

## **Pengembangan Aplikasi *Mobile* Pendeteksi Penyakit Daun Tanaman Jagung dengan Metode *Mobile Application Development Life Cycle* (MADLC)**

**Tri Sandhika Jaya<sup>1</sup>, Panji Andhika Pratomo<sup>2</sup>, Fathurrahman Kurniwan Ikhsan<sup>3</sup>**

<sup>1,2</sup> Program Studi Manajemen Informatika, Politeknik Negeri Lampung

<sup>3</sup> Program Studi Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Lampung

e-mail: [sandi@polinela.ac.id](mailto:sandi@polinela.ac.id), [panjiandhikap@polinela.ac.id](mailto:panjiandhikap@polinela.ac.id), [fathurrahman@polinela.ac.id](mailto:fathurrahman@polinela.ac.id)

### **Abstrak**

*Meningkatnya pertumbuhan populasi global dan permintaan pangan telah membuat peningkatan produksi tanaman pangan seperti jagung menjadi prioritas utama. Namun, tantangan dalam mempertahankan produktivitas tanaman tersebut adalah karena penyakit daun yang dapat mempengaruhi pertumbuhan dan hasil. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan aplikasi mobile yang dapat mendeteksi penyakit daun jagung secara efektif dengan menggunakan pendekatan mobile application development life cycle (MADLC). Metodologi MADLC berfungsi sebagai panduan dalam pengembangan aplikasi seluler ini. Langkah-langkah MADLC meliputi perencanaan, analisis kebutuhan, desain, pengembangan, pengujian, dan implementasi. Aplikasi yang dihasilkan dirancang untuk berjalan di platform seluler, memungkinkan petani dan profesional pertanian dengan mudah mendeteksi penyakit daun jagung dari perangkat seluler mereka. Pembangunan aplikasi ini bertujuan untuk menggali informasi tentang berbagai jenis penyakit daun yang umumnya menyerang tanaman jagung. Selain itu, kemampuan deteksi penyakit dirancang dan diimplementasikan dengan menggunakan teknik pengolahan citra dan kecerdasan buatan. Hasil pengujian menunjukkan bahwa aplikasi dapat secara akurat mendeteksi beberapa jenis penyakit daun pada jagung dari foto yang diambil dengan kamera ponsel. Penelitian ini berkontribusi pada pertanian dan teknologi dengan memasukkan prinsip-prinsip MADLC ke dalam pengembangan aplikasi seluler yang mendukung deteksi dini penyakit tanaman. Aplikasi ini diharapkan dapat memungkinkan petani dan pelaku pertanian untuk mendeteksi penyakit pada tanaman jagung dengan cepat dan akurat, memungkinkan tindakan pengendalian penyakit secara dini dan meningkatkan produktivitas pertanian.*

### **Abstract**

*Increasing global population growth and food demand have made increasing the production of food crops such as maize a top priority. However, the challenge in maintaining the productivity of these plants is due to foliar diseases which can affect growth and yield. Hence, this research aims to create a mobile app capable of efficiently identifying corn leaf diseases by following the mobile application development life cycle (MADLC) approach. The MADLC methodology serves as a guide in the development of this mobile application. MADLC steps include planning, requirements analysis, design, development, testing, and implementation. The resulting application is designed to run on mobile platforms, enabling farmers and agricultural professionals to easily detect corn leaf disease from their mobile devices. The development of this application aims to gather information about various types of foliar diseases that commonly attack corn plants. In addition, disease detection capabilities are designed and implemented using image processing and artificial intelligence techniques. The test results show that the application can accurately detect several types of leaf diseases in corn from photos taken with a cellphone camera. This research contributes to agriculture and technology by incorporating MADLC principles into the development of mobile applications that support early detection of plant diseases. This application is expected to enable farmers and agricultural actors to detect diseases in corn plants quickly and accurately, enabling early disease control measures and increasing agricultural productivity.*

**Keywords:** *mobile applications, leaf disease detection, maize, image processing, artificial intelligence*

## 1. PENDAHULUAN

Proses siklus hidup pengembangan perangkat lunak adalah tipikal struktur atau kerangka kerja yang digunakan dalam pengembangan produk perangkat lunak[1]. Berbagai model siklus hidup telah didefinisikan. Beberapa model tersebut adalah model air terjun, model spiral, model prototipe. Setiap model dijelaskan oleh satu set fungsi[2]. Tahapan atau aktivitas pengembangan mungkin berbeda untuk setiap model, tetapi semua model mencakup desain, persyaratan, analisis, desain, dll. Dalam model air terjun, lebih ditekankan pada proses langkah demi langkah. Model spiral menekankan penilaian risiko, sedangkan model prototipe menekankan pendekatan inkremental pada setiap tahap proses pengembangan.

