

PROTOTYPE* APLIKASI IDENTIFIKASI PENYAKIT PADA DAUN JAGUNG BERBASIS ANDROID MENGGUNAKAN METODE *SCRUM

Walidy Rahman Hakim^{1*}, Agung Ahmad Ilyas², Beny Riswanto³, M. Hasymi Somaida⁴, Dede Yusuf⁵

^{1,2,3,4}STMIK Komputama Majenang; Jl. Majenang – Cimanggu KM 08 No. 99, Gandarukem,

Cilempuyang, Kec. Cimanggu, Kabupaten Cilacap, Jawa Tengah 53256; Telp. (0280) 6265594

⁵Universitas Al-Irsyad Cilacap; Jalan Cerme No. 24 Cilacap Jawa Tengah 53223; Telp. (0280) 532975

Keywords:

prototype;
daun jagung;
Scrum;
pertanian digital.

Correspondent Email:

walidyhakim@gmail.com

Abstrak. Jagung merupakan tanaman pangan yang penting dan menjadi komoditas strategis di sejumlah negara termasuk Indonesia, menempati posisi ketiga sebagai sumber karbohidrat setelah padi dan gandum. Produktivitas jagung dapat terpengaruh oleh berbagai faktor, salah satunya adalah serangan penyakit pada daun jagung yang menjadi hambatan serius dalam produksi. Penelitian ini bertujuan mengembangkan *prototype* aplikasi Android untuk identifikasi penyakit pada daun jagung dengan pendekatan metode *Scrum*. Permasalahan utama yang diangkat adalah keterbatasan petani dalam mengenali penyakit tanaman secara dini. Metode *Scrum* diterapkan dalam proses pengembangan perangkat lunak secara iteratif melalui *sprint-sprint* terstruktur. *Prototype* aplikasi dibangun menggunakan Figma. Hasil dari pengembangan menunjukkan bahwa pendekatan *Scrum* efektif dalam membangun *prototype* yang dapat digunakan untuk proses identifikasi awal penyakit daun jagung secara mandiri oleh petani. Implementasi sistem ini menunjukkan potensi besar dalam mendukung transformasi digital di sektor pertanian. Aplikasi menunjukkan kinerja yang baik dalam pengujian awal, dengan waktu proses yang singkat dan antarmuka yang dapat dipahami oleh pengguna. *Prototype* ini dapat menjadi dasar untuk pengembangan lebih lanjut menuju aplikasi yang siap pakai di lingkungan pertanian.



Copyright © [JITET](http://www.jitet.org) (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)

Abstract. Corn is an important food crop and a strategic commodity in a number of countries, including Indonesia, ranking third as a source of carbohydrates after rice and wheat. Corn productivity can be affected by various factors, one of which is disease attack on corn leaves, which poses a serious obstacle to production. This study aims to develop an Android application prototype for identifying corn leaf diseases using the *Scrum* methodology. The primary issue addressed is farmers' limited ability to detect plant diseases early. The *Scrum* methodology is applied in the software development process through iterative, structured sprints. The application prototype was built using Figma. The results of the development indicate that the *Scrum* approach is effective in building a prototype that can be used for the initial identification of corn leaf diseases by farmers independently. The implementation of this system shows great potential in supporting digital transformation in the agricultural sector. The application performed well in initial Testing, with short processing times and an interface that is understandable to users. This prototype can serve as a foundation for further development toward a ready-to-use application in agricultural environments.

1. PENDAHULUAN

Jagung merupakan tanaman pangan yang penting dan menjadi komoditas strategis di sejumlah negara termasuk Indonesia, menempati posisi ketiga sebagai sumber karbohidrat setelah padi dan gandum [1]. Produktivitas jagung dapat terpengaruh oleh berbagai faktor, salah satunya adalah serangan penyakit pada daun jagung yang menjadi hambatan serius dalam produksi [2]. Produksi jagung sering kali menurun akibat gangguan penyakit, terutama pada bagian daun, yang berdampak langsung terhadap hasil panen [3]. Identifikasi dini terhadap penyakit daun sangat penting untuk menghindari penyebaran penyakit secara masif dan untuk meningkatkan produktivitas. Selama ini, proses identifikasi penyakit daun jagung masih dilakukan secara manual melalui pengamatan visual oleh petani atau ahli pertanian. Pendekatan ini tidak hanya memerlukan keahlian khusus, tetapi juga rawan terhadap kesalahan subjektif serta memakan waktu yang cukup lama [3]. Untuk itu, dibutuhkan sistem berbasis teknologi yang dapat membantu proses identifikasi secara cepat, akurat, dan efisien.

Untuk menjawab tantangan tersebut, penelitian ini bertujuan mengembangkan sebuah *prototype* aplikasi Android yang dapat membantu pada pengembangan aplikasi identifikasi penyakit pada daun jagung [4]. Pengembangan sistem dilakukan dengan pendekatan *Scrum*, yaitu metodologi Agile yang berfokus pada iterasi pendek dan kolaborasi tim. *Scrum* memungkinkan pengembangan aplikasi dilakukan secara bertahap melalui *sprint* yang terukur dan dapat dievaluasi secara rutin [5].

Fokus penelitian ini bukan hanya pada akurasi teknis model klasifikasi citra, tetapi pada proses perancangan, pembangunan, dan evaluasi awal *prototype* sistem. Diharapkan dengan pendekatan ini, *prototype* yang dihasilkan bisa menjawab kebutuhan pengguna dan dapat dijadikan dasar pengembangan sistem lebih lanjut.

2. TINJAUAN PUSTAKA

2.1. Android

Merupakan *operating system* yang bersifat terbuka, dengan Google mengeluarkan kode sumbernya di bawah Lisensi Apache.

Ketersediaan kode sumber terbuka dan lisensi izin pada Android memungkinkan pengembang perangkat lunak untuk melakukan modifikasi secara bebas dan mendistribusikannya, memperluas komunitas pembuat perangkat, operator nirkabel, dan pengembang aplikasi [6]. Android memiliki komunitas pengembang aplikasi yang besar, yang melengkapi fungsionalitas perangkat biasanya ditulis dalam bahasa pemrograman Java. Oktober 2013, jumlah aplikasi Android melebihi satu juta, sekitar 50 miliar unduhan dari Google Play Store, platform utama distribusi aplikasi Android. Survei pada April-Mei 2013 menunjukkan bahwa Android adalah platform paling diminati oleh pengembang aplikasi bergerak, digunakan oleh 71% dari mereka. Google melaporkan lebih dari satu miliar pengguna aktif bulanan Android di Google I/O 2014, naik dari 583 juta pada Juni 2013. Android telah menjadi sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengungguli Symbian pada tahun 2010, dan menjadi pilihan utama bagi perusahaan teknologi yang mencari solusi sistem operasi yang murah, dapat disesuaikan, dan ringan untuk perangkat berteknologi tinggi. Android juga mendorong perkembangan komunitas pengembang aplikasi melalui penggunaan kode sumber terbuka sebagai dasar untuk proyek pembuatan aplikasi baru atau penggunaan pada perangkat dengan sistem operasi lain.

