

TUGAS AKHIR

**IMPLEMENTASI APLIKASI MOBILE UNTUK KLASIFIKASI
JENIS DAUN ALPUKAT MENGGUNAKAN MOBILENETV3**



**RISNA AMANDA
213510221**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS ISLAM RIAU
PEKANBARU
2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir berjudul “Implementasi Aplikasi Mobile untuk Klasifikasi Jenis Daun Alpukat Menggunakan *MobileNetV3*” dengan baik dan lancar.

Tugas akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan program studi pada jenjang Strata 1. Penelitian ini bertujuan mengembangkan sistem klasifikasi jenis tanaman alpukat berdasarkan citra daun dengan memanfaatkan arsitektur *MobileNet-V3*, yang dikenal memiliki efisiensi tinggi dan kebutuhan komputasi yang ringan.

Penulis menyadari bahwa penyelesaian tugas akhir ini tidak terlepas dari bantuan, arahan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Ir. Kurnia Hastuti, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Islam Riau.
2. Ibu Dr. Ir. Des Suryani, M.Sc selaku Ketua Program Studi Teknik Informatika Universitas Islam Riau.
3. Bapak Anggi Hanafiah, S.Kom., M.Kom selaku Sekretaris Program Studi Teknik Informatika.
4. Ibu Ana Yulianti, S.T., M.Kom selaku Dosen Pembimbing Penulis yang telah memberikan ilmu, arahan, dan telah sabar dalam memberikan bimbingan di sela-sela kesibukan beliau dalam penyelesaian skripsi ini.
5. Bapak Rizdqi Akbar Ramadhan, S.Kom., M.Kom sebagai pembimbing akademik yang senantiasa memberikan nasihat, dukungan, dan semangat selama masa studi di Program Studi Teknik Informatika.
6. Saya mengucapkan terima kasih kepada seluruh dosen Teknik Informatika yang telah memberikan bimbingan dan ilmu pengetahuan selama masa perkuliahan.
7. Kepada kedua orang tua tercinta, Sukatno dan Ismiati, yang dengan kasih sayang, ketulusan, pengorbanan, serta doa yang tidak pernah putus telah

menjadi sumber kekuatan dan penyemangat utama bagi penulis dalam setiap langkah perjalanan akademik. Kepada adik penulis, Tiara Dwi Lestari, yang selalu memberikan keceriaan, dorongan, dan semangat sehingga proses ini dapat dilalui dengan lebih ringan. Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

8. Ucapan terima kasih penulis sampaikan kepada pasangan penulis, Dwi Aryo Sentanu, yang senantiasa memberikan ketenangan, dorongan, serta ketulusan dalam menemani setiap proses dan tantangan selama penulis menyelesaikan tugas akhir ini. Dukungan beliau menjadi kekuatan berarti dalam menyelesaikan studi ini.
9. Terima kasih kepada sepupu penulis, Jeni Andriani, Adinda Rizki Feisha, dan Balqis Maipasha yang telah menemani dan mendukung penulis di masa-masa akhir penyelesaian skripsi ini. Ucapan terima kasih juga penulis sampaikan kepada seluruh sepupu, pakde, bude, paman, dan bibi lainnya yang tidak dapat disebutkan satu per satu, namun dukungan dan doa sangat penting dalam proses ini.
10. Sahabat-sahabat penulis, Mita Maiwisa dan Syahra Eliza, yang sejak lama menjadi teman perjalanan, selalu memberikan dukungan, kebersamaan, serta semangat dalam berbagai keadaan, turut menjadi bagian berharga dalam penyelesaian tugas akhir ini.

Akhir kata, penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya dalam bidang kecerdasan buatan dan teknologi pertanian digital.

Penulis

(Risna Amanda)

ABSTRAK

Perkembangan teknologi pengolahan citra dan *deep learning* memberikan peluang besar dalam mendukung identifikasi tanaman secara otomatis. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi jenis daun alpukat menggunakan arsitektur MobileNetV3 serta mengimplementasikannya pada aplikasi mobile berbasis Flutter. Data primer terdiri dari 900 citra daun alpukat yang berasal dari lima kelas, yaitu Aligator, Madu, Kendil, Super, dan SW01. Dataset dibagi menjadi 80% data latih, 10% data validasi, dan 10% data uji. Proses augmentasi diterapkan hanya pada data latih dengan lima variasi augmentasi untuk setiap citra sehingga setiap kelas menjadi 720 citra dan total data latih bertambah menjadi 3.600 citra. Seluruh citra dinormalisasi dan diubah ukurannya menjadi 224×224 piksel agar sesuai dengan kebutuhan input MobileNetV3. Model dilatih menggunakan pendekatan *transfer learning* dengan *optimizer* Adam, *learning rate* 0.0001, dan *loss function* Categorical Crossentropy. Hasil pengujian menunjukkan bahwa MobileNetV3-Large memberikan performa terbaik dengan akurasi 97%, sedangkan MobileNetV3-Small menghasilkan akurasi 90%. Berdasarkan akurasi tersebut, MobileNetV3-Large dipilih sebagai model utama yang diintegrasikan ke dalam aplikasi mobile karena mampu mengklasifikasikan jenis daun alpukat secara lebih efektif dan andal. Implementasi model ke aplikasi memungkinkan pengguna untuk melakukan deteksi secara real-time melalui kamera atau unggahan gambar. Secara keseluruhan, penelitian ini berhasil menghasilkan model klasifikasi jenis daun alpukat yang akurat, efisien, dan siap digunakan pada perangkat mobile. Sistem ini diharapkan dapat membantu masyarakat, petani, serta pelaku usaha tanaman dalam mengidentifikasi varietas alpukat dengan cepat dan mudah, serta menjadi dasar bagi pengembangan sistem klasifikasi tanaman lain di masa mendatang.

Kata kunci: MobileNetV3, klasifikasi daun alpukat, deep learning, augmentasi data, aplikasi mobile.

ABSTRACT

The advancement of image processing and deep learning technologies offers significant potential for developing automated plant identification systems. This study aims to develop a classification system for avocado leaf types using the MobileNetV3 architecture and implement it in a mobile application built with Flutter. The primary dataset consists of 900 images of avocado leaves representing five classes: Aligator, Madu, Kendil, Super, and SW01. The dataset was divided into 80% training data, 10% validation data, and 10% testing data. Data augmentation was applied exclusively to the training set using five augmentation techniques, resulting in 720 images per class and increasing the total number of training images to 3,600. All images were normalized and resized to 224×224 pixels to match the input requirements of the MobileNetV3 model. The model was trained using a transfer learning approach with the Adam optimizer, a learning rate of 0.0001, and the Categorical Crossentropy loss function. Experimental results show that MobileNetV3-Large achieved the highest performance with an accuracy of 97%, while MobileNetV3-Small obtained an accuracy of 90%. Based on these results, MobileNetV3-Large was selected as the primary model for integration into the mobile application due to its superior classification performance and reliability. The implemented system enables real-time detection through the device camera or uploaded images. Overall, this study successfully produced an accurate, efficient, and mobile-ready avocado leaf classification model. The system is expected to assist users, farmers, and agricultural practitioners in identifying avocado varieties quickly and conveniently, as well as serve as a foundation for developing similar plant classification systems in the future.

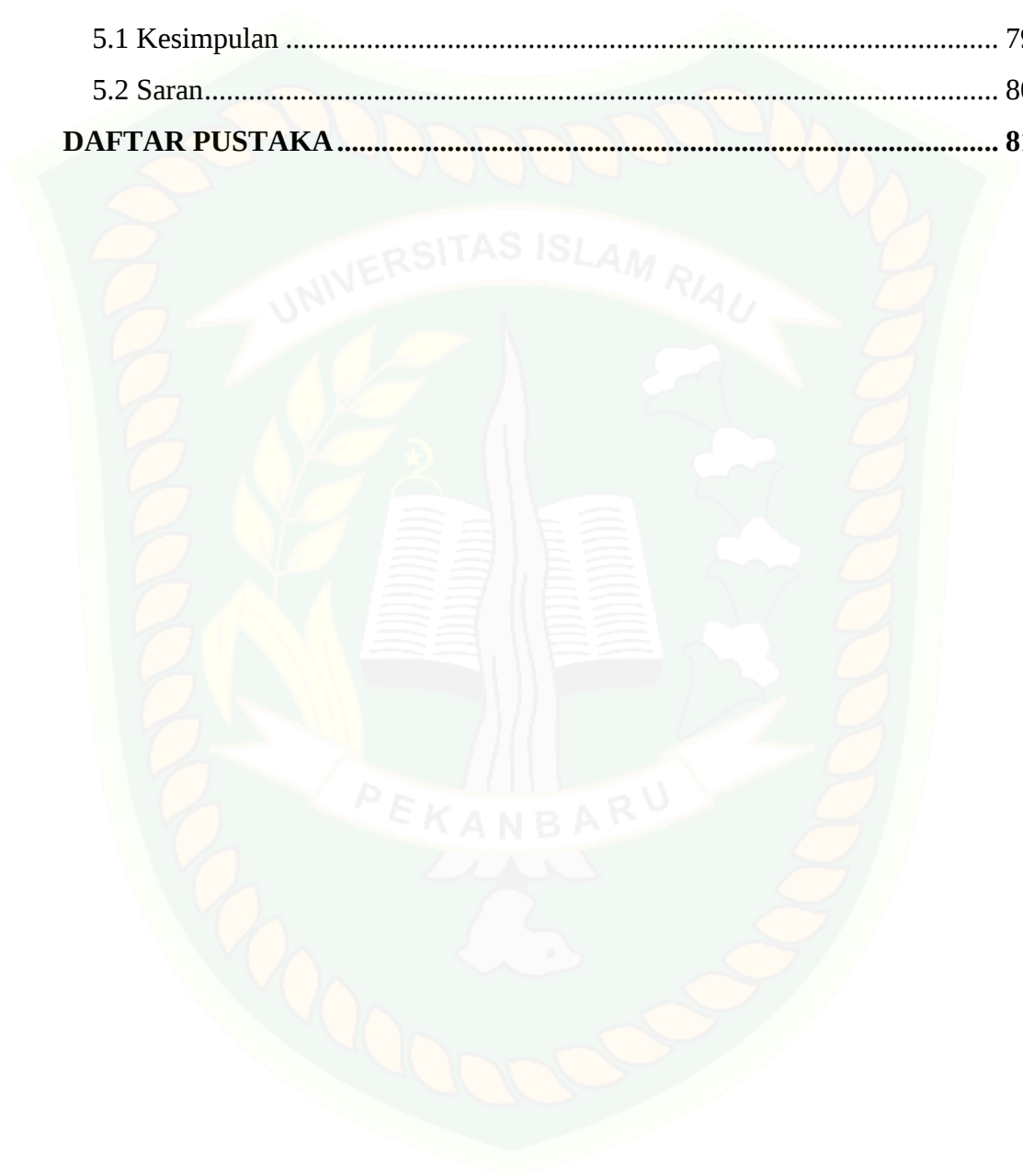
Keywords: MobileNetV3, avocado leaf classification, deep learning, data augmentation, mobile application.

DAFTAR ISI

KATA PENGANTAR.....	i
ABSTRAK	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	3
1.4 Batasan Masalah.....	4
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	4
BAB II LANDASAN TEORI	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	10
2.2.1 Tanaman Alpukat	10
2.2.2 Jenis Alpukat	11
2.2.3 Tahapan Pengolahan Citra Digital	13
2.2.4 Klasifikasi	18
2.2.5 <i>Convolutional Neural Network</i> (CNN)	19
2.2.6 MobileNet-V3	22
2.2.7 Python	24
2.2.8 Google Colab	25
2.2.9 <i>Flask</i>	26
2.2.10 SQLite	27
2.2.11 Flutter	27
2.2.12 Pengujian dengan Menggunakan <i>Confusion Matrix</i>	28
2.3 Kerangka Pemikiran.....	30
BAB III METODOLOGI PENELITIAN	33
3.1 Analisa Sistem yang Sedang Berjalan.....	33

