

BAB II. TINJAUAN PUSTAKA

2.1. Tanaman dan Penyakit Daun Jeruk

Jeruk (*Citrus spp.*) merupakan komoditas hortikultura bernilai ekonomi cukup tinggi dan berperan besar bagi pendapatan petani di Indonesia (Putri Br Sitepu, Fajriani, and Sulistyono 2022). Budidayanya tersebar di banyak daerah karena permintaan buah yang tinggi, baik di pasar domestik maupun ekspor (Arzam et al. 2023). Keberhasilan produksi sangat ditentukan oleh kesehatan tanaman, terutama daun sebagai indikator utama proses fisiologis seperti fotosintesis dan metabolisme (Bandiyah Sri et al. 2023). Beberapa penyakit umum pada daun jeruk antara lain:

1. *Citrus Canker*, disebabkan oleh bakteri *Xanthomonas axonopodis*, yang menimbulkan luka cokelat dan bercak berair di permukaan daun.
2. *Greening (CVPD/Citrus Vein Phloem Degeneration)*, yang disebabkan oleh bakteri *Liberibacter* dan dibawa oleh serangga kutu daun, ditandai dengan gejala daun kecil, kuning, dan pertumbuhan tanaman yang terhambat.
3. Bercak Daun *Alternaria*, akibat jamur *Alternaria citri*, yang menyebabkan bercak hitam atau cokelat berbentuk bulat pada daun.



Gambar 2.1 Beberapa Contoh Kondisi Pada Daun Jeruk

(Sumber: PemrogramanMatlab.com, 2023)

2.2. Pengolahan Citra Digital dalam Pertanian

Pengolahan citra digital adalah teknik memproses gambar dalam format digital dengan tujuan memperbaiki kualitas visual sekaligus menggali informasi yang terkandung di dalamnya, teknik ini banyak dipakai untuk mendukung analisis otomatis karena dapat mengonversi data visual menjadi informasi yang siap diolah oleh sistem komputer.

Di sektor pertanian, teknik ini dimanfaatkan untuk mengenali kondisi kesehatan tanaman lewat analisis perubahan visual pada daun akibat serangan penyakit berupa perbedaan warna, tekstur, atau bentuk yang menjadi penanda adanya infeksi patogen (Ashillah et al. 2025).

2.3. Artificial Intelligence

Artificial Intelligence (AI) atau kecerdasan buatan adalah cabang ilmu komputer yang mengembangkan sistem dengan kemampuan menyerupai kognisi manusia, seperti belajar, mengenali pola, dan mengambil keputusan secara otomatis. Karena mampu mengolah data berskala besar menjadi prediksi atau klasifikasi yang akurat, AI kini banyak diterapkan di berbagai sektor, termasuk pertanian (Marsella et al. 2023) (Simatupang et al., 2025).

Pendekatan dalam AI mencakup *machine learning* dan *deep learning* yang mempelajari pola dari data latih tanpa pemrograman eksplisit. Dalam penelitian ini, AI dimanfaatkan untuk menganalisis citra daun jeruk dan mengenali gejala penyakit secara otomatis, sehingga diagnosis menjadi lebih cepat, objektif, dan efisien daripada pengamatan manual (Saputra 2025).

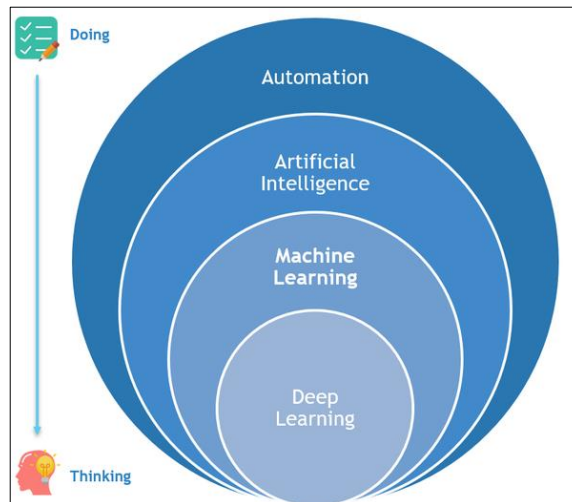
2.4. Machine Learning

Sebagai bagian dari AI, *Machine Learning* berfokus pada pengembangan sistem yang dapat belajar dari data dan meningkatkan kinerjanya sendiri tanpa pemrograman eksplisit. Cara kerjanya adalah membangun model berdasarkan pola atau hubungan yang ditemukan pada data pelatihan, sehingga sistem tersebut bisa memprediksi atau mengklasifikasikan data baru (Trisno, Raharja, and Kartika 2023).

Secara garis besar, pendekatan dalam *machine learning* terbagi menjadi *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Dari ketiganya, *supervised learning* paling sering dipakai untuk tugas klasifikasi citra karena mengandalkan data berlabel (Nurhalizah, Ardianto, and Purwono 2024). Pada pengolahan citra, pendekatan ini digunakan untuk secara otomatis mengenali karakteristik visual seperti warna, bentuk, dan tekstur objek (Riyadi et al. 2025).

2.5. Deep Learning

Deep Learning merupakan pengembangan dari *machine learning* yang memanfaatkan jaringan saraf tiruan berlapis untuk mempelajari pola dan representasi data secara otomatis tanpa memerlukan perancangan fitur secara manual. Metode ini mampu mengolah data dalam jumlah besar serta mengenali hubungan yang kompleks sehingga menghasilkan performa prediksi dan klasifikasi yang lebih akurat dibandingkan pendekatan konvensional. Karena kemampuannya, *deep learning* banyak diterapkan dalam berbagai bidang salah satunya analisis citra digital (Jamiah Nurhakiki and Yahfizham Yahfizham 2024).



Gambar 2.2 Diagram Hubungan Hierarki Antara *Artificial Intelligence*, *Machine Learning* dan *Deep Learning*

(Sumber: AbdUmar.com, 2021)

2.6. Convolutional Neural Network

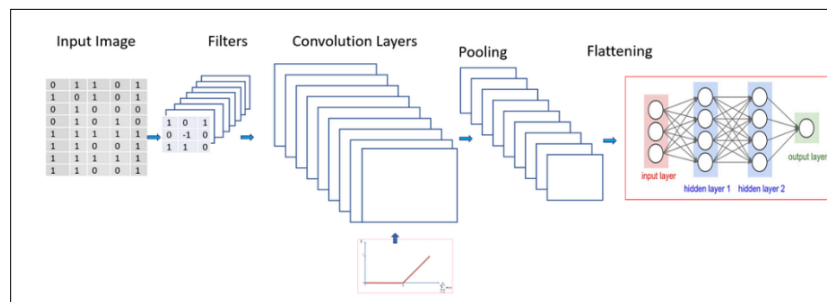
Convolutional Neural Network (CNN) ialah metode *deep learning* yang dikembangkan untuk memproses data citra. Model ini mempelajari pola visual seperti tepi, tekstur, dan bentuk secara otomatis dari data latih tanpa perancangan fitur manual, sehingga klasifikasinya lebih akurat daripada metode tradisional (Gilang Perwati, Suarna, and Suprpti 2024).

CNN diturunkan dari *Artificial Neural Network* (ANN) dan lebih efisien untuk data dua dimensi karena menggunakan operasi konvolusi dan *pooling* yang mengurangi jumlah parameter sekaligus menjaga informasi spasial citra. Pada penelitian ini, CNN digunakan untuk mempelajari karakteristik visual daun jeruk sebagai dasar deteksi penyakit pada aplikasi *mobile* (Fadllullah and Erdina 2025).

Secara umum, CNN bekerja memproses citra secara bertahap dimulai dari ekstraksi fitur menggunakan konvolusi, pengurangan dimensi melalui aktivasi dan *pooling*, hingga klasifikasi dengan *fully connected layer*, sehingga sistem dapat mengenali dan menentukan kelas objek secara otomatis (I. Maulana, Khairunisa, and Mufidah 2023). Adapun tahapan utama proses CNN adalah sebagai berikut:

- *Input Image*, citra dimasukkan sebagai data awal dalam bentuk matriks piksel.
- *Filters/Kernel*, filter digunakan untuk mendeteksi pola seperti tepi, garis, dan tekstur pada citra.

- *Convolution Layer (Feature Map)*, proses konvolusi menghasilkan *feature map* sebagai representasi fitur penting.
- *Activation (ReLU)*, menambahkan non-linearitas agar model mampu mempelajari pola kompleks.
- *Pooling*, mengurangi dimensi data untuk efisiensi komputasi dan mencegah *overfitting*.
- *Flattening*, mengubah fitur dua dimensi menjadi vektor satu dimensi.
- *Fully Connected & Training*, fitur diklasifikasikan dan parameter diperbarui melalui *backpropagation*.



Gambar 2.3 Konsep CNN

(Sumber Medium.com, 2024)

2.7. Arsitektur MobileNetV2

MobileNetV2 merupakan arsitektur dirancang khusus sebagai arsitektur CNN untuk perangkat mobile yang terbatas dari segi memori dan komputasi, sehingga menghasilkan model yang ringan dan cepat tanpa mengorbankan akurasi klasifikasi citra. Efisiensi perhitungannya dicapai dengan menekan jumlah parameter dan operasi konvolusi tanpa menurunkan kualitas representasi fitur, menjadikannya pilihan yang sesuai untuk deteksi penyakit secara *real-time* pada aplikasi Android (Hakiki and Rachmat 2025).

Di luar *accuracy*, proses pelatihan model juga memakai *loss function* untuk mengukur seberapa besar kesalahan prediksi. Semakin kecil nilainya, semakin baik kemampuan model dalam mempelajari pola pada data. Penelitian ini memilih fungsi *Categorical Crossentropy* karena klasifikasi dilakukan terhadap lima kelas citra daun jeruk, di mana fungsi ini menghitung selisih antara hasil prediksi dan label sebenarnya (Sutarti and Fariza Syaqqialloh 2025).

$$Loss = - \sum_{i=1}^C y_i \log(\hat{y}_i)$$

Gambar 2.4 Fungsi *Categorical Crossentropy*

(Sumber: Medium.com, 2023)

Keterangan:

- C = jumlah kelas.
- y_i = label sebenarnya pada kelas ke- i .
- $\hat{y}_i$ = probabilitas hasil prediksi model pada kelas ke- i .

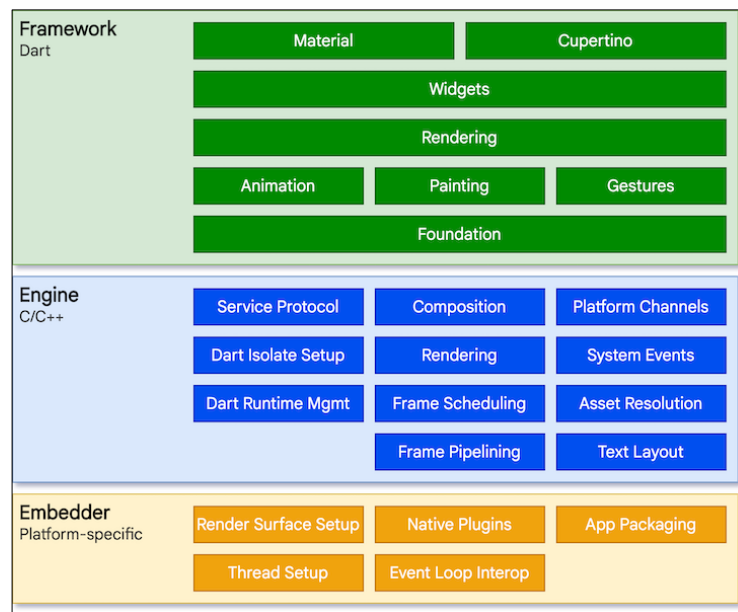
2.8. Flutter Untuk Pengembangan Aplikasi Mobile

Flutter adalah *framework* lintas platform buatan Google yang memungkinkan aplikasi Android dan iOS dibangun dari satu basis kode menggunakan bahasa Dart. Karena antarmukanya dirender langsung lewat Skia Graphics Engine, *framework* ini menghasilkan performa tinggi, tampilan yang konsisten, dan responsivitas yang baik di berbagai perangkat (Fau 2024).

Pada aplikasi berbasis citra seperti sistem deteksi penyakit daun ini, Flutter menawarkan sejumlah *plugin* penting seperti integrasi kamera, penyimpanan lokal, dan komunikasi API guna mengirim serta menerima data hasil klasifikasi. Fitur *hot reload* turut mempercepat siklus pengembangan dan pengujian antarmuka, sehingga waktu implementasi menjadi lebih singkat (AL-atraqchi, 2022). Ada 2 alasan pemilihan Flutter pada penelitian ini dijabarkan sebagai berikut:

1. Integrasi Kamera, mendukung pengambilan citra secara langsung melalui perangkat *mobile*.
2. Kinerja Ringan, berjalan optimal pada perangkat Android kelas menengah sehingga sesuai untuk penggunaan di lapangan.

Sebagaimana aplikasi *native* pada umumnya, arsitektur Flutter tersusun atas tiga lapisan, yaitu *Embedder* di lapisan paling bawah, *Engine* di tengah, dan *Framework* di lapisan teratas (Irfankha 2025).



Gambar 2.5 Layer Pada Flutter

(Sumber: gepcode.com, 2021)

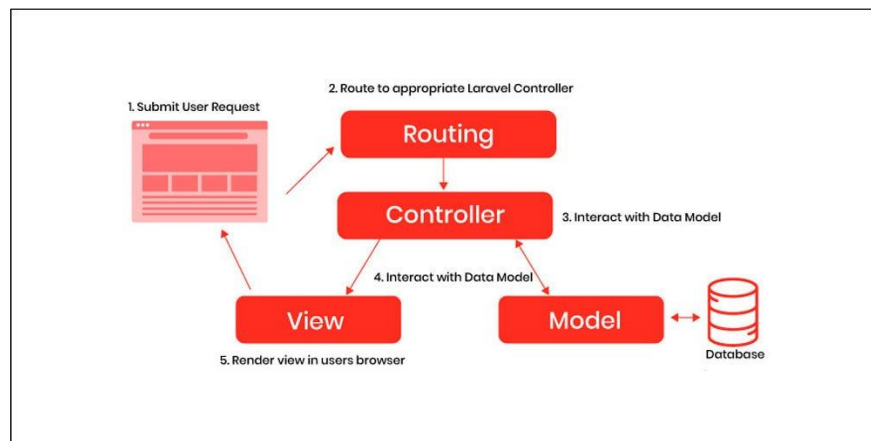
2.9. Laravel Sebagai Backend

Laravel adalah *framework* berbasis PHP untuk pengembangan web yang mengusung pola *Model, View* dan *Controller* (MVC), pertama kali diluncurkan oleh Taylor Otwell pada 2011. Struktur pengembangannya yang terorganisasi, dilengkapi fitur modern seperti *routing, middleware*, ORM Eloquent, validasi data, dan autentikasi pengguna, membuat pembangunan *backend* menjadi lebih aman dan mudah dikelola (Zaman 2025).

Dalam pengembangan Riwayat pendukung deteksi penyakit berbasis citra, Laravel berperan sebagai *backend* yang menangani

1. Pengelolaan akun pengguna dan autentikasi.
2. Penyimpanan gambar hasil deteksi yang diunggah melalui aplikasi *mobile*.
3. Penyimpanan dan pengelolaan riwayat hasil klasifikasi (label serta nilai *confidence*) yang dikirim dari aplikasi.
4. Penyediaan layanan API pendukung lainnya, seperti pengelolaan data profil, riwayat deteksi, dan fitur tambahan lainnya.

Laravel juga memudahkan pembuatan API yang dapat diintegrasikan dengan *frontend* seperti Flutter, sehingga komunikasi antara *client* dan *server* berjalan efisien dan aman sekaligus mendukung skalabilitas aplikasi ke depannya (Putra et al. 2025).



Gambar 2.6 MVC Pada Laravel

(Sumber: Monocubed.com, 2025)

2.10. Basis Data

Basis data adalah kumpulan data yang disusun secara sistematis di dalam sistem komputer agar mudah diakses, dikelola, dan diperbarui. Dalam pengembangan aplikasi *web* maupun *mobile*, ia berperan sebagai tempat penyimpanan seluruh informasi sistem, mulai dari data pengguna hingga data transaksional. Pada model relasional, data disimpan dalam tabel-tabel yang saling terhubung lewat *primary key* sebagai identitas unik tiap baris, serta *foreign key* yang menjembatani antar tabel (Hascarjani 2025).

2.11. Integrasi Model CNN, Laravel, Flutter dan Basis Data

Integrasi antara aplikasi *mobile*, *backend server*, dan model *deep learning* diperlukan untuk membangun sistem deteksi penyakit daun yang terkoordinasi dan berjalan secara otomatis. Pada penelitian ini, Flutter berperan sebagai *frontend* yang menangkap citra daun dan menampilkan hasil deteksi, Laravel menjadi *backend* yang mengelola komunikasi data dan penyimpanan, sementara model CNN berbasis MobileNetV2 menjalankan tugas klasifikasi sehingga sistem dapat memberi hasil prediksi secara *real-time* (Edoazani et al. 2025).

Tabel 2.1 Integrasi Model CNN, *Laravel*, *Flutter* dan Basis Data

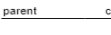
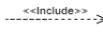
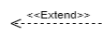
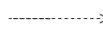
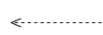
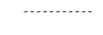
No.	Komponen	Peran	Fungsi Utama
1.	Model CNN (MobileNetV2)	Mesin klasifikasi	Melakukan inferensi, klasifikasi dan menentukan jenis penyakit
2.	Laravel (<i>Backend</i>)	Server API	Menerima data,

			mengelola penyimpanan, dan menghubungkan model
3.	Flutter (<i>Frontend</i>)	Antarmuka pengguna	Mengambil gambar daun dan menampilkan hasil
4.	Basis Data	Penyimpanan data	Menyimpan data pengguna

2.12. Unified Modeling Language (UML)

2.12.1. Use Case Diagram

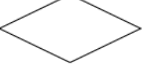
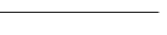
Use Case Diagram berfungsi menggambarkan bagaimana aktor berinteraksi dengan sistem sekaligus menunjukkan fungsi-fungsi yang tersedia di dalamnya, sehingga membantu menetapkan kebutuhan fungsional dan cakupan sistem yang akan dibangun (Simamora, Dalimunthe, and Purnama 2026).

Simbol	Nama	Keterangan
 Actor	Actor	Entitas yang berinteraksi dengan sistem
 Use Case	Use Case	Aktivitas yang dapat dilakukan actor pada sistem
 parent child	Assosiation	Hubungan antara actor dengan use case
	System	Sistem yang sedang dikembangkan
	Include	Suatu use case termasuk bagian dari use case lain
	Extend	Satu use case dapat diperluas dengan use case lain
	Dependency	Ketergantungan antar elemen-elemen diagram
	Generalization	Satu actor atau use case merupakan generalisasi dari yang lain
	Realization	Implementasi dari satu use case oleh yang lain
	Collaboration	Dua atau lebih actor dan use case yang terhubung
	Note	Penjelasan tambahan terkait elemen-elemen diagram
	Anchor	Hubungan teks note dengan elemen-elemen diagram lain

Gambar 2.7 Use Case Diagram

2.12.2. Activity Diagram

Activity diagram memodelkan alur kerja atau rangkaian aktivitas dalam sistem, yaitu urutan langkah yang dijalankan pengguna maupun sistem, sehingga alur proses lebih mudah dipahami (Widyatmoko and Pamungkas 2022).

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan/ Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	Penggabungan/ Join	Yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
	Status Akhir	Akhir dari suatu sistem
	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2.8 Activity Diagram

(Sumber: Dicoding.com, 2021)

2.12.3. Diagram Alur Sistem (Flowchart)

Flowchart adalah diagram visual yang menggambarkan urutan proses, aliran data, dan hubungan antar komponen dalam suatu sistem, sehingga cara kerja sistem secara keseluruhan lebih mudah dipahami. Diagram ini memaparkan langkah operasional dari input hingga *output* secara terstruktur, yang pada gilirannya mempermudah proses analisis, perancangan, dan implementasi perangkat lunak (Noija et al. 2023).

No	Simbol	Bentuk	Keterangan
1	Terminal		Untuk menyatakan awal atau akhir suatu program
2	Input / Output		Menunjukkan operasi masukan atau operasi keluaran
3	Proses		Menunjukkan proses pengolahan data
4	Keputusan		Untuk menyatakan keputusan dari pilihan berdasarkan kondisi tertentu
5	Persiapan		Memberikan konstanta atau nilai awal pada variabel
6	Proses terdefinisi		Menunjukkan proses yang detail proses ini dijelaskan terpisah
7	Penghubung		Untuk menghubungkan bagian diagram alir pada halaman lain
8	Penghubung		Untuk menghubungkan bagian diagram alir dalam satu halaman
9	Arah		Digunakan untuk menunjukkan arah aliran proses

Gambar 2.9 Flowchart

(Sumber: Dicoding.com, 2021)

2.13. Penelitian Terdahulu

Merujuk Tabel 2.2, kelima penelitian terdahulu memperlihatkan tiga celah utama yang belum terjawab secara bersamaan. Dari sisi mode inferensi, (Riswandi et al. 2021) masih bergantung pada proses *cloud* (Google Colab) dengan cakupan hanya 3 kelas, sehingga belum dapat berjalan *offline* penuh. Dari sisi akurasi, (Kurniawati, Wulanningrum, and Sahertian 2023) hanya mencapai 94%, sementara (Gilang, Khaira, and Hutabarat 2025) mencapai sekitar 98% namun objek deteksinya adalah kulit buah, bukan daun. Dari sisi dataset dan implementasi, (Soleha, Hidayat, and Zulus 2025) hanya menggunakan 594 citra dan tidak seimbang antar kelas, sedangkan (Acarya, Muhaimin, and Hindrayani 2024) menggunakan arsitektur EfficientNetB3 yang lebih berat dari MobileNetV2; keduanya masih berhenti pada tahap eksperimen model dan belum diwujudkan menjadi aplikasi *mobile* yang utuh.

Penelitian ini berupaya menutup ketiga celah tersebut sekaligus. Model diinferensikan sepenuhnya secara *on-device* melalui *TensorFlow Lite* sehingga tidak bergantung pada *cloud*, dengan cakupan 5 kelas kondisi daun jeruk. Model dilatih menggunakan dataset *Orange Leaf Disease* sebanyak 38.432 citra dengan rasio 80:10:10 untuk data *training*, *validation*, dan *testing*, jauh lebih besar dibandingkan penelitian (Soleha, Hidayat, and Zulus 2025), dan menghasilkan akurasi 99,87%, melampaui (Kurniawati, Wulanningrum, and Sahertian 2023). Model tersebut juga telah diintegrasikan penuh ke dalam aplikasi *mobile* berbasis Flutter dan Laravel, serta teruji secara fungsional melalui *Black Box Testing*, sehingga penelitian ini menghasilkan sistem yang siap digunakan, bukan sekadar eksperimen model seperti pada penelitian (Acarya, Muhaimin, and Hindrayani 2024)

Tabel 2.2 Penelitian Terdahulu

No.	Peneliti dan Tahun	Judul	Fitur	Keterbatasan
1.	(Riswandi et al. 2021)	Klasifikasi Penyakit Pada Citra Daun Jeruk Menggunakan Arsitektur MobileNet Berbasis Mobile Platform	MobileNet, tiga kelas proses lewat <i>cloud</i> API	Hanya terdapat tiga kelas, bergantung <i>cloud</i> (bukan <i>offline</i> penuh)
2.	(Kurniawati, Wulanningrum and Sahertian 2023)	Perancangan Sistem Deteksi Penyakit Daun Jeruk Siam	CNN, 3 kelas, epoch 35	Akurasi hanya 94%, kelas terbatas, belum diimplementasi jadi

				aplikasi <i>mobile</i> utuh
3.	(Acarya, Muhaimin, and Hindrayani 2024)	Identifikasi Penyakit Daun Jeruk Siam Menggunakan <i>Convolutional Neural Network</i> (CNN) dengan Arsitektur EfficientNet	EfficientNet B3, 6 kelas, dataset asli dari kebun	Model lebih berat dari MobileNetV2, belum ada implementasi aplikasi Android/iOS
4.	(Soleha, Hidayat, and Zulius 2025)	<i>Hyperparameter</i> Tuning Dalam Deteksi Penyakit Daun Jeruk Dengan Pendekatan Transfer Learning	Sudah terintegrasi MobileNetV2 tuning hyperparameter, 5 kelas, dataset Kaggle + augmentasi system	Dataset kecil (594 gambar) tidak seimbang, hanya eksperimen model, belum jadi aplikasi <i>mobile</i>
5.	(Gilang, Khaira, and Hutabarat 2025)	Rancang Bangun Aplikasi Android untuk Klasifikasi Penyakit Tanaman Jeruk Menggunakan CNN dan Metode <i>Extreme Programming</i>	Akurasi MobileNetV2 TFLite, metode XP, tampilkan rekomendasi penanganan tinggi (~98%)	Objek deteksi kulit buah (bukan daun), akurasi dibawah 99%