

FARMSHIELD: RANCANG BANGUN APLIKASI SISTEM PENDETEKSI HAMA KUTU PUTIH UNTUK MENDUKUNG PETANI TOMAT BERBASIS WEBSITE

Luthfi Alviani¹, Fikri Binaul Umah², Muhammad Rifki Munawar³, Alicia Maharani⁴

^{1,2,3,4}IPB University/Teknologi Rekayasa Komputer; Fakultas Sekolah Vokasi; Jl. Kumbang No.14, Kelurahan Babakan, Kecamatan Bogor Tengah, Kota Bogor, Jawa Barat/Fax (0251) 3829101

Keywords:

Pendetksi, Kutu Putih,
Tomat, Website, Hama

Corespondent Email:

binaulumah@apps.ipb.ac.id



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstrak — Penurunan hasil panen tomat sering terjadi akibat keterlambatan identifikasi hama, khususnya kutu putih yang sulit dikenali secara manual oleh petani. Kondisi ini mendorong perlunya sistem deteksi yang cepat, akurat, dan mudah diakses. Penelitian ini mengembangkan FarmShield, sebuah platform berbasis web yang memanfaatkan model *machine learning* untuk mendeteksi hama kutu putih melalui kamera secara *real-time*. Metode yang digunakan meliputi observasi lapangan, studi literatur, perancangan sistem dengan Laravel dan Firebase, serta pengujian fitur melalui uji fungsional, *usability*, dan kompatibilitas. Hasil penelitian menunjukkan bahwa sistem mampu menampilkan deteksi kutu putih dengan *bounding box* dan akurasi yang sesuai, serta menyediakan fitur pendukung seperti *dashboard* analisis, forum petani, radio edukasi, dan ensiklopedia pertanian yang semuanya berjalan stabil. Pengujian membuktikan bahwa antarmuka mudah dipahami dan seluruh fitur utama dapat digunakan tanpa *error*. FarmShield memberikan kontribusi penting dalam membantu petani melakukan identifikasi hama lebih awal dan meningkatkan efisiensi pengambilan keputusan. Secara keseluruhan, platform ini berpotensi dikembangkan lebih lanjut untuk meningkatkan akurasi deteksi dan memperluas cakupan hama.

Abstract. *The decline in tomato crop yields is often caused by delayed identification of pests, particularly mealybugs, which are difficult for farmers to recognize manually. This issue highlights the need for a detection system that is fast, accurate, and easily accessible. This study develops FarmShield, a web-based platform that utilizes a machine learning model to detect mealybugs in real-time through an integrated camera. The research methods include field observation, literature study, system design using Laravel and Firebase, and feature testing through functional, usability, and compatibility evaluations. The results show that the system is capable of displaying mealybug detection with bounding boxes and reliable accuracy, as well as providing supporting features such as an analytical dashboard, farmer forum, educational radio, and agricultural encyclopedia, all of which operate smoothly. Testing confirms that the interface is easy to use and all core features function without errors. FarmShield offers significant benefits by helping farmers perform early pest identification and make more informed decisions. Overall, the platform has strong potential for further development, including improving detection accuracy and expanding coverage to other pests.*

1. PENDAHULUAN

Tomat merupakan salah satu komoditas hortikultura yang sangat penting di Indonesia karena banyak ditanami dan sering dikonsumsi oleh masyarakat. Namun, hasil panen tomat sering menurun karena serangan hama dan penyakit tanaman yang sulit dikenali dengan cepat oleh para petani. Penelitian mengindikasikan bahwa keterlambatan dalam identifikasi menjadi salah satu alasan kegagalan panen, karena petani sering kali hanya bergantung pada pengamatan manual yang tidak selalu tepat [1].

Selain itu, kurangnya pemahaman tentang gejala penyakit atau hama sering kali membuat banyak petani mengambil keputusan yang salah, seperti memanen lebih awal atau menghapus tanaman yang baru menunjukkan tanda-tanda awal serangan hama [2].

Salah satu hama yang dapat menyebabkan kerusakan serius pada berbagai komoditas adalah kutu putih (*mealybug*). Hama ini menyerang dengan cara mengisap cairan dari daun, pucuk, atau batang tanaman, sehingga dapat menyebabkan daun melipat, tanaman kerdil, serta menurunkan hasil panen secara signifikan [3]. Penelitian mengenai kutu putih menunjukkan bahwa hama ini sangat *invasif*, bisa membentuk koloni besar, dan dapat berakibat merusak tanaman lain, termasuk singkong dan berbagai tanaman hortikultura.

Walaupun begitu, informasi mengenai deteksi awal kutu putih pada tanaman tomat masih sangat terbatas, terutama dalam bentuk platform digital yang mudah dijangkau oleh petani. Dengan kemajuan teknologi, berbagai penelitian mulai mengembangkan sistem diagnosis hama dan penyakit tanaman yang berbasis web atau kecerdasan buatan. Sebagai contoh, riset telah membuat aplikasi berbasis web untuk mendiagnosis penyakit tomat menggunakan metode aturan *if-then*, agar petani dapat mengenali penyakit secara mandiri.

Penelitian lain menerapkan metode *Certainty Factor* untuk mendiagnosis hama dan penyakit pada tanaman tomat, dengan memperoleh tingkat akurasi hingga 90%, yang menunjukkan potensi sistem pakar dalam

membantu petani dalam pengambilan keputusan [4]. Teknologi berbasis kecerdasan buatan seperti *Convolutional Neural Network* (CNN) juga banyak dipakai untuk pengelompokan penyakit tanaman tomat, dengan akurasi validasi mencapai 82% pada sepuluh jenis penyakit yang berbeda.

Sementara itu, pada komoditas lain seperti padi, aplikasi berbasis Android untuk mendekripsi hama melalui citra telah mampu membantu petani dalam mengenali hama dengan akurasi 85% dan mempercepat proses identifikasi di lapangan [5].

Walaupun penelitian-penelitian ini menunjukkan kemajuan yang signifikan dalam penggunaan teknologi digital untuk pertanian, terdapat kesenjangan yang bisa diidentifikasi. Kebanyakan penelitian masih berfokus pada diagnosis penyakit tanaman atau deteksi hama pada komoditas selain tomat, sementara sistem berbasis web yang khusus untuk deteksi hama kutu putih pada tanaman tomat sangat jarang ditemukan. Selain itu, belum ada platform yang dirancang khusus untuk memenuhi kebutuhan petani tomat, yang memungkinkan identifikasi hama dengan cepat, memberikan informasi visual tentang gejala, serta memberikan rekomendasi penanganan yang mudah dimengerti.

Berdasarkan masalah di atas, maka dibutuhkan sebuah web *FarmShield*, yaitu sebuah aplikasi sistem pendekripsi hama kutu putih untuk mendukung petani tomat berbasis website dengan cara yang lebih cepat, akurat, dan mudah diakses. Sistem ini dirancang untuk menggantikan metode identifikasi manual yang selama ini masih sering digunakan oleh petani dan terkadang mengakibatkan kesalahan.

