

Perancangan Sistem Informasi Deteksi Penyakit Daun Padi Menggunakan Metode Agile

Faruq Aziz¹, Daniati Uki Eka Saputri², Nurul Khasanah³, Taopik Hidayat⁴

^{1,2}Sistem Informasi, Universitas Nusa Mandiri, Jakarta Timur, Indonesia 17411

³Informatika, Universitas Nusa Mandiri, Jakarta Timur, Indonesia 17411

⁴Sains Data, Universitas Nusa Mandiri, Jakarta Timur, Indonesia 17411

e-mail: ¹faruq.fqs@nusamandiri.ac.id, ²daniati.due@nusamandiri.ac.id, ³nurul.nuk@nusamandiri.ac.id,
⁴taopik.tao@nusamandiri.ac.id

*Corresponding author

Submitted Date: April 18, 2025

Revised Date: May 29, 2025

Reviewed May 09, 2025

Accepted Date: June 20, 2025

Abstract

The development of a rice leaf disease detection information system using the Agile method aims to provide an innovative solution for fast and accurate plant disease identification. The system can detect four major rice leaf disease classes: bacterial blight, brownspot, blast, and healthy conditions. The development process follows an iterative approach, starting from understanding user needs to system implementation and testing. Black-box testing was applied to ensure that all features, such as image upload and disease classification, function according to specifications. Evaluation results indicate that the system achieves high accuracy in disease detection based on the utilized dataset. However, dataset limitations and testing scenarios pose challenges for generalizing results to real-field conditions. Hence, intensive evaluation and dataset updates are crucial for future development. With its user-friendly interface, the system is expected to support farmers in improving productivity and efficiency in rice disease detection.

Keywords: *rice leaf disease detection; Agile method; black-box testing*

Abstrak

Pengembangan sistem informasi deteksi penyakit daun padi berbasis metode Agile bertujuan untuk memberikan solusi inovatif dalam identifikasi penyakit tanaman secara cepat dan akurat. Sistem ini mampu mendeteksi empat kelas penyakit daun padi utama, yaitu *bacterial blight*, *brownspot*, *blast*, dan kondisi *healthy*. Proses pengembangan menggunakan pendekatan iteratif, dimulai dari pemahaman kebutuhan pengguna hingga implementasi dan pengujian sistem. *Black-box testing* diterapkan untuk memastikan setiap fitur, seperti unggah gambar dan klasifikasi penyakit, berfungsi sesuai spesifikasi. Hasil evaluasi menunjukkan bahwa sistem memiliki tingkat akurasi yang tinggi dalam mendeteksi penyakit berdasarkan dataset yang digunakan. Namun, keterbatasan dataset dan skenario uji menjadi tantangan dalam generalisasi hasil untuk kondisi lapangan. Oleh karena itu, evaluasi pengujian intensif dan pembaruan dataset menjadi langkah penting untuk pengembangan di masa mendatang. Dengan antarmuka yang ramah pengguna, sistem ini diharapkan mampu mendukung petani dalam meningkatkan produktivitas dan efisiensi deteksi penyakit tanaman padi.

Kata Kunci: *deteksi penyakit daun padi; metode Agile; black-box testing; sistem informasi; klasifikasi citra*



1. Pendahuluan

Padi (*Oryza sativa*) merupakan salah satu komoditas pangan utama di dunia, terutama di negara-negara Asia termasuk Indonesia (Ramayana, Sadarudin, Rusdiansyah, & Santoso, 2024). Sebagai makanan pokok bagi sebagian besar penduduk, keberlanjutan produksi padi memiliki peran yang sangat vital dalam menjamin ketahanan pangan nasional. Namun, produktivitas padi seringkali terancam oleh berbagai penyakit yang menyerang tanaman ini. Penyakit daun padi seperti *Bacterial Blight*, *Brownspot*, dan *Blast* merupakan tantangan utama yang dapat menyebabkan penurunan hasil panen secara signifikan jika tidak ditangani dengan tepat (Aziz, Ernawan, Fakhreldin, & Adi, 2023).

Penyakit *Bacterial Blight* disebabkan oleh bakteri *Xanthomonas oryzae pv. oryzae*. Patogen ini menyerang jaringan tanaman melalui luka atau stomata, menghasilkan gejala berupa bercak kekuningan yang meluas hingga mengakibatkan kerusakan daun secara menyeluruh. Kerusakan ini tidak hanya mempengaruhi kemampuan fotosintesis tanaman, tetapi juga dapat menyebabkan kerugian ekonomi yang besar bagi petani (Fadil, Yanti, & Khairul, 2023).

Penyakit *Brownspot*, yang disebabkan oleh jamur *Bipolaris oryzae*, ditandai dengan munculnya bercak coklat berbentuk bulat atau oval pada daun. Infeksi ini sering kali mengurangi kemampuan daun untuk melakukan fotosintesis secara optimal, sehingga berdampak langsung pada penurunan produktivitas tanaman. Penyakit ini biasanya menyerang tanaman yang mengalami kekurangan nutrisi atau kondisi lingkungan yang tidak ideal (Abadi et al., 2023).

Sementara itu, penyakit *Blast* yang disebabkan oleh jamur *Magnaporthe oryzae* merupakan salah satu ancaman paling serius pada padi. Penyakit ini menyebabkan gejala berupa lesi berbentuk belah ketupat pada daun yang dapat berkembang menjadi kerusakan yang lebih parah hingga mengakibatkan kematian daun (Rahman, Safni, & Lisnawita, 2023). Penyakit ini juga dapat menyerang bagian lain tanaman, seperti batang dan malai, sehingga menurunkan kualitas dan kuantitas hasil panen secara signifikan.

Dalam upaya mitigasi dampak dari penyakit-penyakit tersebut, deteksi dini menjadi langkah yang cukup penting. Pendekatan tradisional seperti inspeksi visual memerlukan keahlian tinggi dan

sering kali tidak efisien untuk diterapkan dalam skala luas. Keterbatasan ini mendorong perlunya pengembangan sistem informasi berbasis teknologi untuk deteksi penyakit daun padi yang dapat memberikan solusi lebih praktis, cepat, dan akurat.

Salah satu pendekatan yang dapat digunakan dalam pengembangan sistem informasi ini adalah metode Agile (Mishra & Alzoubi, 2023). Metode Agile dikenal sebagai pendekatan pengembangan perangkat lunak yang bersifat iteratif dan adaptif. Pendekatan ini menekankan pada kolaborasi intensif, iterasi pendek, dan respon cepat terhadap perubahan (Dudhat & Abbasi, 2021). Dengan menggunakan prinsip-prinsip Agile, pengembangan sistem informasi dapat dilakukan secara lebih efisien dan fleksibel sesuai dengan kebutuhan pengguna.