Pada November 2013, Android mendominasi pangsa pasar telepon pintar global dengan 64% (Maret 2013), didukung oleh produk-produk Samsung. Pada Juli 2013, ada lebih dari 11.868 perangkat Android berbeda dengan berbagai versi. Keberhasilan Android juga menjadikannya sasaran dalam perseteruan paten antara perusahaan teknologi.

2.2. Jagung Pakan

Salah satu komoditas penting dalam sektor pertanian, khususnya dalam industri peternakan, jagung pakan merujuk pada jagung yang secara khusus digunakan sebagai bahan utama dalam pakan ternak, terutama unggas, babi, dan sapi. Peran jagung pakan sangat vital karena menjadi sumber energi utama bagi hewan ternak, terutama karena kandungan karbohidratnya yang tinggi.

2.3. Penyakit Daun Jagung

Penyakit daun jagung adalah salah satu tantangan terbesar dalam membudidayakan jagung karena setiap penyakit daun memiliki ciri, karakteristik, gejala, dan penanganan yang berbeda-beda. Jika salah dalam mengidentifikasi penyakitnya, maka akan berdampak besar bagi tanaman jagung itu sendiri. Dari berbagai jenis-jenis penyakit pada daun jagung yang dapat mempengaruhi produksi jagung [7]. Dalam penelitian ini hanya berfokus pada beberapa jenis penyakit daun jagung yang ditemukan secara umum dalam budidaya jagung, antara lain karat daun, hawar daun, bintik daun.

2.4. Figma

Platform desain grafis berbasis web yang memungkinkan pengguna untuk membuat berbagai macam konten visual seperti antarmuka pengguna, prototipe, dan desain kolaboratif lainnya. Figma didirikan pada tahun 2012 oleh Dylan Field dan Evan Wallace. Platform ini dirancang untuk memudahkan pengguna dari berbagai latar belakang, baik yang berpengalaman dalam desain grafis maupun yang masih pemula, untuk menghasilkan desain yang menarik dengan cepat dan mudah, serta mendukung kolaborasi real-time di antara tim.

2.5. Scrum

Adalah kerangka kerja ringan yang membantu orang, tim, dan organisasi menghasilkan nilai melalui solusi adaptif untuk masalah yang kompleks. Metodologi ini menekankan pada transparansi, inspeksi, dan adaptasi untuk mencapai hasil yang diinginkan. *Scrum* merupakan salah satu metode yang mengimplementasikan Agile sebagai pengembangan perangkat lunak [8]. *Scrum* diciptakan oleh Jeff Sutherland sekitar tahun 1993 dengan maksud sebagai cara untuk mengembangkan dan mengelola proyek yang sejalan dengan prinsip Agile. Kemudian, proses pengembangan *Scrum* dilanjutkan oleh Ken Schwaber dan Beedle. *Scrum* dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, dengan tetap menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif.

Kerangka kerja yang ada di *Scrum* memiliki peran penting dalam keberhasilan penggunaan metode ini. Aturan main yang disediakan juga

akan mengharmonisasikan setiap kegiatan, peran, kelengkapan artefak, interaksi setiap elemennya. *Event* dilakukan untuk menciptakan rutinitas dan mengurangi pertemuan luar *Scrum*. Artefak *Scrum* menunjukkan nilai bisnis dan tugas untuk meningkatkan transparansi dan kemampuan untuk menginspeksi dan mengadaptasi. Tujuan dari struktur *Scrum* adalah untuk memastikan bahwa semua orang memiliki pemahaman yang sama tentang informasi penting [8].

Scrum memiliki prinsip-prinsip yang jika diimplementasikan dan diperlihatkan dalam Tim *Scrum*, maka fondasi-fondasi *Scrum* seperti keterbukaan, peninjauan, dan penyesuaian akan terwujud dan hubungan saling percaya antarpeserta akan terbentuk [9]. Nilai-nilai tersebut diantaranya sebagai berikut:

a. Komitmen

Setiap kelompok dalam *Scrum* wajib dapat mengelola dan memonitor diri mereka tanpa bantuan eksternal. Setiap individu dalam kelompok harus menyadari tanggung jawab serta peran yang telah disetujui bersama.

b. Fokus

Framework Scrum mengurangi gangguan, sehingga tiap anggota tim dapat mencurahkan perhatian mereka pada tugas yang harus diselesaikan.

c. Keberanian

Dengan keberanian anggota Tim *Scrum* harus melakukan yang terbaik untuk mencapai hasil maksimal dalam mengatasi masalah.

d. Keterbukaan

Setiap individu dalam Tim *Scrum* perlu bersikap transparan mengenai semua aktivitas yang mereka lakukan kepada rekan-rekan tim serta di luar tim. Keterbukaan ini mencakup pekerjaan yang sedang dilakukan dan perencanaan ke depan serta tantangan yang dihadapi.

e. Rasa Hormat

Setiap individu *Scrum Team* wajib menghargai serta menghormati kontribusi yang dihasilkan oleh rekan-rekan tim. Setiap orang membawa beragam kemampuan dan keahlian yang unik. Susunan tim dalam kerangka *Scrum* bertujuan untuk memaksimalkan kelincahan,

inovasi, dan hasil kerja. Struktur *Scrum* Team telah terbukti meningkatkan efektivitas tim dalam menangani berbagai jenis tugas dan pekerjaan yang rumit.

f. Product owner

Adalah individu yang memiliki tugas untuk menetapkan rincian dan meningkatkan nilai komersial dari aplikasi yang akan dikembangkan, pemilik produk juga bertanggung jawab untuk mengelola daftar produk yang perlu dikelola oleh tim pengembangan yang meliputi hal-hal ini yaitu Menjelaskan dengan jelas mengenai komponen dari daftar produk, Mengatur daftar produk menggunakan cara terbaik untuk mencapai tujuan dan misi, meningkatkan nilai bisnis dari pekerjaan tim pengembang, memastikan *backlog* produk terlihat transparan dan jelas bagi semua pihak dan menunjukkan pekerjaan yang akan dilakukan lalu memastikan bahwa tim pengembang hingga batas waktu tertentu agar memahami *backlog* produk.

g. Development team

Tim pengembangan terdiri dari berbagai profesional yang ahli, termasuk desainer, pemrogram, penguji, dan lain-lain. Di akhir setiap sprint, tim pengembangan bertanggung jawab untuk membuat produk atau hasil yang dapat dirilis. Mereka pula yang memegang tanggung jawab atas daftar produk yang telah disampaikan oleh pemilik produk sebelumnya, di mana setiap anggota memiliki tanggung jawab terhadap setiap item yang telah dibagi dan memahami tugas yang harus dilakukan selanjutnya [9].