Tujuan dari penelitian ini adalah untuk menghadirkan sebuah sistem yang dapat membantu petani dalam melakukan pemantauan awal serangan hama, mengurangi kesalahan dalam identifikasi, serta mendukung pengambilan keputusan agar kerugian dari hasil panen dapat dikurangi.

2. TINJAUAN PUSTAKA

2.1 Hama Kutu Putih

Kutu Putih (*Phenacoccus manihoti*) adalah hama yang menghisap cairan dari tanaman, menyerang daun, batang, dan pucuk. Kehadiran kutu putih membaut daun menjadi keriput, menguning, serta menghambat pertumbuhan. Hal ini juga menyebabkan timbulnya koloni putih di bagian daun dan batang. Karakteristik ini menjadi tanda visual yang penting untuk mengidentifikasi hama dalam tanaman *hortikultura*, termasuk tomat [6]. Penelitian menunjukkan bahwa kutu putih memiliki jenis mulut yang dapat menusuk dan menyedot, dan menyebabkan kerusakan yang khas, seperti penundaan pertumbuhan pucuk dan penurunan kesehatan tanaman [7].

2.2 Website

Website adalah suatu kumpulan informasi yang terdiri dari berbagai halaman yang saling terhubung. Halaman-halaman ini digunakan untuk menyajikan informasi dalam bentuk teks, gambar baik yang tetap maupun yang bergerak, animasi, suara, dan gabungan dari semua elemen tersebut, baik yang berbeda maupun yang tetap. Semua ini membentuk sebuah struktur yang saling terhubung, di mana setiap halaman terhubung melalui jaringan halaman lainnya. Keterkaitan antara satu halaman web dengan halaman web lainnya disebut *hyperlink*, dan teks yang berfungsi sebagai penghubung disebut *hypertext* [8].

2.3 Laravel Framework

Berdasarkan penelitian yang dilakukan oleh Prasena dan Sama, Laravel merupakan suatu *framework* pemrograman *open source* yang digunakan oleh banyak pengembang di seluruh dunia [9]. Laravel juga dikenal sebagai salah satu aplikasi untuk memaksimalkan penggunaan bahasa pemrograman PHP seperti sistem *template*, manajemen jalur yang teratur, serta arsitektur yang mudah untuk dikembangkan. dengan lisensi MIT dan metode kerja MVC (*Model View Controller*), laravel menyediakan sintaks yang mudah dan efektif sehingga dapat meningkatkan mutu aplikasi, menekan biaya pengembangan dan pemeliharaan, serta memberikan pengalaman pengguna yang lebih baik dan optimal [10].

2.4 Figma

Figma merupakan salah satu alat desain yang umumnya digunakan untuk menciptakan tampilan aplikasi *mobile*, *dekstop*, situs web, dan lainnya. Figma dapat diakses di sistem operasi Windows, Linux, atau Mac dengan koneksi internet. Salah satu keunggulan Figma adalah memungkinkan lebih dari satu orang untuk bekerja secara bersamaan pada proyek yang sama meskipun berada di lokasi yang berbeda. Ini bisa dianggap sebagai kerja tim, dan kemampuan aplikasi Figma inilah yang menjadikannya pilihan utama bagi banyak desainer UI/UX untuk membuat prototipe situs web atau aplikasi dengan cepat dan efisien [11].

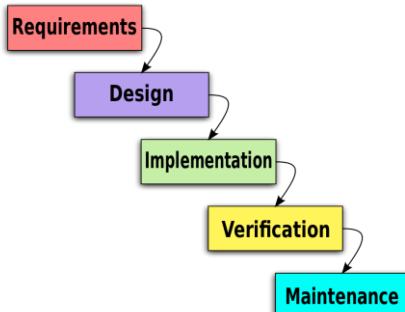
2.5 Firebase

Firebase merupakan layanan yang ditawarkan oleh Google untuk mendukung aplikasi Android, iOS, dan *website* dalam penyimpanan dan menyelaraskan data dengan mudah [12]. Salah satu alat yang sangat populer adalah *Firebase Realtime Database*, yaitu *database* berbasis JSON yang dapat melakukan pembaruan data secara langsung setiap kali ada perubahan. Dengan sistem sinkronasi *Real-time*, aplikasi tidak perlu terus-menerus mengajukan permintaan hanya untuk memperoleh data terbaru. Fitur ini menjadi semakin efektif berkat dukungan dari berbagai *library* Firebase sehingga proses pengolahan data menjadi lebih cepat praktis dan sesuai untuk kebutuhan aplikasi masa kini [13].

2.6 Metode Waterfall

Firebase Metode waterfall, yang juga disebut sebagai metode air terjun atau siklus hidup klasik adalah model pengembangan perangkat lunak yang berlangsung dengan cara yang linier dan berurutan. [14] Model yang dikenal sebagai Linear *Sequential* ini fokus pada proses pengembangan yang dimulai dari identifikasi kebutuhan pengguna, dilanjutkan dengan tahap perencanaan, pembuatan model, konstruksi sistem, hingga proses distribusi kepada pengguna. Pengembang harus memahami dengan lebih mendalam mengenai cara proses pengembangan sistem ketika menggunakan model *waterfall* serta ciri-ciri yang dimiliki oleh model *waterfall* itu sendiri. Semua langkah ini ditutup dengan fase

dukungan dan pemeliharaan untuk perangkat lunak yang telah selesai dikembangkan [15].



Gambar 1. Metode *Waterfall*

3. METODE PENELITIAN

3.1 Pengambilan Data

Dalam tahapan yang bertujuan untuk mengumpulkan informasi secara menyeluruh, perlu dilakukan metode atau langkah pengumpulan data dalam tahap pembuatan sistem web untuk mendukung petani tomat dalam deteksi hama kutu putih. Maka dapat dilakukan seperti berikut:

3.1.1. Observasi Lapangan

Penelitian ini dilakukan observasi atau pengamatan secara langsung dengan para petani *hortikultura*. Dalam penelitian ini, wawancara juga dilakukan untuk menilai cara penyemprotan pestisida yang masih dilakukan secara manual, yang mengakibatkan kelelahan, kemungkinan terpapar pestisida, serta pemberoran bahan.

3.1.2. Studi Literatur

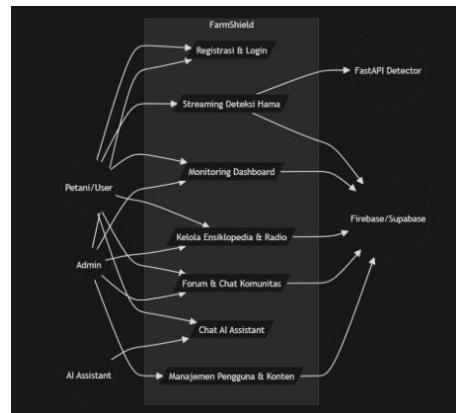
Informasi diambil dari jurnal, buku, artikel akademis, dan laporan teknis. Data ini digunakan sebagai landasan teori mengenai hama kutu putih, sistem berbasis daring, serta ide-ide lain yang mendukung proses pengembangan *FarmShield*.