Merancang produk sesuai dengan kebutuhan pengguna untuk berbagai siklus pengembangan standar merupakan bagian penting dari proses pengembangan[3]. Proses pengembangan tertentu adalah penting. Siklus Hidup Perkembangan Siklus hidup perkembangan dapat dipengaruhi jika perkembangan terjadi secara bertahap. Jika pelanggan terlibat dalam proses pengembangan, persyaratan pelanggan lebih baik dipenuhi pada akhir proses pengembangan, yang mengarah pada kepuasan pelanggan yang lebih baik[4].

Aplikasi seluler merujuk pada perangkat lunak yang dibuat khusus untuk digunakan pada perangkat kecil, seperti ponsel pintar dan tablet, serta komputasi nirkabel, bukan pada komputer desktop atau laptop[5]. Laporan Statistika baru-baru ini menunjukkan bahwa pada tahun 2017 smartphone memiliki pangsa 77% secara global, dan lebih dari 32% populasi global menggunakan smartphone[6]. Perangkat seluler dilengkapi dengan daya komputasi, yang mengarah pada pergeseran dari komputer desktop.

Penggunaan perangkat mobile sudah mencakup banyak bidang kehidupan manusia salah satunya pertanian seperti mendeteksi penyakit tanaman melalui citra daun. Penyakit tanaman merupakan masalah serius di negara agraris seperti Indonesia. Setiap tahun, produksi tanaman mengalami kerugian besar akibat serangan penyakit[7]. Sangat sulit bagi mata manusia untuk mendeteksi penyakit tanaman. Oleh karena itu, membangun sistem deteksi penyakit otomatis sangatlah penting[8].

Pada penelitian ini bertujuan untuk merancang aplikasi pendeteksi penyakit daun tanaman menggunakan pendekatan metode Mobile Application Development Life Cycle (MADLC) yang merupakan metode pengembangan perangkat lunak khusus untuk aplikasi mobile.

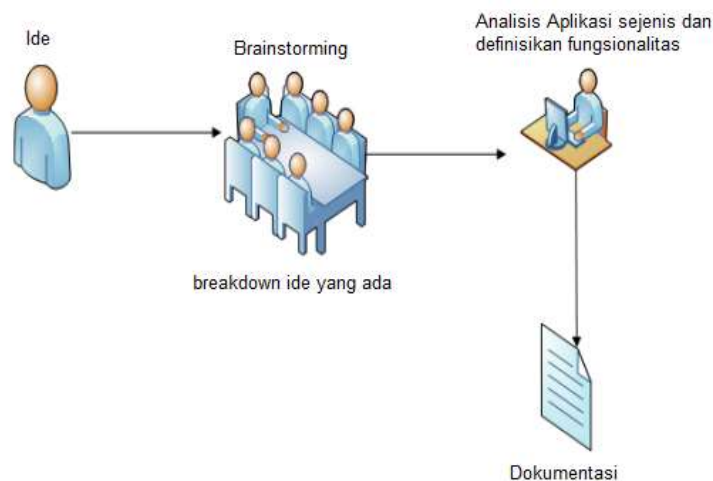
## 2. METODOLOGI PENELITIAN

Aplikasi seluler memiliki fitur yang kompleks tidak seperti aplikasi desktop. *Mobile Application Development Life Cycle* (MADLC) berikut disediakan untuk memungkinkan pendekatan sistematis untuk pengembangan sistem.

### 2.1. Identification Phase

Pada tahap awal, ide-ide dikumpulkan dan dikelompokkan. Fokus utama pada fase ini adalah menghasilkan ide-ide baru atau memperbaiki aplikasi yang sudah ada [9]. Ide-ide ini dapat berasal dari pelanggan atau pengembang. Jika pelanggan sendiri yang

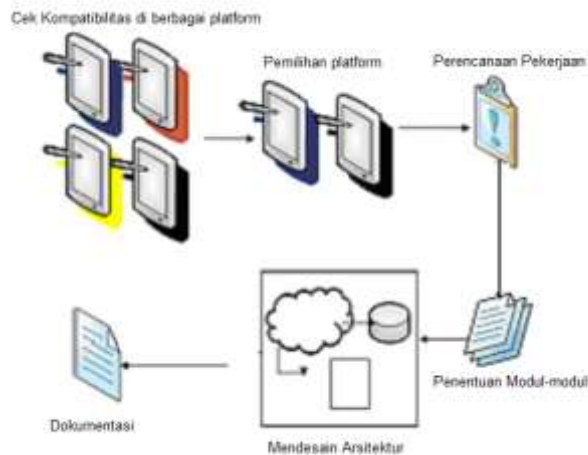
memiliki ide, ide tersebut diprivatisasi dan dianalisis lebih lanjut. Pengembang dapat menghasilkan ide untuk aplikasi baru. Daftar ideal yang disaring dipertimbangkan oleh tim ide aplikasi seluler, yang terdiri dari perwakilan bisnis dan TI, untuk kemungkinan memulai proyek di seputar ide tersebut. Untuk menentukan kebaruan suatu ide, kami mencari aplikasi yang ada di salah satu platform standar. Jika ada program serupa di pasar, popularitas dan fungsi yang didukung dari program diteliti dan dibandingkan. Perbedaan dari implementasi yang ada didokumentasikan. Jika tidak ada aplikasi serupa di platform seluler mana pun, ide tersebut harus didokumentasikan bersama dengan fitur dasarnya. Tugas penting lainnya pada tahap ini adalah menentukan waktu yang dibutuhkan untuk mengembangkan aplikasi[10]. Pengumpulan kebutuhan pokok juga harus dihentikan. Pekerjaan yang dilakukan oleh tim ide bergerak kemudian harus didokumentasikan dan dikomunikasikan kepada tim desain.



Gambar 1. Alur Kerja *Identification Phase*

## 2.2. Design Phase

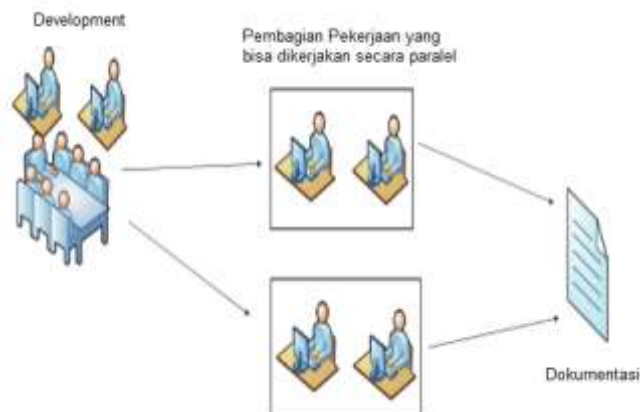
Pada tahap ini, tim pengembang aplikasi seluler mengubah ide menjadi desain awal aplikasi. Kelayakan pengembangan aplikasi untuk semua platform seluler sedang dieksplorasi[11]. Alternatifnya, platform seluler target spesifik diidentifikasi. Perlu diputuskan apakah aplikasi yang dikembangkan akan dirilis sebagai versi gratis atau sebagai versi uji coba dengan fitur terbatas atau hanya sebagai versi lanjutan. Pada tahap ini, fungsionalitas aplikasi dipecah menjadi modul dan kemudian digabungkan menjadi prototipe. Persyaratan fungsional ditentukan. Arsitektur perangkat lunak aplikasi dibuat. Prototipe dan modul terkait kemudian ditentukan. Bagian yang krusial dalam proses desain adalah membuat storyboard UI interaktif. Storyboard ini menggambarkan bagaimana aplikasi berjalan. Pekerjaan tim desain didokumentasikan dan diteruskan ke tim pengembangan untuk pengkodean.



Gambar 2. Alur Kerja *Design Phase*

### 2.3. *Development Phase*

Pada langkah ini, aplikasi mulai diimplementasikan dengan kode. Pengembangan modul yang berbeda dari prototipe yang sama dapat dilakukan secara bersamaan. Proses pengembangan dapat dibagi menjadi dua tahap: pengkodean kebutuhan fungsional dan pengkodean kebutuhan [12]. Mulailah dengan mengembangkan kode untuk fungsionalitas inti. Pengembangan paralel dari modul yang saling independen dari prototipe yang sama dimungkinkan. Selanjutnya, modul-modul ini dapat diintegrasikan. Pada tahap kedua, antarmuka pengguna dirancang untuk mendukung sebanyak mungkin platform sistem operasi seluler; bukan praktik yang baik untuk aplikasi yang sama memiliki tampilan dan nuansa yang berbeda pada platform yang berbeda. Desain harus menggunakan kumpulan minimum komponen antarmuka yang ada di semua platform sistem operasi seluler. Terakhir, dokumentasi dari tahap pengembangan akan diteruskan ke tahap *prototyping*.



Gambar 3. Alur Kerja *Development Phase*

#### 2.4. Prototyping Phase

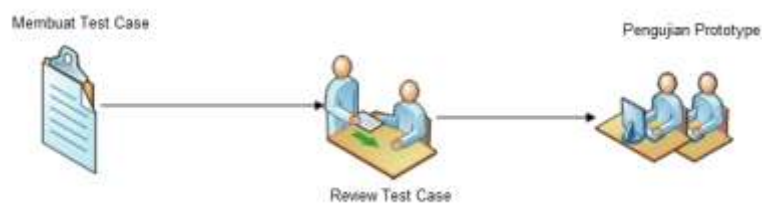
Pada langkah ini, kita menganalisis persyaratan fungsional dari setiap prototipe. Prototipe diuji dan dikirim ke pelanggan untuk mendapatkan umpan balik. Setelah menerima umpan balik dari pelanggan, perubahan yang diperlukan akan diterapkan selama tahap pengembangan [13]. Ketika prototipe kedua sudah siap, prototipe tersebut diintegrasikan dengan prototipe pertama, diuji, dan dikirim ke pelanggan. Ulangi tahap pengembangan, pembuatan prototipe, dan pengujian hingga prototipe akhir siap. Prototipe akhir dikirim ke klien untuk umpan balik akhir. Pekerjaan yang dilakukan selama fase pembuatan prototipe ini didokumentasikan dan kemudian diteruskan ke fase pengujian.



Gambar 4. Alur Kerja *Prototyping Phase*

#### 2.5. Testing Phase

Pengujian adalah salah satu langkah terpenting dalam setiap model siklus hidup pengembangan. Uji prototipe dilakukan di emulator/simulator lalu uji di perangkat nyata [14]. Emulator/simulator biasanya termasuk dalam SDK. Pengujian dengan perangkat nyata, misalnya sehubungan dengan pengembangan OS Android, harus dilakukan dengan beberapa versi sistem operasi, beberapa model ponsel dengan ukuran layar yang berbeda. Uji kasus dicatat dan diteruskan ke klien untuk umpan balik.



Gambar 5. Alur Kerja *Testing Phase*

Beberapa penelitian sebelumnya telah mengkaji pengembangan aplikasi mobile. Penelitian pertama berjudul “Desain dan Implementasi Aplikasi Mobile untuk Kuliah di Politeknik Negeri Lampung dengan Basis Teknologi Android” yang dilakukan oleh Kurniawan dan Jaya (2016). Penelitian ini membahas tentang pengembangan aplikasi konferensi telepon dengan menggunakan metode pengembangan air terjun. Perbedaan utamanya terletak pada metode yang digunakan. Jika penelitian terkait menggunakan metode *Waterfall*, maka penelitian ini menggunakan metode MADLC.

Penelitian lain yang relevan berjudul “Pengembangan Aplikasi Android MVTE dengan Pendekatan RAD” yang dilakukan oleh Amrullah dan rekan-rekannya pada tahun 2021. Penelitian ini membahas tentang pengembangan aplikasi *Mobile Voice to Text* (MVTE) dengan menggunakan metode RAD (*Rapid Application Development*). Perbedaan utamanya terletak pada metode yang digunakan, karena penelitian ini menggunakan metode MADLC.

Penelitian berikutnya berjudul “Pengembangan Aplikasi Berbasis Android untuk Pengenalan Huruf Hijaiyah” yang dilakukan oleh Gunawan (2019) dalam jurnal tersebut membahas tentang pengembangan aplikasi pengenalan huruf Hijaiyah menggunakan platform Android. Perbedaan utamanya terletak pada metode yang digunakan. Meskipun penelitian sebelumnya berfokus pada penggunaan metode SDLC (*Software Development Life Cycle*), namun penelitian ini mengadopsi metode MADLC.

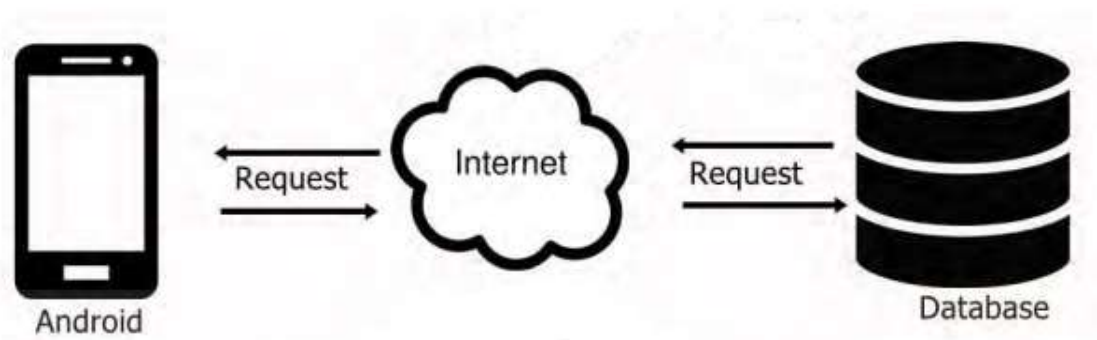
### 3. HASIL DAN ANALISIS

#### 3.1. Identification Phase

*Identification phase* bertujuan mendefinisikan komponen-komponen aplikasi yang akan dibangun nantinya, dan melibatkan pengguna selama proses desain untuk memastikan bahwa pengguna mendapatkan hasil yang benar yang mereka butuhkan. Hasil dari brainstorming dan diskusi antara user dan tim peneliti menghasilkan bahwa objek sistem adalah tanaman jagung. Kemudian sistem dikembangkan dengan model *Convolutional Neural Network* (CNN) sebagai algoritma dasar. Selain fungsi identifikasi penyakit, aplikasi juga dikembangkan dengan penambahan fitur rekomendasi perawatan tanaman, dimana data yang digunakan berasal dari pakar tanaman jagung.

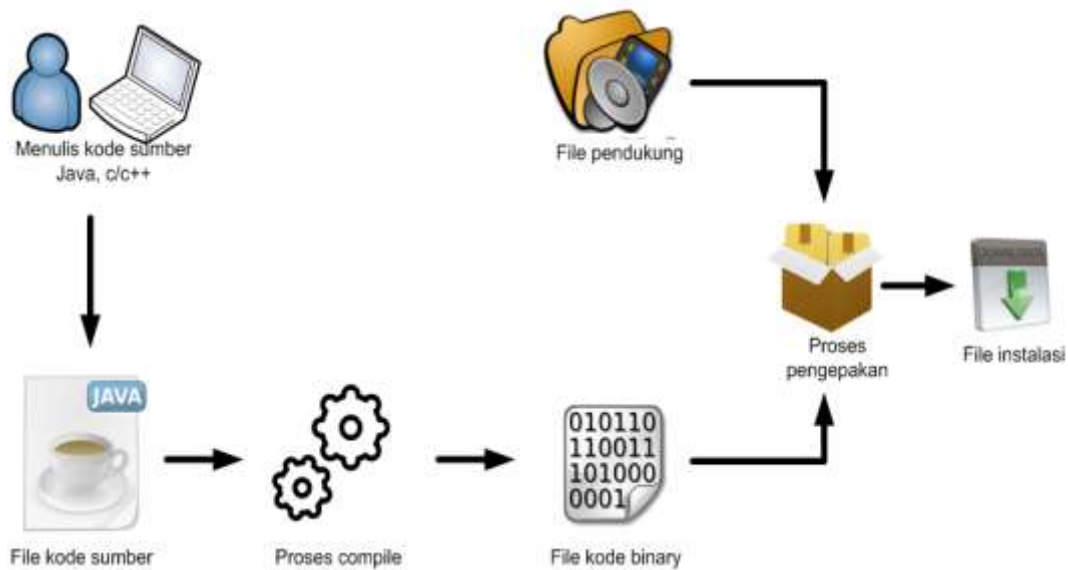
#### 3.2 Design Phase

Pada fase ini terbentuk rancangan arsitektur fungsional sistem, arsitektur aplikasi, dan desain awal aplikasi. Rancangan arsitektur sistem ditunjukkan pada Gambar 8.



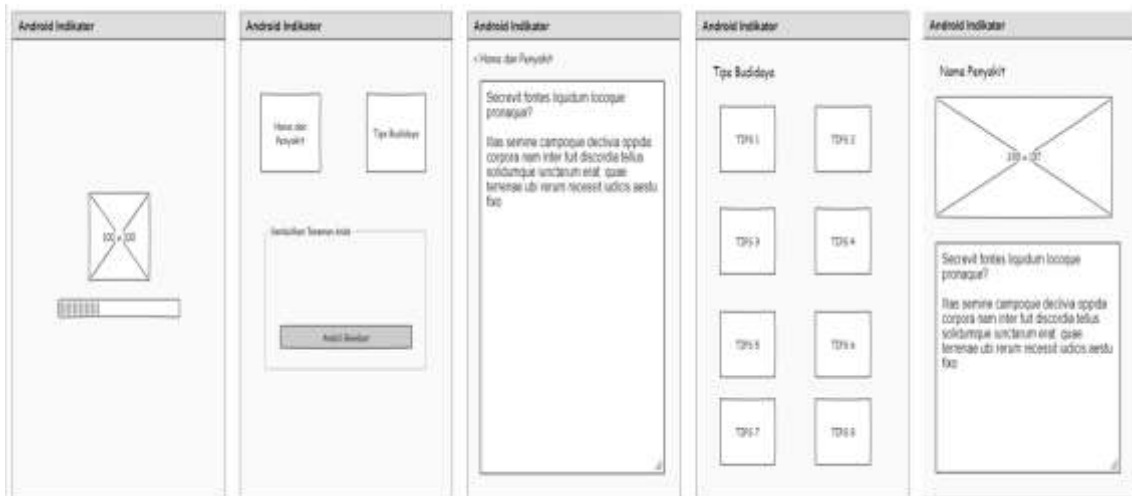
Gambar 8. Arsitektur Sistem

Kemudian pada fase ini menghasilkan arsitektur aplikasi android seperti pada Gambar 9.



Gambar 9. Arsitektur Aplikasi

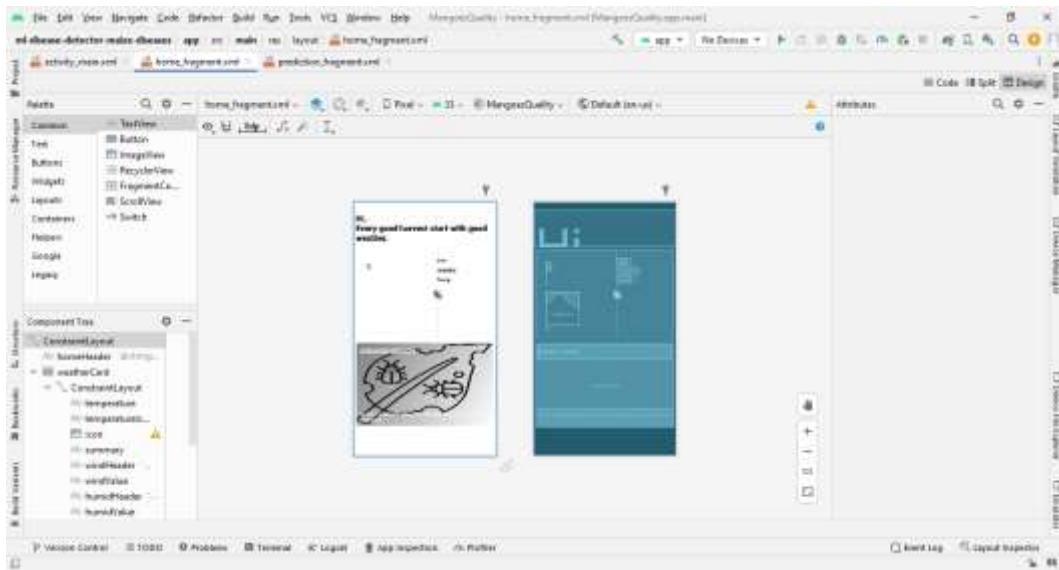
Untuk desain awal aplikasi dapat dilihat pada Gambar 10.



Gambar 10. Desain Awal Aplikasi

### 3.3. Development Phase

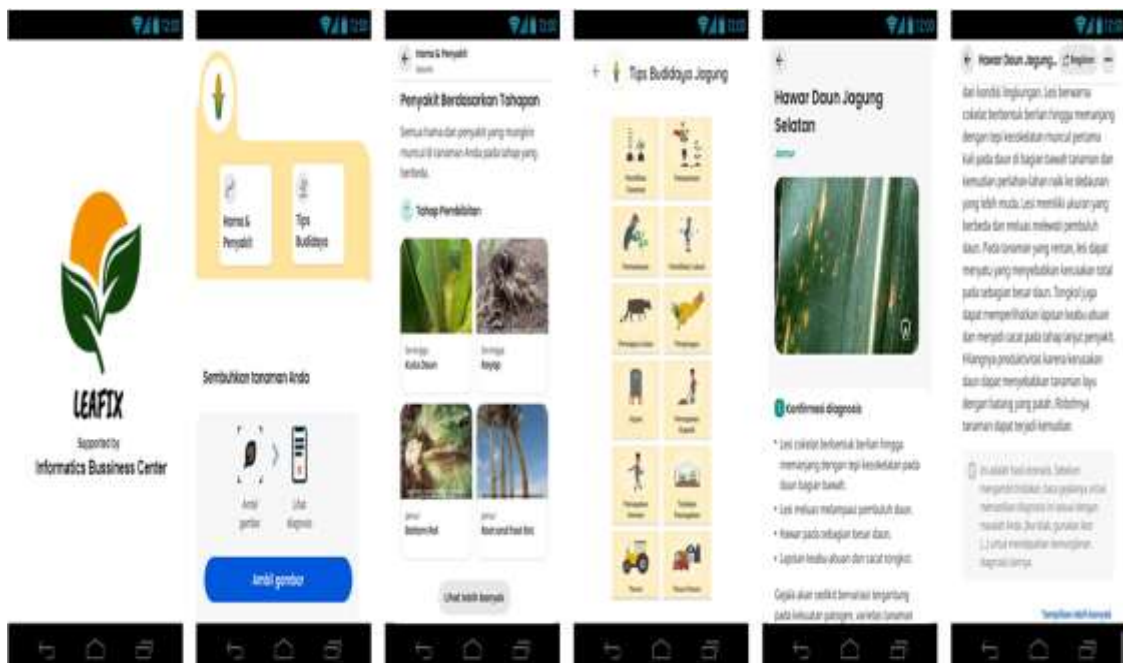
Pada tahap ini dilakukan pengkodean menggunakan Bahasa JAVA dengan *tools* Android Studio seperti pada Gambar 11.



Gambar 11. Pengodean dengan Android Studio

### 3.4 Prototyping Phase

Setelah tahap pengkodean maka hasilnya akan masuk ke tahap prototipe. Hasil tahap pengkodean dapat dilihat pada Gambar 12.



Gambar 12. Purwarupa Aplikasi



Tahap ini juga melibatkan pengguna untuk berinteraksi dengan aplikasi untuk mengetahui respon pengguna. User yang terlibat sebanyak 20 orang.

### 3.5 Testing Phase

Setelah purwarupa selesai, maka masuk tahap pengujian. Pengujian menggunakan metode blackbox testing dengan teknik error gussy [15]. Teknik error gussy merupakan salah satu teknik untuk melihat bagaimana aplikasi menangani error yang disebabkan oleh pengguna. Dalam pengujian melibatkan 20 user dengan berbagai tingkatan umur dengan tujuan melihat seberapa mudah aplikasi bisa digunakan oleh user. Hasil Rekapitulasi dari pengujian yang dilakukan seperti pada Tabel 1.

Tabel 1. Rekapitulasi Pengujian

User	Menu Utama	Menu Hama & Penyakit	Menu Tips Budidaya	Menu Diagnosis Penyakit
1	Sukses	Sukses	Sukses	Sukses
2	Sukses	Sukses	Sukses	Sukses
3	Sukses	Sukses	Sukses	Sukses
4	Sukses	Sukses	Sukses	Sukses
5	Sukses	Sukses	Sukses	Sukses
6	Sukses	Sukses	Sukses	Sukses
7	Sukses	Sukses	Sukses	Sukses
8	Sukses	Sukses	Sukses	Sukses
9	Sukses	Sukses	Sukses	Sukses
10	Sukses	Sukses	Sukses	Sukses
11	Sukses	Sukses	Sukses	Sukses
12	Sukses	Sukses	Sukses	Gagal
13	Sukses	Sukses	Sukses	Sukses
14	Sukses	Sukses	Sukses	Sukses
15	Sukses	Sukses	Sukses	Sukses
16	Sukses	Sukses	Sukses	Sukses
17	Sukses	Sukses	Sukses	Gagal
18	Sukses	Sukses	Sukses	Sukses
19	Sukses	Sukses	Sukses	Sukses
20	Sukses	Sukses	Sukses	Sukses

## 4. KESIMPULAN

Berdasarkan hasil dan pembahasan, maka dapat disimpulkan bahwa aplikasi pendeteksi penyakit daun tanaman jagung berhasil dibangun dengan menerapkan metode *mobile application development life cycle*. Berdasarkan hasil uji dengan teknik error gussy didapatkan ada 2 kasus yang gagal dari 20 kasus yang diujikan.

## DAFTAR PUSTAKA

- [1] J. M. F. Sihombing, “Keterlibatan Pengguna Dalam Pengembangan Perangkat Lunak Dan Keamanan Dalam Pengembangan Perangkat Lunak,” *Prosiding SAINTEK: Sains dan Teknologi*, vol. 2, no. 1, pp. 240–249, 2023.
- [2] F. N. Hasanah and R. S. Untari, *Buku Ajar Rekayasa Perangkat Lunak*. 2020.
- [3] A. Herdiansah, R. I. Borman, and S. Maylinda, “Sistem Informasi Monitoring dan Reporting Quality Control Proses Laminating Berbasis Web Framework Laravel,” *Jurnal Tekno Kompak*, vol. 15, no. 2, p. 13, 2021, doi: 10.33365/jtk.v15i2.1091.
- [4] J. S. H. Brata and Dwi Soediantono, “Total Quality Manufacturing (TQM) and Recommendations for Its Application in the Defense Industry: A Literature Review,” *International Journal of Social and Management Studies (Ijosmas)*, vol. Vol. 3 No., no. 3, pp. 50–62, 2022.
- [5] S. Ahdan, A. R. Putri, and A. Sucipto, “Aplikasi M-Learning Sebagai Media Pembelajaran Conversation Pada Homey English,” *Sistemasi*, vol. 9, no. 3, p. 493, 2020, doi: 10.32520/stmsi.v9i3.884.
- [6] M. T. Sembiring and Carine, “Peninjauan Prospek Smartphone Global: Studi Pustaka,” *TALENTA Conference Series: Energy & Engineering*, vol. 2, no. 3, pp. 274–284, 2019, doi: 10.32734/ee.v2i3.741.
- [7] T. S. Jaya and D. Sahlinal, “Potato Leaf Disease Detection with Convolutional Neural Network Method,” in *International Conference On Agriculture and Applied Science (ICoAAS)*, Politeknik Negeri Lampung, 2022, pp. 50–58. doi: <https://doi.org/10.25181/icoaas.v3i3.2867>.
- [8] T. S. Jaya, “Klasifikasi nanas layak jual dengan metode naïve bayes classifier dan k-nearest neighbor,” *Jurnal Sistem Informasi*, vol. 13, no. 1, pp. 2221–2229, 2021.
- [9] S. Ahdan, A. Sucipto, and Y. Agus Nurhuda, “Game untuk Menstimulasi Kecerdasan Majemuk pada Anak (Multiple Intelligence) Berbasis Android Game to Stimulate Children’s Multiple Intelligence Based on Android,” *Senter 2019*, no. November, pp. 554–568, 2019.
- [10] A. P. Ruminto, R. S. Sianturi, and K. C. Brata, “Perancangan Pengalaman Pengguna Mobile Learning menggunakan Metode Human Centered Design (Studi Kasus: Madrasah Ibtidaiyah Hidayatul Athfal),” ... *Informasi dan Ilmu Komputer e-ISSN*, vol. 6, no. 3, pp. 1253–1263, 2022.
- [11] R. Aditya, V. H. Pranatawijaya, and P. B. A. A. Putra, “Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype,” *Journal of Information Technology and Computer Science*, vol. 1, no. 1, pp. 47–57, 2021.
- [12] M. Purba, L. L. Tarigan, and D. E. br Jabat, “Pengembangan Sistem Informasi Perpustakaan Berbasis Web,” *Informasi dan Teknologi Ilmiah(INTI)*, vol. 1, no. 3, pp. 2225–2234, 2013.
- [13] A. C. Wardhana, C. Kartiko, W. A. Saputra, and T. Fani, “User Experience Lifecycle pada Aplikasi Knowledge Management System Inovasi Desa,” *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, p. 99, 2022, doi: 10.30865/mib.v6i1.3431.
- [14] A. Panduwinata, P. P. Arhandi, and B. S. D. Nugraha, “Penerapan Metode Gamifikasi Pada Aplikasi Mosam (Money Sampah) Berbasis Mobile,” vol. 12, pp. 1032–1044, 2020.

JSI : Jurnal Sistem Informasi (*E-Journal*), VOL. 16, NO. 1, Oktober 2024

ISSN Print : 2085-1588

ISSN Online : 2355-4614

LINK: <https://jsi.ejournal.unsri.ac.id/index.php/jsi/index>

- [15] T. S. Jaya, “Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analisis,” *JURNAL INFORMATIKA : Jurnal Pengembangan IT*, vol. 03, no. 02, pp. 45–48, 2018.