2.6. Scrum Master

Salah satu individu yang bertanggung jawab untuk memperkenalkan dan mendukung implementasi kerangka kerja *Scrum* adalah *Scrum* Master. *Scrum* Master juga bertanggung jawab untuk membantu seluruh anggota tim memahami teori, praktik, peraturan, dan nilai-nilai yang terkandung dalam kerangka kerja *Scrum*. Salah satu tanggung jawab *Scrum* Master adalah memastikan bahwa setiap anggota menerapkan *Scrum* dengan benar dan efisien. Selain itu, *Scrum* Master membantu pemilik produk dalam menentukan cara terbaik untuk mengelola *backlog* produk yang telah ditentukan, dan jika diperlukan, membantu

mengatur acara *Scrum* [9]. *Sprint* adalah inti dari kerangka kerja *scrum*, *sprint* berlangsung selama satu hingga empat minggu, sepanjang siklus pengembangan produk, durasi sprint tetap sama, Setelah *sprint* sebelumnya selesai, *sprint* baru akan dimulai. *Sprint* dianggap selesai jika produk yang dihasilkan memenuhi kriteria penerimaan.

a. Product Backlog

Adalah kumpulan kebutuhan proyek yang diurutkan berdasarkan prioritas. Kebutuhan ini dapat mencakup fungsi, fitur, peningkatan, dan perbaikan yang akan dilakukan. Mereka juga dapat berfungsi sebagai dasar untuk semua perubahan yang harus dilakukan pada produk tersebut. Untuk membuat produk layak, pemilik produk menjelaskan urutan, isi, dan ketersediaan stok produk. Stok produk terus berubah. Produk *backlog* terus berubah seiring dengan evolusi produk dan konteks penggunaannya. Daftar produk akan tetap ada selama produk tersebut masih ada.

b. Sprint Planning

Adalah fase analisis yang menekankan pada kebutuhan pengguna yang ingin dipenuhi. Lama waktu untuk *Sprint Planning* dapat bervariasi, berdasarkan kompleksitas dan skala syarat pengguna yang akan dipenuhi. *Sprint Planning* merencanakan apa yang harus dilakukan dalam satu *sprint*. Perencanaan ini dilakukan secara kolektif oleh setiap anggota Tim *Scrum*.

c. Sprint Backlog

Adalah kumpulan item dari *Sprint Backlog* adalah proyeksi dari Tim Pengembangan tentang fitur-fitur yang akan dimasukkan ke dalam *increment* yang akan datang dan dipilih untuk digunakan selama sprint guna mencapai *Sprint Goal*. serta tugas-tugas yang harus diselesaikan untuk menyelesaikan *increment* tersebut. Tugas yang dinilai oleh *Development Team* merupakan hal penting untuk meraih *Sprint Goal* dicantumkan dalam *Sprint Backlog*. *Sprint Backlog* harus mencakup setidaknya satu perbaikan proses dengan prioritas tinggi berdasarkan hasil dari *Sprint Retrospective* yang telah dilakukan sebelumnya untuk memastikan perbaikan yang berkesinambungan [9].

d. *Sprint Execution*

adalah proses di dalam *Scrum* di mana tim bekerja untuk menyelesaikan tugas-tugas dan kegiatan yang diperlukan untuk mencapai tujuan *sprint* (*sprint goal*). Proses ini mencakup perencanaan tugas, pelaksanaan, manajemen tugas, serta komunikasi antar anggota tim selama *sprint* berlangsung. Tujuan utama dari *Sprint Execution* adalah untuk memastikan bahwa *Product Backlog Items* (PBIs) yang dipilih dapat diselesaikan dalam waktu *sprint* yang telah ditentukan, biasanya 1-4 minggu.

e. *Daily Scrum*

Pada umumnya memiliki durasi sekitar lima belas menit untuk setiap pertemuan dan berlangsung selama *sprint*. Untuk menerapkan *Daily Scrum*, jumlah anggota dan jenis pembahasan dapat diubah. Pada titik ini, kemajuan tim membuat rencana kerja untuk 24 jam berikutnya. *Daily Scrum* memeriksa pekerjaan yang telah dilakukan sebelumnya untuk meningkatkan kerja sama dan kinerja tim [9]. Pada *Daily Scrum*, orang berbicara tentang apa yang telah dilakukan sejak pertemuan tim sebelumnya, masalah yang ditemui selama pengerjaan, rencana untuk menyelesaikannya, dan rencana kerja yang disusun untuk mencapai sesuatu sebelum pertemuan tim berikutnya. Tujuan dari langkah ini adalah untuk mengetahui berapa banyak pekerjaan yang harus diselesaikan selama sisa waktu *sprint* [9].

f. *Sprint Review*

Pada akhir *sprint*, peninjauan *sprint* dilakukan untuk mengevaluasi peningkatan dan meninjau item *backlog* produk yang belum diselesaikan. Peninjauan ini juga menilai proses kerja tim untuk menentukan perbaikan apa yang perlu dilakukan untuk proses pengembangan di *sprint* berikutnya [9].

g. *Sprint Retrospective*

Adalah momen bagi tim *Scrum* untuk menganalisis kinerja mereka dan merumuskan strategi untuk meningkatkan *sprint* yang akan datang. Sebelum perencanaan *sprint* selanjutnya, kegiatan ini dilakukan setelah *Review Sprint*, *Scrum* Master memastikan *Sprint Retrospective* berjalan dengan baik bahwa setiap anggota memiliki pemahaman yang jelas tentang tujuannya. Sebelum *Sprint Retrospective* selesai, tim *Scrum* harus