3.2 Pembuatan Sistem Informasi

3.2.1. Analisis Kebutuhan

a. Use Case Diagram

Diagram *use case* ini memperlihatkan hubungan antara beberapa penggunaanya dengan sistem *FarmShield*. Pengguna yang terlibat terdiri dari Petani/Operator, Data *Scientist*, Pengguna Jarak Jauh, dan Koperasi/Kelompok Tani seperti pada gambar di bawah ini.



Gambar 2 Use Case Diagram

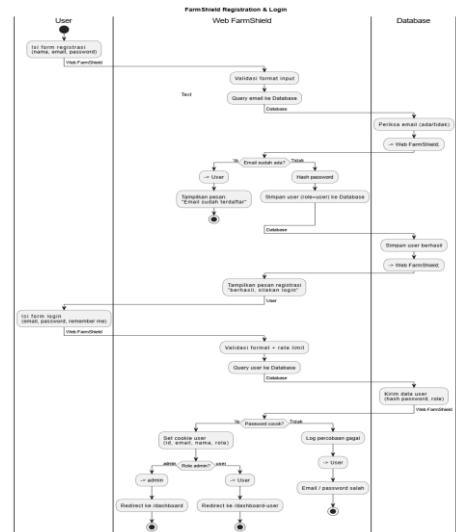
Petani/User dapat melakukan Registrasi & Login, mengakses Streaming Deteksi Hama, Monitoring Dashboard, fitur Forum dan Chat Komunitas, serta memperoleh informasi melalui Ensiklopedia dan Radio Petani.

Admin memiliki akses yang lebih luas, mencakup Registrasi & Login, Monitoring Dashboard, Forum dan Chat, serta pengelolaan konten dan pengguna melalui fitur Manajemen Pengguna & Konten.

AI Assistant berfungsi sebagai sistem pendukung percakapan melalui fitur Chat AI Assistant yang terhubung dengan basis data sistem.

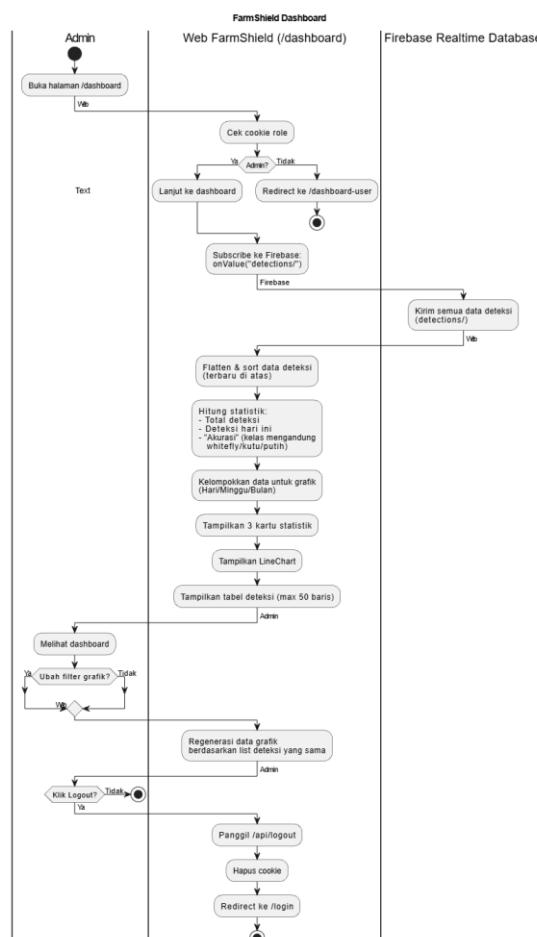
Sistem juga berinteraksi dengan layanan eksternal, yaitu FastAPI Detector untuk proses Streaming Deteksi Hama, serta Firebase/Supabase sebagai layanan penyimpanan data untuk sebagian besar fitur di dalam platform.

b. Activity Diagram



Gambar 3 Activity Diagram Registrasi dan Login

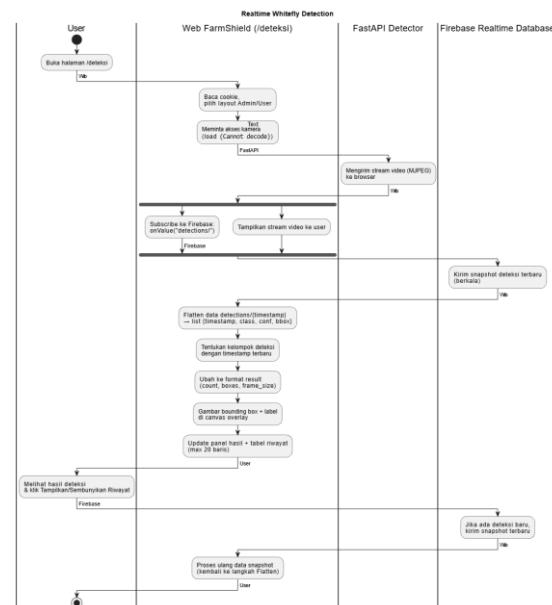
Dari Gambar 3 *activity diagram* di atas, proses dimulai ketika pengguna melakukan registrasi dengan mengisi data diri. Sistem memvalidasi input, mengecek email di database, lalu menyimpan akun baru jika email belum terdaftar. Setelah registrasi berhasil, pengguna dapat melakukan login. Pada proses login, sistem memvalidasi data, mencocokkan password dengan data di database, lalu memberikan akses sesuai role pengguna. Jika sesuai, pengguna diarahkan ke dashboard masing-masing, sedangkan jika salah, sistem menampilkan pesan error.



Gambar 4 Activity Diagram Dashboard Statistik (Admin)

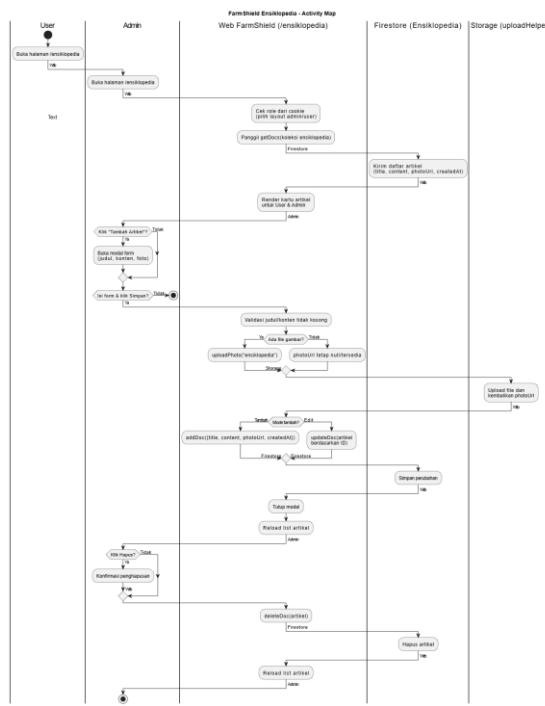
Dari Gambar 4 *activity diagram* di atas, proses dimulai ketika admin membuka halaman dashboard. Sistem memeriksa cookie untuk memastikan role admin, kemudian mengambil seluruh data deteksi dari Firebase. Data tersebut diurutkan, dihitung statistiknya, dan dikelompokkan untuk menghasilkan grafik serta kartu informasi. Dashboard kemudian

menampilkan grafik, total deteksi, akurasi, dan tabel data terbaru. Jika admin mengubah filter grafik, sistem meregenerasi visualisasi berdasarkan data yang sama. Ketika admin memilih logout, sistem menghapus cookie dan mengarahkan kembali ke halaman login.



