

Aplikasi Deteksi Penyakit pada Daun Tomat Berbasis Android Menggunakan Model Terlatih *Tensorflow Lite*

Android Based Application for Tomato Plant Leaf Disease Detection Using the Tensorflow Lite

Yuyun Hana Natbais*, Ari Bangkit Sanjaya Umbu

Program Studi Fisika, Fakultas Sains dan Teknik, Universitas Nusa Cendana, Kota Kupang 85001, Indonesia

*E-mail: yuyunnatbais01@gmail.com

Diterima: 5 Mei 2023; Disetujui: 22 Juni 2023

ABSTRAK

Tanaman Tomat (*Solanum lycopersicum* Mill) merupakan salah satu jenis tanaman yang dapat tumbuh di seluruh wilayah Indonesia sehingga sering dibudidayakan. Salah satu tantangan dalam budidaya tomat yaitu penyakit daun. Penyakit daun dapat menyebabkan turunnya kualitas dan kuantitas buah tomat yang dihasilkan. Berdasarkan permasalahan tersebut, penulis melakukan perancangan aplikasi deteksi penyakit daun tomat dengan menggunakan *Convolutional Neural Network* (CNN). Dalam penelitian ini penyakit daun tomat diklasifikasikan menjadi sembilan jenis penyakit. *Image processing* dilakukan dengan mengimplementasikan *Algoritma Machine Learning*. *Datasheet* jenis dan gambar penyakit daun tomat diambil dari *Kaggle Competition page*. *Web Teachable Machine* digunakan untuk melatih model menjadi *Tensorflow lite*. Selanjutnya dibangun menjadi aplikasi android dengan basis bahasa pemrograman Java menggunakan *Android Studio*. Aplikasi yang berhasil dibangun diberi nama *Tomato Leaf Disease*. Aplikasi ini dapat mendeteksi gambar yang berasal dari galeri maupun hasil tangkapan kamera. Berdasarkan hasil uji coba, aplikasi yang dibangun dapat mengidentifikasi penyakit daun dengan baik. Tingkat akurasi sangat bergantung pada spesifikasi kamera, sudut pengambilan gambar, dan pencahayaan saat pengambilan gambar.

Kata kunci: android; deteksi penyakit; *teachable machine*; *tensorflow lite*; tomat

ABSTRACT

The tomato plant (*Solanum lycopersicum* Mill) is one plant that can grow in all regions of Indonesia, so it is often cultivated. One of the challenges in tomato cultivation is leaf disease. Leaf disease can lead to a decrease in the quality and quantity of tomato fruits produced. Based on these problems, the author designed an application for detecting tomato leaf disease using the *Convolution Neural Network* (CNN). This study classified tomato leaf diseases into nine types of diseases. *Image processing* is done by implementing *Machine Learning Algorithms*. *Datasheet* types and pictures of tomato leaf disease taken from the *Kaggle Competition page*. The *Web Teachable Machine* is used to train models into *Tensor flow lite*. Furthermore, it is built into an *Android application* based on the *Java programming language* using *Android Studio*. The successfully built application is named *Tomato Leaf Disease*. This application can detect images that come from the gallery as well as the camera captures. Based on the results of the trials, the built application can identify leaf disease well. The accuracy rate depends on camera specifications, shooting angles, and picture exposure.

Keywords: android; disease detection; *teachable machine*; *tensorflowlite*; tomato

PENDAHULUAN

Tanaman tomat (*Solanum lycopersicum* Mill.) merupakan tanaman jenis hortikultura yang mudah tumbuh di hampir seluruh Indonesia karena tumbuhan jenis ini dapat tumbuh dengan baik pada ketinggian rata-rata 200-700 mdpl. Salah satu tantangan dalam budidaya tomat yaitu penyakit daun. Penyakit daun dapat menyebabkan turunnya kualitas dan kuantitas buah tomat yang dihasilkan (Givari and Halilintar, 2021).

Tomat merupakan salah satu tanaman hortikultura yang sangat menjanjikan dalam pemasarannya, karena tanaman tomat banyak diolah menjadi produk makanan dan minuman sehingga permintaan pasarnya cukup tinggi. Namun penyakit dan hama yang menyerang tanaman tomat menyebabkan kualitas dan kuantitas yang dipanen tidak maksimal. Hal ini tentunya menyebabkan

ketidakseimbangan antara permintaan pasar dan pasokan, sehingga menyebabkan harga tomat meningkat (Putra et al., 2020). Penyakit-penyakit yang menyerang tanaman tomat bisa disebabkan oleh jamur, bakteri maupun virus. Hingga saat ini sebagian besar petani mendeteksi jenis penyakit tomat hanya berdasarkan pengamatan secara langsung dan kemudian mengambil tindakan seadanya tanpa mencari tahu lagi lebih lanjut guna penanganan yang tepat (Alviansyah et al., 2017).

Metode *Convolution Neural Network* merupakan metode yang sistem kerjanya terinspirasi dari visual cortex otak yang dibuat khusus untuk memproses suatu data dengan struktur grid (Putri et al., 2022). Sehingga dengan bantuan *CNN* daun tomat dapat diklasifikasi menjadi 9 kelas penyakit yaitu *Bacterial spot*, *Early blight*, *Late blight*, *Leaf mold*, *Mosaic virus*, *Septoria leaf spot*, *Spider mites*, *Target spot* dan *Yellow leaf curl virus* (Fuentes et al., 2017). *Convolution*

Neural Network telah terbukti menjadi alat yang sangat potensial dalam pemrosesan citra (Dhakshaya and Auxillia, 2019).

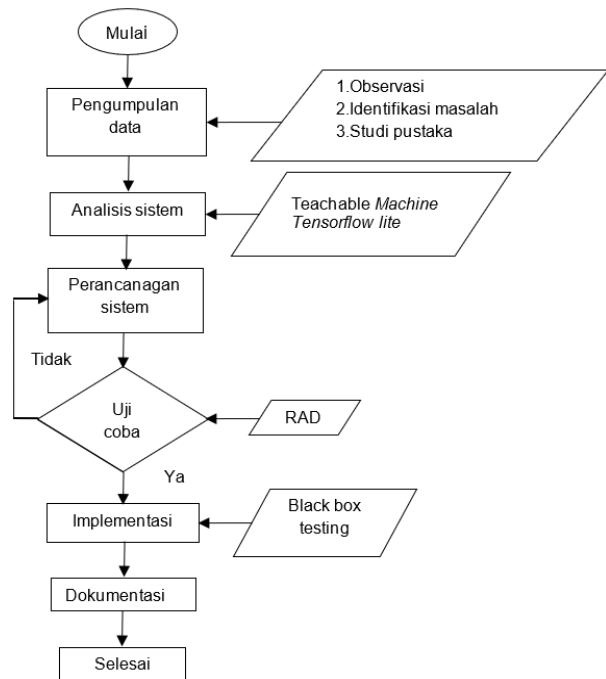
Penelitian terdahulu tentang klasifikasi penyakit menggunakan model terlatih *Tensorflow lite* seperti ini sudah dilakukan dengan judul “Implementasi Metode *Convolution Neural Network* untuk Mendeteksi Penyakit Pada Citra Daun Tomat”. Penelitian ini membahas tentang mendeteksi penyakit pada daun tomat, menggunakan metode pengembangan sistem *Waterfall*, algoritma *Convolution Neural Network (CNN)*, metode *Deep learning*, menggunakan Bahasa pemrograman *Python* dan dilakukan dengan memanfaatkan platform *Google Collab*. Pembangunan aplikasi dibangun dengan basis bahasa pemrograman *Kotlin*. Aplikasi ini menghasilkan akurasi ketika menggunakan gambar hasil tangkapan kamera dan dan mengambil gambar dari galeri masing-masing 84% dan 94% (Kotta et al., 2022). Penelitian lainnya yang berjudul “Sistem Pendeteksi Penyakit Pada Daun Tanaman Singkong Menggunakan *Deep Learning* Dan *Tensorflow* Berbasis Android”. Penelitian ini membahas tentang pendeteksi penyakit pada daun singkong menggunakan metode *Prototyping*, *Augmentasi Citra Tensorflow* dengan bantuan *Library package Keras*, pelatihan data menggunakan metode *Stratified K-Fold Cross Validation*, serta basis bahasa pemrograman untuk bangun aplikasi adalah bahasa *Kotlin*. Hasil akurasi yang diperoleh sebesar 86% (Faturrahman et al., 2022). Penelitian yang sama juga dengan judul “implementasi *tensorflow lite* pada *teachable* untuk identifikasi tanaman aglaonema berbasis android”. Penelitian ini membahas tentang membuat aplikasi berbasis android yang dapat membantu mendeteksi jenis tanaman aglonema dengan menggunakan datasheet yang diambil secara langsung menggunakan kamera *Handphone*. Hasil foto diproses dengan menggunakan tools Adobe Photoshop CS6 dan web *Teachable machine* untuk menghasilkan model *Tensorflow Lite* (Baihaqi et al., 2022).

Image processing atau pengolahan citra digital adalah metode untuk mengolah gambar dua dimensi menggunakan computer. Citra digital didefinisikan sebagai sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks dan direpresentasikan dengan deret bit tertentu (Alviansyah et al. 2017). *Image processing* ini sangat bermanfaat karena dapat meningkatkan kualitas citra, menghilangkan cacat pada citra, dan mengidentifikasi objek (Simbolon et al., 2022). Sehingga dapat juga digunakan dalam bidang pertanian untuk mendeteksi secara terperinci penyakit yang menyerang tanaman pertanian.

Penelitian dengan judul “aplikasi deteksi penyakit pada daun tomat berbasis android menggunakan model terlatih *tensorflow lite*”, menerapkan metode *Image processing* dengan mengimplementasikan *Algoritma Machine Learning* dan membutuhkan data sheet yang diambil dari *Kaggle Competition page* lalu memanfaatkan alat web *Teachable Machine* untuk melatih model menjadi *Tensorflow lite* kemudian dibangun menjadi aplikasi android dengan basis bahasa pemrograman *Java* menggunakan *Android Studio*. Tujuan penelitian ini menghasilkan aplikasi berbasis android yang diberi nama *Tomato Leaf Disease* untuk mendeteksi penyakit yang menyerang daun tomat.

METODOLOGI

Gambaran mengenai tahapan pelaksanaan penelitian ini dijelaskan menggunakan sebuah *flowchart* yang ditunjukan pada Gambar 1.



Gambar 1. Tahapan penelitian

Pengumpulan Data

Pembuatan model *machine learning* membutuhkan jumlah data yang banyak karena kurangnya data untuk dilatih dapat menyebabkan *overfitting* (model terlalu fokus pada pola spesifik kemudian mengabaikan pola yang umum) dan *underfitting* (data yang digunakan sedikit serta memiliki banyak *noise* sehingga model gagal memahami pola pada data). Dalam pembuatan model *machine learning* juga dibutuhkan data yang seimbang untuk setiap kelasnya agar performa model *machine learning* bisa mencapai hasil yang lebih baik (Elwirehardja et al., 2023).

Pada penelitian ini Data yang digunakan berupa data sheet yang diambil dari situs *Kaggle Competition page*. Datasheet merupakan kumpulan data yang mencantumkan suatu nilai untuk setiap variabel (Gunawan et al., 2022). Data yang digunakan berisi 125.000 citra berbentuk JPG dari 10 jenis tanaman berbeda dan Setiap citra pada datasheet ini sudah dilabeli dengan bantuan *Convolution Neural Network (CNN)*. Namun dalam penelitian ini hanya digunakan datasheet tanaman tomat yang terdiri dari 10 kelas berbeda dengan jumlah citra pada setiap kelasnya bervariasi rentang 2.000 sampai 7.000 jumlah citra. Oleh karena itu penulis memilih untuk setiap kelasnya menggunakan 2.000 citra pada setiap kelasnya agar jumlah data seimbang dan waktu untuk melatih data tidak terlalu lama. Pada penelitian ini jumlah keseluruhan data adalah 20.000 citra JPG.

Analisis Sistem

Penelitian ini menggunakan *Teachable machine* dan *Tensorflow lite* yang memiliki peranan yang sangat penting dalam menentukan keberhasilan aplikasi.

Teachable Machine

Teachable machine adalah alat berbasis web yang digunakan untuk membangun model pembelajaran kecerdasan buatan sederhana. Klasifikasi data pada *Teachable machine* dapat dilakukan dengan menambahkan kelas berdasarkan klasifikasi biner (Jeong, 2020). Data latih dan data uji dapat diinput melalui webcam maupun menggunakan file dalam penyimpanan perangkat.

Teachable Machine digunakan untuk mengolah 20.000 citra JPG menjadi model terlatih *Tensorflow lite*. *Teachable*

machine dipilih karena dapat membuat model *machine learning* dengan cepat, mudah dan coding Android studio yang sangat ringkas. Pelatihan model menggunakan *Standar image model* yaitu 224×224px yang kemudian data citra diupload ke dalam 10 kelas berbeda dengan label atau nama kelas sesuai dengan datasheet yang digunakan. Dalam penelitian ini ada 10 nama kelas yaitu daun tomat terserang *Bacterial spot*, daun tomat terserang *Early blight*, daun tomat sehat, daun tomat terserang *Late blight*, daun tomat terserang *Leaf mold*, daun tomat terserang *Mosaic virus*, daun tomat terserang *Septoria leaf spot*, daun tomat terserang *Spidermites*, daun tomat terserang *taget spot* dan daun tomat terserang *Yellow leaf curl virus*. Sebelum *Train model* dilakukan, terlebih dahulu ditentukan nilai *Epoch* pada menu *Advance*. Nilai ini merupakan jumlah *Train model* yang akan dilakukan. Pemilihan nilai *Epoch* yang besar akan memberikan hasil prediksi yang lebih akurat (Peryanto et al., 2020). Pada tahap ini nilai *Epoch* yang digunakan sebesar 50. Selanjutnya ditentukan nilai *Batch size*, nilai *Batch size* merupakan jumlah sampel data yang dilatih ke *Neural Network*. Dalam penelitian ini penulis menggunakan 20.000 *datasheet* dan nilai *Batch size* 16. Algoritma ini menggunakan 16 sampel data pertama dari 20.000 data untuk dilatih oleh *Neural Network* hingga selesai. Selanjutnya data kedua dari 20.000 data diambil dan proses ini berlanjut sampai 16 sampel data ke 1.250. Angka tersebut merupakan hasil dari jumlah *datasheet* dibagi nilai *Batch size* 20.000/16. Pada tahap akhir dilakukan penentuan nilai *Learning rate* sebelum dilakukan *training model*. Nilai *Learning rate* merupakan parameter dari algoritma *Gradient descent* yang fungsinya untuk mengoptimalkan iterasi yang digunakan pada *machine learning* untuk menentukan hasil yang terbaik (minimal kurva). *Epoch*, *Batch size* dan *Learning rate* dalam *machine learning* disebut *hyperparameter tuning*. Nilai dari parameter ini sangat berpengaruh pada performa model, jika nilai yang digunakan besar maka akurasi model akan semakin baik dan nilai *loss* akan semakin kecil (Julianto et al., 2022). Selanjutnya model dilatih dan dilakukan *Export model* ke *tensorflow lite* dengan tipe model konversi *Floating Point*.

Tensorflow lite

Tensorflow lite adalah *framework* dari *tensorflow* yang dibuat khusus untuk mengeksekusi model-model kompleks yang dapat dikembangkan di android (Aqil et al., 2021). *Tensorflow lite* dapat berjalan tanpa adanya koneksi internet dan lebih fleksibel terutama ketika menggunakan model *custom* untuk *machine learning model* (Taufiq and Pratama, 2021). Setelah melakukan *training* di *Teachable Machine*, akan diperoleh file zip dengan nama *converted_tflite*. File zip berisikan beberapa folder seperti folder *train* yang berisi 10 kelas citra yang sudah dilatih dan folder *valid* berisi 10 kelas citra yang sudah dilatih. Dalam pembangunan model *machine learning* folder *train* berfungsi untuk melatih atau membangun model dan folder *valid* merupakan kumpulan data yang digunakan untuk mengoptimasi atau menguji kinerja saat melatih model (Putra, 2020). Selain itu terdapat juga file text dokumen bernama *label* dan *TFLITE* file yang bernama *model_unquant.tflite*. Kedua file inilah yang kemudian digunakan dalam perancangan sistem.

Metode Pengembangan Sistem

Pada penelitian ini digunakan metode *Rapid Application Development* (RAD). Metode RAD merupakan proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. RAD terdiri dari tiga tahapan yaitu *Requirements Planning*

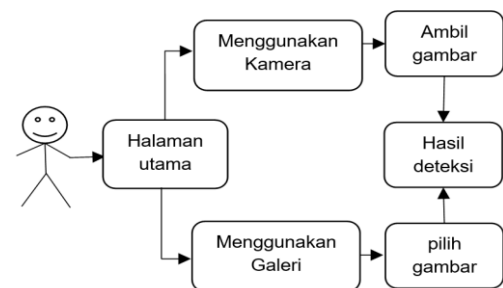
(Perancangan kebutuhan), *Design Workshop* (Proses perancangan) dan *implementation* (Penerapan) (Hariyanto et al., 2021). RAD juga merupakan strategi siklus hidup yang berfungsi sebagai penyedia pengembangan yang jauh lebih cepat dan mendapatkan hasil dengan kualitas yang lebih baik dibandingkan dengan hasil yang dicapai melalui siklus tradisional (Recky, 2023).

Perancangan Sistem Aplikasi

Perancangan system ini terdiri dari *Use Case Diagram* dan *Activity Diagram*.

Use Case Diagram

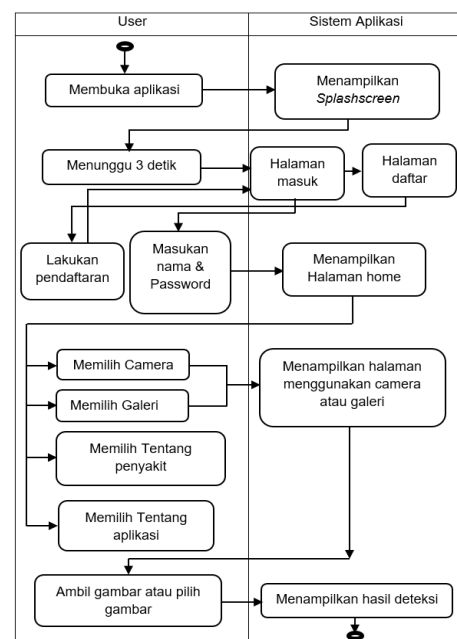
Use case Diagram merupakan diagram yang menceritakan bagaimana sebuah sistem dipakai oleh *user* (T. Bayu and Syarifuddin, 2020). Dalam aplikasi ini *user* diarahkan ke halaman utama untuk memilih menu kamera atau galeri. Menu kamera digunakan untuk mengambil gambar secara langsung menggunakan kamera perangkat (*Handphone*) dan menu galeri digunakan untuk mengambil gambar yang bersumber dari penyimpanan perangkat (*Handphone*). Proses ini seperti ditunjukkan pada Gambar 2.



Gambar 2. Use case diagram

Activity Diagram

Activity diagram merupakan diagram yang menggambarkan aliran kerja dalam sistem (T. Bayu and Syarifuddin, 2020). Diagram menampilkan dua tabel yang memisahkan antara alur aktivitas *user* dan alur aktivitas aplikasi. *Activity Diagram* ini ditunjukkan dalam Gambar 3.

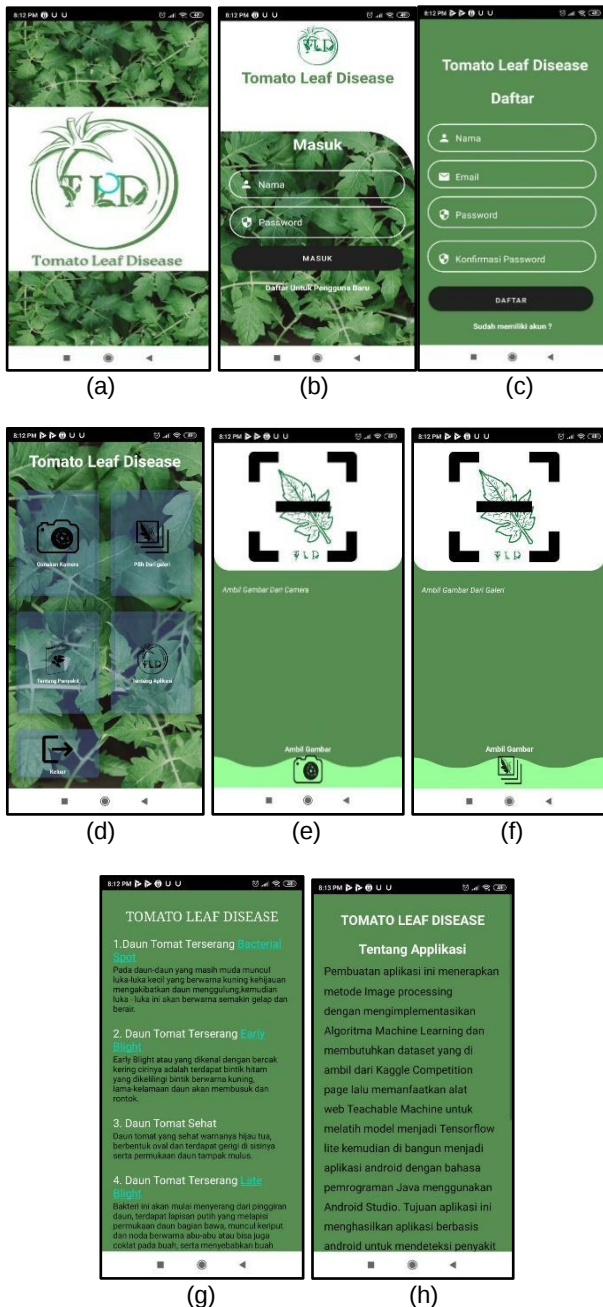


Gambar 3. Activity diagram

HASIL DAN PEMBAHASAN

Desain Aplikasi

Data-data yang dikumpulkan selanjutnya dianalisis dan dilakukan perancangan sistem yang berbasis Bahasa pemrograman Java dengan menggunakan software Android studio. Hasilnya dari perancangan ini berupa sebuah aplikasi berbasis android yang didesain secara menarik seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Hasil Desain : a) tampilan *splash screen*, b) tampilan halaman masuk, c) tampilan halaman daftar d) tampilan halaman *home*, e) tampilan menu gunakan camera, f) tampilan menu gunakan galeri, g) tampilan menu tentang penyakit, h) tampilan menu tentang aplikasi

Tampilan Deteksi

Pada tampilan deteksi terdapat dua pilihan menu yaitu kamera dan galeri. Kedua pilihan menu ini telah diuji coba untuk mendeteksi penyakit daun pada tanaman tomat. Menu kamera digunakan untuk mengambil gambar secara

langsung dengan memanfaatkan kamera perangkat dan menu galeri digunakan untuk mengambil gambar dari penyimpanan perangkat. Hasil uji coba kedua pilihan menu ditunjukkan pada Gambar 5.



Gambar 5. Tampilan deteksi : a) menggunakan kamera, b) menggunakan galeri

Implementasi

Pada tahap ini hasil program yang telah dianalisis dan dirancang kemudian diimplementasikan sehingga sistem siap untuk diuji lanjut.

Pengujian

Pengujian ini menggunakan metode *Black box testing*. Metode *Black box testing* ini berorientasi pada spesifikasi fungsional dari perangkat lunak (Djuanda and Dewi, 2020). Dari pengujian dapat didefinisikan kumpulan kondisi input dan melakukan pengujian terhadap spesifikasi fungsional dari aplikasi yang dibuat.

Pada pengujian lanjut ini digunakan tiga perangkat (*Handphone Android*) yang berbeda dan gambar uji diambil dari datasheet. Setiap pengujian digunakan sepuluh jenis gambar penyakit dan untuk setiap jenis penyakit dilakukan pengulangan beberapa kali. Pengulangan ini bertujuan untuk mengetahui kestabilan aplikasi dalam mendeteksi penyakit. Hasil pengujian ditunjuk pada Tabel 1-10 secara berturut-turut.

Tabel 1. Hasil pengujian jenis penyakit *septoria leaf spot*

Merek HP	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual kamera 12MP dan 5MP	<i>Septoria leaf spot</i> 50.1%, Tomat sehat 31.4%, <i>Late Blight</i> 16.1%	<i>Septoria leaf spot</i> 99.6%, <i>Bacterial spot</i> 0.4%	Benar
Galaxy A03	Kamera utama 48MP dan Depth 2MP	<i>Septoria leaf spot</i> 94.7%, <i>Bacterial spot</i> 2.7%	<i>Septoria leaf spot</i> 99.7%, <i>Bacterial spot</i> 0.3%	Benar
Redmi 10A	Kamera utama 50MP	<i>Septoria leaf spot</i> 99.1%, Tomat sehat 0.9%	<i>Septoria leaf spot</i> 99.9%, <i>Bacterial spot</i> 0.1%	Benar

Tabel 2. Hasil pengujian jenis penyakit *mosaic virus*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual kamera 12MP dan 5MP	<i>Mosaic virus</i> 94.2%, <i>Septoria Leaf spot</i> 5.1%, Target Spot 0.1%	<i>Mosaic virus</i> 100%	Benar
Galaxy A03	Kamera utama 48MP dan <i>Depth</i> 2MP	<i>Mosaic virus</i> 99.6%, <i>Leaf mold</i> 0.3%	<i>Mosaic virus</i> 100%	Benar
Redmi 10A	Kamera utama 50MP	<i>Mosaic virus</i> 100%	<i>Mosaic virus</i> 100%	Benar

Tabel 3. Hasil pengujian jenis penyakit *bacterial spot*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual kamera 12MP dan 5MP	<i>Bacterial spot</i> 100%	<i>Bacterial spot</i> 99.6%	Benar
Galaxy A03	Kamera utama 48MP dan <i>Depth</i> 2MP	<i>Bacterial spot</i> 100%	<i>Bacterial spot</i> 100%	Benar
Redmi 10A	Kamera utama 50MP	<i>Bacterial spot</i> 100%	<i>Bacterial spot</i> 100%	Benar

Tabel 4. Hasil pengujian tomat sehat

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual kamera 12MP dan 5MP	Tomat sehat 99.9%, Target spot 0.1%	Tomat sehat 100%	Benar
Galaxy A03	Kamera utama 48MP dan <i>Depth</i> 2MP	Tomat sehat 100%	Tomat sehat 100%	Benar
Redmi A10	Kamera utama 50MP	Tomat sehat 100%	Tomat sehat 100 %	Benar

Tabel 5. Hasil pengujian jenis penyakit *late blight*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual Kamera 12MP dan 5MP	<i>Late blight</i> 97.4%, <i>Bacterial spot</i> 2.6%	<i>Late blight</i> 100%	Benar
Galaxy A03	Kamera utama 48Mp dan <i>Depth</i> 2MP	<i>Late blight</i> 99.9%, <i>Target spot</i> 0.1%	<i>Late blight</i> 100%	Benar
Redmi A10	Kamera utama 50MP	<i>Late blight</i> 99.1%, <i>Leaf mold</i> 0.1%	<i>Late blight</i> 100%	Benar

Tabel 6. Hasil Pengujian jenis penyakit *leaf mold*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual kamera 12MP dan 5MP	<i>Leaf mold</i> 100%	<i>Leaf mold</i> 100%	Benar
Galaxy A03	Kamera utama 48MP dan <i>Depth</i> 2MP	<i>Leaf mold</i> 100%	<i>Leaf mold</i> 100%	Benar
Redmi A10	Kamera utama 50MP	<i>Leaf mold</i> 100%	<i>Leaf mold</i> 100%	Benar

Dari hasil pengujian yang ditunjukkan pada Tabel 1-10 diperoleh hasil bahwa secara keseluruhan aplikasi yang dibangun dapat mendeteksi jenis penyakit daun pada tanaman tomat dengan cukup akurat. Dari pengujian menggunakan kamera perangkat dapat dilihat bahwa terjadi 4 kali kesalahan deteksi pada jenis penyakit *Early blight*. Kesalahan ini terjadi pada ketiga perangkat uji (*Handphone*). Sistem tidak menunjukkan hasil deteksi yang benar dengan menunjukan hasil deteksinya merupakan penyakit *Leaf*

mold. Namun ketika menggunakan sumber penyimpanan perangkat (*Gallery*) sistem menampilkan hasil deteksi yang tepat terhadap jenis penyakit *Early Blight*. Nilai probabilitas pada beberapa percobaan juga mengalami variasi dan satu kesalahan pada pengujian jenis penyakit *Spider mites*. Hal ini disebabkan karena spesifikasi kamera, sudut pengambilan gambar dan pencahayaan saat pengambilan gambar.

Tabel 7. Hasil Pengujian penyakit *Early Blight*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual kamera 12MP dan 5 MP	<i>Leaf mold</i> 50%, <i>Early blight</i> 5.7%, <i>Light blight</i> 36.3%, <i>Bacterial spot</i> 8%	<i>Early blight</i> 94.6%, <i>Light blight</i> 5.4%	Salah
Galaxy A03	Kamera utama 48MP dan Depth 2MP	<i>Leaf mold</i> 96.3%, <i>Early blight</i> 3.7%	<i>Early blight</i> 94.9%, <i>Light blight</i> 5%	Salah
Redmi A10	Kamera utama 50MP	<i>Leaf mold</i> 100%	<i>Early blight</i> 96.4%, <i>Light blight</i> 3.5%	Salah

Tabel 8. Hasil pengujian jenis penyakit *Spider mites*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiomi Redmi 6 Pro	Dual kamera 12MP dan 5 MP	<i>Light blight</i> 98.8%, <i>Spider mites</i> 0.5%,Tomat sehat 0.7%	<i>Spider mites</i> 100%	Salah
Galaxy A03	Kamera utama 48MP dan Depth 2MP	<i>Spider mites</i> 85.7%, Tomat sehat 14.3%	<i>Spider mites</i> 100%	Benar
Redmi A10	Kamera utama 50MP	<i>Spider mites</i> 99.9%, <i>Leaf curl virus</i> 0.1%	<i>Spider mites</i> 100%	Benar

Tabel 9. Hasil pengujian jenis penyakit *Target Spot*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiomi Redmi 6 Pro	Dual kamera 12MP dan 5MP	<i>Target spot</i> 100%	<i>Target spot</i> 0.99% Tomat sehat 0.1%	Benar
Galaxy A03	Kamera utama 48MP dan Depth 2MP	<i>Target spot</i> 100%	<i>Target spot</i> 0.99% Tomat sehat 0.1%	Benar
Redmi A10	Kamera utama 50MP	<i>Target spot</i> 99.9% Tomat sehat 0.1%	<i>Target spot</i> 99.9% Tomat sehat 0.1%	Benar

Tabel 10. Hasil pengujian jenis penyakit *Yellow Leaf Curl Virus*

Merek Hp	Spesifikasi Kamera	Probabilitas Jenis Penyakit		Benar / Salah
		Kamera	Galeri	
Xiaomi Redmi 6 Pro	Dual Kamera 12MP dan 5MP	<i>Yellow leaf curl virus</i> 100%	<i>Yellow leaf curl virus</i> 100%	Benar
Galaxy A03	Kamera utama 48MP dan Depth 2MP	<i>Yellow leaf virus curl</i> 100%	<i>Yellow leaf curl virus</i> 100%	Benar
Redmi A10	Kamera utama 50MP	<i>Yellow leaf curl virus</i> 100%	<i>Yellow leaf curl virus</i> 100%	Benar

Ketika hasil yang ditampilkan sesuai maka user dapat memilih “Klik hasil untuk mengetahui lebih lanjut” dan sistem akan diarahkan ke halaman Google. Pilihan ini dapat digunakan bila user ingin mencari informasi terkait jenis penyakit. Sistem yang dibangun akan mengarahkan user secara langsung ke halaman jenis penyakit hasil deteksi. Pada aplikasi ini terdapat halaman tentang penyakit yang menyajikan penjelasan singkat mengenai sembilan jenis penyakit dan satu daun tomat sehat.

Pada aplikasi ini disediakan juga *link* untuk membawa pengguna pada web yang menyajikan penjelasan mengenai penyakit-penyakit tersebut agar user dapat mencari tahu penyebab dan cara penanganan penyakit tersebut. Bila dibandingkan dengan penelitian-penelitian sebelumnya perbedaan keakuratan sistem yang telah dibangun tidak signifikan. Namun keakuratan penelitian sebelumnya belum dapat dikatakan relevan, dikarenakan cara pengujian aplikasi penelitian terdahulu hanya menggunakan variasi gambar tidak mempertimbangkan spesifikasi kamera dan hanya memberikan informasi terkait penyakit tanpa memberikan solusi terkait penanganan penyakit.

KESIMPULAN

Setelah dilakukan pengujian dan analisis dapat disimpulkan bahwa aplikasi deteksi penyakit daun tomat berbasis android dengan bantuan model terlatih *Tensorflow lite* berhasil dibangun. Aplikasi dapat beroperasi untuk mendeteksi jenis penyakit daun dengan menggunakan gambar yang berasal dari galeri maupun tangkapan kamera perangkat. Agar dapat memperoleh hasil deteksi yang tepat maka perlu memperhatikan kualitas kamera, sudut pengambilan gambar. Sedangkan untuk gambar yang bersumber dari penyimpanan perangkat (*Gallery*) perlu untuk diperhatikan sumber gambar yang digunakan.

DAFTAR PUSTAKA

- Alviansyah, Fahri., Ruslianto, Ikhwan., & Diponegoro, Muhammad. (2017). Identifikasi Penyakit Pada Tanaman Tomat Berdasarkan Warna Dan Bentuk Daun Dengan Metode Naive Bayes Classifier Berbasis Web. *Jurnal Coding Sistem Komputer Untan* 5(1):23–32.
- Aqil, M., F. Tabri., N. N. Andayani., S. Panikkai., Suwardi., Roy, Efendi., Bunyamin, Z. M., Azrai., & Taufiq, Ratule. (2021). Integration of Smartphone Technology for Maize Recognition. *IOP Conference Series: Earth and Environmental Science* 911(1). doi: 10.1088/1755-1315/911/1/012037.
- Baihaqi, M. B., Yovi, Litanianda., & Triyanto, Andy. (2022). Implementasi Tensor Flow Lite Pada Teachable Untuk Identifikasi Tanaman Aglonema Berbasis Android. *Komputek* 6(1):70. doi: 10.24269/jkt.v6i1.1143.
- Darma Putra, I. Ketut Gede., I. Putu Deva Jayantha Putra., Rahmat Fauzi., & Deden Witasryah. (2020). Classification of Tomato Plants Diseases Using Convolutional Neural Network. *International Journal on Advanced Science, Engineering and Information Technology* 10(5):1821–27. doi: 10.18517/ijaseit.10.5.11665.
- Dhakshaya, S. S., & D. Jeraldin Auxillia. (2019). Classification of ECG Using Convolutional Neural Network (CNN). *2019 International Conference on Recent Advances in Energy-Efficient Computing and Communication, ICRAECC 2019* 19(2). doi: 10.1109/ICRAECC43874.2019.8995096.
- Djuanda, Ihsan., & Dewi, C. N. Puspita. (2020). Metode Rad Untuk Perancangan Sistem Informasi Perwira Tugas Belajar Bagi Pegawai Di Kementerian Pertahanan Republik Indonesia. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)* 1(2).
- Elwirehardja, Gregorius N., Suparyanto, Teddy., & Pardamean, Bens. (2023). Pengenalan Konsep Machine Learning Untuk Pemula. *Publisher : Instipr Press. ISBN: 978-623-5979-10-6*
- Faturrachman, Mirza., Indra, Yustia., & Somantri. (2022). Sistem Pendeteksi Penyakit Pada Daun Tanaman Singkong menggunakan Deep Learning dan Tensorflow Berbasis Android. *IJIS Indonesian Journal on Information System* 7(September).
- Fuentes, Alvaro., Sook, Yoon., Sang, C. Kim., & Dong, S. Park. (2017). A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* 17(9). doi: 10.3390/s17092022.
- Givari El Mirzaq, M. F. ., & Halilintar, R. (2021). Sistem Pakar Diagnosa Penyakit Tanaman Tomat Menggunakan Metode Certainty Factor. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 5(2), 230–235. <https://doi.org/10.29407/inotek.v5i2.1134>
- Gunawan, P. Fuyi., Andjarwirawan, Justinus., & Purbowo, A. Nathania. (2022). Penerapan Metode Convolutional Neural Network Untuk Clothing Image Recognition. *Jurnal Infra* 10(1).
- Hariyanto, D., Sastra, R., & Putri, F. E. P. E. P. (2021). Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan. *JUPITER (Jurnal Penelitian Ilmu Dan Teknik Komputer)*, 13(1), 110–117. Retrieved from <https://jurnal.polsri.ac.id/index.php/jupiter/article/view/3253>
- Jeong, Hyunja. (2020). Feasibility Study of Google's Teachable Machine in Diagnosis of Tooth-Marked Tongue. *Journal of Dental Hygiene Science* 20(4):206–12. doi: 10.17135/jdhs.2020.20.4.206.
- Julianto, Afis., Sunyoto, Andi., & Wibowo, F. Wahyu. (2022). Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi. *Teknimedia: Teknologi Informasi Dan Multimedia*, 3(2), 98-105. <https://doi.org/10.46764/teknimedia.v3i2.77>
- Kotta, Chrisno R., Debby, Paseru., & Michael, Sumampouw. (2022). Implementasi Metode Convolutional Neural Network Untuk Mendeteksi Penyakit Pada Citra Daun Tomat. *Jurnal Pekommas Vol. 7 No 2*:123–32.
- Peryanto, Ari., Yudhana, Anton., & Umar, Rusydi. (2020). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Format: Jurnal Ilmiah Teknik Informatika* 8(2):138. doi: 10.22441/format.2019.v8.i2.007.
- Putra, J. W. Gotama. (2020). Pengenalan Konsep Pembelajaran Mesin Dan Deep Learning Edisi 1.4 (17 Agustus 2020). 4:45–46.
- Recky, Arianto. (2023). Perancangan Aplikasi Pengenalan dan Pembelajaran Pertanian Milenial Hidroponik Berbasis Android Flutter & Dart Dengan Metode RAD . *OKTAL : Jurnal Ilmu Komputer dan Sains*, 2(03), 928–935. Retrieved from <https://journal.mediapublikasi.id/index.php/oktal/article/view/1103>
- Simbolon, H. F. S. ., Zulham Effendi, & Irwansyah. (2022). Rekognisi Pola Daun Gulma di Perkebunan Melalui Proses Pencitraan (*Image Processing*). *Jurnal Agro Estate*, 6(2), 66–76. <https://doi.org/10.47199/jae.v6i2.104>

- Taufiq, Arif, M. P., & Pratama, Ahmad R. (2021). Rancang Bangun Aplikasi Android 'Kuliah Apa?' Berbasis Flutter Dan TensorFlow Lite. *Automata* 2(1).
- T. Bayu Kurniawan, & Syarifuddin. (2020). Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman Pada Cafeteria No Caffe di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman Php dan Mysql. *Jurnal Tikar*, 1(2), 192-206. https://doi.org/10.51742/teknik_informatika.v1i2.153