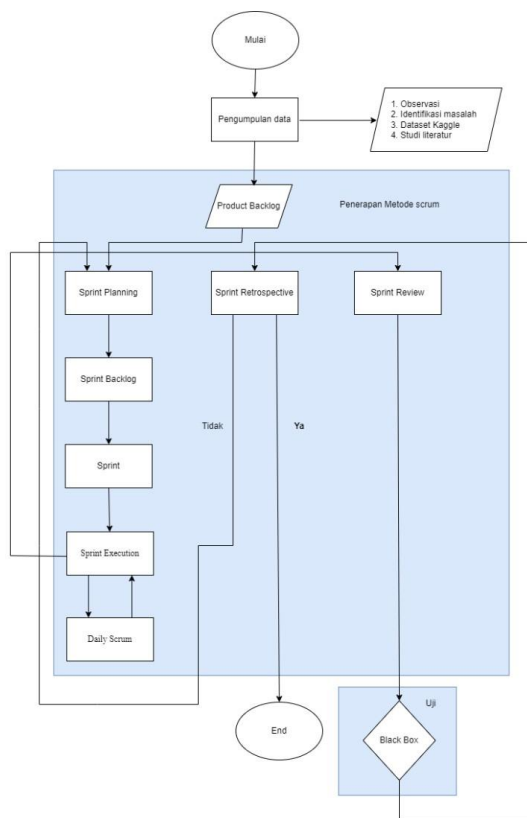
menyepakati langkah-langkah perbaikan yang akan diterapkan dalam *sprint* mendatang [9]. Perbaikan yang dilakukan oleh *Scrum Team* pada *sprint* yang akan datang mencerminkan kemampuan beradaptasi dari tim tersebut. Meski perbaikan bisa dilakukan kapan saja di sepanjang *sprint*, *Sprint Retrospective* memberikan kesempatan formal untuk melakukan pemeriksaan dan penyesuaian.

2.7. *Black Box Testing*

Merupakan pendekatan di mana seorang penguji hanya mengetahui tujuan dari perangkat lunak yang diuji. Mereka tidak memahami detail operasional dari perangkat lunak tersebut. Dengan demikian, pengujian hanya bergantung pada hasil yang dihasilkan dari input yang diberikan tanpa memperhatikan bagaimana atau apa yang terjadi di dalamnya. Dengan kata lain, *Black Box Testing* menitikberatkan pada aspek fungsional dari perangkat lunak. Pengujian dapat melibatkan definisi kondisi input yang berbeda dan mengevaluasi apakah perangkat lunak sesuai dengan spesifikasi fungsionalnya [5].

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan pengembangan sistem perangkat lunak dengan metode *Scrum* yang merupakan kerangka kerja Agile yang menekankan pengembangan iteratif dan kolaboratif melalui *sprint* yang terstruktur yang dapat dikembangkan lagi untuk mendeteksi penyakit pada daun jagung secara mandiri oleh petani [10].



Gambar 1 alur penelitian

3.1. Pengumpulan Data

Pengumpulan data mencakup data kualitatif yang memberikan konteks lebih mendalam mengenai masalah dan kondisi di lapangan, data kuantitatif yang esensial untuk analisis statistik dan pemodelan, serta kombinasi ini memastikan penelitian memiliki landasan yang kuat untuk di bangun sebuah *prototype* lalu dilakukan pengembangan aplikasi identifikasi penyakit daun jagung kedepannya.

a. Observasi Lapangan

Melakukan observasi langsung ke lapangan PT. Benih Citra Asia untuk memahami kondisi sebenarnya dari tanaman jagung, bagaimana penyakit terdeteksi diidentifikasi secara manual, berbagai faktor yang mempengaruhi kesehatan tanaman dan jenis penyakit yang ditemukan.

Tabel 1. jenis Penyakit

No	Jenis Penyakit	Gambar
1	Bercak daun abu-abu	

2	Hawar daun	
3	Karat jagung	
4	Daun Sehat	

Tabel 1 menyajikan 4 data jenis daun jagung yang memiliki ciri khusus diambil dari kaggle, yaitu bercak daun abu-abu, hawar daun, karat daun dan daun sehat

b. Identifikasi Masalah

Pemahaman Masalah di PT. Benih Citra Asia Identifikasi dan deskripsi masalah yang dihadapi dalam identifikasi penyakit daun jagung di lapangan, termasuk tantangan dan kebutuhan spesifik, Hasil identifikasi menggambarkan beberapa penyakit utama yang sering menyerang daun tanaman jagung, yaitu bercak daun, karat daun, dan hawar daun. Bercak daun disebabkan oleh Jamur *Cercospora nicotianae* yang berkembang pada kondisi kelembaban tinggi, sedangkan karat daun disebabkan oleh cendawan *Puccinia polysora* dan *Puccinia sorghi* yang menyebar melalui angin [11]. Hawar daun, di sisi lain disebabkan oleh jamur *Helminthosporium turcicum* yang dapat bertahan di tanaman dan tanah. Pengendalian penyakit- penyakit ini meliputi penggunaan fungisida yang tepat, rotasi tanaman, dan pemilihan varietas yang tahan terhadap penyakit tersebut.

c. Dataset Dari Kaggle

Dataset yang sudah disediakan dari kaggle ini mencakup gambar daun jagung dengan berbagai penyakit yang didokumentasikan dengan baik, Dimana pada proses ini penulis memperoleh *dataset* mengenai penyakit pada tanaman daun jagung khususnya pada penampang daunnya yang diambil dari kaggle. *Dataset* ini terdiri dari 4 kelas yaitu *Blight*, *Common Rust*, *Gray Leaf Spot*, dan *Healthy*

dengan total citra keseluruhan sebanyak 4188 citra dengan tipe file JPG.

d. Studi Literatur

Studi literatur ini mencakup pencarian referensi dan informasi dari berbagai sumber seperti jurnal, konferensi, dan dokumentasi teknis yang terkait dengan deteksi penyakit pada tanaman. Tujuan dari studi literatur ini adalah untuk memvalidasi konsep, mengumpulkan wawasan terkini, dan menemukan metode yang efektif dalam menerapkan teknologi untuk meningkatkan deteksi penyakit pada tanaman jagung [3]. Analisis studi sebelumnya yang relevan dengan identifikasi penyakit tanaman menggunakan teknologi serupa untuk memahami metodologi, hasil, dan kesimpulan yang dapat diambil.

3.2. Product owner

Untuk menentukan spesifikasi aplikasi dan memaksimalkan nilai bisnis, adalah tanggung jawab pemilik produk. Mereka juga bertanggung jawab untuk mengelola stok produk yang harus dilakukan tim pengembang yang terdiri dari hal-hal berikut:

- a. Menjelaskan dengan tegas mengenai konten dari *product backlog*.
- b. Mengatur *product backlog* dengan metode yang paling efektif untuk meraih tujuan dan misi.
- c. Mengoptimalkan nilai bisnis dari tugas-tugas yang dilaksanakan oleh tim pengembang.
- d. Memastikan bahwa *product backlog* terlihat, terbuka, dan jelas untuk semua pihak, sambil menunjukkan pekerjaan *Scrum Team* yang akan dilakukan berikutnya.
- e. Memastikan bahwa *product backlog* telah dimengerti oleh tim pengembang hingga tingkat tertentu.

3.3. Development team

Tim pengembang terdiri dari berbagai profesi terampil seperti *desainer*, *pemrogram*, *penguji*, dan lain-lain. Tim ini berfokus pada pembuatan produk atau hasil yang dapat diluncurkan pada akhir setiap *sprint*. Mereka juga bertanggung jawab atas daftar *backlog produk* yang telah disediakan oleh pemilik produk sebelumnya, di mana setiap anggota mempunyai tanggung jawab atas *backlog* yang

telah dibagi serta memahami apa yang harus dikerjakan selanjutnya [9].

3.4. Scrum Master

Adalah individu yang memiliki tugas untuk memperkenalkan serta mendukung penerapan kerangka kerja *Scrum*. Mereka mendampingi semua anggota tim dalam memahami konsep, praktik, peraturan, dan nilai-nilai yang terkandung dalam *Scrum*. *Scrum Master* juga bertanggung jawab untuk mengawasi dan memimpin kerangka kerja *Scrum*. Tanggung jawab mereka juga mencakup memastikan bahwa setiap anggota tim *Scrum* menerapkan *Scrum* dengan cara yang tepat. Jika diminta atau diperlukan, *Scrum Master* membantu *Product Owner* menemukan cara terbaik untuk mengelola daftar produk yang telah ditentukan sebelumnya. Selain itu, *Scrum Master* membantu mengatur acara *Scrum*. [9].

Sprint adalah inti dari kerangka kerja *Scrum* dan berlangsung selama satu hingga empat minggu. Durasi *sprint* tetap sama selama seluruh proses pengembangan produk. Setelah menyelesaikan *sprint* yang sebelumnya, *sprint* berikutnya akan dimulai. Jika hasil produk yang diperoleh dari *sprint* memenuhi kriteria penerimaan, maka *sprint* dianggap telah selesai.

3.5. Product Backlog

Adalah kumpulan prioritas dari kebutuhan dalam sebuah proyek yang dapat mencakup peningkatan fungsi dan perbaikan fitur dari fitur yang akan dilakukan. Mereka juga dapat berfungsi sebagai sumber dari semua perubahan yang harus dilakukan pada produk. *Owner* produk menentukan urutan, komposisi, dan ketersediaan *backlog* produk. *Backlog* ini fleksibel dan terus berubah untuk menjaga kelayakan produk. Ini berkembang seiring dengan kemajuan produk dan kondisi penggunaannya.

3.6. Sprint Planning

Fase analisis yang berfokus pada kebutuhan pengguna adalah perencanaan jalur, *Sprint Planning* bervariasi dalam jangka waktu. Dalam hal kebutuhan pengguna yang akan ditangani, kerangka kerja *Scrum* luas dan rumit. *Sprint Planning* merencanakan apa yang harus dilakukan dalam satu *sprint*. Perencanaan ini

dilakukan secara bersama-sama oleh setiap anggota Tim *Scrum*.

3.7. *Sprint Backlog*

Merupakan sekumpulan item dari *Sprint Backlog* yaitu estimasi dari Tim Pengembangan tentang fitur-fitur yang akan ditambahkan ke dalam *increment* selanjutnya, dan produk *backlog* yang dipilih untuk dicapai selama *sprint* untuk mencapai *Sprint Goal*, serta tugas-tugas yang harus diselesaikan untuk menyelesaikan *increment* itu. *Sprint Backlog* menunjukkan pekerjaan yang harus dilakukan oleh Tim Pengembangan untuk mencapai *Sprint Goal*. Untuk memastikan peningkatan berkelanjutan, *Sprint Backlog* berisi paling sedikit satu proses peningkatan yang sangat penting dari hasil pertemuan *Sprint Retrospective* sebelumnya [9].

3.8. *Sprint Execution*

Adalah proses di dalam *Scrum* di mana tim bekerja untuk menyelesaikan tugas-tugas dan kegiatan yang diperlukan untuk mencapai tujuan *sprint* (*sprint goal*). Proses ini mencakup perencanaan tugas, pelaksanaan, manajemen tugas, serta komunikasi antar anggota tim selama *sprint* berlangsung. Tujuan utama dari *Sprint Execution* adalah untuk memastikan bahwa *Product Backlog Items* (PBIs) yang dipilih dapat diselesaikan dalam waktu *sprint* yang telah ditentukan, biasanya 1-4 minggu.

3.9. *Daily Scrum*

Proses ini umumnya berlangsung sekitar 15 menit setiap kali diadakan selama periode *sprint*. Dalam praktiknya, waktu *Daily Scrum* dapat bervariasi bergantung pada jumlah anggota serta topik yang dibahas. Pada titik ini, tim pengembang menyiapkan agenda untuk 24 jam ke depan. *Daily Scrum* bertujuan untuk meningkatkan kolaborasi dan efisiensi tim dengan meninjau tugas yang telah diselesaikan sejak *Daily Scrum* terakhir. Diskusi dalam *Daily Scrum* meliputi: apa yang telah diselesaikan sejak pertemuan tim terakhir, hambatan yang dihadapi selama proses pengerjaan, strategi untuk menyelesaikan hambatan tersebut, dan rencana kerja yang dibuat sebelum pertemuan tim berikutnya untuk mencapai tujuan. Sasaran fase ini adalah untuk memahami sisa tugas yang harus diselesaikan selama *sprint* [9].