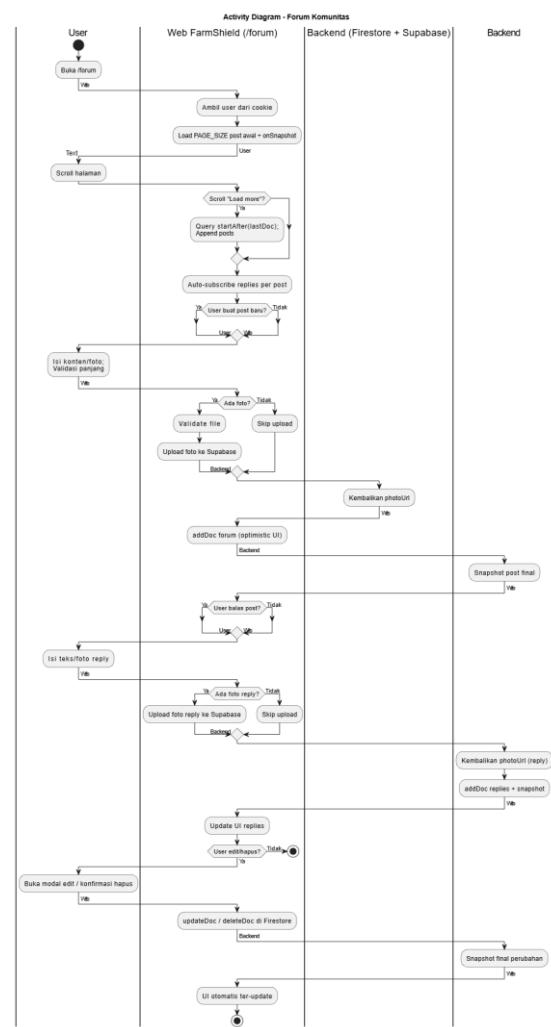
Gambar 5 Activity Diagram Deteksi Kutu Putih Real-Time

Dari Gambar 5 *activity diagram* di atas, proses dimulai ketika pengguna membuka halaman deteksi. Sistem membaca cookie untuk menentukan layout, kemudian meminta akses kamera dan menampilkan stream video dari FastAPI ke browser. Secara bersamaan, sistem berlangganan data deteksi terbaru dari Firebase. Setiap snapshot deteksi yang diterima akan diflatten, dipilih berdasarkan timestamp terbaru, lalu diubah ke format yang siap ditampilkan. Sistem menggambar bounding box dan label pada overlay video, serta memperbarui panel hasil dan riwayat deteksi. Proses ini berulang setiap kali Firebase mengirimkan snapshot baru sehingga pengguna dapat melihat hasil deteksi hama secara real-time.



Gambar 6 Activity Diagram Ensiklopedia

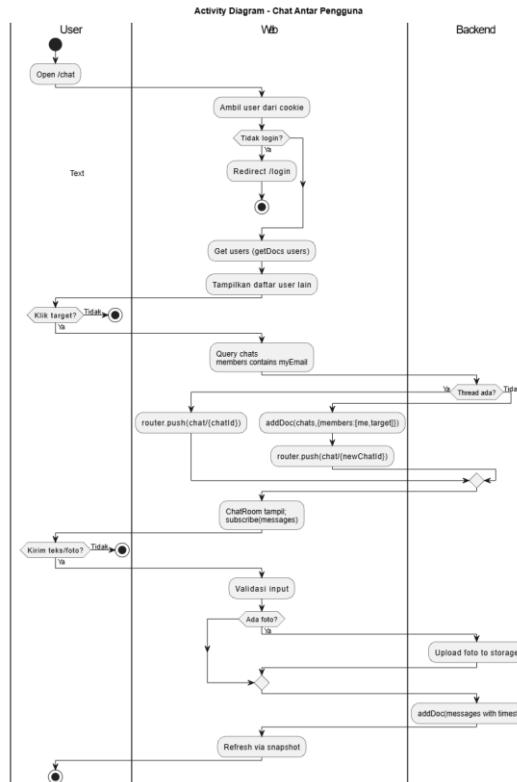
Dari Gambar 6 *activity diagram* di atas, proses dimulai ketika pengguna membuka halaman ensiklopedia. Sistem memeriksa role melalui cookie lalu mengambil daftar artikel dari Firestore dan menampilkannya kepada pengguna maupun admin. Jika admin menambahkan atau mengedit artikel, sistem memvalidasi input, mengunggah gambar bila tersedia, lalu menyimpan data baru atau pembaruan ke Firestore. Setelah itu daftar artikel diperbarui. Ketika admin menghapus artikel, sistem mengonfirmasi aksi tersebut, menghapus data dari Firestore, dan memuat ulang daftar artikel.



Gambar 7 Activity Diagram Forum Komunitas

Dari Gambar 7 *activity diagram* di atas, proses dimulai ketika pengguna membuka halaman forum. Sistem mengambil data pengguna dari cookie lalu memuat daftar posting secara real-time. Ketika pengguna melakukan scroll dan memilih “Load more”, sistem menampilkan posting tambahan dari basis data. Saat pengguna membuat posting baru, sistem memproses isi teks maupun foto, mengunggah foto ke Supabase jika ada, lalu menyimpan posting ke Firestore dan menampilkan hasilnya secara otomatis. Pengguna juga dapat memberikan balasan dengan mekanisme serupa, yaitu validasi dan unggah foto (jika ada) kemudian penyimpanan reply ke basis data dan pembaruan tampilan forum secara real-time. Apabila pengguna mengedit atau menghapus posting/reply, sistem melakukan update atau delete di Firestore, dan perubahan langsung tercermin pada UI. Dengan

demikian, seluruh aktivitas forum berlangsung secara interaktif dan diperbarui otomatis tanpa perlu memuat ulang halaman.