Penelitian terdahulu juga menunjukkan relevansi penerapan teknologi dalam mendukung sektor pertanian. Misalnya, penelitian oleh (Pramadhana, Amalia, & Ghazali, 2024) di mana mengembangkan sistem informasi berbasis *e-commerce* untuk pemasaran hasil pertanian di Kabupaten Indramayu. Penelitian ini menunjukkan bahwa sistem *e-commerce* mampu meningkatkan efisiensi pemasaran, mengurangi rantai distribusi, dan memberikan manfaat ekonomi yang lebih besar bagi petani. Hasil penelitian tersebut menggarisbawahi pentingnya penggunaan teknologi digital dalam mendukung pertanian berkelanjutan dan peningkatan kesejahteraan petani.

Penelitian lain yang relevan adalah studi oleh (Siagian, Dewi, & Lubis, 2024). Penelitian ini menjelaskan pengembangan *website* desa sebagai platform *online* yang menyediakan informasi dalam format teks dan gambar yang dapat diakses melalui internet. *Website* ini dirancang untuk memperkenalkan Desa Aeklung kepada publik di tingkat nasional maupun internasional. Fitur-fitur yang disediakan meliputi halaman utama, profil desa, proyek desa, struktur organisasi, berita, data penduduk, surat-menyurat, serta kolom komentar dan saran. Perancangan *website* dilakukan untuk menggantikan sistem manual yang masih digunakan di kantor administrasi desa, di mana penyampaian informasi dilakukan secara tatap muka.

Dengan menerapkan metode Agile, proses pengembangan *website* dilakukan secara iteratif dan responsif terhadap kebutuhan pengguna.



Penelitian ini menekankan bahwa implementasi *website* berbasis Agile mampu mempermudah masyarakat dalam mengakses informasi desa secara lebih cepat dan efisien, sehingga meningkatkan transparansi dan keterlibatan masyarakat.

Penelitian lainnya yang relevan adalah studi oleh (Syaputri, Putra, Syahrani, Dwian, & Purwani, 2024). Penelitian ini menyajikan tinjauan literatur sistematis mengenai penggunaan metode *Waterfall* dan Agile dalam pengembangan sistem informasi antara tahun 2019 hingga 2024. Melalui analisis terhadap berbagai literatur yang diperoleh menggunakan perangkat lunak *Publish or Perish*, penelitian ini mengidentifikasi bahwa metode *Waterfall* lebih sering digunakan pada proyek dengan kebutuhan yang stabil karena sifatnya yang terstruktur dan berurutan. Sebaliknya, metode Agile lebih disukai dalam lingkungan yang dinamis di mana fleksibilitas dan pengembangan iteratif menjadi kebutuhan utama. Penelitian ini memberikan wawasan yang berharga mengenai kelebihan dan kekurangan kedua metode tersebut, serta memberikan rekomendasi tentang penerapan masing-masing metode sesuai dengan kebutuhan proyek.

Berdasarkan latar belakang tersebut, penelitian ini difokuskan pada perancangan sistem informasi berbasis *website* untuk mendeteksi penyakit daun padi menggunakan metode Agile. Sistem dikembangkan dengan memanfaatkan teknologi pengolahan citra berbasis kecerdasan buatan, khususnya model deteksi YOLO yang dimodifikasi untuk mengklasifikasikan empat kategori utama: *Bacterial Blight*, *Brownspot*, *Blast*, dan *healthy* (Kappali et al., 2024).

Sistem informasi ini dikembangkan menggunakan teknologi modern seperti PHP versi 8.2, *framework CodeIgniter 3*, dan *Bootstrap 5*. PHP 8.2 dipilih karena kemampuannya yang andal dalam pengolahan data serta fitur-fitur terbaru yang meningkatkan performa aplikasi (Engebretth & Sahu, 2023). *CodeIgniter 3* digunakan sebagai *framework* karena kesederhanaannya dan efisiensi dalam pengembangan aplikasi berbasis *website* (Ahmed, Bello, Jauro, & Dawaki, 2024). Sementara itu, *Bootstrap 5* memberikan dukungan desain antarmuka yang responsif dan *user-friendly*, sehingga memudahkan pengguna dalam mengakses dan memanfaatkan sistem (Al Salmi, 2023).

Sistem ini memanfaatkan Kecerdasan Buatan atau *Artificial Intelligence* (AI) dengan model deteksi berbasis YOLO (*You Only Look Once*) yang telah ditingkatkan untuk mendeteksi penyakit daun padi (Aziz, Ernawan, et al., 2023). YOLO mampu mengklasifikasikan empat kategori utama (*bacterial blight*, *brownspot*, *blast*, dan *healthy*) dengan akurasi 94% dan waktu deteksi rata-rata 40 detik. Peningkatan model dilakukan melalui modifikasi *anchors*, *depth multiple*, dan *learning rate* pada jaringan *Convolutional Neural Network* (CNN).

Dengan demikian, tujuan penelitian ini adalah merancang dan membangun sistem informasi deteksi penyakit daun padi berbasis Agile yang bersifat responsif, efisien, serta mudah diakses oleh petani. Sistem ini diharapkan dapat membantu proses deteksi dini penyakit daun padi, mendukung produktivitas pertanian, dan pada akhirnya memperkuat ketahanan pangan nasional.

2. Metodologi Penelitian

a. Sistem Informasi sebagai Fondasi Teknologi Pertanian

Sistem informasi didefinisikan sebagai kombinasi antara teknologi, prosedur, dan manusia yang berfungsi untuk mengolah data menjadi informasi yang bernilai guna mendukung proses pengambilan keputusan (Riswandha et al., 2025). Dalam konteks modern, sistem informasi tidak hanya terbatas pada lingkungan bisnis, tetapi juga telah menjadi bagian integral dalam pengelolaan sektor pertanian.

Pemanfaatan sistem informasi di bidang pertanian dikenal dengan istilah *Agricultural Information System* (AIS), yang berfungsi untuk memfasilitasi pertukaran informasi antara petani, peneliti, dan instansi pertanian (Azizi, 2024). Melalui sistem ini, berbagai data seperti kondisi lahan, cuaca, dan kesehatan tanaman dapat diolah menjadi informasi yang membantu petani dalam pengambilan keputusan secara cepat dan berbasis data (*data-driven decision making*).