3.10. *Sprint Review*

Sprint Review dilakukan di akhir *sprint* untuk mengevaluasi peningkatan dan mempertimbangkan item *Product Backlog* yang sudah serta belum selesai. Evaluasi dan pertimbangan tidak hanya terpaku pada *output*, tetapi juga mencakup penilaian terhadap cara kerja tim agar bisa diidentifikasi perbaikan yang diperlukan untuk proses pengembangan di *sprint* berikutnya [9].

3.11. *Sprint Retrospective*

Scrum Team memiliki kesempatan untuk meninjau kembali *sprint* dan membuat rencana untuk meningkatkan *sprint* berikutnya. [9]. *Sprint Retrospective* diadakan setelah *Sprint Review* dan sebelum perencanaan *Sprint* yang mendatang. Scrum Master bertanggung jawab untuk memastikan bahwa *Sprint Retrospective* ini dilaksanakan dan semua anggota memahami maksudnya. Sebelum sesi *Sprint Retrospective* selesai, *Scrum Team* harus mencapai kesepakatan mengenai perbaikan yang akan mereka laksanakan pada *sprint* mendatang. Perbaikan yang diambil oleh *Scrum Team* pada *sprint* berikutnya merupakan hasil dari proses penyesuaian oleh tim itu sendiri. Meskipun perbaikan bisa dilakukan kapan saja dalam *sprint*, *Sprint Retrospective* merupakan momen resmi yang menyoroti pada pemeriksaan dan penyesuaian.

4. HASIL DAN PEMBAHASAN

4.1. *Product backlog*

Proses ini disusun sebelum dilaksanakannya pembentukan *sprint* yang didiskusikan oleh tim *Scrum* yang di sepakati [12]. Tahapan yang dilaksanakan sebelum *Sprint* yang diikuti oleh seluruh anggota tim *Scrum* untuk mendefinisikan mengenai semua kebutuhan ketika diadakannya *Scrum*. Tim *Scrum* yang dibentuk adalah sebagai berikut:

a. *Product Owner*

Penentuan pemilik produk didasarkan pada pihak yang paling bertanggung jawab dalam pemberian informasi, pemberian informasi kebutuhan, serta penyediaan data terkait pengembangan system sesuai dengan yang diinginkan. Pada kasus ini ditentukan *product owner* adalah manager PT. Benih Citra Asia.

b. Scrum Master

Merupakan pihak yang mengatur pengelolaan dari seluruh proses *Scrum* kepada seluruh tim *Scrum* yang ada sehingga dapat berjalan dengan baik. Pada penelitian ini ditentukan *Scrum* master adalah ketua peneliti penelitian ini.

c. Tim Scrum

Merupakan seluruh anggota tim yang bertugas untuk menjalankan proyek berupa proses analisis bisnis, analisis system, pengembangan sistem, dan pengujian sistem sesuai keinginan *product owner*.

Tabel 2 Struktur Produk *Backlog*

NO	<i>Backlog</i> Item	Prioritas
1	Menu Dasbor	Tinggi
2	Menu Identifikasi Dengan Kamera	Tinggi
3	Hasil Deteksi	Tinggi
4	Menu Bantuan	Sedang
5	Menu Tentang	Sedang

Tabel 2 menjelaskan prioritas dari berbagai kebutuhan pengguna terkait *prototype* aplikasi identifikasi penyakit daun jagung, serta item *backlog* yang terkait dengan setiap kebutuhan tersebut.

4.2. Sprint Planning

Sprint Planning merupakan fase di mana tim melakukan peninjauan terhadap *Product Backlog* untuk setiap *sprint*, yang umumnya berlangsung selama 2-4 minggu [13]. Dalam kasus ini, *Sprint* Planning dibagi menjadi 4 *sprint* dengan durasi 1 minggu untuk setiap *sprint*. Berikut *sprint* planning :

Tabel 3 Daftar *Sprint Planning*

<i>Sprint</i>	Durasi	Item Produk <i>Backlog</i>
<i>Sprint</i> 1	1 Minggu	Analisa dan klasifikasi <i>dataset</i> gambar daun jagung
<i>Sprint</i> 2	1 Minggu	Menu Dasbor, Identifikasi via Kamera, Identifikasi Dari Galeri
<i>Sprint</i> 3	1 Minggu	Hasil Deteksi, Rekomendasi Pengobatan

<i>Sprint</i> 5	1 Minggu	Menu Bantuan dan Tentang
-----------------	----------	--------------------------

Dalam *Sprint* Planning ini, Setiap *sprint* akan berfokus pada *prototype* aplikasi identifikasi penyakit berbasis android.

4.3. Sprint Backlog

Merupakan hasil dari tahapan *sprint* planning yang berisikan daftar *product backlog* dengan pemecahan pekerjaan beserta role dalam penyelesaiannya [14]. Struktur rancangan *sprint backlog* adalah sebagai berikut:

a. Sprint 1

Tabel 4 *Sprint* 1

Estimasi Durasi (Hari)	Tugas
3	Analisa <i>Dataset</i> Gambar daun jagung
3	Klasifikasi <i>Dataset</i> Gambar daun jagung

Pada tabel 4 analisa *dataset* gambar daun jagung dari kaggle dilakukan selama 3 hari dan klasifikasi *dataset* gambar daun jagung dilakukan 3 hari.

b. Sprint 2

Tabel 5 *Sprint* 2

Estimasi Durasi (Hari)	Tugas
3	Menu <i>Dashboard</i>
2	Menu identifikasi via kamera
1	Menu identifikasi via galeri

Pada tabel 5 menu *dashboard* dikerjakan selama 3 hari, menu identifikasi via kamera dikerjakan selama 2 hari dan menu identifikasi via galeri dikerjakan selama 1 hari.

a. Hasil akhir untuk menu *dashboard* pada gambar 2 *Dashboard*.



Gambar 2. Dashboard

- b. Hasil akhir untuk menu identifikasi via kamera pada gambar 3 berikut.



Gambar 3. Identifikasi via kamera

- c. Hasil akhir untuk menu identifikasi via galeri pada gambar 4 berikut



Gambar 4. Identifikasi via galeri

c. *Sprint 3*

Tabel 6 *Sprint 3*

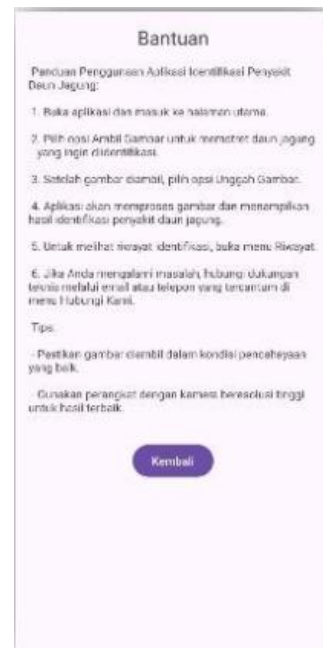
Estimasi Durasi (Hari)	Tugas
4	Hasil Deteksi
2	Rekomendasi Pengobatan

Pada tabel 6 hasil deteksi dikerjakan selama 4 hari dan rekomendasi pengobatan dikerjakan selama 2 hari.

- 1) Hasil akhir deteksi daun jagung dan rekomendasi pengobatan menjadi satu tampilan seperti pada gambar 5 berikut.



Gambar 5. Hasil deteksi dan rekomendasi



Gambar 6. Menu Bantuan

d. *Sprint 4*

Tabel 7 *Sprint 4*

Estimasi Durasi (Hari)	Tugas
3	Menu Bantuan
3	Menu Tentang

Sprint 4 berfokus pada pengembangan fitur "Bantuan" dan "Tentang" dalam aplikasi. Fitur ini bertujuan untuk memberikan panduan penggunaan aplikasi dan informasi tentang aplikasi serta tim pengembang. Pada tabel 7 menu bantuan dikerjakan selama 3 hari dan menu tentang dikerjakan selama 3 hari.

1) Hasil akhir menu bantuan seperti pada gambar 6 berikut.



Gambar 7. Menu Tentang

4.4. *Sprint Execution*

Selama pelaksanaan *Sprint*, desain sistem dilakukan secara mendetail dengan tujuan mendefinisikan arsitektur produk dan antar muka pengguna yang akan dikembangkan menggunakan *platform* figma.

4.5. *Daily Scrum*

Setiap hari, para pengembang menyediakan waktu selama 1 hingga 15 menit sebelum atau setelah menjalankan tugas *sprint* untuk berkomunikasi dengan *Scrum* master. Beberapa hal yang dibahas sebagai berikut:

- Laporan tentang pekerjaan yang telah diselesaikan pada hari sebelumnya.
- Rencana pekerjaan yang akan dilakukan pada hari tersebut.
- Identifikasi permasalahan atau hambatan yang mungkin dihadapi selama proses pengembangan.

4.6. *Sprint Review*

Sprint Review pada *sprint* 4 dilakukan di mana tim pengembang mempresentasikan hasil kerja kepada *product owner* untuk mendapatkan umpan balik dan persetujuan yaitu tanggapan mengenai apakah hasil kerja sudah sesuai dengan ekspektasi, serta persetujuan untuk melanjutkan ke tahap berikutnya. Persetujuan ini menunjukkan bahwa hasil kerja dianggap sudah memenuhi kriteria yang diharapkan, berikut ini Hasil *Sprint* 4:

- Halaman Bantuan selesai dengan memberikan panduan komprehensif kepada pengguna.
- Halaman Tentang telah diimplementasikan dengan informasi lengkap tentang aplikasi dan tim pengembang.

4.7. *Sprint Retrospective*

Sprint Retrospective membahas proses yang dilakukan selama *sprint*, mengidentifikasi keberhasilan dan kendala yang dihadapi, serta memperbaiki strategi untuk *sprint* berikutnya, berikut Hasil *Sprint Retrospective*:

- Tim berhasil menyelesaikan tugas-tugas utama seperti halaman Bantuan dan halaman Tentang dengan baik, sesuai dengan rencana *sprint*. Namun, tim menemukan beberapa kendala, seperti adanya hambatan dalam komunikasi yang mempengaruhi efisiensi kerja.
- Hasil diskusi dalam *Retrospective* ini menunjukkan bahwa komunikasi antar anggota tim perlu ditingkatkan. Sebagai tindak lanjut, tim memutuskan untuk mengadopsi strategi baru seperti pertemuan harian yang lebih terstruktur atau penggunaan alat kolaborasi yang lebih efektif. Selain itu, tim mungkin juga memutuskan untuk

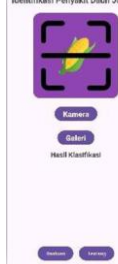
mengubah beberapa pendekatan dalam pengelolaan waktu atau distribusi tugas agar *sprint* berikutnya dapat berjalan lebih lancar.

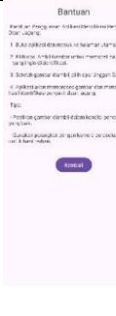
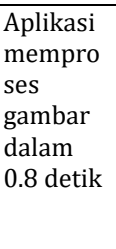
Tujuan utama dalam penelitian ini adalah fungsionalitas dan struktur prototipe. Evaluasi sistem dilakukan dengan pendekatan *black-box Testing* terhadap fungsionalitas aplikasi, serta uji kelayakan awal antarmuka oleh pengguna. Dengan pendekatan *Scrum*, proses pengembangan dapat dimonitor secara terstruktur dan menghasilkan prototipe yang dapat divalidasi secara bertahap.

4.8. *Blackbox Testing*

Pengujian ini berfokus pada pemeriksaan fungsionalitas aplikasi dengan memasukkan input tertentu dan memeriksa apakah output yang dihasilkan sesuai dengan yang diharapkan tanpa memperhatikan bagaimana sistem mengolah input tersebut (Fajar et al., 2024). Berikut adalah table pengujian *blackbox Testing*:

Tabel 8 Hasil *Blackbox Testing*

Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Keterangan
1. Masuk ke aplikasi dan menampilkan dasbor	Aplikasi berhasil menampilkan dasbor dengan semua menu yang tersedia		Fungsionalitas dasbor berjalan sesuai rencana
2. Mengambil gambar daun jagung menggunakan kamera	Kamera terbuka dan gambar berhasil diambil		Fungsionalitas kamera berjalan dengan baik

3. Mengambil dan memilih gambar daun jagung dari galeri	Gambar berhasil diambil dari galeri		Pengambilan gambar dari galeri berhasil
4. Menampilkan gambar dan hasil deteksi yang serta Menampilkan informasi rekomendasi pengobatan	Gambar yang diambil muncul dengan jelas di layar		Tampilan gambar sesuai harapan dan hasil deteksi sesuai dengan gambar input
5. Mengakses halaman bantuan	Halaman bantuan ditampilkan dengan informasi yang jelas		Informasi di halaman bantuan lengkap dan dapat dipahami
6. Mengakses halaman "Tentang Aplikasi"	Halaman "Tentang Aplikasi" menampilkan informasi aplikasi dengan benar		Informasi aplikasi ditampilkan dengan benar
7. Pengujian Kecepatan Identifikasi	Aplikasi dapat memproses gambar dan memberikan hasil		Kinerja aplikasi optimal dalam hal kecepatan

	dalam < 1 detik		
8. Pengujian Pengunaan Memori	Aplikasi menggunakan memori kurang dari 110 MB		Penggunaan memori sesuai target
9. Pengujian Fungsi Offline	Aplikasi tetap dapat digunakan meskipun tanpa koneksi internet		Aplikasi bekerja sesuai spesifikasi offline