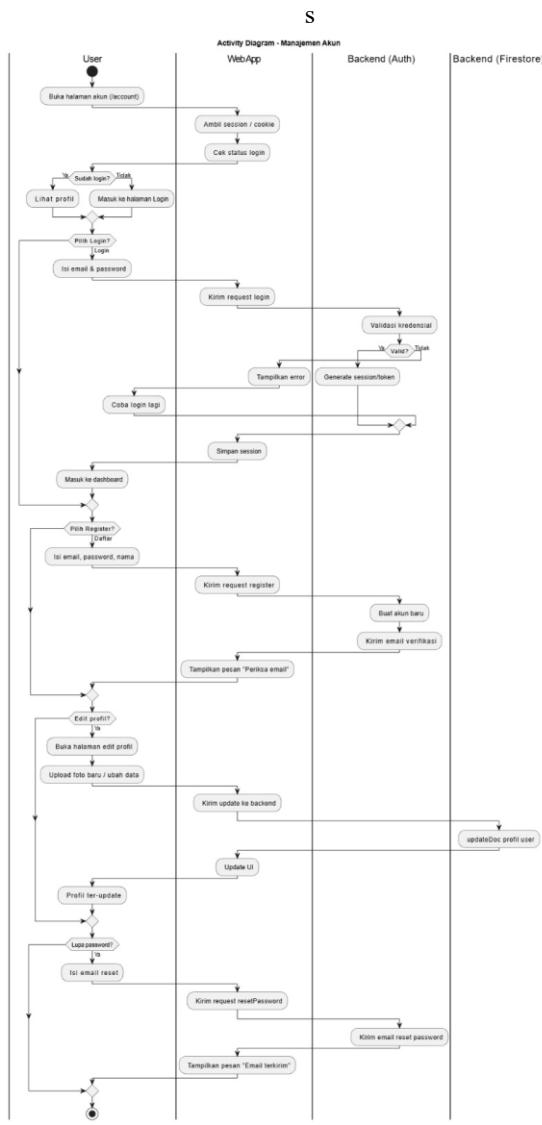
Gambar 8 Activity Diagram Chat Antar Pengguna

Dari Gambar 8 *activity* diagram di atas, proses dimulai ketika pengguna membuka halaman chat. Sistem mengambil informasi pengguna dari cookie, dan apabila pengguna belum login, sistem akan mengarahkan ke halaman login. Setelah login, sistem mengambil daftar pengguna lain dari basis data dan menampilkannya. Ketika pengguna memilih target chat, sistem mengecek apakah thread chat antara kedua pengguna sudah ada. Jika belum, sistem membuat thread chat baru dan membuka ruang percakapan. Pada ruang chat, sistem melakukan *subscribe* agar pesan baru tampil secara real-time. Ketika pengguna mengirim teks atau foto, sistem memvalidasi input; jika ada foto maka foto terlebih dahulu diunggah ke storage, lalu pesan disimpan ke basis data dengan *timestamp*. Snapshot real-time membuat pesan baru langsung tampil tanpa memuat ulang halaman. Dengan demikian komunikasi antar pengguna berlangsung secara interaktif dan diperbarui otomatis oleh sistem.



Gambar 9 Activity Diagram Chat AI

Dari Gambar 9 *activity* diagram di atas, proses dimulai ketika pengguna membuka halaman *chat AI assistant*. Sistem memuat riwayat percakapan dari *localStorage* dan menampilkannya pada area chat. Setelah itu pengguna dapat mengetik pesan untuk dikirim. Ketika pesan dikirim, sistem menyimpan pesan tersebut ke *state* dan *localStorage*, kemudian membentuk riwayat percakapan terakhir untuk dikirim ke AI Backend. Selanjutnya sistem memanggil layanan AI dengan membawa pesan terbaru dan riwayat chat. Jika AI memberikan respons dengan sukses, sistem menambahkan respons tersebut ke *state* dan *localStorage* lalu menampilkannya di layar. Jika terjadi kegagalan, sistem menambahkan pesan error sebagai *fallback* ke *state* dan *localStorage* kemudian menampilkan indikator error. Setelah respons ditampilkan, sistem kembali menunggu interaksi berikutnya dari pengguna.



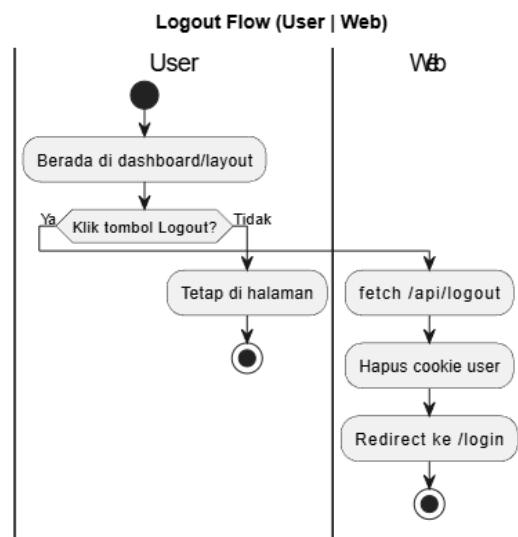
Gambar 10 Activity Diagram Akun

Dari Gambar 10 activity diagram di atas, proses dimulai ketika pengguna membuka halaman manajemen akun. Sistem mengambil status sesi atau cookie untuk mengecek apakah pengguna sudah login. Jika belum, pengguna diarahkan ke halaman login untuk memasukkan email dan password. Sistem memvalidasi kredensial di backend; apabila valid, sesi login disimpan dan pengguna diarahkan ke dashboard, sedangkan jika gagal sistem menampilkan pesan error.

Pengguna juga dapat melakukan registrasi akun baru dengan mengisi email, password, dan nama. Permintaan registrasi dikirim ke backend, kemudian sistem membuat akun baru dan mengirim email verifikasi kepada pengguna.

Setelah *login*, pengguna dapat mengedit profil melalui halaman edit profil dengan mengubah data atau mengunggah foto baru. Update profil dikirim ke backend dan disimpan pada basis data Firestore, kemudian tampilan profil diperbarui secara otomatis.

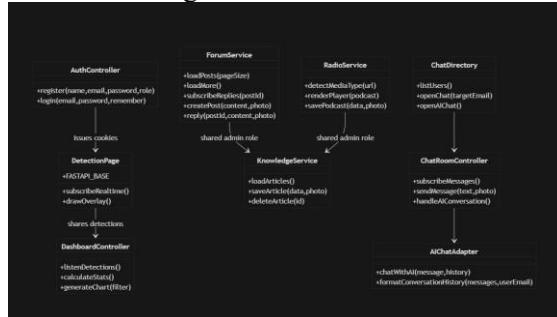
Jika pengguna lupa *password*, mereka dapat mengirim permintaan reset *password* dengan mengisi email. Backend akan mengirim email reset *password* dan sistem menampilkan pemberitahuan bahwa email terkirim. Seluruh proses kembali menunggu interaksi pengguna berikutnya.



