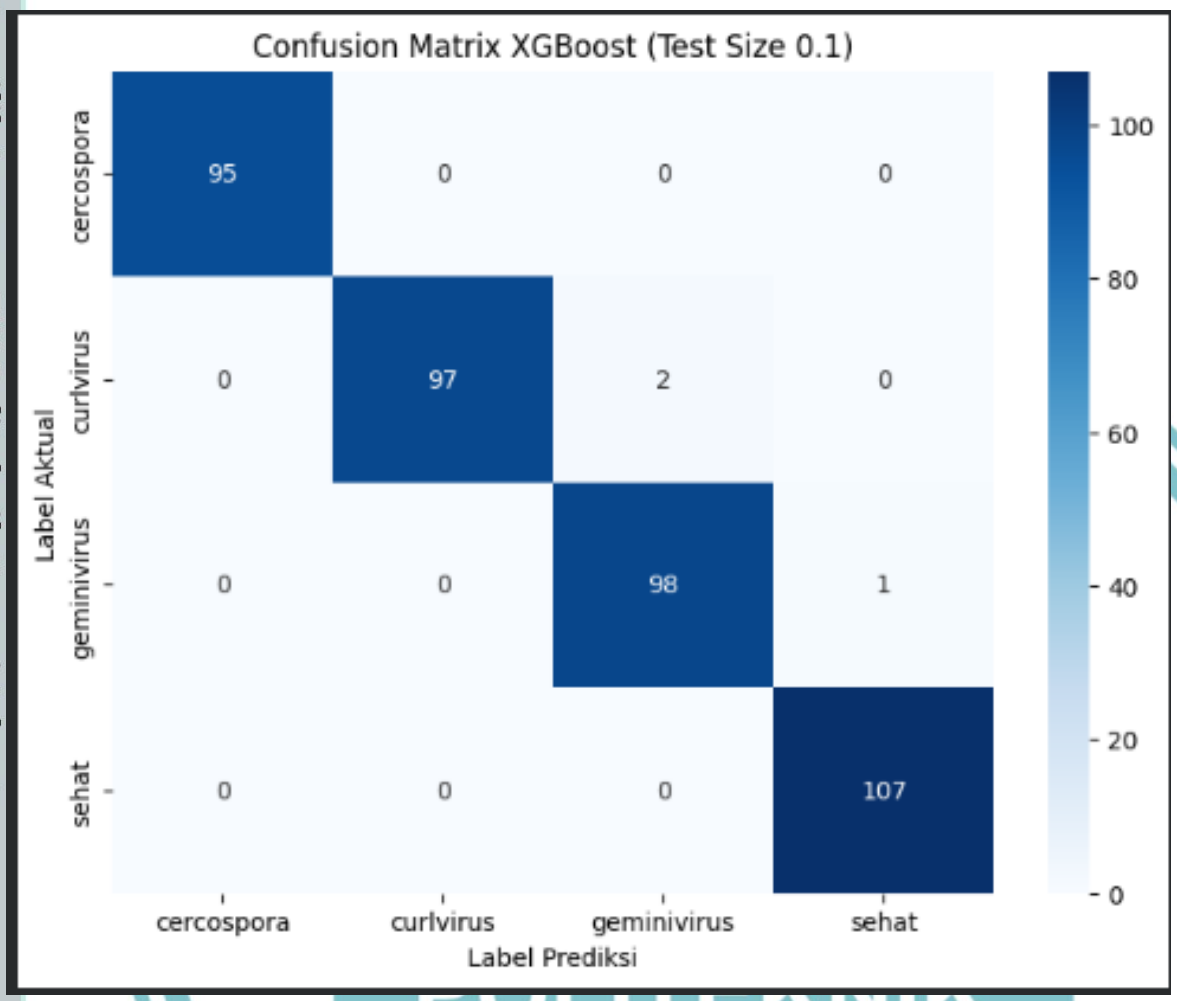
(lanjutan)

4	4. Daun Sehat (Healthy) - Gejala: Permukaan daun mulus, warna hijau merata, tidak ada bercak, dan bentuk morfologi daun sempurna/normal.		

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 3 Confussion Matrix



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Lampiran 4 Lembar Justifikasi Pakar dan Validasi Data Lapangan

Kelas Cercospora

No. Sampel	Cercospora	GeminiVirus	CurlVirus	Sehat	Catatan Pakar
1	1	1	0	0	
2	1	1	1	0	
3	1	0	1	0	
4	1	0	0	0	
5	1	1	0	0	
6	1	1	1	0	
7	1	1	1	0	
8	1	1	1	0	
9	1	1	1	0	
10	1	0	0	0	
11	1	0	1	0	
12	1	1	1	0	
13	1	1	1	0	
14	1	1	0	0	
15	1	0	0	0	
16	1	0	0	0	
17	1	1	0	0	
18	1	1	0	0	
19	1	1	0	0	
20	1	1	0	0	
21	1	1	0	0	
22	1	1	0	0	
23	1	1	0	0	
24	1	1	1	0	
25	1	1	0	0	
26	1	1	0	0	
27	1	1	0	0	
28	1	1	1	0	
29	1	1	1	0	
30	1	0	1	0	
31	1	0	1	0	
32	1	0	1	0	
33	1	0	1	0	
34	1	1	0	0	
35	1	1	1	0	
36	1	1	0	0	
37	1	1	0	0	
38	1	1	0	0	
39	1	1	0	0	

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(lanjutan)

40	1	0	1	0	
41	1	1	0	0	

Kelas GeminiVirus

No. Sampel	Cercospora	GeminiVirus	CurlVirus	Sehat	Catatan Pakar
1	0	0	1	0	
2	0	0	1	0	
3	0	1	1	0	
4	1	0	1	0	
5	0	0	0	1	
6	1	1	0	0	
7	1	1	0	0	
8	1	1	0	0	
9	1	1	0	0	
10	1	1	1	0	
11	0	1	0	0	
12	0	1	1	0	
13	0	1	1	0	
14	0	1	0	0	
15	1	1	0	0	
16	1	1	0	0	
17	1	1	0	0	
18	0	0	0	0	
19	1	1	0	0	
20	1	1	0	0	
21	1	1	0	0	
22	1	1	0	0	
23	0	0	0	1	
24	1	1	0	0	
25	1	1	0	0	
26	1	0	1	0	
27	1	1	0	0	
28	1	1	0	0	
29	1	1	1	0	
30	1	0	0	0	
31	1	1	1	0	
32	0	0	1	0	
33	1	1	0	0	
34	1	1	0	0	
35	1	1	0	0	

(lanjutan)

36	1	1	0	0	
37	1	1	0	0	
38	1	1	0	0	
39	0	0	0	1	
40	0	1	0	0	
41	1	1	1	0	
42	0	0	1	0	
43	0	0	1	0	
44	0	0	1	0	
45	1	1	1	0	
46	1	0	0	0	
47	0	1	1	0	
48	0	0	0	1	
49	1	1	1	0	

Kelas Curl Virus

No. Sampel	Cercospora	GeminiVirus	CurlVirus	Sehat	Catatan Pakar
1	0	0	0	1	
2	0	0	0	1	
3	0	0	1	0	
4	0	0	1	0	
5	0	0	1	0	
6	0	0	1	0	
7	0	0	1	0	
8	0	0	1	0	
9	0	0	1	0	
10	0	0	1	0	
11	0	0	1	0	
12	0	0	1	0	
13	0	0	1	0	
14	0	0	1	0	
15	0	0	1	0	
16	0	0	1	0	
17	0	0	1	0	
18	0	0	0	1	
19	0	0	1	0	
20	0	0	0	1	
21	0	0	1	0	
22	0	0	1	0	
23	0	0	0	1	

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(lanjutan)

24	0	0	1	0	
25	0	0	1	0	
26	0	0	1	0	
27	0	0	0	1	
28	0	0	1	0	
29	0	0	1	0	
30	0	0	1	0	
31	0	0	1	0	
32	0	0	1	0	
33	0	0	1	0	
34	0	0	1	0	
35	0	0	1	0	
36	0	0	0	1	
37	0	0	0	1	
38	0	0	1	0	
39	0	0	1	0	
40	0	0	1	0	
41	1	1	1	0	
42	0	0	1	0	
43	0	0	1	0	
44	0	0	0	1	
45	1	1	1	0	
46	0	0	1	0	
47	0	0	1	0	
48	0	0	1	0	
49	0	0	1	0	
50	0	0	1	0	
51	1	1	1	0	
52	0	0	1	0	
53	0	0	0	1	
54	0	0	0	1	
55	0	0	0	1	
56	0	0	1	0	
57	0	0	0	1	

Kelas Sehat

No. Sampel	Cercospora	GeminiVirus	CurlVirus	Sehat	Catatan Pakar
1	0	0	0	1	
2	0	0	0	1	pupuk daun
3	0	0	0	1	

(lanjutan)

4	1	0	0	0	
5	1	0	0	0	
6	1	0	0	0	
7	1	0	0	0	
8	1	0	0	0	
9	0	0	0	1	
10	0	0	0	1	
11	0	0	0	1	
12	0	0	1	0	
13	1	0	0	0	
14	0	1	0	0	
15	1	0	0	0	
16	0	0	0	1	
17	0	0	0	1	
18	0	0	0	1	
19	0	0	0	1	
20	0	0	1	0	
21	1	0	0	0	
22	0	0	0	1	
23	0	0	0	1	
24	1	0	0	0	
25	0	0	0	1	
26	0	1	0	0	
27	1	0	0	0	
28	0	0	0	1	
29	0	0	0	1	
30	0	0	0	1	

© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta