

Deteksi Penyakit Tanaman Padi Melalui Google Gemini API pada Aplikasi Mobile

Ciayu Pahlawati Suryani^{1*}, Imam Maruf Nugroho², Yudhi Raymond Ramadhan³

¹Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana Purwakarta, Indonesia

^{2,3}Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana Purwakarta, Indonesia

Email: ¹ciayupahlawati26@wastukencana.ac.id, ²imam.ma@wastukencana.ac.id,

³yudhi.raymond@wastukencana.ac.id

INFORMASI ARTIKEL

Histori artikel:

Naskah masuk, 26 Juni 2026

Direvisi, 6 Juli 2026

Diterima, 31 Juli 2026

Kata Kunci:

Google Gemini API, deteksi penyakit tanaman padi, Flutter, kecerdasan buatan multimodal, aplikasi mobile.

ABSTRAK

Abstract- Rice diseases are among the major factors contributing to reduced agricultural productivity, highlighting the need for rapid and accessible early detection systems for farmers. This study aimed to design and implement a mobile application for rice disease detection using the Google Gemini API based on leaf image analysis. The research adopted a software engineering approach employing the Extreme Programming (XP) methodology, which consisted of seven stages: problem identification, literature review, planning, design, coding, testing, and manuscript preparation. The application was developed using Flutter, Dart, Firebase Authentication, Cloud Firestore, and the Google Gemini API as an Artificial Intelligence as a Service (AIaaS) platform. The results demonstrated that the application successfully identified four major rice diseases, namely rice blast, bacterial leaf blight, brown spot, and tungro, while providing comprehensive information on symptoms, causes, treatment recommendations, and preventive measures. System evaluation was conducted through 56 functional test cases, including unit testing, widget testing, integration testing, pipeline testing, and edge-case testing, achieving a 100% success rate. The findings indicate that integrating the Google Gemini API significantly simplifies the development of AI-based rice disease detection applications by eliminating the need for independent model training while providing a practical multimodal AI solution to support the digital transformation of smart agriculture.

Abstrak- Penyakit tanaman padi merupakan salah satu faktor utama yang menyebabkan penurunan produktivitas pertanian sehingga diperlukan sistem deteksi dini yang cepat dan mudah diakses oleh petani. Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi mobile untuk mendeteksi penyakit tanaman padi menggunakan Google Gemini API berbasis analisis citra daun. Metode penelitian menggunakan pendekatan rekayasa perangkat lunak dengan Extreme Programming (XP) yang meliputi tujuh tahapan, yaitu identifikasi masalah, studi literatur, planning, design, coding, testing, dan penyusunan artikel ilmiah. Aplikasi dikembangkan menggunakan Flutter, Dart, Firebase Authentication, Cloud Firestore, serta Google Gemini API sebagai layanan Artificial Intelligence as a Service (AIaaS). Hasil penelitian menunjukkan bahwa aplikasi mampu mengidentifikasi empat penyakit utama tanaman padi, yaitu blas, hawar daun bakteri, bercak cokelat, dan tungro, serta menghasilkan informasi mengenai gejala, penyebab, rekomendasi penanganan, dan langkah pencegahan. Evaluasi sistem dilakukan melalui 56 skenario pengujian yang mencakup unit testing, widget testing, integration testing, pipeline testing, dan edge case testing, dengan tingkat keberhasilan 100%. Temuan penelitian menunjukkan bahwa pemanfaatan Google Gemini API mampu menyederhanakan pengembangan aplikasi deteksi penyakit tanpa memerlukan proses pelatihan model klasifikasi secara mandiri, sehingga memberikan alternatif implementasi AI multimodal yang praktis untuk mendukung digitalisasi sektor pertanian.

Copyright © 2026 LPPM - STMIK IKMI Cirebon
This is an open access article under the CC-BY license

Penulis Korespondensi:

Ciayu Pahlawati Suryani

Program Studi Teknik Informatika

Sekolah Tinggi Teknologi Wastukencana Purwakarta

Indonesia

1. Pendahuluan

Tanaman padi (*Oryza sativa* L.) merupakan komoditas pangan strategis yang memiliki peran penting dalam mendukung ketahanan pangan nasional. Sebagian besar masyarakat Indonesia menjadikan beras sebagai makanan pokok sehingga keberlanjutan produksi padi menjadi salah satu prioritas pembangunan sektor pertanian. Data Badan Pusat Statistik menunjukkan bahwa luas panen padi di Indonesia pada tahun 2023 mencapai sekitar 10,21 juta hektare dengan produksi sebesar 53,98 juta ton gabah kering giling (Badan Pusat Statistik [BPS], 2024). Meskipun demikian, produktivitas tanaman padi masih menghadapi berbagai tantangan, salah satunya adalah serangan penyakit tanaman yang dapat menyebabkan penurunan hasil panen secara signifikan.

Penyakit tanaman padi merupakan faktor pembatas produksi yang dapat menurunkan kualitas maupun kuantitas hasil panen. Beberapa penyakit yang umum menyerang tanaman padi di Indonesia antara lain penyakit blas, hawar daun bakteri, bercak cokelat, dan tungro (Nurkayah et al., 2024). Penyakit blas yang disebabkan oleh *Pyricularia oryzae* bahkan dilaporkan mampu menyebabkan kehilangan hasil hingga 90% pada varietas yang rentan (Dagsy & As, 2025). Selain itu, serangan penyakit dan hama secara umum diperkirakan menyebabkan kehilangan hasil panen yang cukup besar setiap tahunnya (Rijal et al., 2024).

Kondisi tersebut menunjukkan bahwa deteksi dini penyakit tanaman padi merupakan aspek penting dalam mendukung keberhasilan pengendalian penyakit dan peningkatan produktivitas pertanian.

Permasalahan yang sering dihadapi di lapangan adalah proses identifikasi penyakit masih dilakukan secara manual berdasarkan pengamatan visual oleh petani atau melalui konsultasi dengan penyuluh pertanian. Metode tersebut memiliki beberapa keterbatasan, antara lain ketergantungan pada pengalaman individu, kemiripan gejala antarpengakit, serta keterbatasan jumlah penyuluh pertanian yang tersedia di berbagai wilayah (Kuswanto, 2020). Akibatnya, proses diagnosis sering terlambat dilakukan sehingga penyakit telah menyebar lebih luas sebelum tindakan pengendalian diterapkan. Oleh karena itu, diperlukan suatu solusi yang mampu membantu petani melakukan identifikasi penyakit secara cepat, mudah, dan mandiri.

Perkembangan kecerdasan buatan (Artificial Intelligence) membuka peluang baru dalam pengembangan sistem deteksi penyakit tanaman berbasis citra digital. Berbagai penelitian telah memanfaatkan teknik deep learning dan computer vision untuk mengenali gejala penyakit pada daun tanaman padi. Rijal et al. (2024) mengembangkan sistem klasifikasi penyakit padi menggunakan pendekatan Convolutional Neural Network (CNN) dan memperoleh hasil klasifikasi yang baik pada citra daun padi. Penelitian lain yang dilakukan oleh Rizal et al. (2025) mengimplementasikan CNN dan TensorFlow Lite pada perangkat Android untuk mendeteksi penyakit daun padi secara otomatis. Selain itu, Novantara et al. (2025) memanfaatkan teknik optimasi CNN untuk meningkatkan kemampuan deteksi penyakit dan hama pada daun padi. Hasil penelitian tersebut menunjukkan bahwa pendekatan berbasis pembelajaran mendalam memiliki potensi yang besar dalam mendukung diagnosis penyakit tanaman.

Seiring berkembangnya teknologi kecerdasan buatan generatif, pendekatan baru mulai memanfaatkan model multimodal yang mampu memproses teks dan gambar secara bersamaan. Google Gemini merupakan salah satu model kecerdasan buatan multimodal yang memiliki kemampuan memahami konten visual dan menghasilkan respons berbasis konteks secara terpadu (Google AI for Developers, 2026). Beberapa penelitian telah menunjukkan potensi penggunaan Gemini API pada berbagai bidang aplikasi. Heru et al. (2025) memanfaatkan Gemini untuk identifikasi jenis tanaman berdasarkan citra daun, sedangkan Manoj dan Ihsan (2024) mengintegrasikan model deep learning dengan Gemini API untuk meningkatkan pengelolaan hama tanaman. Pada sektor pertanian yang lebih luas, AgroCare yang dikembangkan oleh Shelke et al. (2025) memanfaatkan Gemini API untuk mendukung analisis kondisi tanaman dan pemberian rekomendasi perawatan berbasis kecerdasan buatan.

Meskipun berbagai penelitian terkait deteksi penyakit tanaman dan pemanfaatan Gemini API telah dilakukan, masih terdapat kesenjangan penelitian yang perlu diperhatikan. Sebagian besar penelitian deteksi penyakit padi masih berfokus pada penggunaan CNN yang memerlukan dataset berlabel dalam jumlah besar serta proses pelatihan model yang relatif kompleks dan memerlukan sumber daya komputasi yang tinggi (Rijal et al., 2024; Rizal

et al., 2025). Di sisi lain, penelitian yang memanfaatkan kemampuan multimodal Google Gemini untuk deteksi penyakit tanaman padi berbasis aplikasi mobile masih sangat terbatas. Padahal, pendekatan ini berpotensi menyederhanakan proses pengembangan sistem karena kemampuan analisis citra telah tersedia pada model yang disediakan melalui layanan API tanpa memerlukan proses pelatihan model secara mandiri.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk merancang dan membangun aplikasi mobile berbasis Android yang memanfaatkan Google Gemini API untuk mendeteksi penyakit tanaman padi melalui analisis citra daun. Aplikasi yang dikembangkan diharapkan mampu membantu petani maupun masyarakat dalam melakukan identifikasi penyakit secara mandiri serta memperoleh rekomendasi penanganan yang sesuai sehingga dapat mendukung upaya pengendalian penyakit tanaman secara lebih cepat dan efektif.

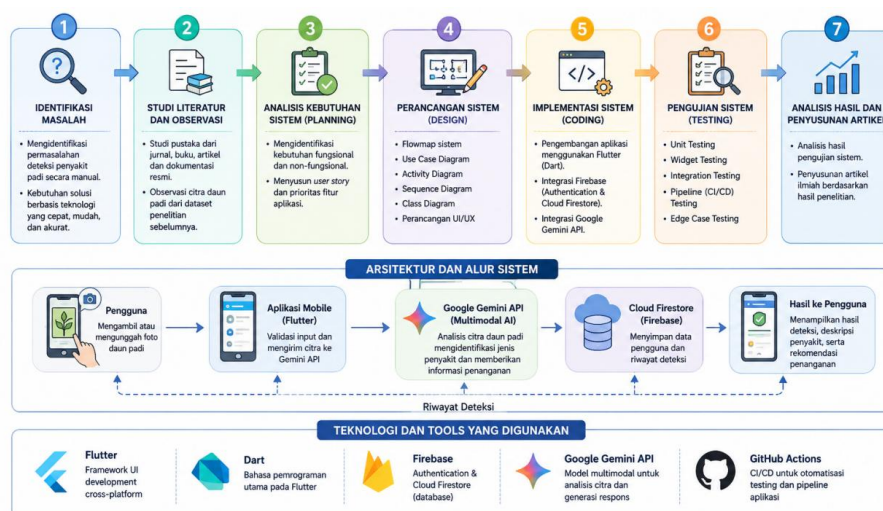
Kebaruan penelitian ini terletak pada pemanfaatan kemampuan multimodal Google Gemini API sebagai layanan Artificial Intelligence as a Service (AIaaS) untuk mendeteksi penyakit tanaman padi langsung pada aplikasi mobile, tanpa melalui proses pengumpulan dataset berlabel maupun pelatihan model klasifikasi secara mandiri sebagaimana umumnya dilakukan pada penelitian berbasis Convolutional Neural Network (CNN) (Rijal et al., 2024; Rizal et al., 2025). Berbeda dengan penelitian tersebut yang berfokus pada

peningkatan akurasi model klasifikasi, kontribusi penelitian ini berada pada aspek integrasi teknologi AI generatif ke dalam arsitektur aplikasi mobile secara utuh, mencakup mekanisme fallback antar-model Gemini serta penyajian informasi gejala, penyebab, rekomendasi penanganan, dan langkah pencegahan secara kontekstual, sehingga menghasilkan pendekatan pengembangan sistem deteksi penyakit tanaman yang lebih ringkas dan mudah diadaptasi dibandingkan pendekatan berbasis pelatihan model secara konvensional.

2. Metode

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan tujuan merancang dan membangun aplikasi mobile untuk mendeteksi penyakit tanaman padi melalui analisis citra daun menggunakan Google Gemini API. Metode pengembangan perangkat lunak yang digunakan adalah Extreme Programming (XP), karena metode ini bersifat iteratif, adaptif, dan sesuai untuk pengembangan aplikasi yang membutuhkan penyesuaian fitur secara bertahap. Dalam konteks penelitian ini, XP digunakan untuk mengelola proses pengembangan mulai dari analisis kebutuhan, perancangan sistem, implementasi aplikasi, sampai pengujian fungsional secara berkelanjutan.

Tahapan penelitian terdiri atas tujuh tahap utama, yaitu identifikasi masalah, studi literatur dan observasi, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, dan analisis hasil dan penyusunan artikel.



Gambar 1. Metode Penelitian

Gambar 1, memperlihatkan bahwa proses penelitian diawali dari pemahaman masalah penyakit tanaman padi, dilanjutkan dengan pengumpulan referensi dan pengamatan citra daun, kemudian diterjemahkan ke dalam tahapan pengembangan perangkat lunak berbasis XP. Setelah aplikasi selesai dikembangkan, sistem diuji menggunakan pendekatan CI/CD testing untuk memastikan bahwa setiap fungsi berjalan sesuai kebutuhan.

Objek penelitian ini adalah tanaman padi (*Oryza sativa* L.) dengan fokus pada empat jenis penyakit utama, yaitu blas, hawar daun bakteri, bercak cokelat, dan tungro. Pemilihan objek tersebut didasarkan pada tingginya dampak serangan penyakit terhadap produktivitas padi dan kebutuhan petani terhadap alat bantu identifikasi penyakit yang mudah digunakan. Data yang digunakan dalam penelitian ini diperoleh melalui studi pustaka dan observasi tidak langsung terhadap citra daun padi yang menunjukkan gejala penyakit. Studi pustaka dilakukan dengan menelaah jurnal ilmiah, buku, artikel, dan dokumentasi resmi yang berkaitan dengan penyakit tanaman padi, kecerdasan buatan, Google Gemini API, Flutter, serta metode XP.

Observasi citra dilakukan untuk memahami karakteristik visual penyakit, seperti perubahan warna daun, bercak, pola kerusakan, dan kondisi visual lain yang menjadi dasar dalam penyusunan kebutuhan sistem.

Untuk kebutuhan pengujian fungsional dan edge case testing, disusun kumpulan citra uji yang mencakup representasi citra daun padi bergejala blas, hawar daun bakteri, bercak cokelat, dan tungro, serta citra kondisi khusus seperti gambar buram, gambar bukan objek daun, dan gambar berukuran besar. Citra uji tersebut diperoleh melalui observasi tidak langsung dan penelusuran pustaka sebagaimana dijelaskan sebelumnya, dan digunakan untuk memverifikasi bahwa aplikasi dapat menerima, memproses, serta menampilkan keluaran Google Gemini API sesuai skenario pengujian yang telah dirancang. Prosedur validasi dilakukan secara kualitatif dengan membandingkan keluaran aplikasi berupa nama penyakit, gejala, dan penyebab terhadap deskripsi pada literatur rujukan mengenai karakteristik masing-masing penyakit. Evaluasi performa diagnosis secara kuantitatif menggunakan dataset citra berlabel dan metrik seperti accuracy, precision, recall, dan F1-score tidak dilakukan pada tahap ini karena berada di luar ruang lingkup pengujian fungsional sistem,

dan diposisikan sebagai arah pengujian lanjutan pada penelitian berikutnya.

Tahap planning dilakukan untuk mengidentifikasi kebutuhan pengguna dan kebutuhan sistem. Pada tahap ini, kebutuhan fungsional aplikasi dirumuskan dalam bentuk fitur utama, yaitu registrasi pengguna, login, pengambilan atau pengunggahan foto daun padi, proses deteksi penyakit, penampilan hasil deteksi, penyimpanan riwayat deteksi, tampilan informasi penyakit, pengelolaan profil, penggantian kata sandi, dan logout. Selain itu, kebutuhan nonfungsional juga ditentukan, meliputi kemudahan penggunaan, dukungan pada perangkat Android, koneksi internet untuk akses Gemini API dan Firebase, serta kemampuan aplikasi dalam menangani variasi kualitas gambar.

Tahap design dilakukan dengan menyusun rancangan alur kerja sistem dan rancangan antarmuka pengguna. Perancangan sistem menggunakan flowmap dan Unified Modeling Language (UML), yang meliputi use case diagram, activity diagram, sequence diagram, dan class diagram. Use case diagram digunakan untuk menggambarkan interaksi pengguna dengan fitur utama aplikasi. Activity diagram digunakan untuk memodelkan alur aktivitas setiap proses, seperti registrasi, login, deteksi penyakit, riwayat deteksi, dan ganti kata sandi. Sequence diagram digunakan untuk menjelaskan urutan komunikasi antarobjek, sedangkan class diagram digunakan untuk memodelkan struktur kelas utama dalam aplikasi, seperti UserModel, DeteksiModel, AuthService, DeteksiService, dan GeminiService. Rancangan antarmuka meliputi halaman splash screen, login, registrasi, beranda, deteksi penyakit, hasil deteksi, riwayat, detail riwayat, informasi penyakit, profil, dan ganti kata sandi.

Tahap coding merupakan proses implementasi rancangan sistem ke dalam aplikasi mobile. Aplikasi dikembangkan menggunakan framework Flutter dengan bahasa pemrograman Dart. Flutter dipilih karena mendukung pengembangan aplikasi lintas platform dengan satu basis kode, sedangkan Dart digunakan sebagai bahasa utama untuk membangun logika aplikasi dan antarmuka pengguna. Platform target dalam penelitian ini adalah Android. Aplikasi juga menggunakan Firebase Authentication untuk proses autentikasi pengguna dan Cloud Firestore untuk penyimpanan data akun serta riwayat deteksi. Integrasi kecerdasan buatan dilakukan

menggunakan Google Gemini API sebagai layanan analisis citra daun padi.

Alur kerja deteksi penyakit dimulai ketika pengguna memilih gambar daun padi melalui kamera atau galeri. Gambar tersebut kemudian dikirimkan ke GeminiService bersama instruksi analisis. Google Gemini API memproses citra dan mengembalikan respons yang berisi nama penyakit, deskripsi gejala, penyebab, rekomendasi penanganan, pencegahan, dan tingkat keyakinan. Respons tersebut diproses oleh aplikasi dalam format terstruktur, kemudian ditampilkan pada halaman hasil deteksi dan disimpan ke Cloud Firestore sebagai riwayat pengguna. Model utama yang digunakan adalah Gemini 2.5 Flash Lite, sedangkan Gemini 2.5 Flash dan Gemini 1.5 Flash digunakan sebagai model alternatif apabila model utama tidak dapat diakses.

Secara konseptual, alur pemrosesan sistem dapat dinyatakan sebagai berikut:

Input Citra Daun → Mobile App → Gemini API
→ Respons Diagnosis → Penyimpanan Firestore
→ Tampilan Hasil

Dalam penelitian ini tidak digunakan persamaan matematis untuk pelatihan model, karena aplikasi tidak membangun model klasifikasi secara mandiri.

Proses analisis citra dilakukan dengan memanfaatkan model multimodal yang tersedia melalui Google Gemini API.

Oleh sebab itu, fokus metode tidak diarahkan pada proses pelatihan, validasi, dan optimasi model, melainkan pada integrasi layanan kecerdasan buatan ke dalam aplikasi mobile serta pengujian fungsi aplikasi. Pendekatan ini berbeda dari penelitian berbasis CNN yang membutuhkan dataset berlabel dan proses pelatihan model. Justifikasi penggunaan Gemini API adalah kemampuannya dalam menganalisis gambar secara langsung melalui layanan API sehingga pengembangan aplikasi menjadi lebih ringkas dan dapat difokuskan pada kebutuhan pengguna akhir.

Lingkungan implementasi terdiri atas perangkat keras dan perangkat lunak. Perangkat keras minimum yang digunakan meliputi prosesor Intel Core i5 atau setara, RAM 8 GB, media penyimpanan SSD 256 GB, serta koneksi internet stabil. Perangkat lunak yang digunakan

meliputi Windows 10 atau Windows 11, Visual Studio Code, Flutter SDK 3.29.0, Dart 3.7.0, Firebase Authentication, Cloud Firestore, Google Gemini API, dan manajer paket Flutter. Seluruh komponen tersebut digunakan untuk mendukung pengembangan, integrasi, penyimpanan data, autentikasi, dan pengujian aplikasi.

Tahap testing dilakukan menggunakan pendekatan CI/CD testing. Pengujian mencakup unit test, widget test, integration test, pipeline test, dan edge case test. Unit test digunakan untuk menguji fungsi pada AuthService, GeminiService, dan Deteksi Model. Widget test digunakan untuk menguji komponen antarmuka seperti halaman login, registrasi, deteksi, hasil deteksi, riwayat, informasi penyakit, profil, dan ganti kata sandi

Integration test digunakan untuk menguji alur penggunaan aplikasi secara menyeluruh, seperti registrasi hingga login, pemilihan foto hingga hasil deteksi, serta deteksi hingga penyimpanan riwayat. Pipeline test dilakukan melalui GitHub Actions untuk memastikan pemeriksaan format kode, analisis statis, pengujian otomatis, dan pembuatan laporan cakupan pengujian berjalan secara berkelanjutan. Edge case test digunakan untuk menguji kondisi khusus, seperti gambar tidak jelas, gambar bukan daun padi, kegagalan koneksi, dan ukuran gambar yang besar.

3. Hasil dan Pembahasan

3.1 Hasil Implementasi Sistem

Penelitian ini berhasil menghasilkan sebuah aplikasi mobile berbasis Android yang mampu mendeteksi penyakit tanaman padi melalui analisis citra daun menggunakan Google Gemini API. Aplikasi dikembangkan menggunakan framework Flutter dengan bahasa pemrograman Dart sehingga memiliki antarmuka yang responsif dan dapat dijalankan pada berbagai perangkat Android. Seluruh data pengguna dan riwayat deteksi disimpan menggunakan Cloud Firestore, sedangkan proses autentikasi dilakukan melalui Firebase Authentication. Integrasi layanan kecerdasan buatan dilakukan menggunakan Google Gemini API yang berfungsi sebagai mesin analisis citra daun padi.

Alur penggunaan aplikasi dimulai ketika pengguna melakukan autentikasi menggunakan akun yang telah terdaftar. Selanjutnya pengguna dapat mengambil gambar daun padi secara langsung menggunakan kamera atau memilih gambar dari galeri perangkat. Citra tersebut kemudian dikirimkan ke Google Gemini API melalui modul GeminiService. Sistem

memberikan prompt yang berisi instruksi kepada model untuk melakukan identifikasi penyakit berdasarkan karakteristik visual daun. Hasil analisis dikembalikan dalam bentuk informasi terstruktur yang meliputi nama penyakit, tingkat keyakinan, penyebab, gejala, rekomendasi penanganan, serta langkah pencegahan. Seluruh hasil deteksi selanjutnya disimpan ke Cloud Firestore sehingga pengguna dapat melihat kembali riwayat diagnosis yang pernah dilakukan.

Implementasi aplikasi terdiri atas beberapa modul utama, yaitu modul autentikasi pengguna, modul deteksi penyakit, modul riwayat deteksi,

modul informasi penyakit, serta modul pengelolaan profil pengguna. Pembagian modul tersebut bertujuan untuk meningkatkan keterpisahan fungsi (separation of concerns) sehingga setiap komponen dapat dikembangkan maupun dipelihara secara independen. Selain itu, penelitian ini juga menerapkan pola layanan (servicelayer)

melalui AuthService, DeteksiService, dan Gemini Service, sehingga komunikasi antara antarmuka pengguna, basis data, dan layanan Gemini API berlangsung secara modular.

Arsitektur aplikasi yang dikembangkan menunjukkan bahwa Google Gemini API berperan sebagai komponen inti dalam proses analisis citra. Berbeda dengan pendekatan konvensional yang melakukan pelatihan model CNN secara lokal, aplikasi ini memanfaatkan model multimodal yang telah disediakan oleh Google sehingga proses inferensi dilakukan secara langsung melalui layanan API. Pendekatan tersebut menyederhanakan proses pengembangan aplikasi karena peneliti tidak perlu melakukan proses pengumpulan dataset dalam jumlah besar, pelabelan data, pelatihan model, maupun optimasi parameter sebagaimana umumnya dilakukan pada penelitian berbasis deep learning.

3.2 Pembahasan

Salah satu temuan penting dalam penelitian ini adalah bahwa pemanfaatan Google Gemini API memungkinkan pembangunan sistem deteksi penyakit tanaman padi tanpa melalui proses pelatihan model klasifikasi secara mandiri. Selama ini sebagian besar penelitian deteksi penyakit tanaman memanfaatkan pendekatan Convolutional Neural Network (CNN) yang mengharuskan peneliti melakukan penyusunan dataset, pelabelan data, pelatihan model, validasi, dan optimasi parameter (Rijal et al., 2024; Rizal et al., 2025). Tahapan tersebut

memerlukan waktu yang relatif panjang serta sumber daya komputasi yang cukup besar. Pada penelitian ini, seluruh proses analisis citra dilakukan oleh model multimodal Google Gemini sehingga pengembangan sistem lebih difokuskan pada integrasi layanan kecerdasan buatan ke dalam aplikasi mobile.

Temuan kedua menunjukkan bahwa model multimodal tidak hanya menghasilkan klasifikasi penyakit, tetapi juga mampu memberikan penjelasan kontekstual mengenai kondisi tanaman. Berdasarkan hasil implementasi, respons yang dihasilkan Gemini API tidak terbatas pada identifikasi jenis penyakit, melainkan juga memuat informasi penyebab penyakit, karakteristik gejala, rekomendasi penanganan, serta langkah pencegahan. Karakteristik tersebut merupakan keunggulan model large language model (LLM) berbasis multimodal yang mampu mengintegrasikan kemampuan computer vision dan natural language generation dalam satu proses inferensi (Google AI for Developers, 2026).

Dengan demikian, aplikasi tidak hanya berfungsi sebagai alat klasifikasi, tetapi juga sebagai media penyedia informasi yang dapat membantu pengguna memahami kondisi tanaman secara lebih komprehensif. Temuan berikutnya berkaitan dengan strategi implementasi layanan AI melalui mekanisme fallback model. Sistem dikembangkan menggunakan tiga model Gemini secara berurutan, yaitu Gemini 2.5 Flash Lite sebagai model utama, Gemini 2.5 Flash sebagai model alternatif pertama, dan Gemini 1.5 Flash sebagai model alternatif kedua. Strategi tersebut meningkatkan keandalan sistem karena proses analisis citra tetap dapat berlangsung apabila salah satu model mengalami keterbatasan akses atau gangguan layanan. Pendekatan seperti ini masih jarang diterapkan pada penelitian deteksi penyakit tanaman yang umumnya hanya menggunakan satu model klasifikasi sehingga keberlangsungan layanan sangat bergantung pada model tersebut.

Selain itu, penelitian ini menunjukkan bahwa integrasi Flutter, Firebase, dan Google Gemini API menghasilkan arsitektur aplikasi yang bersifat modular dan mudah dikembangkan. Flutter memungkinkan pengembangan antarmuka pengguna secara konsisten pada perangkat Android, sedangkan Firebase menyediakan mekanisme autentikasi dan penyimpanan data berbasis cloud. Kombinasi tersebut mendukung implementasi sistem yang lebih sederhana dibandingkan pengembangan aplikasi berbasis server lokal. Dari perspektif

rekayasa perangkat lunak, pendekatan ini memperlihatkan bahwa layanan cloud computing dan Artificial Intelligence as a Service (AIaaS) dapat diintegrasikan secara efektif untuk membangun aplikasi deteksi penyakit tanaman. Apabila dibandingkan dengan penelitian sebelumnya, terdapat beberapa perbedaan mendasar sebagaimana disajikan pada Tabel 1.

Tabel 1. Perbandingan penelitian dengan penelitian terdahulu

Aspek	Rijal et al. (2024)	Rizal et al. (2025)	Penelitian ini
Pendekatan AI	CNN	CNN + TensorFlow Lite	Google Gemini API
Pelatihan Model	Ya	Ya	Tidak
Dataset Berlabel	Diperlukan	Diperlukan	Tidak diperlukan pada sisi aplikasi
Mobile Application	Ya	Ya	Ya
Analisis Kontekstual	Terbatas	Terbatas	Ya
Rekomendasi Penanganan	Tidak	Terbatas	Ya
Model Multimodal	Tidak	Tidak	Ya

Tabel 1 menunjukkan bahwa penelitian ini memiliki karakteristik yang berbeda dibandingkan penelitian terdahulu. Inovasi utama bukan terletak pada pengembangan algoritma klasifikasi baru, melainkan pada pemanfaatan model multimodal sebagai layanan kecerdasan buatan yang terintegrasi dengan aplikasi mobile. Pendekatan tersebut

memperluas ruang implementasi AI pada bidang pertanian karena pengembang tidak lagi dibatasi oleh kebutuhan pelatihan model secara mandiri. Oleh karena itu, kontribusi penelitian ini berada pada aspek integrasi teknologi AI generatif ke dalam sistem deteksi penyakit tanaman yang praktis dan mudah diimplementasikan.



Gambar 2. Perbandingan Karakteristik Penelitian Terdahulu

Gambar 2, dapat diinterpretasikan bahwa pendekatan berbasis Google Gemini API memiliki cakupan kemampuan yang lebih luas karena selain mendukung identifikasi penyakit, juga mampu menghasilkan penjelasan gejala, rekomendasi penanganan, serta respons berbasis

bahasa alami tanpa memerlukan proses pelatihan model pada sisi pengembang

3.3 Evaluasi Pengujian Sistem

Evaluasi sistem dilakukan untuk memastikan bahwa seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan yang telah ditetapkan pada

tahap analisis sistem. Pengujian menggunakan pendekatan Continuous Integration/Continuous Deployment (CI/CD) yang mengintegrasikan proses pengujian otomatis pada setiap perubahan kode program. Pendekatan ini dipilih karena mampu meningkatkan kualitas perangkat lunak melalui proses validasi yang dilakukan secara berulang selama pengembangan (Fowler, 2018).

Pengujian terdiri atas lima jenis, yaitu unit testing, widget testing, integration testing, pipeline testing, dan edge case testing, sebagaimana disajikan pada Tabel 2. Unit testing digunakan untuk menguji fungsi-fungsi utama pada setiap kelas, seperti proses

otentikasi pengguna, komunikasi dengan Google Gemini API, serta pengelolaan data deteksi. Widget testing bertujuan memastikan bahwa setiap antarmuka pengguna dapat ditampilkan dan merespons interaksi sesuai rancangan. Selanjutnya, integration testing dilakukan untuk menguji integrasi antar modul mulai dari proses login hingga penyimpanan hasil deteksi. Alur testing memverifikasi keberhasilan proses build, analisis kode, dan otomatisasi pengujian melalui GitHub Actions. Adapun edge case testing digunakan untuk menguji ketahanan aplikasi terhadap kondisi khusus, seperti gambar tidak valid, ukuran file besar, dan gangguan koneksi internet.

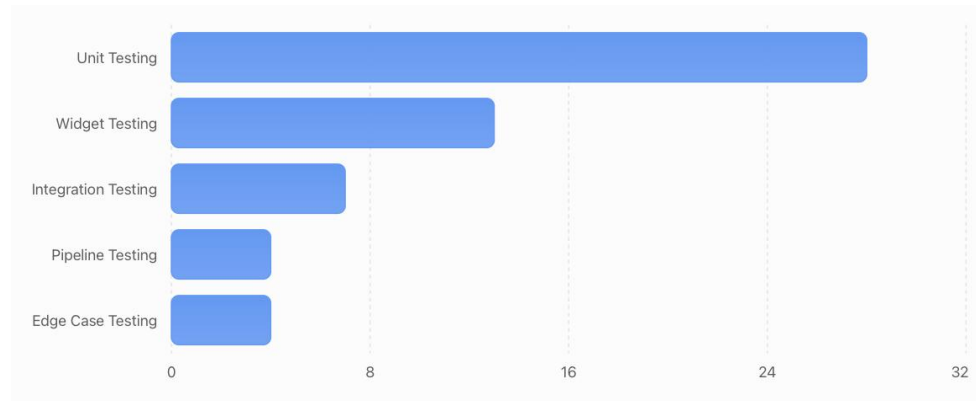
Tabel 2. Ringkasan hasil pengujian sistem

Jenis Pengujian	Jumlah Test Case	Berhasil	Tingkat Keberhasilan
Unit Testing	28	28	100%
Widget Testing	13	13	100%
Integration Testing	7	7	100%
Pipeline Testing	4	4	100%
Edge Case Testing	4	4	100%
Total	56	56	100%

Berdasarkan Tabel 2, seluruh skenario pengujian berhasil dijalankan tanpa ditemukan kegagalan fungsi. Hasil tersebut menunjukkan bahwa aplikasi telah memenuhi seluruh kebutuhan fungsional yang dirumuskan pada tahap planning. Keberhasilan seluruh pengujian mengindikasikan bahwa integrasi antara Flutter, Firebase, dan Google Gemini API berjalan secara konsisten sehingga proses autentikasi, analisis citra, penyimpanan data, serta penyajian informasi kepada pengguna dapat dilakukan secara terpadu. Meskipun seluruh pengujian memperoleh tingkat keberhasilan sebesar 100%, hasil tersebut tidak dapat diinterpretasikan sebagai ukuran akurasi model kecerdasan buatan dalam mengenali penyakit tanaman. Pengujian pada penelitian ini berfokus pada aspek rekayasa perangkat lunak (software functional testing), yaitu menguji apakah setiap fitur aplikasi berjalan sesuai spesifikasi. Oleh karena itu, hasil pengujian lebih merepresentasikan kualitas implementasi sistem dibandingkan performa model Gemini dalam melakukan klasifikasi

penyakit. Evaluasi terhadap akurasi diagnosis memerlukan dataset referensi berlabel dan pengukuran menggunakan metrik seperti accuracy, precision, recall, dan F1-score, yang berada di luar ruang lingkup penelitian ini.

Sebagai gambaran awal mengenai keandalan diagnosis, keluaran aplikasi terhadap citra uji pada masing-masing kategori penyakit (blas, hawar daun bakteri, bercak cokelat, dan tungro) dibandingkan secara kualitatif dengan deskripsi gejala dan penyebab penyakit yang terdapat pada literatur rujukan (Nurkayah et al., 2024; Dagsy & As, 2025), dan hasilnya menunjukkan kesesuaian antara nama penyakit yang diidentifikasi Gemini API dengan karakteristik gejala pada rujukan tersebut. Perbandingan ini bersifat kualitatif dan berskala kecil sehingga belum dapat menggantikan evaluasi kuantitatif berbasis dataset citra berlabel serta metrik accuracy, precision, recall, F1-score, dan confusion matrix, yang tetap direkomendasikan sebagai evaluasi lanjutan untuk memperkuat validitas kemampuan diagnosis sistem.



Gambar 3. Keberhasilan Pengujian Sistem

Berdasarkan Gambar 3, menunjukkan bahwa seluruh kategori pengujian berhasil dijalankan tanpa kegagalan. Unit testing memiliki jumlah skenario terbesar karena menguji fungsi-fungsi inti aplikasi, sedangkan pipeline testing dan edge case testing digunakan untuk memastikan stabilitas sistem pada kondisi operasional yang berbeda.

3.4 Pembahasan dan Perbandingan dengan Penelitian Sebelumnya

Hasil penelitian menunjukkan bahwa Google Gemini API dapat diintegrasikan secara efektif ke dalam aplikasi mobile sebagai layanan analisis citra daun padi. Temuan ini memperlihatkan adanya perubahan paradigma dalam pengembangan sistem deteksi penyakit tanaman. Penelitian sebelumnya umumnya menerapkan pendekatan deep learning berbasis CNN, yang mengharuskan peneliti melakukan pengumpulan dataset, pelabelan data, pelatihan model, validasi, serta optimasi parameter sebelum model dapat digunakan

(Rijal et al., 2024; Rizal et al., 2025). Pada penelitian ini, seluruh proses tersebut digantikan oleh pemanfaatan model multimodal yang telah tersedia melalui Google Gemini API.

Pendekatan tersebut memberikan beberapa keuntungan. Pertama, waktu pengembangan aplikasi menjadi lebih singkat karena pengembang tidak perlu membangun model klasifikasi dari awal. Kedua, kebutuhan sumber daya komputasi selama proses pengembangan menjadi lebih rendah karena tidak diperlukan

proses pelatihan model yang umumnya membutuhkan GPU. Ketiga, sistem mampu menghasilkan keluaran yang lebih informatif karena model Gemini tidak hanya mengidentifikasi kemungkinan penyakit, tetapi juga menyajikan penjelasan mengenai gejala, penyebab, rekomendasi penanganan, dan

langkah pencegahan. Kemampuan menghasilkan respons kontekstual tersebut merupakan karakteristik model multimodal berbasis large language model yang tidak dimiliki oleh sebagian besar model klasifikasi konvensional (Google AI for Developers, 2026). Namun demikian, penelitian ini juga menunjukkan adanya beberapa keterbatasan. Berbeda dengan model CNN yang dapat dijalankan secara lokal setelah proses pelatihan selesai, implementasi Gemini API sepenuhnya bergantung pada koneksi internet dan ketersediaan layanan cloud. Kondisi tersebut menyebabkan proses deteksi tidak dapat dilakukan apabila perangkat tidak memiliki akses internet atau ketika layanan API mengalami gangguan. Untuk mengurangi risiko tersebut, penelitian ini menerapkan mekanisme fallback model, yaitu menggunakan Gemini 2.5 Flash Lite sebagai model utama, kemudian secara otomatis beralih ke Gemini 2.5 Flash dan Gemini 1.5 Flash apabila model sebelumnya tidak tersedia. Strategi ini meningkatkan keandalan sistem dan menjadi salah satu kontribusi implementatif yang belum banyak dibahas pada penelitian terdahulu. Jika dibandingkan dengan penelitian Rijal et al. (2024), penelitian ini memiliki pendekatan yang berbeda dalam proses pengembangan sistem.

Penelitian Rijal et al. berfokus pada peningkatan kemampuan klasifikasi citra menggunakan CNN sehingga keberhasilan penelitian diukur berdasarkan performa model terhadap dataset pengujian. Sebaliknya, penelitian ini menitikberatkan pada integrasi layanan kecerdasan buatan multimodal ke dalam aplikasi mobile sehingga keberhasilan penelitian dievaluasi melalui kualitas implementasi sistem dan keterpenuhan kebutuhan fungsional pengguna. Dengan demikian, kedua penelitian memiliki tujuan yang berbeda dan saling melengkapi.

Penelitian Rizal et al. (2025) juga mengembangkan aplikasi Android menggunakan TensorFlow Lite sehingga model dapat dijalankan secara lokal pada perangkat pengguna. Pendekatan tersebut memiliki keunggulan pada aspek kemandirian sistem karena tidak memerlukan koneksi internet setelah model ditanamkan ke dalam aplikasi. Akan tetapi, setiap peningkatan performa model mengharuskan pengembang melakukan proses pelatihan ulang dan mendistribusikan model terbaru kepada pengguna. Sebaliknya, pada penelitian ini, pembaruan kemampuan model dapat diperoleh secara langsung melalui layanan Google Gemini tanpa mengubah aplikasi yang telah dipasang pada perangkat pengguna.

Karakteristik ini memberikan fleksibilitas yang lebih tinggi terhadap perkembangan teknologi kecerdasan buatan.

3.5 Implikasi Teoritis dan Praktis

Penelitian ini memberikan beberapa implikasi baik dari sisi teoritis maupun praktis. Dari perspektif teoritis, penelitian memperluas pemanfaatan model large language model multimodal pada bidang pertanian.

Selama ini implementasi AI untuk deteksi penyakit tanaman didominasi oleh pendekatan computer vision berbasis CNN. Hasil penelitian menunjukkan bahwa model multimodal juga dapat dimanfaatkan sebagai mesin analisis citra melalui layanan API sehingga membuka peluang penelitian lanjutan mengenai integrasi foundation model dalam sistem informasi pertanian. Dengan demikian, penelitian ini memperkaya kajian mengenai penerapan Artificial Intelligence as a Service (AIaaS) sebagai alternatif pengembangan aplikasi berbasis kecerdasan buatan.

Dari perspektif praktis, aplikasi yang dikembangkan dapat dimanfaatkan sebagai alat bantu identifikasi awal penyakit tanaman padi oleh petani maupun penyuluh pertanian. Kemampuan aplikasi dalam menghasilkan informasi mengenai jenis penyakit, penyebab, gejala, rekomendasi penanganan, dan langkah pencegahan memungkinkan pengguna memperoleh informasi secara cepat tanpa harus menunggu proses konsultasi secara langsung. Selain itu, penyimpanan riwayat deteksi pada Cloud Firestore memungkinkan pengguna mendokumentasikan perkembangan kondisi tanaman sebagai bahan evaluasi pada musim tanam berikutnya.

Walaupun demikian, penelitian ini masih memiliki beberapa keterbatasan. Sistem belum melakukan evaluasi kuantitatif terhadap tingkat akurasi diagnosis menggunakan dataset berlabel sehingga performa model Gemini belum dapat dibandingkan secara statistik dengan model CNN maupun metode deep learning lainnya. Oleh karena itu, penelitian selanjutnya disarankan mengombinasikan evaluasi fungsional dengan evaluasi performa model menggunakan

metrik accuracy, precision, recall, F1-score, dan confusion matrix. Selain itu, pengujian lapangan yang melibatkan petani dan pakar tanaman perlu dilakukan untuk mengukur tingkat penerimaan pengguna (user acceptance) serta kesesuaian hasil diagnosis dengan kondisi nyata di lapangan.

4. Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan aplikasi mobile untuk deteksi penyakit tanaman padi berbasis Google Gemini API menggunakan framework Flutter.

Aplikasi berhasil diimplementasikan untuk mendukung identifikasi empat jenis penyakit utama pada tanaman padi melalui analisis citra daun yang diproses oleh Google Gemini API, serta memberikan informasi mengenai gejala, penyebab, rekomendasi penanganan, dan langkah pencegahan. Hasil pengujian fungsional menunjukkan bahwa seluruh fitur aplikasi berjalan sesuai dengan kebutuhan sistem, dengan tingkat keberhasilan 100% pada 56 skenario pengujian yang meliputi unit testing, widget testing, integration testing, pipeline testing, dan edge case testing. Temuan penelitian menunjukkan bahwa pemanfaatan model multimodal berbasis Artificial Intelligence as a Service (AIaaS) dapat menyederhanakan proses pengembangan aplikasi karena tidak memerlukan pelatihan model klasifikasi secara mandiri, sekaligus mampu menghasilkan keluaran yang lebih informatif dibandingkan pendekatan klasifikasi konvensional. Kelebihan penelitian ini terletak pada integrasi Google Gemini API dengan platform mobile dan layanan cloud sehingga sistem mudah dikembangkan dan diperbarui. Namun demikian, penelitian ini masih terbatas pada evaluasi aspek fungsional aplikasi dan belum mengukur performa diagnosis menggunakan metrik klasifikasi maupun validasi lapangan. Oleh karena itu, penelitian selanjutnya disarankan melakukan evaluasi kuantitatif terhadap akurasi diagnosis menggunakan dataset berlabel,

membandingkan kinerja Gemini API dengan model deep learning lainnya, serta melibatkan pakar pertanian dan petani dalam pengujian lapangan untuk meningkatkan validitas dan keberterimaan sistem.

Daftar Pustaka

Badan Pusat Statistik. (2024). *In 2023, paddy harvested area was approximately 10.21 million hectares with 53.98 million tons of dry unhusked paddy production*. <https://www.bps.go.id>

Dagsy, I. R. D., & As'ari, H. (2025). Isolasi dan uji antagonistik bakteri endofit akar tanaman padi terhadap pertumbuhan fungi *Pyricularia oryzae* penyebab penyakit blas. *Biosfer: Jurnal Biologi dan Pendidikan Biologi*, 10(2), 147–155. <https://doi.org/10.23969/biosfer.v10i2.32984>

Google AI for Developers. (2026). *Image understanding*. <https://ai.google.dev/gemini-api/docs/image-understanding>

Heru, G., Luthfi, A. A., Aden, A. S., Galang, A. F. P. G., Fachri, A., & Riza, F. (2025). Pemanfaatan kecerdasan buatan untuk identifikasi jenis tanaman menggunakan citra daun. *Jurnal Pendidikan Teknologi Informasi dan Vokasional*, 7.

Kuswanto, J. (2020). Sistem pakar untuk perlindungan tanaman padi menggunakan metode *forward chaining*. *Jurnal Ilmiah Edutic: Pendidikan dan Informatika*, 7(1), 31–39. <https://doi.org/10.21107/edutic.v7i1.8805>

Manoj, M., & Ihsan, F. (2024). Integrating deep learning with the Gemini API for improved pest

management. In *Proceedings of the 2024 3rd International Conference on Automation, Computing and Renewable Systems* (pp. 846–851). IEEE.

Novantara, P., Firmansyah, R. L., & Arismawati, M. (2025). Deteksi hama penyakit daun padi dengan menggunakan teknik optimasi deep learning convolutional neural network. *bit-Tech*, 7(3), 975–983.

Nurkayah, N., Sidiq, A. N., & Murdini, L. A. (2024). Identifikasi prevalensi dan karakterisasi penyakit tanaman padi (*Oryza sativa* L) di Kecamatan Sumber Harta, Kabupaten Musi Rawas. *Indonesian Research Journal on Education*, 4(3), 660–664. <https://doi.org/10.31004/irje.v4i3.847>

Rijal, M., Yani, A. M., & Rahman, A. (2024). Deteksi citra daun untuk klasifikasi penyakit padi menggunakan pendekatan deep learning dengan model CNN. *Jurnal Teknologi Terpadu*.

Rizal, F., Sa'idi, M. L., & Hidayat, M. U. R. (2025). Implementasi CNN dan TensorFlow Lite untuk deteksi penyakit daun padi berbasis Android. *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, 6(2), 194–203. <https://doi.org/10.33650/coreai.v6i2.13244>

Shelke, S., Vanjari, N. N., Ghadage, M. R., Wadekar, S. Y., & Tagare, D. S. (2025). *AgroCare: AI-powered crop management system*. *International Journal of Scientific Research in Engineering and Management*, 9(5). <https://doi.org/10.55041/IJSREM48399>