Gambar 11 Logout Flow

Dari Gambar 11 activity diagram, proses menunjukkan alur logout pengguna dari sistem berbasis web. Proses dimulai ketika pengguna berada pada halaman dashboard atau layout. Pada tahap ini, pengguna diberikan pilihan untuk menekan tombol Logout. Jika pengguna tidak menekan tombol tersebut, maka pengguna tetap berada di halaman dan proses berakhir tanpa perubahan sesi. Namun, jika pengguna menekan tombol Logout, alur proses berlanjut ke sisi sistem. Sistem kemudian melakukan permintaan *fetch /api/logout* untuk memproses permintaan keluar, dilanjutkan dengan penghapusan cookie pengguna guna mengakhiri sesi login, dan diakhiri dengan pengalihan (redirect) ke halaman login. Dengan demikian, proses logout selesai sepenuhnya dan pengguna kembali ke halaman login.

c. Class Diagram

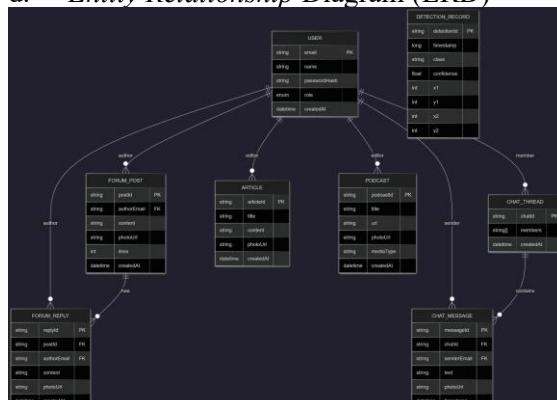


Gambar 12 Class Diagram

Pada Gambar 8 Class diagram FarmShield terdiri dari tiga bagian utama yakni otentikasi, informasi, dan komunikasi. Layanan otentikasi ditangani oleh AuthController untuk registrasi dan login, kemudian DetectionPage dan DashboardController memproses hasil deteksi hama menjadi data statistik dan grafik.

Layanan informasi mencakup ForumService dan RadioService yang terhubung dengan KnowledgeService untuk pengelolaan konten pengetahuan, posting forum, dan media radio. Layanan komunikasi diatur oleh ChatDirectory dan ChatRoomController untuk percakapan antar pengguna, sedangkan komunikasi dengan AI difasilitasi oleh AIChatAdapter yang menangani integrasi ke layanan AI eksternal.

d. Entity Relationship Diagram (ERD)



Gambar 13 ERD

Pada Gambar 9 ERD di atas diperlihatkan bahwa sistem ini terdiri dari delapan tabel utama, yaitu User, Detection_Record, Forum_Post, Article, Podcast, Chat_Thread, Forum_Reply, dan Chat_Message. Tabel User menjadi pusat relasi karena seluruh aktivitas dalam sistem berkaitan langsung dengan pengguna. Pengguna dapat melakukan deteksi hama yang tercatat pada tabel

Detection_Record. Selain itu, pengguna dapat membuat posting pada Forum_Post serta memberikan balasan yang tersimpan pada Forum_Reply, sehingga membentuk hubungan satu-ke-banyak antara User dan aktivitas forum. Untuk penyediaan informasi dan edukasi, pengguna dapat mengakses tabel Article dan Podcast, yang memuat konten ensiklopedia dan siaran radio/podcast pertanian. Pada fitur komunikasi, tabel Chat_Thread menyimpan daftar percakapan antar pengguna maupun percakapan dengan AI, sedangkan tabel Chat_Message menyimpan isi pesan dalam setiap thread.

3.2.3 Implementasi

Dalam pengembangan *website* ini, Laravel dipilih sebagai *framework* utama, HTML digunakan sebagai fondasi halaman, dan CSS untuk meningkatkan estetika antarmuka. Di sisi lain, Firebase berperan sebagai penyimpanan data yang menyimpan dan mengatur informasi secara terstruktur dan langsung.

3.2.4 Pengujian

Pada tahap akhir, dilakukan proses pengujian terhadap sistem yang telah diimplementasikan dengan mengoperasikan website serta fitur utama berupa pendekripsi kutu putih, sehingga dapat dipastikan bahwa seluruh fitur berfungsi sesuai kebutuhan dan mampu mencapai kinerja yang diharapkan.

4. HASIL DAN PEMBAHASAN

4.1 Analisis Sistem

a. Analisis Sistem Yang Digunakan Saat ini

Sistem yang ada saat ini di kalangan petani *hortikultura* masih menggunakan metode manual, dan belum ada alat deteksi hama secara otomatis dan *website* mengenai informasi hama kutu putih juga masih terbatas.

b. Analisis Sistem Yang Akan Diterapkan

Kebutuhan fungsional menjelaskan tentang proses serta peran yang beroperasi dalam sistem, yang dirancang untuk membantu mengatasi masalah yang dihadapi oleh pengguna. Fungsi-fungsi yang ada pada *website* ini adalah sebagai berikut:

1. Menunjukkan hasil deteksi hama secara *real-time* dengan indikator dan tingkat akurasi.

2. Menyajikan data dan tren deteksi melalui *dashboard* analisis.
3. Menyediakan wadah diskusi antara petani lewat Forum Petani.
4. Menampilkan konten audio pendidikan pertanian melalui fitur Radio Petani.
5. Menyediakan informasi tentang hama penyakit tanaman, serta cara penanganannya melalui Ensiklopedia Pertanian.

4.2 Implementasi

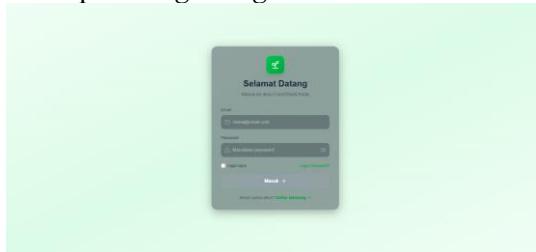
a. Tampilan Page Home



Gambar 14 Page Home

Gambar 10 di atas menunjukkan tampilan halaman awal dari website Farmshield yang di mana terdapat dua pilihan yaitu masuk untuk deteksi dan masuk untuk melihat *dashboard*.

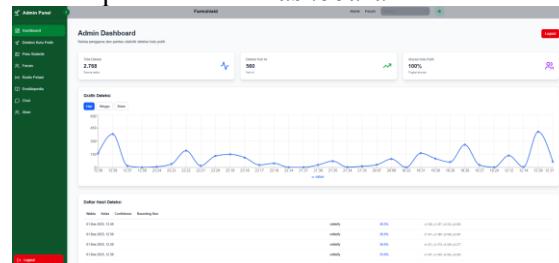
b. Tampilan Login Page



Gambar 15 Login Page

Gambar 11 di atas merupakan tampilan dari halaman *login* di mana terdapat email yang akan digunakan dan *password* untuk *login*.

c. Tampilan Admin Dashboard



Gambar 16 Dashboard

Gambar 12 di atas menunjukkan hasil tampilan *login* yaitu *dashboard*, di sini dapat

melihat akurasi dan grafik deteksi yang telah di deteksi. Terdapat juga beberapa pilihan menu untuk melihat halaman lainnya.