Tabel 1. Penelitian Terkait

No	Peneliti & Tahun	Judul / Fokus Penelitian	Metode / Teknologi	Hasil Utama	Keterbatasan	Relevansi dengan Penelitian Ini
1	Pramadhana, Amalia, & Ghozali (2024)	Pengembangan sistem informasi e-commerce hasil pertanian di Kabupaten Indramayu	Agile, Web-based System	Sistem e-commerce meningkatkan efisiensi distribusi hasil panen dan keuntungan petani	Tidak mencakup aspek analisis atau deteksi berbasis AI	Menunjukkan potensi integrasi teknologi web dalam pertanian
2	Siagian, Dewi, & Lubis (2024)	Pengembangan website desa berbasis Agile	Agile (Scrum)	Website adaptif terhadap kebutuhan pengguna dan meningkatkan transparansi	Fokus pada informasi administratif, bukan sistem analitik	Menguatkan efektivitas Agile dalam pengembangan sistem berbasis pengguna
3	Syaputri, Putra, Syahrani, Dwian, & Purwani (2024)	Kajian sistematis perbandingan metode Waterfall dan Agile (2019–2024)	Systematic Literature Review	Agile lebih fleksibel dan efisien pada proyek dengan kebutuhan dinamis	Tidak ada implementasi sistem nyata	Memberikan dasar metodologis pemilihan Agile untuk penelitian ini
4	Aziz, Ernawan, Fakhreldin, & Adi (2023)	Deteksi penyakit daun padi menggunakan model YOLO yang dimodifikasi	YOLO, CNN	Akurasi 94% dalam mendeteksi 4 kategori penyakit padi	Tidak memiliki integrasi sistem informasi berbasis web	Menjadi rujukan model deteksi citra yang digunakan dalam penelitian ini
5	Kappali et al. (2024)	Pengembangan model YOLOv5 untuk klasifikasi penyakit tanaman	YOLO V5, Deep Learning	Model YOLO menunjukkan akurasi dan kecepatan deteksi tinggi	Tidak fokus pada penerapan dalam sistem pengguna (user interface)	Menjadi acuan teknis pengembangan model AI pada penelitian ini

Sumber: Penelitian Mandiri, 2025

b. Teknologi Pengolahan Citra dan Kecerdasan Buatan (AI)

Kemajuan bidang AI dan pengolahan citra digital (*image processing*) memungkinkan komputer mengenali pola visual dalam citra daun tanaman untuk mendeteksi penyakit.

Teknik yang paling banyak digunakan adalah *Convolutional Neural Network* (CNN), yaitu arsitektur deep learning yang meniru cara kerja sistem penglihatan manusia dalam mengenali pola (Aziz, Ernawan, et al., 2023; Zhao et al., 2024). CNN secara otomatis mengekstraksi fitur visual dari citra seperti bentuk, tekstur, dan warna tanpa perlu rekayasa fitur manual.

c. Metode Agile dalam Pengembangan Sistem Informasi

Metode Agile merupakan pendekatan pengembangan perangkat lunak yang menekankan pada iterasi pendek, kolaborasi intensif, dan kemampuan adaptasi terhadap perubahan kebutuhan pengguna (Mishra & Alzoubi, 2023; Sarkar, Moharana, Rakhra, & Cheema, 2024).

Dalam praktiknya, Agile sering diimplementasikan melalui kerangka kerja seperti *Scrum* atau *Extreme Programming* (XP).

Karakteristik utama pendekatan ini (Dudhat & Abbasi, 2021) adalah:

1. Iteratif dan inkremental: sistem dikembangkan dalam sprint singkat dengan fokus pada fitur prioritas tinggi;
2. Kolaboratif: melibatkan komunikasi aktif antara pengembang dan pengguna;
3. Adaptif: memungkinkan perubahan kebutuhan tanpa mengganggu keseluruhan proses proyek.

Dalam metodologi pengembangan perangkat lunak, *black-box testing* digunakan untuk mengevaluasi apakah fungsi sistem telah bekerja sesuai dengan spesifikasi tanpa memeriksa kode internal (Barraood, Mohd, Baharom, & Almogahed, 2023). Pengujian ini memfokuskan pada hubungan input–output sistem dan cocok untuk memvalidasi sistem berbasis antarmuka pengguna.

Black-box testing sering digunakan dalam pendekatan Agile karena memungkinkan evaluasi cepat pada setiap iterasi fitur yang selesai. Dalam konteks sistem deteksi penyakit daun padi, pengujian dilakukan terhadap fungsi-fungsi utama seperti proses unggah gambar, klasifikasi hasil deteksi, dan penyajian hasil diagnosis.



d. Penelitian Terkait

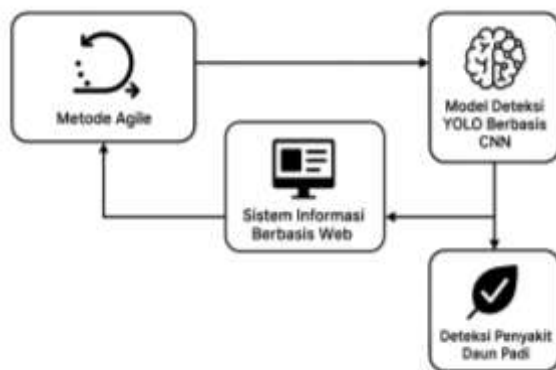
Penelitian terdahulu menunjukkan peningkatan signifikan dalam penerapan sistem informasi berbasis web dan metode Agile di berbagai bidang, termasuk pertanian dan layanan publik seperti terlihat pada Tabel 1. Penelitian terkait.

e. Kerangka Konseptual Penelitian

Kerangka konseptual penelitian ini seperti terlihat pada Gambar 1 menunjukkan integrasi antara metode Agile, sistem informasi berbasis *website*, dan model deteksi YOLO berbasis CNN. Metode Agile digunakan sebagai pendekatan iteratif yang memungkinkan penyesuaian sistem sesuai kebutuhan pengguna.

Sistem informasi *website* berfungsi sebagai platform untuk mengelola dan menampilkan hasil deteksi penyakit daun padi, sedangkan model YOLO bertugas mengidentifikasi empat kategori penyakit berdasarkan citra daun.

Hasil deteksi diintegrasikan ke dalam sistem dan dievaluasi secara berkelanjutan melalui siklus Agile, sehingga menghasilkan sistem yang adaptif, akurat, dan berorientasi pada pengguna.



Sumber: Penelitian Mandiri, 2025

Gambar 1. Kerangka Penelitian

f. Metodologi

Penelitian ini menggunakan metode Agile dalam proses pengembangan sistem. Beberapa tahap utama dalam Agile, yaitu *plan*, *design*, *develop*, *test*, *deploy*, *review*, dan *launch*. Pendekatan ini dipilih karena fleksibilitasnya dalam mengakomodasi perubahan kebutuhan dan kemampuannya untuk menghasilkan prototipe yang dapat diuji secara iteratif.



Sumber: (Rantung, 2024)

Gambar 2. Metode Agile

Tahap kesatu adalah *plan*, yang bertujuan untuk merencanakan kebutuhan sistem secara menyeluruh. Dalam tahap ini, dilakukan pengumpulan data dan analisis kebutuhan dari para pemangku kepentingan, seperti petani dan penyuluh pertanian. Fokus utama adalah mengidentifikasi fitur-fitur yang dibutuhkan, termasuk kemampuan deteksi empat kondisi utama penyakit daun padi (*Bacterial Blight*, *Brownspot*, *Blast*, dan *Healthy*). Teknologi yang digunakan, yaitu PHP 8.2, *CodeIgniter 3*, dan *Bootstrap 5*, juga ditentukan pada tahap ini. Selain itu, disusun *product backlog* yang berisi daftar fitur utama untuk memastikan pengembangan dilakukan secara terarah dan efisien.

Tahap kedua adalah *design*, yang berfokus pada perancangan *antarmuka* pengguna atau *User Interface* (UI) dan arsitektur sistem. Pada tahap ini, dibuat prototipe dalam bentuk *wireframe* dan *mockup* untuk memastikan antarmuka yang responsif dan mudah digunakan (Aziz, Saputri, Khasanah, & Hidayat, 2023). Selain itu, dirancang alur kerja sistem, seperti proses unggah gambar daun padi dan analisis otomatis menggunakan pengolahan citra. Prinsip desain modern diterapkan agar sistem tidak hanya fungsional tetapi juga estetis dan ramah pengguna.

Tahap ketiga adalah *develop*, di mana sistem mulai *dikembangkan* secara iteratif. Proses ini meliputi implementasi modul utama, seperti fitur deteksi penyakit berbasis pengolahan citra dengan Python (Quaranta, Calefato, & Lanubile, 2021) yang diintegrasikan ke *backend* *CodeIgniter*. Antarmuka yang responsif dikembangkan menggunakan *Bootstrap 5* untuk memastikan pengalaman pengguna yang optimal. Selain itu, sistem autentikasi dan otorisasi diterapkan untuk menjaga keamanan data. Selama tahap ini, praktik

terbaik dalam pengkodean diikuti untuk mengurangi risiko *bug* dan memastikan efisiensi sistem.

Tahap keempat *adalah test*, yang bertujuan untuk memastikan bahwa sistem berfungsi sesuai dengan kebutuhan pengguna. Pengujian dilakukan pada berbagai level menggunakan *black-box testing* (Barraood et al., 2023; Cani, Hannie, & Ridha, 2023), mulai dari pengujian unit untuk memastikan setiap modul bekerja dengan baik, hingga pengujian integrasi untuk memverifikasi komunikasi antar-modul. Pengujian sistem menyeluruh dilakukan menggunakan data uji, termasuk gambar daun padi dengan berbagai kondisi penyakit. Selain itu, pengujian antarmuka dilakukan untuk memastikan bahwa sistem mudah digunakan dan memenuhi kebutuhan pengguna. *Bug* yang ditemukan selama pengujian diperbaiki sebelum sistem dilanjutkan ke tahap berikutnya.

Tahap kelima adalah *deploy*, di mana sistem diunggah ke server agar dapat diakses oleh pengguna. Infrastruktur server disiapkan, termasuk konfigurasi *hosting* dan aktivitas log. Setelah kode aplikasi dan file pendukung diunggah, dilakukan konfigurasi domain agar sistem dapat diakses melalui internet. Pengujian akhir dilakukan di lingkungan server untuk memastikan semua fungsi bekerja dengan baik, termasuk kecepatan akses dan stabilitas sistem.

Tahap keenam adalah *review*, yang melibatkan evaluasi sistem berdasarkan umpan balik dari pengguna. Pada tahap ini, masukan dari petani dan penyuluh pertanian dikumpulkan untuk mengevaluasi fungsionalitas, kinerja, dan kemudahan penggunaan sistem. Umpan balik tersebut dianalisis untuk mengidentifikasi area yang memerlukan peningkatan. Revisi dan pengembangan tambahan dilakukan berdasarkan hasil evaluasi ini untuk memastikan sistem terus berkembang sesuai kebutuhan pengguna.

Tahap terakhir adalah *launch*, di mana sistem resmi dirilis untuk digunakan oleh masyarakat luas. Pada tahap ini, disediakan panduan pengguna dalam bentuk dokumen dan video tutorial untuk mempermudah adopsi sistem. Pelatihan diberikan kepada petani dan penyuluh pertanian agar mereka dapat memanfaatkan sistem secara maksimal. Selain itu, dilakukan promosi melalui media sosial dan kanal komunikasi lainnya untuk meningkatkan kesadaran dan jumlah pengguna. Pemantauan kinerja sistem dilakukan

secara berkala untuk memastikan stabilitas dan keberlanjutan sistem dalam jangka panjang.

Setiap tahap pengembangan dilakukan secara fleksibel untuk menghasilkan sistem informasi deteksi penyakit daun padi yang memenuhi kebutuhan pengguna dan memberikan dampak positif pada sektor pertanian.

3. Hasil Pembahasan

1. Plan (Perencanaan)

Pada tahap perencanaan, fokus utama adalah mengidentifikasi kebutuhan pengguna dan merancang strategi pengembangan yang terarah. Dimulai dari mengidentifikasi masalah yang dihadapi.

Tabel 2. Analisis Kebutuhan

Kategori	Deskripsi Kebutuhan	Detail Implementasi
Fungsional	Sistem mampu mendeteksi penyakit daun padi dari gambar yang diunggah oleh pengguna.	<ul style="list-style-type: none">- Implementasi model AI untuk klasifikasi penyakit.- Mendukung format gambar populer seperti JPEG dan PNG.- Menampilkan langkah-langkah pengelolaan.
	Sistem memberikan rekomendasi solusi penanganan berdasarkan hasil deteksi penyakit.	<ul style="list-style-type: none">- Memberikan referensi sumber terpercaya untuk edukasi lebih lanjut.- Menggunakan Bootstrap untuk desain responsif.
	Sistem menyediakan antarmuka yang mudah digunakan dan responsif di berbagai perangkat.	<ul style="list-style-type: none">- Tampilan sederhana untuk akses pengguna tanpa pelatihan.- Waktu respons maksimal: 5 detik per gambar.
Non-Fungsional	Sistem memiliki waktu respons cepat dalam melakukan deteksi penyakit.	<ul style="list-style-type: none">- Optimasi algoritma AI dan server hosting.
	Sistem mampu menangani unggahan banyak pengguna secara bersamaan.	<ul style="list-style-type: none">- Infrastruktur cloud hosting.- Penggunaan caching dan load balancing.
	Sistem aman dari serangan, seperti unggahan file berbahaya.	<ul style="list-style-type: none">- Validasi unggahan file.- Protokol HTTPS untuk transmisi data terenkripsi.

Analisis ini dapat dilihat seperti pada Tabel 1 dimana kebutuhan fungsional mencakup kemampuan sistem untuk mendeteksi penyakit dari gambar dan memberikan rekomendasi penanganan yang relevan. Kebutuhan non-fungsional memastikan performa, keamanan, dan skalabilitas



sistem, termasuk kecepatan respons dan kemampuan menangani banyak pengguna secara bersamaan.

Tabel 3. Perencanaan

Komponen	Deskripsi	Detail
Tujuan	Mengembangkan sistem deteksi penyakit daun padi berbasis gambar.	- Mengurangi waktu identifikasi. - Meningkatkan produktivitas petani.
Target Pengguna	Petani, penyuluh pertanian, dan komunitas agrikultur.	Fokus pada pengguna yang tidak memiliki latar belakang teknologi tinggi.
Fitur Utama	1. Deteksi penyakit. 2. Rekomendasi perawatan. 3. Responsivitas antarmuka.	Menggunakan teknologi AI untuk akurasi tinggi.
Teknis	Sistem menggunakan teknologi terkini yang kompatibel dengan standar industri.	- PHP 8.2 untuk <i>backend</i> . - <i>Logging</i> untuk manajemen data. - Dataset menggunakan publik dataset dari Paddy Disease (Kappali et al., 2024) dengan 4 kelas (<i>bacterial blight</i> , <i>healthy</i> , <i>brownspot</i> , <i>blast</i>)
Data dan Algoritma	Sistem memerlukan dataset gambar daun padi yang mencakup berbagai jenis penyakit untuk pelatihan model AI.	- Algoritma AI yang digunakan bersumber dari penelitian penulis sebelumnya (Aziz, Ernawan, et al., 2023)
Operasional	Sistem harus mudah diakses oleh petani dan penyuluh pertanian di lokasi pedesaan.	- Optimalisasi untuk koneksi internet lambat. - Antarmuka sederhana untuk pengguna awam.
Legal dan Etika	Sistem harus mematuhi aturan terkait privasi dan penggunaan data.	- Menampilkan kebijakan privasi. - Tidak menyimpan data gambar tanpa izin pengguna.

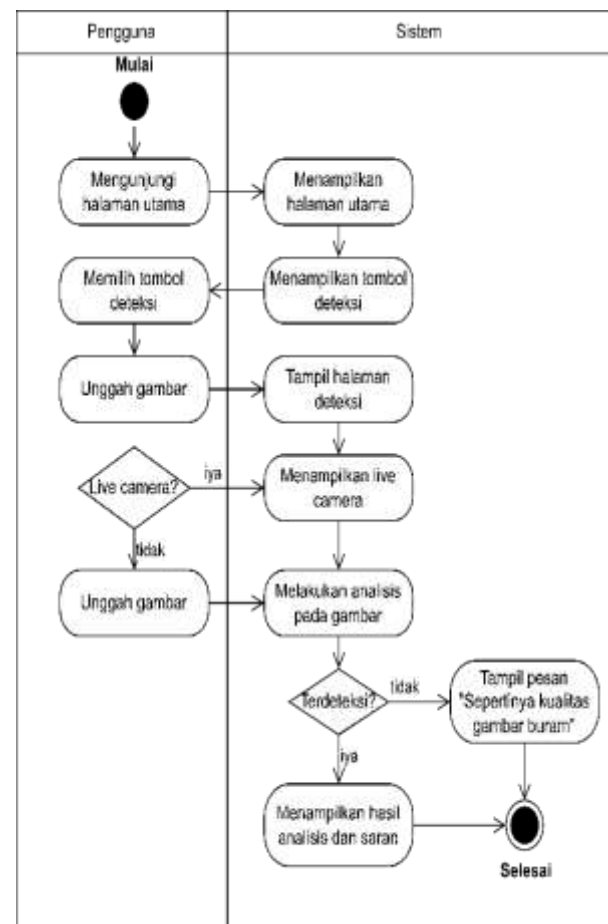
Kemudian setelah analisis kebutuhan selesai ditentukan, berikutnya adalah melakukan pendefinisian sistem dan fitur-fitur seperti terlihat pada Tabel 2 terkait perencanaan lebih lanjut seperti menegaskan pemilihan teknologi terkini yang digunakan untuk memastikan kompatibilitas dan efisiensi. Sistem juga dirancang agar mudah diakses oleh petani di lokasi pedesaan dengan koneksi internet yang terbatas. Selain itu, dataset yang representatif dan pemenuhan aspek legal terkait privasi pengguna menjadi prioritas untuk memastikan keandalan dan keberlanjutan sistem.

2. Design (Desain)

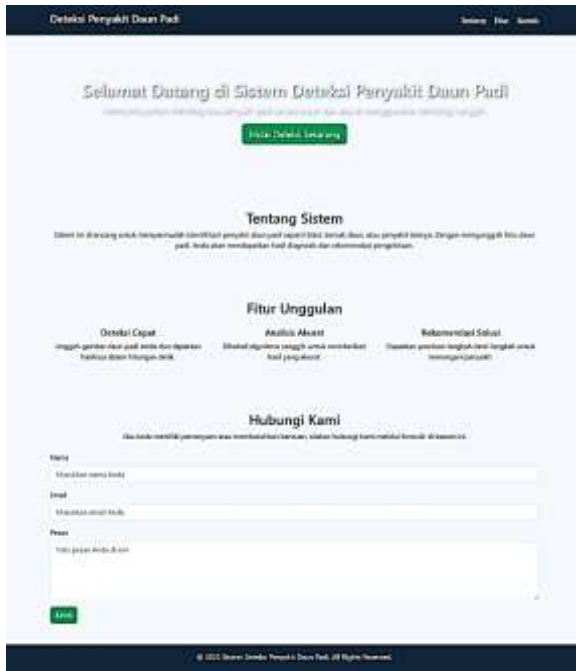
Tahap desain bertujuan untuk merancang struktur dan antarmuka sistem yang efektif, efisien,

dan ramah pengguna. Desain ini mencakup dua komponen utama, yaitu UML diagram aktivitas (Nurrahman, Asep, & Aziz, 2020) dan desain arsitektur sistem. Alur dari bagaimana penggunaan sistem oleh pengguna dapat dilihat seperti pada Gambar 2.

Desain antarmuka pengguna difokuskan pada kemudahan akses dan pengalaman yang ramah bagi pengguna awam, seperti petani. Tata letak antarmuka dirancang responsif menggunakan sistem *grid* dari *Bootstrap*, sehingga tampilannya optimal diberbagai ukuran layar. Navigasi antarmuka dibuat sesederhana mungkin, dengan menu utama yang mencakup fitur deteksi, informasi penyakit, dan kontak bantuan. Skema warna hijau dipilih sebagai tema utama, mencerminkan hubungan erat dengan dunia agrikultur. Selain itu, elemen visual seperti ikon dan gambar digunakan untuk memberikan kejelasan pada fungsi setiap bagian antarmuka. Rancangan antarmuka dapat dilihat seperti pada Gambar 3.



Gambar 3. Aktivitas Diagram



Gambar 4. Antarmuka Halaman Utama



Gambar 5. Antarmuka Halaman Deteksi

Gambar 4 menampilkan halaman antarmuka utama sistem untuk mendeteksi penyakit daun padi. Halaman ini dirancang dengan pendekatan antarmuka yang sederhana dan responsif, sehingga memudahkan pengguna, termasuk petani, untuk mengoperasikannya. Pada halaman ini, pengguna dapat mengunggah gambar daun padi melalui fitur unggah file yang tersedia. Setelah gambar diunggah, sistem secara otomatis memproses gambar tersebut dan menampilkan hasil deteksi dalam bentuk klasifikasi penyakit daun padi, yaitu *bacterial blight*, *brownspot*, *blast*, atau *healthy*.

3. Develop (Pengembangan)

Tahap pengembangan dimulai setelah rancangan desain, termasuk antarmuka pengguna (UI) dan diagram UML aktivitas, telah selesai dibuat. Pada tahap ini, fokus utama adalah mengimplementasikan sistem berdasarkan spesifikasi yang telah dirancang.

Proses pengembangan dimulai dari membangun komponen *backend* menggunakan PHP 8.2, yang dipilih karena kemampuannya

dalam menangani proses *server-side* secara cepat dan efisien. *Backend* bertugas untuk memproses data yang diterima dari antarmuka pengguna, seperti file gambar yang diunggah, dan menghubungkannya dengan model AI. AI ini menggunakan model deteksi yang telah dilatih untuk mengenali berbagai jenis penyakit daun padi (Aziz, Ernawan, et al., 2023), sehingga dapat memberikan hasil deteksi secara akurat.

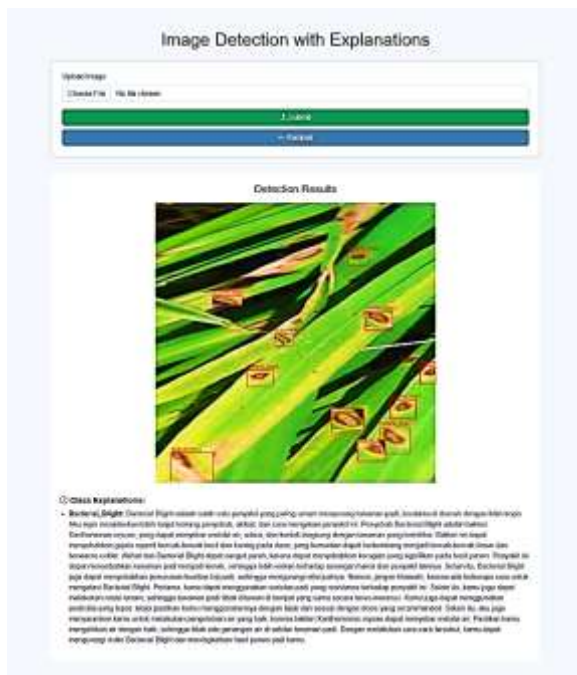
Pada sisi *frontend*, *framework* Bootstrap digunakan untuk memastikan desain yang responsif sesuai dengan prototipe UI yang telah dirancang sebelumnya. Elemen antarmuka seperti formulir unggahan gambar, halaman hasil analisis, dan navigasi utama diimplementasikan sesuai rancangan UI. Penyesuaian juga dilakukan agar tampilan tetap intuitif bagi pengguna awam, seperti petani seperti terlihat pada Gambar 5.

Sistem pengembangan dilakukan secara iteratif berdasarkan konsep Metode Agile, yang memberikan panduan alur proses dari unggahan gambar hingga penyajian hasil deteksi. Setiap fungsi diuji pada tahap pengembangan untuk memastikan bahwa sistem berjalan sesuai dengan diagram aktivitas yang dirancang. Seperti alur validasi file unggahan, pengiriman data ke AI, dan pengambilan hasil analisis diuji secara bertahap untuk memastikan integritas sistem.



Gambar 6. Tampilan Awal

Selain itu, diterapkan mekanisme *error handling* untuk menangani potensi kesalahan, seperti unggahan file yang tidak sesuai format atau kegagalan komunikasi. *Log error* disimpan untuk memudahkan proses debugging dan memastikan sistem tetap andal. Hasil dari tahap pengembangan ini adalah sebuah sistem yang siap diuji lebih lanjut pada tahap pengujian, dengan komponen utama yang telah diimplementasikan sesuai dengan spesifikasi desain. Hasil analisis sistem dapat terlihat seperti pada Gambar 6.



Gambar 7. Tampilan Hasil Analisis

4. Test (Pengujian)

Pengujian sistem *website* pada proyek deteksi penyakit daun padi dilakukan menggunakan pendekatan *black-box testing*, di mana pengujian difokuskan pada fungsionalitas sistem tanpa memperhatikan struktur internal atau kode program yang ada di balik sistem tersebut.

Tabel 4. *Black-box Testing* Deteksi

No	Skenario Pengujian	Input	Ekspektasi Output	Keterangan Hasil
1	Pengujian dengan gambar daun padi sehat	Gambar daun padi sehat	Klasifikasi: healthy	Gambar daun sehat seharusnya terdeteksi dengan benar sebagai healthy.

2	Pengujian dengan gambar daun padi terinfeksi bakterial	Gambar daun padi bakterial blight	Klasifikasi: bacterial_blight	Gambar daun dengan infeksi bakterial harus terdeteksi sebagai bacterial blight.
3	Pengujian dengan gambar daun padi terinfeksi bercak coklat	Gambar daun padi brownspot	Klasifikasi: brownspot	Gambar daun yang terinfeksi bercak coklat harus terdeteksi sebagai brownspot.
4	Pengujian dengan gambar daun padi terinfeksi blast	Gambar daun padi blast	Klasifikasi: blast	Gambar daun dengan infeksi blast harus terdeteksi sebagai blast.
5	Pengujian dengan gambar daun padi yang rusak parah	Gambar daun rusak (kombinasi penyakit)	Klasifikasi salah satu jenis penyakit yang dominan (misal: bacterial_blight)	Menguji ketahanan sistem terhadap gambar daun rusak parah dengan lebih dari satu infeksi.
6	Pengujian dengan gambar daun yang tidak terdeteksi (false positive)	Gambar daun padi yang sehat tetapi salah terdeteksi	Klasifikasi: healthy atau salah klasifikasi	Menguji untuk memastikan tidak ada false positive dalam klasifikasi.
7	Pengujian dengan gambar daun blur atau buram	Gambar daun padi blur atau buram	Klasifikasi sesuai dengan gejala yang tampak, meskipun kualitas gambar rendah	Menguji sensitivitas sistem terhadap gambar dengan kualitas rendah.
8	Pengujian dengan gambar daun padi yang sangat besar (zoom in)	Gambar daun padi zoom-in	Klasifikasi sesuai dengan jenis penyakit yang terdeteksi pada gambar	Menguji apakah sistem dapat mengenali penyakit dengan gambar yang terzoom.
9	Pengujian dengan gambar daun padi yang sangat kecil (zoom out)	Gambar daun padi zoom-out	Klasifikasi sesuai dengan jenis penyakit yang terdeteksi pada gambar	Menguji apakah sistem dapat mengenali penyakit pada gambar yang terlalu kecil.



Pengujian ini bertujuan untuk memastikan bahwa *website* dapat berfungsi sesuai dengan spesifikasi dan memenuhi kebutuhan pengguna. Setiap fitur, seperti unggah gambar daun padi, proses deteksi penyakit, dan hasil keluaran (misalnya klasifikasi jenis penyakit), diuji dengan berbagai skenario untuk memastikan bahwa input yang diberikan menghasilkan output yang sesuai. Tim penguji akan memberikan berbagai jenis data input yang mencakup berbagai variasi gambar daun padi, baik yang sehat maupun yang terinfeksi, untuk mengevaluasi apakah sistem dapat mengenali dan mengklasifikasikan penyakit dengan benar.

Hasil dari pengujian ini kemudian dianalisis untuk mengetahui apakah *website* berfungsi dengan baik tanpa adanya bug atau kesalahan dalam operasionalnya. Pendekatan *black-box testing* memastikan bahwa pengalaman pengguna pada *website* berjalan lancar dan sesuai harapan tanpa perlu memeriksa kode yang ada di belakangnya seperti ditunjukkan pada Tabel 3.

5. Deploy (Penerapan)

Pada tahap *deploy*, sistem yang telah selesai dikembangkan dan diuji akan dipindahkan ke lingkungan produksi. Proses ini melibatkan instalasi *website* deteksi penyakit daun padi di server yang dapat diakses oleh pengguna akhir. Setelah *deployment*, dilakukan konfigurasi server dan pengaturan domain agar *website* dapat berjalan dengan baik di platform yang telah dipilih.

6. Review (Tinjauan)

Selanjutnya, tahap *review* dilakukan untuk menilai keseluruhan kinerja sistem sebelum diluncurkan secara resmi. Tim pengembang dan *stakeholder* mengevaluasi hasil implementasi dan memastikan bahwa semua fitur berjalan sesuai dengan kebutuhan pengguna. Pada tahap ini, umpan balik dari pengujian pengguna (*user testing*) dan hasil pengujian fungsionalitas (seperti *black-box testing*) digunakan untuk memperbaiki kekurangan atau *bugs* yang mungkin masih ada. Jika ditemukan masalah, dilakukan perbaikan dan pengujian ulang.

7. Launch (Peluncuran)

Setelah semua tahapan *review* selesai dan sistem dinyatakan siap, tahap *launch* dilakukan. *Website* deteksi penyakit daun padi resmi

diluncurkan dan dapat diakses oleh pengguna. Pada tahap ini, promosi dan informasi mengenai *website* dapat disebarluaskan kepada target audiens, seperti petani atau ahli pertanian. Tim pengembang akan terus memantau kinerja *website* setelah peluncuran, untuk memastikan tidak ada masalah teknis dan memberikan dukungan jika diperlukan.

4. Kesimpulan

Berdasarkan hasil perancangan sistem informasi deteksi penyakit daun padi menggunakan metode Agile, dapat disimpulkan bahwa pendekatan ini terbukti sangat efektif dalam menghasilkan sistem yang sesuai dengan kebutuhan pengguna. Sistem berhasil mengimplementasikan deteksi penyakit daun padi pada empat kelas utama (*Bacterial Blight*, *Brownspot*, *Blast*, dan *Healthy*) dengan dukungan pengujian intensif melalui *black-box testing*.

Penerapan metode Agile dengan siklus iteratif memungkinkan tim pengembang beradaptasi secara cepat terhadap perubahan serta masukan dari pengguna. Hal ini tidak hanya meningkatkan fleksibilitas proses pengembangan, tetapi juga memastikan bahwa sistem yang dihasilkan lebih optimal, relevan, dan mudah digunakan.

Dengan demikian, penelitian ini menegaskan bahwa metode Agile berperan penting dalam kesuksesan perancangan sistem informasi deteksi penyakit daun padi, karena mampu menggabungkan aspek teknis (akurasi deteksi berbasis AI) dengan kebutuhan praktis di lapangan. Sistem yang dihasilkan diharapkan dapat membantu petani melakukan deteksi penyakit secara lebih cepat, akurat, dan aplikatif, sehingga berkontribusi pada peningkatan produktivitas pertanian dan ketahanan pangan nasional.

Referensi

- Abadi, A. L., Syib'li, M. A., Aini, L. Q., Sektiono, A. W., Choliq, F. A., & Trianti, I. (2023). *Pengelolaan Penyakit Tumbuhan Terpadu (Integrated Plant Disease Management)*. Universitas Brawijaya Press.
- Ahmed, M. K., Bello, A. H., Jauro, S. S., & Dawaki, M. (2024). A comparative analysis of performance optimization techniques for benchmarking PHP frameworks: Laravel and Codeigniter. *Dutse Journal of Pure and*



- Applied Sciences*, 10(3c), 284–295.
<https://doi.org/10.4314/dujopas.v10i3c.27>
- Al Salmi, H. (2023). Comparative CSS frameworks. *Multi-Knowledge Electronic Comprehensive Journal for Education and Science Publications (MECSJ)*, 66. Retrieved from https://mecsjs.com/uplode/images/photo/hat4_.pdf
- Aziz, F., Ernawan, F., Fakhreldin, M., & Adi, P. W. (2023). YOLO Network-Based for Detection of Rice Leaf Disease. *2023 International Conference on Information Technology Research and Innovation (ICITRI)*, 65–69.
<https://doi.org/10.1109/ICITRI59340.2023.10249843>
- Aziz, F., Saputri, D. U. E., Khasanah, N., & Hidayat, T. (2023). Penerapan UI/UX dengan Metode Design Thinking (Studi Kasus: Warung Makan). *Jurnal Infortech*, 5(1), 1–8.
- Azizi, J. (2024). Application of Artificial Intelligence (AI) In-Farm. *International Journal of Artificial Intelligence and Machine Learning*, 4(2), 83–91.
<https://doi.org/10.51483/IJAIML.4.2.2024.83-91>
- Barraood, S. O., Mohd, H., Baharom, F., & Almogahed, A. (2023). Verifying Agile Black-Box Test Case Quality Measurements: Expert Review. *IEEE Access*, 11, 106987–107003.
<https://doi.org/10.1109/ACCESS.2023.3320576>
- Cani, Y. M., Hannie, H., & Ridha, A. A. (2023). Pengujian Black Box Testing Pada Sistem Pendukung Keputusan Penerima Beasiswa di SMK Tarbiyatul Ulum Karawang. *Jurnal Ilmiah Wahana Pendidikan*, 9(9), 754–760.
<https://doi.org/10.5281/zenodo.8084698>
- Dudhat, A., & Abbasi, M. A. (2021). Discussion of Agile Software Development Methodology and its Relevance to Software Engineering. *ADI Journal on Recent Innovation*, 3(1), 105–114.
<https://doi.org/10.34306/ajri.v3i1.536>
- Engbreth, G., & Sahu, S. K. (2023). *PHP 8 Basics: For Programming and Web Development*. Berkeley, CA: Apress.
<https://doi.org/10.1007/978-1-4842-8082-9>
- Fadil, M., Yanti, Y., & Khairul, U. (2023). Penapisan aktinobakteria rhizosfer padi sebagai agens pengendali hayati *Xanthomonas oryzae* pv. *Oryzae* pathogen penyebab penyakit hawar daun bakteri. *Jurnal AGRO*, 10(1), 1–15.
<https://doi.org/10.15575/19798>
- Kappali, H., TV, S., G, M., K, P., R, S. K., & M, S. K. (2024). A Squeeze Net Based Paddy Disease Classification. *2024 5th International Conference on Circuits, Control, Communication and Computing (I4C)*, 34–37.
<https://doi.org/10.1109/I4C62240.2024.10748530>
- Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*, 14(4), 1504–1522. <https://doi.org/10.1007/s13198-023-01958-5>
- Nurrahman, D., Asep, A., & Aziz, F. (2020). Rancang Bangun Sistem Informasi Akuntansi Penjualan Kredit Pada Pd. Lajuar Motor Karawang. *Community Development Journal: Jurnal Pengabdian Masyarakat*, 1(2), 41–47.
- Pramadhana, D., Amalia, D. R., & Ghozali, A. L. (2024). SMART E-COMMERCE UNTUK STRATEGI PEMASARAN GUNA PENINGKATAN PENJUALAN HASIL PERTANIAN (SiPeHtanI). *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 10(2), 40–47.
<https://doi.org/10.24014/rmsi.v10i2.31937>
- Quaranta, L., Calefato, F., & Lanubile, F. (2021). KGTorrent: A Dataset of Python Jupyter Notebooks from Kaggle. *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 550–554.
<https://doi.org/10.1109/MSR52588.2021.00072>
- Rahman, F. A., Safni, I., & Lisnawita, L. (2023). Kelimpahan Jamur Non-Patogenik Pada Rhizosfer Daerah Endemik Patogen *Magnaporthe grisea* Penyebab Penyakit Blas Pada Tanaman Padi (*Oryza sativa* L.). *Agro Bali : Agricultural Journal*, 6(2), 395–404.
<https://doi.org/10.37637/ab.v6i2.1145>



- Ramayana, A. S., Sadarudin, Rusdiansyah, & Santoso. (2024). *Buku Referensi Padi Ladang Spesifik Wilayah Tropika Basah*. Penerbit NEM.
- Rantung, V. P. (2024). *AGILE SOFTWARE DEVELOPMENT Panduan Komprehensif untuk Tim Pengembang*. Jakarta: Lakeisha.
- Riswandha, M. N., Khairati, F., Husni, A. F., Dengen, A., Impran, I. A., CIISA, F. W., S. Kom, M. M., ... M.Kom, C., S. Kom. (2025). *DASAR SISTEM INFORMASI MODERN*. CV Rey Media Grafika.
- Sarkar, T., Moharana, B., Rakhra, M., & Cheema, G. S. (2024). Comparative Analysis of Empirical Research on Agile Software Development Approaches. *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1–6. <https://doi.org/10.1109/ICRITO61523.2024.10522134>
- Siagian, A. L., Dewi, A. R., & Lubis, F. R. (2024). Implementasi Metode Agile Untuk Perancangan Website Di Desa Aeklung Doloksanggul. *Innovative: Journal Of Social Science Research*, 4(5), 3939–3948. <https://doi.org/10.31004/innovative.v4i5.15244>
- Syaputri, L., Putra, E. G., Syahrani, E., Dwian, E., & Purwani, F. (2024). PERBANDINGAN EFEKTIVITAS METODE WATERFALL DAN AGILE DALAM PENGEMBANGAN SISTEM INFORMASI SEBUAH SYSTEMATIC LITERATURE REVIEW. *Journal of Scientech Research and Development*, 6(2), 262–273. <https://doi.org/10.56670/jsrd.v6i2.585>
- Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57(4), 99. <https://doi.org/10.1007/s10462-024-10721-6>