Hasil pengujian menunjukkan bahwa *prototype* aplikasi identifikasi penyakit daun jagung berbasis android berhasil memenuhi semua skenario pengujian yang direncanakan yaitu dapat menampilkan dasbor dengan baik, memungkinkan pengguna memilih menu dengan benar, serta berhasil mengakses kamera dan galeri untuk mengambil gambar daun jagung. Gambar yang diambil ditampilkan dengan jelas dan menampilkan deteksi penyakit. Selain itu, halaman bantuan dan "Tentang Aplikasi" dapat diakses tanpa kendala, dan informasi penyakit ditampilkan dengan benar setelah proses deteksi. Secara keseluruhan, aplikasi berfungsi dengan optimal sesuai dengan spesifikasi dan harapan, memberikan pengalaman pengguna yang memuaskan.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan sebuah *prototype* aplikasi Android identifikasi penyakit pada daun jagung menggunakan metode *Scrum*. Melalui 4 *sprint* pengembangan pada *sprint backlog*, *prototype* berhasil memenuhi kebutuhan dasar pengguna, yakni kemudahan identifikasi penyakit secara cepat dan mandiri. Implementasi *Scrum* memungkinkan proses pengembangan berjalan

secara sistematis dan adaptif terhadap kebutuhan pengguna. Aplikasi menunjukkan kinerja yang baik dalam pengujian awal, dengan waktu proses yang singkat dan antarmuka yang dapat dipahami oleh pengguna. *Prototype* ini dapat menjadi dasar untuk pengembangan lebih lanjut menuju aplikasi yang siap pakai di lingkungan pertanian. Hasil ini menunjukkan bahwa pendekatan *Scrum* efektif digunakan untuk pengembangan aplikasi pertanian berbasis Android. *Scrum* memberikan struktur kerja yang fleksibel dan terukur, sehingga *prototype* dapat dikembangkan dan divalidasi secara bertahap. Keterlibatan pengguna sejak awal menjadi kunci keberhasilan desain antarmuka yang sesuai kebutuhan lapangan.

Penulis pun menyadari masih banyak kekurangan yang perlu dilakukan pengembangan lebih lanjut yaitu melakukan pengujian implementasi skala besar pada lingkungan secara langsung, menambahkan fitur algoritma klasifikasi, serta pengembangan integrasi dengan sistem rekomendasi pengendalian penyakit berbasis data di lapangan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada PT. Benih Citra Asia yang telah memberikan kesempatan dan dukungan selama proses pengembangan dan pengujian *prototype* aplikasi ini. Ucapan terima kasih juga disampaikan kepada seluruh rekan yang telah memberikan masukan dan motivasi selama proses penelitian berlangsung.

DAFTAR PUSTAKA

- [1] P. P. Produksi, J. D. I. Indonesia, N. S. Nguru, P. S. Agribisnis, F. Pertanian, and U. T. Tunggadewi, "Prospek peningkatan produksi jagung di indonesia," 2024.
- [2] I. P. Putra and D. Alamsyah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network," vol. 2, no. 2, pp. 102–112, 2022.
- [3] I. Wirabowo, I. Susilawati, T. Informasi, U. Mercu, and B. Yogyakarta, "Implementasi Convolution Neural Network (CNN) untuk Deteksi Penyakit pada Daun Jagung Berbasis Citra Digital," vol. 5, no. 1, pp. 233–241, 2025.
- [4] P. S. Rosiana, A. Voutama, and A. A. Ridha, "Perancangan Ui/Ux Sistem Informasi Pembelian Hasil Tani Berbasis Mobile Dengan Metode Design Thinking," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 246–253, 2023, doi: 10.23960/jitet.v11i3.3048.
- [5] Y. Ardilla *et al.*, "Pengembangan Aplikasi Mobile Dengan Menggunakan Scrum Model Untuk Deteksi Dini Stunting," vol. 18, no. x, pp. 623–636, 2023.
- [6] F. Aditya, A. D. Putra, and A. Surahman, "Rancang Bangun Aplikasi Penjualan Berbasis Android (Studi Kasus: Pada Toko Murah Jaya Alumunium)," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 3, no. 3, pp. 316–329, 2022, doi: 10.33365/jatika.v3i3.2037.
- [7] M. A. Setyawan, U. Nusantara, P. Guru, and R. Indonesia, "RUANG WARNA HSV DAN FITUR TEKSTUR DENGAN ALGORITMA K-NN," 2022.
- [8] R. Gutama, "Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP)," 2020.
- [9] K. Schwaber and J. Sutherland, "Panduan Scrum," *Diakses tanggal*, vol. 2, 2020.
- [10] B. K. Fajar, M. Lulu, L. Usman, P. Kidul, and K. Banyumas, "BERBASIS ANDROID MENGGUNAKAN METODE," vol. 12, no. 3, 2024.
- [11] D. Iswantoro and D. Handayani UN, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," *J. Ilm. Univ. Batanghari Jambi*, vol. 22, no. 2, p. 900, 2022, doi: 10.33087/jiubj.v22i2.2065.
- [12] D. Y. Sylfania, F. P. Juniawan, and I. Zaliman, "Implementasi Aplikasi Perpustakaan SMAN 1 Koba Dengan Metode Scrum Berbasis Android," vol. 8, no. 1, pp. 69–77, 2024.
- [13] R. A. Kusyanto, N. A. Prasetyo, P. Kidul, and K. Banyumas, "MITRA OBRE-JEK BERBASIS ANDROID," vol. 12, no. 3, pp. 1907–1918, 2024.
- [14] H. Prasetyo, N. Radita, and A. Tirtana, "Aplikasi Manajemen Pendakian Berbasis Android Menggunakan Metode Scrum Hiking Management Application Based on Android Using Scrum Method," vol. 11, no. November, pp. 180–189, 2022, doi: 10.34148/teknika.v11i3.514.