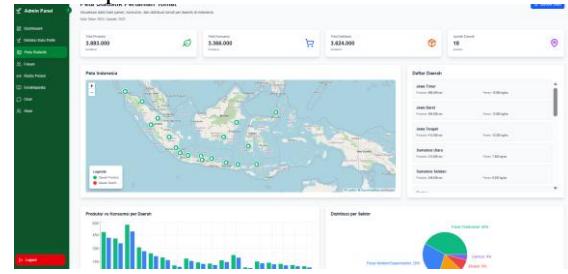
d. Tampilan Halaman Deteksi Kutu Putih



Gambar 17 Halaman Deteksi Kutu Putih

Gambar 13 halaman tampilan deteksi kutu putih yang nanti akan digunakan untuk mendekripsi dengan fitur kamera yang sudah di integrasikan dengan *machine learning* yang telah dibuat. Nanti hasilnya akan terlihat di sini pada bagian hasil deteksi.

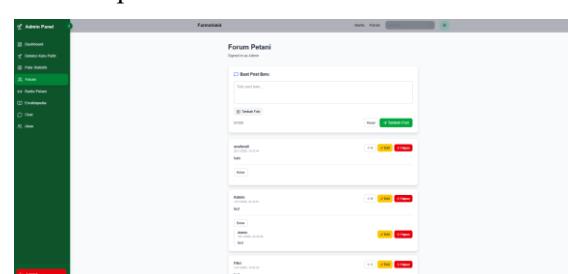
e. Tampilan Halaman Peta Statistik



Gambar 18 Halaman Peta Statistik

Gambar di atas merupakan tampilan halaman dari menu peta statistik pertanian tomat untuk melihat visualisasi data panen, konsumsi, dan distribusi tomat per daerah di Indonesia.

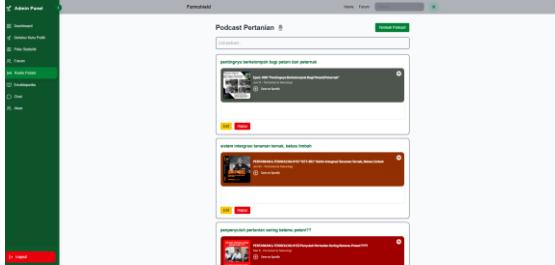
f. Tampilan Halaman Forum Petani



Gambar 19 Halaman Forum Petani

Gambar 15 di atas menampilkan halaman forum petani sebagai tempat diskusi antar pengguna terkait hal yang berkaitan dengan masalah pertanian tomat.

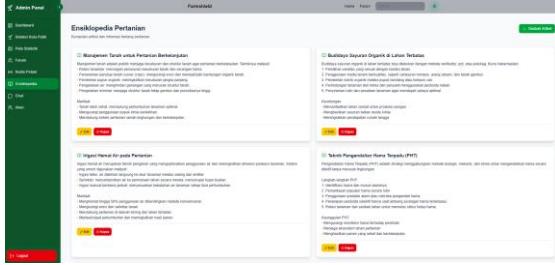
g. Halaman Tampilan Radio Petani



Gambar 20 Halaman Radio Petani

Gambar 16 menunjukkan halaman tampilan dari *podcast* pertanian yang dapat didengarkan untuk menambah wawasan atau informasi tentang pertanian.

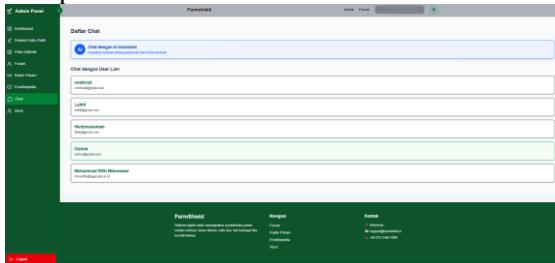
h. Tampilan Halaman Ensiklopedia



Gambar 21 Halaman Ensiklopedia Pertanian

Gambar 17 merupakan halaman tampilan menu ensiklopedia digunakan untuk melihat kumpulan artikel dan informasi tentang pertanian.

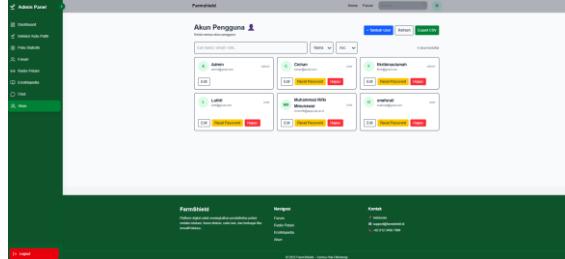
i. Tampilan Halaman Chat



Gambar 22 Halaman Chat

Gambar 18 menunjukkan halaman tampilan daftar *user* untuk *chat* antar pengguna dan chat dengan *AI assistant* (Gemini).

j. Tampilan Halaman Akun



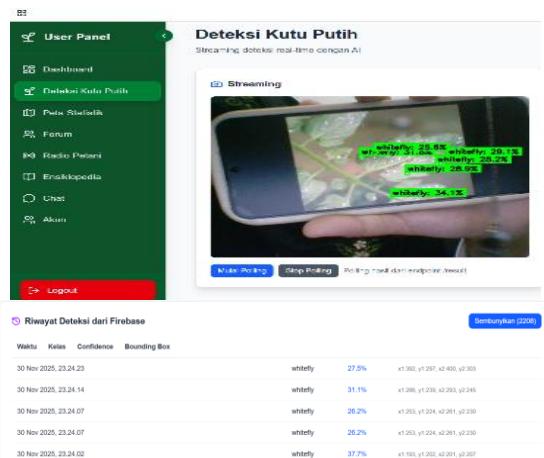
Gambar 23 Halaman Akun

Gambar 19 halaman tampilan menu akun untuk menampilkan daftar akun pengguna yang sudah melakukan pendaftaran pada *website* farmshield untuk dapat melakukan *login*.

4.3 Pengujian

Pengujian memastikan seluruh fitur FarmShield berjalan sesuai kebutuhan. Pengujian meliputi *autentikasi*, deteksi hama, forum, radio, profil pengguna, performa, kompatibilitas, serta *usability*. Hasil pengujian menunjukkan bahwa fungsi *Sign In*, *Logout*, dan Registrasi bekerja dengan baik, termasuk validasi terhadap *password* salah, email tidak terdaftar, dan *field* kosong.

Fitur Forum Petani berfungsi sesuai rancangan, di mana posting valid berhasil ditampilkan dan posting kosong ditolak. Fitur Komentar dapat disimpan dan dapat ditampilkan. Pada Fitur Ensiklopedia menampilkan daftar artikel secara lengkap dan tanpa *error*. Radio Petani juga bekerja dengan baik, termasuk fungsi pencarian episode. Fitur Profil Pengguna menampilkan data akun dengan benar, dan fitur Ubah *Password* berhasil mengganti *password* pengguna. Pengujian kompatibilitas menunjukkan *website* berjalan stabil pada Chrome, Firefox, dan Edge. Pengujian *usability* menunjukkan antarmuka mudah digunakan dan dipahami oleh pengguna.



Gambar 24 Pengujian Deteksi Kutu Putih

Pengujian dilakukan pada fitur Deteksi Kutu Putih menggunakan kamera secara langsung (*streaming*). Pada tampilan hasil uji, sistem berhasil mendekripsi beberapa objek kutu putih pada tanaman, yang ditunjukkan dengan *bounding box* berwarna hijau beserta label kelas dan nilai akurasi. Setiap objek yang terdeteksi menampilkan persentase akurasi, seperti whitefly: 34.15%, whitefly: 28.79%, dan whitefly: 26.31%.

5. KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa FarmShield berhasil dikembangkan sebagai *website* pendekripsi hama kutu putih dengan dukungan kamera dan model *machine learning*, serta dilengkapi fitur *dashboard*, forum, ensiklopedia, radio, dan chat yang semuanya berfungsi dengan baik. Sistem ini memiliki kelebihan berupa proses deteksi yang cepat dan *real-time*, antarmuka yang mudah digunakan, serta tersedianya berbagai fitur edukasi dan komunikasi bagi petani.

Namun, sistem masih memiliki beberapa kekurangan, seperti akurasi deteksi yang perlu ditingkatkan dan ketergantungan pada koneksi internet, sementara *dataset* yang digunakan juga masih terbatas. Ke depannya, FarmShield dapat dikembangkan dengan menambah *dataset* agar akurasi model meningkat, menambahkan fitur rekomendasi otomatis dan mode *offline*, serta memperluas kemampuan deteksi untuk hama dan penyakit lain pada tanaman tomat.

6. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa atas berkat-Nya yang diberikan sehingga artikel jurnal ini. Ucapan terima kasih juga ditujukan kepada Sekolah Vokasi IPB University, para dosen, dan teman-teman di Teknologi Rekayasa Komputer untuk dukungan yang telah diberikan. Penulis juga sangat menghargai para petani hortikultura, serta kepada Jurnal Informatika dan Teknik Elektro Terapan yang menjadi wadah untuk publikasi. Semoga penelitian ini bermanfaat bagi perkembangan ilmu pengetahuan dan sistem informasi di bidang pendidikan.

7. DAFTAR PUSTAKA

- [1] R. Soekarta, N. Nurdjan, dan A. Syah, “Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN),” vol. 8, no. 2, 2023.
- [2] M. Niawati, F. Ardhy, T. H. Andika, P. Bintoro, U. A. Pringsewu, dan J. A. Yani, “Aplikasi Diagnosis Penyakit Pada Tumbuhan Tomat Berbasis Website”.
- [3] T. Prakoso, H. Alpandari, F. Yuliani, dan K. Anwar, “Edukasi Pengendalian Kutu Putih Pada Tanaman Alpukat di Desa Ternadi Kabupaten Kudus,” 2024. [Daring]. Tersedia pada: <https://madaniya.biz.id/journals/content/s/article/view/794>
- [4] D. Adellia dkk., “JEPIN (Jurnal Edukasi dan Penelitian Informatika) Penerapan Metode Certainty Factor pada Sistem Pakar Diagnosa Hama dan Penyakit pada Tanaman Tomat”.
- [5] Nengah Riki dan Tata Sutabri, “Perancangan Aplikasi Pendekripsi Hama Tanaman Padi Berbasis Android,” *Uranus : Jurnal Ilmiah Teknik Elektro, Sains dan Informatika*, vol. 2, no. 4, hlm. 215–222, Nov 2024, doi: 10.61132/uranus.v2i4.510.
- [6] P. Simarmata, M. C. Tobing, dan A. Z. Siregar, “Beberapa Aspek Biologi Kutu Putih (*Paracoccus Marginatus*) (Hemiptera: Pseudococcidae) Pada Terung Di Rumah Kaca,” *Jurnal Agrotek Tropika*, vol. 9, no. 3, hlm. 377, Sep 2021, doi: 10.23960/jat.v9i3.4678.

- [7] W. Sabinus, B. Lena, dan Y. M. S. W. Puu, "Keragaman Jenis Hama Kutu Putih Pada Tanaman Singkong Di Kota Ende," *AGRICA*, vol. 11, no. 1, hlm. 51–59, 2018.
- [8] M. Dody Firmansyah, "Perancangan Web E-Commerce Berbasis Website pada Toko Ida Shoes," 2023.
- [9] Rio Renaldo Prasena dan Hendi Sama, "Studi Komparasi Pengembangan Website Dengan Framework Codeigniter Dan Laravel," 2020.
- [10] M. Nugraha, L. Sakinah, R. A. Setiawan, dan H. Mulyani, "Rancang Bangun Sistem Informasi Penerimaan Mahasiswa Baru Berbasis Web Dengan Menggunakan Framework Laravel," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr 2024, doi: 10.23960/jitet.v12i2.4179.
- [11] Faisal Maulana Akbar, Ayoubi Poerna Wardhanie2, dan Tan Amelia, "Implementasi Re-design UI/UX Website Fumigasi Untuk Meningkatkan Customer Experience," *Journal of Applied Computer Science and Technology*, vol. 4, no. 2, hlm. 90–99, Nov 2023, doi: 10.52158/jacost.v4i2.488.
- [12] M. Imbal Zaki Hasibuan dan T. Triase, "Implementasi Sistem Database Nosql Secara Realtime Menggunakan Firebase Realtime Database Pada Aplikasi Ourticle," *SIBATIK JOURNAL: Jurnal Ilmiah Bidang Sosial, Ekonomi, Budaya, Teknologi, dan Pendidikan*, vol. 2, no. 1, hlm. 1–24, Des 2022, doi: 10.54443/sibatik.v2i1.489.
- [13] M. Imbal Zaki Hasibuan dan T. Triase, "Implementasi Sistem Database Nosql Secara Realtime Menggunakan Firebase Realtime Database Pada Aplikasi Ourticle," *Sibatik Journal: Jurnal Ilmiah Bidang Sosial, Ekonomi, Budaya, Teknologi, dan Pendidikan*, vol. 2, no. 1, hlm. 1–24, Des 2022, doi: 10.54443/sibatik.v2i1.489.
- [14] A. Reservasi, L. Futsal, B. Web, T. Ardiansah, dan D. Hidayatullah, "Penerapan Metode Waterfall Pada," *Journal of Information Technology, Software Engineering, and Computer Science (ITSECS)*, vol. 1, no. 1, 2023.
- [15] A. A. Wahid, "Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK Oktober (2020) Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi".