

Rancang Bangun Aplikasi Android untuk Klasifikasi Penyakit Tanaman Jeruk Menggunakan CNN dan Metode *Extreme Programming*

Divo Rio Gilang¹, Ulfa Khaira², Benedika Ferdian Hutabarat³

^{1, 2, 3}Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Indonesia
Email: ¹divorg60@gmail.com, ²ulfakhaira@unja.ac.id, ³benedika@unja.ac.id

Abstrak

Jeruk merupakan salah satu komoditas pertanian penting yang memiliki nilai ekonomi tinggi dan permintaan pasar yang besar. Namun, produktivitas dan kualitas jeruk sering terancam oleh penyakit pada kulit, seperti *Citrus Bacterial Spot*, *Citrus Canker*, dan *Huanglongbing* (HLB), yang dapat menurunkan nilai jual dan daya saing jeruk lokal. Deteksi penyakit secara manual memerlukan waktu dan keahlian khusus, sehingga dibutuhkan solusi berbasis teknologi yang cepat dan akurat. Penelitian ini bertujuan merancang dan membangun aplikasi mobile untuk identifikasi penyakit kulit jeruk menggunakan metode *Extreme Programming* (XP) serta mengevaluasi fungsionalitas sistem. Model klasifikasi citra dibangun menggunakan arsitektur *MobileNetV2*. Keunikan riset ini terletak pada integrasi *lightweight deep learning* model dengan pendekatan XP, sehingga menghasilkan aplikasi yang tidak hanya akurat, tetapi juga efisien dan siap diterapkan langsung di perangkat mobile dengan sumber daya terbatas. Evaluasi sistem menunjukkan akurasi validasi yang tinggi serta efisiensi kinerja berdasarkan metrik penggunaan sumber daya perangkat tanpa memaparkan detail teknis. Aplikasi ini mampu mengidentifikasi penyakit kulit jeruk secara cepat dan akurat, sehingga berpotensi membantu petani dalam pengambilan keputusan pengelolaan tanaman.

Kata kunci: aplikasi mobile, extreme programming, identifikasi penyakit jeruk, klasifikasi citra, mobilenetv2.

Abstract

Oranges are an important agricultural commodity with high economic value and strong market demand. However, their productivity and quality are often threatened by peel diseases such as *Citrus Bacterial Spot*, *Citrus Canker*, and *Huanglongbing* (HLB), which can reduce market value and weaken the competitiveness of local citrus products. Manual disease detection requires time and specialized expertise, creating the need for a fast and accurate technology-based solution. This study aims to design and develop a mobile application for citrus peel disease identification using the *Extreme Programming* (XP) method and to evaluate its system functionality. The image classification model is built using the *MobileNetV2* architecture. The novelty of this research lies in the integration of a *lightweight deep learning* model with the XP approach, resulting in an application that is not only accurate but also efficient and suitable for deployment on mobile devices with limited resources. System evaluation shows high validation accuracy and efficient performance based on device resource usage metrics without presenting technical details. This application is capable of identifying citrus peel diseases quickly and accurately, making it a potential decision-support tool for farmers in managing their citrus crops.

Keywords: citrus disease identification, extreme programming, image classification, mobile application, mobilenetv2.

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)



1. PENDAHULUAN

Jeruk adalah salah satu komoditas pertanian penting yang tidak hanya berkontribusi pada pemasukan negara, tetapi juga menjadi sumber penghasilan utama bagi petani, menyediakan bahan baku untuk industri, dan menciptakan lapangan kerja. Jeruk merupakan salah satu jenis buah yang banyak dikonsumsi oleh masyarakat, sehingga permintaannya cukup tinggi[1]. Namun, produktivitas ini menghadapi tantangan serius dari penyakit yang menyerang kulit buah, seperti *Citrus bacterial spot*, *Citrus Canker*, dan HLB. Penyakit seperti *citrus canker* yang disebabkan oleh bakteri *Xanthomonas citri* menjadi ancaman global yang dapat menurunkan kualitas dan grade jeruk, sehingga berdampak negatif pada harga jual dan pendapatan petani[2][3]. Penurunan kualitas ini membuat jeruk lokal sulit bersaing

dan dapat menghambat produksi jika tidak ditangani secara cepat dan tepat[4]. Pemilihan ketiga penyakit ini dalam penelitian didasarkan pada tingkat prevalensi dan dampaknya yang paling sering dilaporkan di sentra produksi jeruk nasional maupun internasional, sehingga deteksi dini menjadi sangat penting bagi keberlanjutan produksi.

Sebagai solusi, diperlukan sebuah sistem deteksi dini penyakit yang mudah diakses oleh petani. Aplikasi mobile menjadi platform strategis mengingat tingginya penetrasi pengguna ponsel di Indonesia, yang mencapai 67,88% pada tahun 2022 [5]. Untuk mengenali pola penyakit pada citra kulit jeruk, penelitian ini memanfaatkan Convolutional Neural Network (CNN), sebuah metode yang terbukti efektif dalam analisis citra tanpa memerlukan ekstraksi fitur manual [6]. Untuk efisiensi pada perangkat mobile, arsitektur MobileNetV2 dengan pendekatan transfer learning diimplementasikan guna mempercepat proses pelatihan model dengan sumber daya komputasi yang terbatas [7].

Sejumlah penelitian terdahulu turut menunjukkan relevansi pendekatan ini. Rahman et al. (2024) mengembangkan aplikasi mobile menggunakan metode Extreme Programming (XP) dan menemukan bahwa XP menawarkan fleksibilitas tinggi serta mendukung komunikasi intensif antara pengembang dan pemangku kepentingan melalui siklus pengujian berulang. Di sisi lain, Sigitta et al. (2023) menerapkan CNN untuk mendeteksi penyakit tanaman tomat melalui citra daun, dengan akurasi mencapai 90% pada arsitektur LeNet-5, yang menegaskan kemampuan CNN dalam pengenalan pola visual pada tanaman. Namun, penelitian-penelitian tersebut masih membahas penerapan XP dan CNN secara terpisah, serta belum mengintegrasikan model deep learning ringan seperti MobileNetV2 dengan pendekatan XP untuk membangun aplikasi deteksi penyakit tanaman yang efisien pada perangkat mobile. Selain itu, belum ada penelitian yang secara spesifik mengembangkan aplikasi identifikasi penyakit kulit jeruk berbasis Android yang siap pakai dan optimal pada perangkat dengan sumber daya terbatas. Gap inilah yang menjadi landasan kontribusi penelitian ini.

Proses pengembangan aplikasi ini menggunakan metode Extreme Programming (XP), sebuah pendekatan Agile yang menekankan fleksibilitas, kecepatan, dan komunikasi intensif untuk menghasilkan perangkat lunak yang sesuai dengan kebutuhan pengguna [8]. Setelah metode pengembangan dijelaskan, platform dan bahasa pemrograman yang digunakan juga diuraikan, yaitu Kotlin sebagai bahasa pemrograman berorientasi objek yang berjalan pada Java Virtual Machine (JVM), sehingga mendukung pengembangan aplikasi Android secara optimal [9].

Secara keseluruhan, tujuan inti dari penelitian ini adalah mengembangkan sebuah aplikasi mobile yang mampu mengidentifikasi penyakit pada kulit jeruk secara otomatis menggunakan model CNN berbasis MobileNetV2. Selain itu, penelitian ini berkontribusi dalam menyediakan alat bantu praktis bagi petani untuk mendeteksi penyakit lebih cepat dan mengurangi potensi kerugian hasil panen. Penelitian ini dibatasi pada platform Android serta berfokus pada beberapa jenis penyakit kulit jeruk yang paling umum, sehingga aplikasi yang dihasilkan diharapkan dapat menjadi solusi awal dalam upaya peningkatan kualitas dan keberlanjutan produksi jeruk nasional.

2. METODE PENELITIAN

2.1 Metode Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui observasi di Dinas Pertanian dan Ketahanan Pangan Kota Jambi untuk memahami kebutuhan sistem identifikasi penyakit kulit jeruk, wawancara dengan petani dan pihak terkait guna memperoleh informasi mendalam mengenai karakteristik penyakit serta kendala yang dihadapi, dan studi literatur dari berbagai sumber ilmiah untuk memperoleh landasan teoritis dalam pengembangan model deteksi berbasis Convolutional Neural Network (CNN) dengan arsitektur MobileNetV2.

2.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk merumuskan spesifikasi sistem yang akan dikembangkan berdasarkan hasil observasi, wawancara, dan studi literatur. Kebutuhan fungsional meliputi kemampuan aplikasi untuk mengunggah atau mengambil gambar kulit jeruk, memproses citra menggunakan model *MobileNetV2* untuk mengidentifikasi jenis penyakit, serta menampilkan informasi hasil deteksi beserta langkah penanganan yang direkomendasikan. Kebutuhan non-fungsional mencakup kemudahan penggunaan (*user friendly*), respons cepat dalam pemrosesan citra, serta kompatibilitas dengan berbagai perangkat Android. Hasil analisis ini menjadi acuan dalam perancangan antarmuka dan arsitektur sistem.

2.3 Desain Sistem

Tahap ini menghadirkan langkah-langkah yang menjelaskan secara detail cara kerja perangkat lunak serta implementasi konsep yang telah dianalisis sejak tahap perencanaan. peneliti bertujuan untuk merinci serangkaian proses penting yang akan dilakukan dalam tahap ini. Beberapa langkah spesifik yang direncanakan oleh peneliti meliputi:

- *Use Case Diagram*, Menggambarkan interaksi antara pengguna dengan fungsionalitas aplikasi deteksi penyakit kulit jeruk dalam bentuk skenario penggunaan[10].
- *Activity diagram*, Memodelkan alur proses aplikasi dari input citra hingga penampilan hasil deteksi secara visual.
- *Class diagram*, Menjelaskan struktur sistem, hubungan antar kelas, atribut, dan metode sebagai acuan implementasi.

2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang banyak digunakan untuk pengenalan citra karena kemampuannya mengekstraksi fitur secara otomatis melalui operasi konvolusi, *pooling*, dan *fully connected layer*. CNN efektif mendeteksi pola visual seperti tekstur, tepi, dan bentuk pada citra, yang sangat relevan untuk mengidentifikasi gejala penyakit pada kulit jeruk.

Penelitian ini menggunakan *MobileNetV2*, sebuah arsitektur CNN yang dirancang untuk perangkat mobile melalui pemanfaatan *depthwise separable convolution* dan *inverted residual block*. *MobileNetV2* mendukung *transfer learning*, yaitu proses memanfaatkan bobot model pra-latih (pretrained) pada dataset besar seperti *ImageNet*, kemudian menyesuaikannya (*fine-tuning*) dengan dataset khusus citra kulit jeruk. Pendekatan ini mempercepat waktu pelatihan dan meningkatkan akurasi meskipun dataset relatif terbatas.

2.5 Implementasi

Tahap implementasi merupakan inti dari metode *Extreme Programming* (XP), di mana pengembangan dilakukan secara iteratif dengan membagi fitur menjadi tugas-tugas kecil yang dapat diselesaikan dan diuji dalam siklus singkat. Model deteksi penyakit dibangun menggunakan Python dan pustaka TensorFlow melalui pendekatan *transfer learning* dengan arsitektur *MobileNetV2*, yang kemudian di-*fine-tuning* menggunakan dataset citra kulit jeruk. Model yang telah dilatih dikonversi ke format TensorFlow Lite (*TFLite*) untuk mengoptimalkan performa pada perangkat mobile. Selanjutnya, aplikasi Android dikembangkan dengan bahasa Kotlin, dan model TFLite diintegrasikan menggunakan Firebase ML Kit agar dapat melakukan deteksi secara langsung di perangkat pengguna. Proses pengembangan menerapkan *continuous integration* untuk memastikan setiap penambahan kode terintegrasi dengan baik, mendeteksi kesalahan sejak dini, dan menjaga stabilitas sistem hingga diperoleh aplikasi yang siap digunakan.

2.6 Pengujian

Tahap pengujian dilakukan menggunakan metode Black Box Testing untuk memverifikasi bahwa seluruh fungsi aplikasi berjalan sesuai spesifikasi tanpa memeriksa struktur internal kode [8]. Pengujian difokuskan pada kesesuaian input dan output sistem, dengan skenario uji yang mencakup kasus uji (test case), rencana pengujian, hasil yang diharapkan, dan hasil aktual, yang disusun dalam tabel untuk memudahkan analisis[11]. Evaluasi fungsionalitas dilakukan dengan menghitung tingkat keberhasilan menggunakan rumus:

$$F = \frac{B}{C}$$

dengan F adalah tingkat fungsionalitas, B jumlah pengujian berhasil, dan C jumlah kasus pengujian. Nilai mendekati 1 menunjukkan aplikasi berfungsi dengan baik.

Selain pengujian fungsional, dilakukan pula pengujian kinerja dan kompatibilitas menggunakan Firebase Test Lab dengan metode RoboTest pada berbagai perangkat Android untuk memastikan aplikasi mampu berjalan stabil dan efisien di berbagai kondisi. Pengujian ini mencakup evaluasi penggunaan memori, konsumsi CPU, serta performa aplikasi secara umum. Selain itu, dilakukan evaluasi performa model deteksi penyakit menggunakan confusion matrix untuk menghitung nilai

akurasi, precision, recall, dan F1-score pada setiap kelas penyakit kulit jeruk. Penggunaan confusion matrix memberikan penilaian yang lebih komprehensif terhadap kemampuan model dalam membedakan setiap jenis penyakit secara tepat.

Hasil dari seluruh pengujian tersebut kemudian dianalisis secara kuantitatif dengan membandingkan hasil aktual terhadap nilai yang diharapkan, menafsirkan persentase keberhasilan fungsi, serta mengevaluasi efisiensi penggunaan sumber daya. Nilai akurasi, precision, recall, dan F1-score ditafsirkan untuk menentukan seberapa baik model mengenali setiap kelas penyakit, sehingga analisis ini memberikan dasar objektif dalam menilai kelayakan, akurasi, dan stabilitas aplikasi secara menyeluruh.

3. HASIL DAN PEMBAHASAN

Dalam proses pengembangan sistem, peneliti menerapkan metode Extreme Programming sebagai pendekatan utama. Metode ini sejalan dengan prinsip perancangan berbasis object-oriented, khususnya ketika dikembangkan oleh tim berukuran kecil dengan komunikasi yang efektif bersama pengguna, sehingga mampu menghasilkan solusi yang sesuai dengan kebutuhan [12]. Secara umum, metode ini terbagi ke dalam beberapa tahapan, yaitu perencanaan (planning), perancangan (design), pengkodean (coding), dan pengujian (testing)[13].

3.1. Analisis Kebutuhan

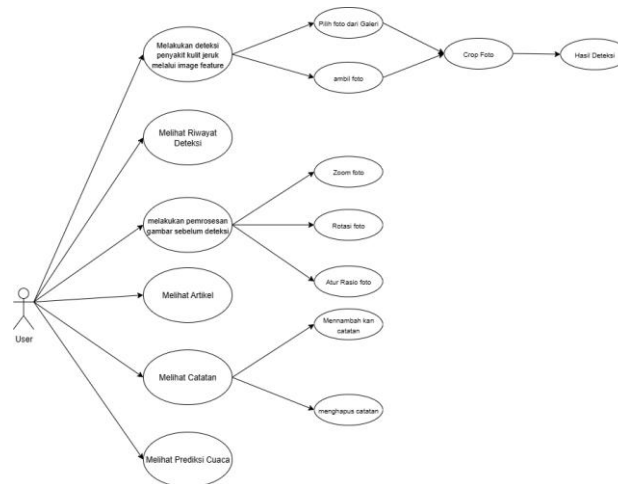
Pada tahap awal penelitian, peneliti menerapkan tahapan *listening* dari metode Extreme Programming (XP). Tahap ini dilakukan melalui wawancara langsung dengan mitra dari Dinas Pertanian dan Ketahanan Pangan Kota Jambi, khususnya dengan salah satu asisten kebun di Balai Benih Hortikultura (BBH), untuk menggali kebutuhan dan permasalahan yang dihadapi dalam pengelolaan tanaman jeruk.

Berdasarkan hasil wawancara, kebutuhan fungsional sistem meliputi kemampuan aplikasi untuk mendeteksi penyakit pada kulit jeruk berbasis *image feature* dengan dukungan fitur pengolahan citra seperti pemotongan (*crop*), rotasi, zoom, dan penyesuaian rasio sebelum analisis. Sistem juga harus mampu menyajikan informasi terkait karakteristik penyakit, metode penanganan, pencegahan, serta panduan penggunaan aplikasi. Selain itu, aplikasi perlu menyediakan fitur penambahan, penghapusan, dan penayangan catatan hasil deteksi, menampilkan daftar catatan, serta menyimpan dan menampilkan riwayat deteksi sebelumnya. Fitur tambahan yang diharapkan meliputi penyajian informasi prakiraan cuaca secara real-time di lokasi pengguna, serta konten edukatif berupa artikel dan video terkait pertanian, perawatan jeruk, dan penyakit tanaman. Seluruh kebutuhan ini dirancang agar aplikasi dapat menjadi alat bantu yang praktis, informatif, dan relevan bagi petani jeruk.

3.2. Desain Sistem

A. Use Case Diagram

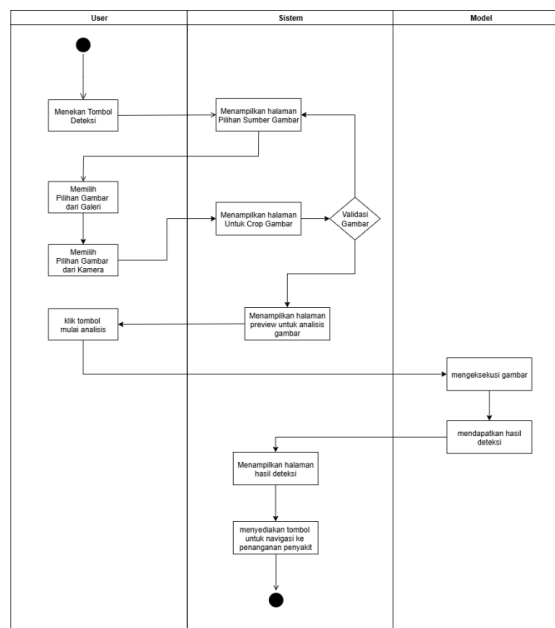
Diagram ini berfungsi untuk menggambarkan alur kontrol sistem antar objek dalam aplikasi [14]. Berdasarkan hasil perancangan, pengguna (user) pada aplikasi ini adalah petani jeruk serta pemangku kepentingan lainnya seperti pegawai BBH dan peneliti di bidang tanaman.



Gambar 1. Use Case Diagram SiTani Jeruk

B. Activity Diagram

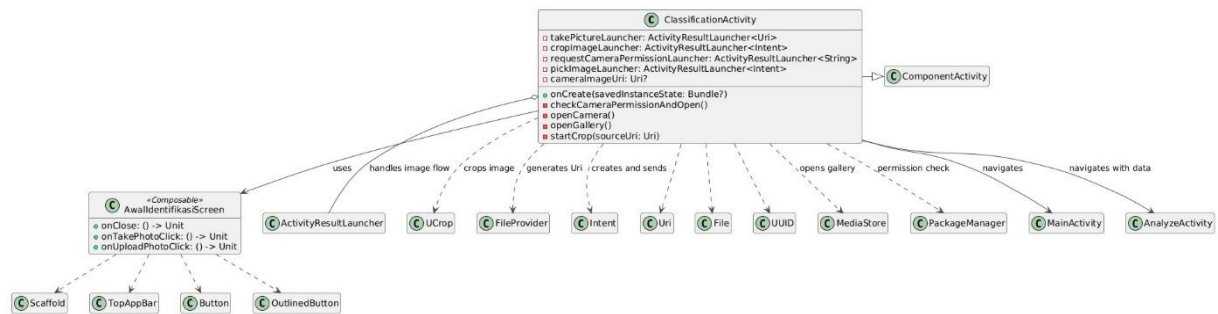
Aktivitas pertama pada Fitur deteksi dimulai dengan pemilihan gambar dari galeri atau kamera, dilanjutkan dengan pemotongan (*crop*) dan pratinjau. Setelah pengguna menekan tombol “Mulai Analisis”, gambar diproses oleh model klasifikasi untuk mengidentifikasi jenis penyakit pada kulit jeruk, kemudian sistem menampilkan hasil deteksi beserta tautan menuju halaman penanganan penyakit yang sesuai.



Gambar 2. Activity Diagram Deteksi

C. Class Diagram

Pengembangan aplikasi difokuskan pada penyusunan struktur dasar dengan mengintegrasikan `ComponentActivity`, `ViewModel`, dan fungsi `Composable`. `ComponentActivity` digunakan untuk menampilkan antarmuka awal melalui `setContent`, `ViewModel` mengelola data dan logika seperti hasil deteksi, cuaca, dan catatan, sedangkan fungsi `Composable` membangun antarmuka seperti halaman deteksi dan artikel. Pendekatan ini menghasilkan arsitektur yang modular, responsif, dan mudah dikembangkan pada iterasi berikutnya. Representasi rancangan sistem ditunjukkan pada class diagram aplikasi SiTaniJeruk berikut.



Gambar 3. Class Diagram Klasifikasi SiTani Jeruk






3.3. Implementasi Model

Model yang digunakan dalam penelitian ini adalah MobileNetV2, yang dipilih sebagai base model dibandingkan arsitektur lain seperti ResNet atau VGG karena beberapa alasan. Dari sisi efisiensi komputasi, MobileNetV2 dirancang khusus untuk perangkat mobile dengan jumlah parameter relatif kecil (sekitar 2,25 juta), sehingga sangat sesuai untuk implementasi pada aplikasi Android yang menjadi target penelitian. Dari sisi performa, MobileNetV2 telah terbukti mampu memberikan akurasi tinggi pada klasifikasi citra ringan meskipun menggunakan jumlah parameter yang lebih sedikit dibandingkan arsitektur lain. Selain itu, MobileNetV2 memiliki struktur depthwise separable convolution yang memungkinkan ekstraksi fitur dilakukan secara efisien sekaligus menjaga kompleksitas model tetap rendah, sehingga proses pelatihan berlangsung lebih cepat namun tetap mampu mengenali fitur-fitur penting pada citra penyakit jeruk.

3.3.1. Persiapan Dataset

Dataset terdiri dari citra kulit jeruk yang dikelompokkan ke dalam lima kelas: *Sehat*, *Greening*, *Citrus Canker*, *Black Spot*, dan *Not Orange*. Masing-masing kelas berisi 150 gambar, sehingga total dataset berjumlah 750 citra berformat *.jpg* atau *.jpeg*. Sebelum pelatihan, citra dinormalisasi dan diubah ukurannya menjadi 224×224 piksel agar sesuai dengan arsitektur MobileNetV2, sehingga memastikan konsistensi input dan memudahkan model dalam mempelajari pola fitur secara optimal.

Tabel 1. Dataset SiTani Jeruk

Black Spot	Citrus Canker	Greening	Fresh	Not Orange
				

3.3.2. Pra-pemrosesan data

Untuk meningkatkan keragaman data dan mencegah overfitting pada model MobileNetV2, dilakukan augmentasi citra menggunakan teknik rotasi (hingga 30°), pergeseran horizontal dan vertikal (30%), shear (20%), zoom (20%), pembalikan horizontal, dan penyesuaian kecerahan. Seluruh citra dinormalisasi dengan membagi nilai piksel terhadap 255, serta diubah ukurannya menjadi 224×224 piksel. Selain itu, 20% dari dataset secara otomatis dialokasikan untuk proses validasi. Pendekatan ini bertujuan memperbesar jumlah data latih, meningkatkan variasi visual, dan memperkuat kemampuan generalisasi model.

Tabel 2. Augmentasi DataSet

Rotasi	Pergeseran	Pemotongan secara miring	Perbesar	pembalikan horizontal
				

3.3.3. Training

Pelatihan dilakukan menggunakan arsitektur MobileNetV2 sebagai base model melalui pendekatan transfer learning untuk memanfaatkan bobot yang telah dilatih sebelumnya pada dataset ImageNet, sehingga proses pelatihan menjadi lebih efisien dan akurasi model dapat ditingkatkan. Bagian klasifikasi asli pada arsitektur ini dihilangkan (`include_top=False`) agar dapat disesuaikan dengan kebutuhan penelitian, kemudian diganti dengan rangkaian lapisan tambahan berupa Global Average Pooling untuk mereduksi dimensi fitur, Batch Normalization untuk menstabilkan distribusi data selama pelatihan, beberapa lapisan Dropout guna mencegah overfitting, dua lapisan Dense dengan aktivasi ReLU sebagai pemrosesan non-linear, serta satu lapisan Dense output dengan aktivasi Softmax yang terdiri dari lima neuron sesuai dengan jumlah kelas yang diklasifikasikan dalam penelitian ini.

Model Architecture: Model: "sequential"		
Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
batch_normalization (BatchNormalization)	(None, 1280)	5,120
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 512)	655,872
batch_normalization_1 (BatchNormalization)	(None, 512)	2,048
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131,328
batch_normalization_2 (BatchNormalization)	(None, 256)	1,024
dropout_2 (Dropout)	(None, 256)	0
predictions (Dense)	(None, 5)	1,285

Gambar 4. Arsitektur Model MobileNetV2 dengan Lapisan Klasifikasi Tambahan

Pelatihan model dilakukan menggunakan beberapa hyperparameter penting yang dipilih melalui pendekatan empiris:

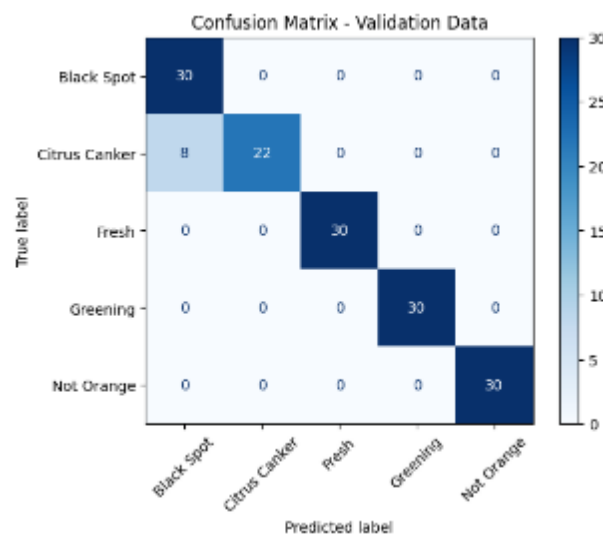
- **Batch Size:** 32, dipilih karena mampu menjaga keseimbangan antara stabilitas gradien dan efisiensi komputasi. Batch size kecil (<16) cenderung menghasilkan gradien yang fluktuatif, sedangkan batch size besar (>64) membutuhkan memori lebih besar dan memperlambat adaptasi model terhadap variasi data.
- **Epochs:** Total 30 epoch dengan strategi *early stopping* jika validasi loss tidak membaik setelah 5 epoch berturut-turut. Proses pelatihan dilakukan dua tahap: (1) *initial training* selama 10 epoch dengan MobileNetV2 dalam kondisi *freeze*, sehingga hanya lapisan klasifikasi tambahan yang dilatih; (2) *fine-tuning* selama 20 epoch dengan membuka sebagian lapisan akhir MobileNetV2 untuk menangkap fitur spesifik dari citra kulit jeruk.
- **Learning Rate & Scheduling:** Optimizer yang digunakan adalah Adam dengan *initial learning rate* 0,001. Untuk menghindari konvergensi prematur, digunakan *ReduceLROnPlateau* dengan faktor 0,1 ketika validasi loss stagnan selama 3 epoch. Strategi ini memperhalus proses konvergensi menuju titik optimal.

- **Dropout Rate:** Tiga lapisan dropout digunakan dengan tingkat 0,5. Strategi ini efektif mencegah overfitting dengan menonaktifkan sebagian neuron secara acak saat pelatihan.
- **Regularisasi:** Selain dropout, lapisan *Batch Normalization* digunakan untuk menstabilkan distribusi aktivasi, mempercepat proses pelatihan, dan meningkatkan generalisasi model.
- **Loss Function & Metrics:** *Categorical crossentropy* dipilih sebagai fungsi kerugian karena klasifikasi bersifat multi-kelas, dengan akurasi (*accuracy*) sebagai metrik utama.
- **Class Imbalance Handling:** Karena jumlah sampel tiap kelas seimbang (150 citra per kelas), pendekatan class weights tetap digunakan untuk memastikan distribusi label proporsional selama pelatihan.

Dengan konfigurasi tersebut, pelatihan dilakukan menggunakan dataset berjumlah 750 citra kulit jeruk yang terbagi ke dalam lima kelas. Sebanyak 20% data otomatis dialokasikan sebagai data validasi. Untuk meningkatkan keragaman data, diterapkan augmentasi berupa rotasi (hingga 30°), pergeseran horizontal dan vertikal (30%), shear (20%), zoom (20%), pembalikan horizontal, serta penyesuaian kecerahan. Seluruh citra dinormalisasi dengan skala [0,1] dan diubah ukurannya menjadi 224×224 piksel agar sesuai dengan arsitektur MobileNetV2.

Pada tahap initial training, seluruh lapisan MobileNetV2 dibekukan (freeze) untuk mempertahankan pengetahuan dari pre-trained ImageNet, sehingga hanya lapisan klasifikasi tambahan yang dilatih selama 10 epoch menggunakan optimizer Adam (learning rate 0,001) dan categorical crossentropy loss. Untuk menangani ketidakseimbangan data, digunakan class weights yang dihitung otomatis berdasarkan distribusi label.

3.3.4. Confusion Matrix



Gambar 5. Confusion Matrix

Gambar menunjukkan confusion matrix pada data validasi model klasifikasi penyakit jeruk. Kelas Black Spot, Fresh (Healthy), Greening, dan Not Orange berhasil dikenali secara sempurna dengan seluruh 30 sampel terklasifikasi benar. Sementara itu, kelas Citrus Canker menunjukkan hasil kurang optimal, dengan hanya 22 dari 30 sampel yang teridentifikasi dengan benar, sedangkan 8 sampel lainnya salah diprediksi sebagai Black Spot. Kondisi ini mengindikasikan adanya kemiripan fitur antara Citrus Canker dan Black Spot yang menyebabkan model mengalami kesulitan dalam membedakan keduanya.

Secara keseluruhan, confusion matrix ini menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan sebagian besar kelas penyakit pada jeruk. Meski demikian, kelemahan pada kelas Citrus Canker menjadi catatan penting untuk pengembangan lebih lanjut, seperti melalui penambahan jumlah data latih pada kelas tersebut atau penerapan teknik augmentasi khusus agar model mampu mengenali perbedaan fitur yang lebih halus antar kedua kelas.

Hasil confusion matrix menunjukkan adanya misklasifikasi antara kelas Citrus Canker dan Black Spot. Fenomena ini dapat dianalisis dari dua sisi:

- **Sisi Visual:** Kedua penyakit sama-sama menampilkan bercak hitam pada permukaan kulit jeruk. Pada beberapa sampel, ukuran, bentuk, dan distribusi bercak terlihat sangat mirip, sehingga secara kasat mata pun sulit dibedakan.
- **Sisi Fitur Ekstraksi Model:** MobileNetV2 mengekstraksi pola tekstur dan warna sebagai fitur utama. Karena kemiripan tekstur bercak hitam pada kedua kelas, model terkadang gagal menemukan fitur diskriminatif yang konsisten. Perbedaan halus, seperti adanya lingkaran kekuningan di sekitar bercak Citrus Canker, mungkin tidak tertangkap optimal jika jumlah data terbatas.

Untuk mengurangi misklasifikasi ini, penelitian lanjutan dapat mempertimbangkan:

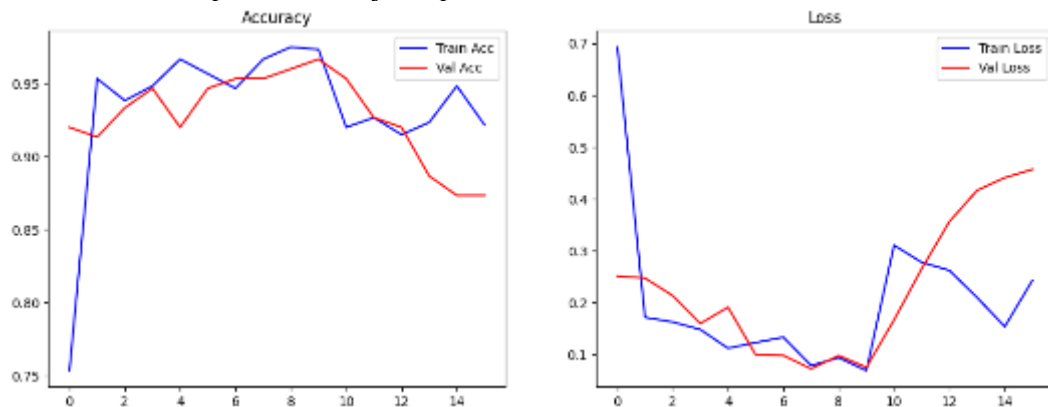
- Penambahan jumlah data latih pada kelas Citrus Canker
- Teknik augmentasi yang menekankan variasi tekstur bercak
- penerapan *attention mechanism* agar model lebih fokus pada pola bercak yang khas.

3.3.5. Hasil Evaluasi Model

Setelah pelatihan selesai, model dievaluasi menggunakan data validasi sebesar 20% dari total data pelatihan. Berdasarkan hasil evaluasi akhir, diperoleh:

- Akurasi validasi akhir: 95.33%
- Loss validasi akhir: 0.1664

Hasil ini menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat baik terhadap citra daun jeruk berdasarkan kategori penyakit maupun kondisi sehat. Nilai loss yang rendah juga menandakan bahwa prediksi model stabil dan tidak mengalami overfitting secara signifikan. Visualisasi akurasi dan loss selama pelatihan disajikan pada Gambar berikut:



Gambar 6. Grafik Akurasi dan Loss Selama Pelatihan Model

Jika dibandingkan dengan penelitian-penelitian sebelumnya, performa model yang digunakan dalam penelitian ini menunjukkan capaian yang lebih baik. Sigitta et al. (2023) menerapkan arsitektur *LeNet-5* untuk mendeteksi penyakit pada tanaman tomat dan memperoleh akurasi sebesar 90%. Sementara itu, model *MobileNetV2* yang digunakan pada penelitian ini mampu mencapai akurasi validasi hingga 95.33%, menunjukkan bahwa arsitektur yang lebih modern dan ringan tersebut memiliki kemampuan ekstraksi fitur yang lebih efektif untuk klasifikasi citra penyakit tanaman.

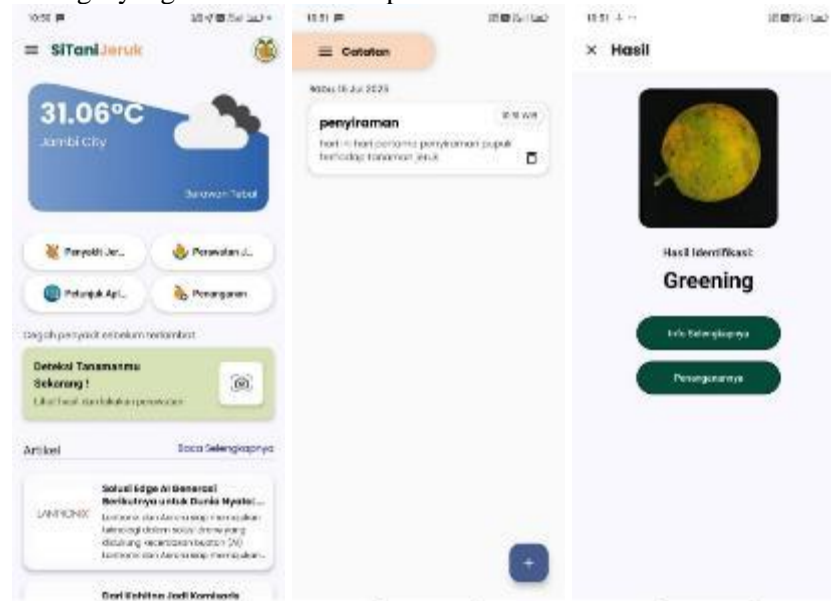
3.3.6. Analisis Keterbatasan Penelitian

Meskipun hasil yang diperoleh cukup baik, penelitian ini memiliki beberapa keterbatasan. Pertama, jumlah dataset masih terbatas dengan total 750 citra, sehingga keragaman data belum cukup untuk mewakili kondisi nyata di lapangan secara menyeluruh. Kedua, variasi kondisi citra seperti pencahayaan, sudut pengambilan gambar, dan latar belakang masih kurang beragam. Hal ini menyebabkan model berpotensi mengalami misklasifikasi ketika menghadapi citra dengan kondisi

berbeda dari data latih. Ketiga, model sangat bergantung pada kualitas input gambar; pencahayaan yang buruk atau adanya objek pengganggu pada latar belakang dapat menurunkan akurasi prediksi, khususnya pada kelas yang memiliki kemiripan visual tinggi, seperti *Citrus Canker* dan *Black Spot*.

3.4. Implementasi Coding

proses pengembangan kode telah menghasilkan sejumlah halaman yang dapat berfungsi dengan baik. Halaman pertama yang ditampilkan adalah beranda, Halaman ini dirancang untuk memberikan akses cepat dan mudah ke berbagai fitur inti dalam aplikasi. sehingga memudahkan pengguna dalam menjelajahi seluruh fungsi yang disediakan oleh aplikasi SiTaniJeruk.



Gambar 7. Halaman Beranda Sitani Jeruk kiri, Halaman Catatan SiTani Jeruk Tengah, dan Halaman Deteksi SiTani Jeruk Kanan

Halaman catatan memungkinkan pengguna mencatat dan memantau aktivitas perawatan tanaman, seperti penyiraman, pemupukan, dan hasil deteksi penyakit, sehingga riwayat perawatan tersimpan rapi untuk memudahkan evaluasi dan pengambilan keputusan.

Halaman deteksi memungkinkan pengguna memilih gambar dari galeri atau kamera, melakukan pemotongan (crop), lalu menganalisisnya menggunakan model klasifikasi untuk menampilkan hasil identifikasi penyakit pada kulit jeruk beserta opsi menuju halaman penanganan.

3.5. Pengujian

3.5.1. Pengujian Black-Box

Pengujian aplikasi SiTaniJeruk menggunakan metode black-box dilakukan oleh dua tester dari Dinas Pertanian dan Ketahanan Pangan untuk menguji enam fungsi utama sesuai user stories. Setiap fungsi utama diuji berdasarkan skenario yang sesuai dengan *user stories*. Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3. Skenario Pengujian

No.	Fitur yang Diuji	Aktivitas Pengujian	Hasil yang Diharapkan	Kesimpulan
1.	Halaman Utama	Membuka aplikasi dan melewati onboarding	Aplikasi menampilkan halaman utama dan klasifikasi	Berhasil
2.	Deteksi Penyakit	Mengambil/unggah gambar, crop, analisis,	Aplikasi menampilkan hasil deteksi dan informasi penanganan	Berhasil

		lihat hasil dan penanganan		
3.	Catatan	Menambah dan menghapus catatan	Aplikasi menampilkan catatan berhasil ditambah atau dihapus	Berhasil
4.	Artikel	Membuka artikel, melihat isi, kembali ke halaman utama	Aplikasi menampilkan artikel dengan benar	Berhasil
5.	Prediksi Cuaca	Membuka halaman prediksi cuaca dan kembali ke utama	Aplikasi menampilkan informasi prediksi cuaca	Berhasil
6.	Riwayat Deteksi	Membuka histori deteksi dan melihat daftar riwayat	Aplikasi menampilkan riwayat sesuai hasil sebelumnya	Berhasil

Adapun perhitungan tingkat keberhasilan pengujian dilakukan dengan menggunakan rumus sebagai berikut:

$$\frac{\sum x}{n} \times 100\% \dots\dots\dots(1)$$

$$\frac{\sum y}{n} \times 100\% \dots\dots\dots(2)$$

Keterangan:

n = Jumlah semua fungsi yang ada

$\sum x$ = Jumlah fungsi yang berhasil berjalan

$\sum y$ = Jumlah fungsi yang gagal berjalan

Diketahui:

$n = 6 \times 2$, $\sum x = 6 \times 2$, $\sum y = 0$

Jika dimasukkan ke dalam rumus:

$$\frac{\sum x}{n} \times 100\% = \frac{12}{12} \times 100\% = 100\%$$

$$\frac{\sum y}{n} \times 100\% = \frac{0}{12} \times 100\% = 0\%$$

Dengan demikian, fungsi yang berhasil dijalankan dinilai dengan persentase 100%, sementara fungsi yang tidak berjalan diberikan nilai 0%.

3.5.2. Pengujian Aspek Performance Efficiency

Dalam Pengujian Aspek Performance Efficiency, Pengujian ini mencakup aspek penggunaan CPU dan memori. Proses pengujian dilakukan menggunakan Firebase Test Lab selama 4 menit 53 detik pada perangkat simulasi Pixel 5 dengan API Level 30. Berikut adalah grafik hasil kinerja penggunaan CPU dan memori saat menjalankan aplikasi SiTaniJeruk :

a) CPU

Hasil pengujian efisiensi kinerja pada aspek penggunaan CPU dapat dilihat pada gambar grafik pengujian berikut:

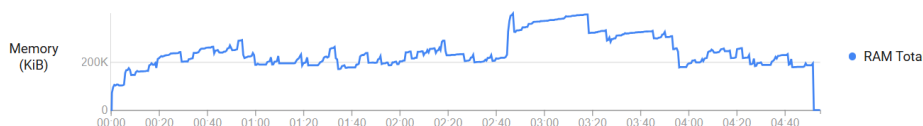


Gambar 8. Hasil Pengujian Penggunaan CPU

Berdasarkan hasil pengujian efisiensi kinerja CPU pada Gambar X, penggunaan CPU aplikasi berkisar antara 0% hingga sekitar 22% selama proses pengujian. Rentang ini masih tergolong rendah untuk aplikasi mobile berbasis pemrosesan citra, terutama karena sebagian besar literatur menyebutkan bahwa penggunaan CPU di bawah 30% umumnya dikategorikan efisien untuk aplikasi yang melakukan komputasi berbasis machine learning pada perangkat bergerak. Dengan demikian, penggunaan CPU sebesar 0–22% pada aplikasi ini menunjukkan bahwa proses deteksi berjalan cukup ringan dan tidak membebani sumber daya perangkat secara berlebihan.

b) Memory

Hasil pengujian efisiensi kinerja pada aspek penggunaan CPU dapat dilihat pada gambar grafik pengujian berikut:



Gambar 9. Hasil Pengujian Penggunaan Memori

Berdasarkan hasil pengujian efisiensi kinerja memori pada Gambar X, aplikasi memulai penggunaan memori sekitar ± 200 KiB dan meningkat bertahap hingga mencapai puncak ± 450 KiB pada menit ke-3 lebih 10 detik. Setelah itu, terjadi fluktuasi ringan hingga akhir pengujian. Pola ini menunjukkan pemanfaatan memori yang dinamis namun tetap dalam batas wajar [15].

Selain itu, hasil pengujian penggunaan CPU dan memori ini juga konsisten dengan penelitian sebelumnya. Sigitta et al. (2023) yang menggunakan arsitektur LeNet-5 pada deteksi penyakit tanaman tomat melaporkan akurasi 90% dengan kebutuhan komputasi yang lebih tinggi, sementara model MobileNetV2 pada penelitian ini mampu mencapai akurasi 95,33% dengan konsumsi CPU yang tetap berada pada rentang efisien (0–22%). Temuan ini menunjukkan bahwa arsitektur MobileNetV2 lebih optimal untuk diterapkan pada perangkat mobile yang memiliki keterbatasan komputasi. Selain itu, penelitian Rahman et al. (2024) menegaskan bahwa metode Extreme Programming (XP) efektif dalam mempercepat proses pengembangan aplikasi melalui iterasi berulang, namun belum mengintegrasikan model deep learning ringan. Oleh karena itu, hasil pengujian ini memperlihatkan bahwa penggabungan XP dan MobileNetV2 pada penelitian ini memberikan kontribusi baru berupa pengembangan aplikasi deteksi penyakit tanaman yang tidak hanya akurat, tetapi juga efisien secara komputasi.

4. KESIMPULAN

Penelitian ini berhasil menghasilkan aplikasi mobile berbasis Android untuk identifikasi penyakit pada tanaman jeruk dengan menerapkan metode Extreme Programming (XP) dan integrasi model Convolutional Neural Network (CNN) yang dikonversi ke TensorFlow Lite. Penerapan XP berperan penting dalam memastikan proses pengembangan berlangsung iteratif, adaptif, dan terstruktur melalui tahapan perencanaan, pengkodean, dan pengujian yang dilakukan secara berulang sehingga seluruh fitur dapat disempurnakan secara berkelanjutan. Aplikasi yang dikembangkan mampu mengklasifikasikan penyakit kulit jeruk dengan akurasi yang baik, berjalan stabil pada berbagai perangkat Android, serta dilengkapi dengan fitur tambahan seperti artikel, berita pertanian, riwayat laporan, dan petunjuk penggunaan. Hasil evaluasi model yang menunjukkan akurasi sebesar 95,33%, efisiensi penggunaan CPU (0–22%), dan konsumsi memori yang rendah menegaskan bahwa arsitektur MobileNetV2 efektif untuk klasifikasi citra penyakit tanaman pada perangkat mobile. Meskipun demikian, pengembangan lebih lanjut masih diperlukan, terutama pada perluasan dataset dengan jumlah citra yang lebih banyak, variasi kondisi lingkungan, dan penambahan jenis penyakit agar akurasi model semakin tinggi. Selain itu, aplikasi ini diharapkan dapat dikembangkan lintas platform seperti iOS serta diuji lebih

komprehensif melalui pengujian *usability* dan *user acceptance* sehingga kualitas, kegunaan, dan penerimaan aplikasi oleh pengguna dapat semakin optimal.

DAFTAR PUSTAKA

- [1] T. R. Izzalqurny, A. Ilmia, and A. Mufidah, "Pemanfaatan Dan Pengolahan Potensi Buah Jeruk Untuk Pengembangan Produk Umkm Desa Gunting Kecamatan Sukorejo," *Disem. J. Pengabd. Kpd. Masy.*, vol. 4, no. 1, pp. 74–84, 2022, doi: 10.33830/diseminasiabdimas.v4i1.1866.
- [2] L. Liu *et al.*, "Clarification of the infection pattern of *Xanthomonas citri* subsp. *citri* on citrus fruit by artificial inoculation," *Plant Methods*, vol. 20, no. 1, pp. 1–13, 2024, doi: 10.1186/s13007-024-01190-7.
- [3] R. Riswandi, R. Jamiah, N. Mardhatillah, and H. P. Hamid, "Klasifikasi Penyakit Pada Citra Daun Jeruk Menggunakan Arsitektur MobileNet berbasis Mobile Platform," *J. Fokus Elektroda Energi List. Telekomun. Komputer, Elektron. dan Kendali*, vol. 6, no. 4, p. 212, 2021, doi: 10.33772/jfe.v6i4.19113.
- [4] A. B. Sembiring, I. M. Sudana, and N. W. Suniti, "Identifikasi Jamur Penyebab Penyakit Kudis pada Buah Jeruk Siam Kintamani (*Citrus nobilis* L.) dan Pengendaliannya Secara Hayati," *J. Agroekoteknologi Trop.*, vol. 10, no. 1, pp. 1–14, 2021, [Online]. Available: <https://ojs.unud.ac.id/index.php/JAT>
- [5] Badan Pusat Statistik, "Persentase Penduduk yang Memiliki/Menguasai Telepon Seluler Menurut Provinsi dan Klasifikasi Daerah, 2021-2023," 2022.
- [6] R. C. Sigitta, R. H. Saputra, and F. Fathulloh, "Deteksi Penyakit Tomat melalui Citra Daun menggunakan Metode Convolutional Neural Network," *Avitec*, vol. 5, no. 1, p. 43, 2023, doi: 10.28989/avitec.v5i1.1404.
- [7] R. Faturrahman, Y. S. U. N. Hariyani, and S. Hadiyoso, "Klasifikasi Jajanan Tradisional Indonesia berbasis Deep Learning dan Metode Transfer Learning," vol. 11, no. 4, pp. 945–957, 2023.
- [8] I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, "Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android," *INOVTEK Polbeng - Seri Inform.*, vol. 5, no. 2, p. 297, 2020, doi: 10.35314/isi.v5i2.1654.
- [9] A. Febriandirza, "Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin," *Pseudocode*, vol. 7, no. 2, pp. 123–133, 2020, doi: 10.33369/pseudocode.7.2.123-133.
- [10] A. Voutama and E. Novalia, "Perancangan Aplikasi M-Magazine Berbasis Android Sebagai Sarana Mading Sekolah Menengah Atas," *J. Tekno Kompak*, vol. 15, no. 1, p. 104, 2021, doi: 10.33365/jtk.v15i1.920.
- [11] R. I. Borman, A. T. Priandika, and A. R. Edison, "Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 3, p. 272, 2020, doi: 10.26418/justin.v8i3.40273.
- [12] R. Wahyudi, A. Junaidi, A. Saryoko, D. Setiawati, S. Santi Winarsih, and G. Setyaningsih, "Pengembangan dan Pengujian Aplikasi 'SIATAP' Sistem Informasi Tanya Apoteker Berbasis Android dan Website," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 3, pp. 210–218, 2020.
- [13] M. F. F. Rahman, K. Darussalam, R. C. Saphira, and F. Purwani, "Implementasi Extreme Programming Dalam Pengembangan Aplikasi Mobile Pengenalan Organisasi Pada Masa Orientasi Mahasiswa," *Just IT J. Sist. Informasi, Teknol. Inform. dan Komput.*, vol. 14, no. 2, pp. 128–132, 2024.
- [14] A. Auric and H. Pangaribuan, "Implementasi Aplikasi E-Wallet Untuk Umkm Menggunakan Metode Agile Dengan Qr Code Pada Tensorflow Berbasis Android," *Comput. Sci. Ind. Eng.*, vol. 8, no. 1, pp. 119–129, 2023, doi: 10.33884/comasiejournal.v8i1.6743.
- [15] Viving Frendiana and Fahmi Febriyansyah, "Aplikasi Android untuk Validasi Berkas Tes Covid-19," *J. Comput. Sci. Vis. Commun. Des.*, vol. 7, no. 1, pp. 57–65, 2022, doi: 10.55732/jikdiskomvis.v7i1.428.