3.2 Metode Penelitian.....	35
3.2.1 Akuisisi Citra	36
3.2.2 Citra <i>Resize</i>	37
3.2.3 Augmentasi	38
3.2.4 Normalisasi Pixel	41
3.2.5 Melatih Model <i>MobileNetV3</i>	41
3.2.6 Evaluasi	44
3.3 Analisa Kebutuhan Sistem	45
3.3.1 Spesifikasi Kebutuhan <i>Hardware Dan Software</i>	45
3.4 Pengembangan Sistem	46
3.5 Perancangan Sistem	47
3.5.1 Arsitektur Sistem.....	47
3.5.2 Diagram Konteks	48
3.5.3 <i>Hierarchy Chart</i>	48
3.5.4 <i>Data Flow Diagram (DFD)</i>	49
3.5.5 Desain Database	51
3.5.6 Perancangan Antarmuka	52
3.5.7 Desain Logika Program.....	54
BAB IV HASIL DAN PEMBAHASAN	56
4.1 Akuisisi Citra.....	56
4.2 Hasil Pelatihan Model	58
4.2.1 Hasil Pelatihan Model <i>MobileNetV3Large</i>	58
4.2.2 Hasil Pelatihan Model <i>MobileNetV3Small</i>	60
4.2.3 Analisis Pelatihan Model	62
4.2.4 Evaluasi Metrix.....	65
4.2.5 Analisis Confusion Matrix	66
4.3 Pengujian Black Box	69
4.3.1 Tampilan Halaman Utama Aplikasi.....	69
4.3.2 Tampilan Halaman Riwayat.....	70
4.3.3 Tampilan Halaman Pengaturan	71
4.4 Hasil Prediksi Sistem.....	72
4.4.1 Prediksi Alpukat Alligator	72

4.4.2	Prediksi Alpukat Kendil.....	73
4.4.3	Prediksi Alpukat Madu	74
4.4.4	Prediksi Alpukat Super	75
4.4.5	Prediksi Alpukat SW01.....	76
BAB V KESIMPULAN DAN SARAN		79
5.1	Kesimpulan	79
5.2	Saran.....	80
DAFTAR PUSTAKA		81



DAFTAR GAMBAR

Gambar 2. 1 Tanaman Alpukat	11
Gambar 2. 2 Pengolahan Citra Digital	14
Gambar 2. 3 Proses Akuisisi Citra	15
Gambar 2. 4 Perbedaan Sebelum Dan Sesudah Resizing	16
Gambar 2. 5 Contoh Dari Augmentasi Data	17
Gambar 2. 6 Proses Convolution Neural Network.....	20
Gambar 2. 7 Gambaran Operasi Konvolusi	21
Gambar 2. 8 Pooling Layer	21
Gambar 2. 9 Fully Connected Layer	22
Gambar 2. 10 Arsitektur MobileNetV3 Large	23
Gambar 2. 11 Arsitektur MobileNetV3 Small	24
Gambar 2. 12 Logo Python	25
Gambar 2. 13 Logo Google Colab	26
Gambar 2. 14 Logo <i>Flask</i>	27
Gambar 2. 15 Logo SQLite.....	27
Gambar 2. 16 Logo Flutter.....	28
Gambar 2. 17 Confusion Matrix	29
Gambar 2. 18 Kerangka Pemikiran.....	32
Gambar 4. 1 Hasil Pengumpulan Data Primer	56
Gambar 4. 2 Hasil Data Split	57
Gambar 4. 3 Hasil Grafik <i>MobileNetV3Large</i>	60
Gambar 4. 4 Hasil Grafik <i>MobileNetV3Small</i>	61
Gambar 4. 5 Hasil Metrix <i>MobileNetV3Large</i>	66
Gambar 4. 6 Hasil Metrix <i>MobileNetV3Small</i>	66
Gambar 4. 7 Confusion Matrix <i>MobileNetV3Large</i>	68
Gambar 4. 8 Confusion Matrix <i>MobileNetV3Small</i>	68
Gambar 4. 9 Tampilan Halaman Utama Aplikasi.....	70
Gambar 4. 10 Tampilan Halaman Riwayat.....	71
Gambar 4. 11 Tampilan Halaman Pengaturan	72
Gambar 4. 12 Hasil Prediksi Alpukat Aligator	73
Gambar 4. 13 Hasil Prediksi Alpukat Kendil.....	74

Gambar 4. 14 Hasil Prediksi Alpukat Madu	75
Gambar 4. 15 Hasil Prediksi Alpukat Super	76
Gambar 4. 16 Hasil Prediksi Alpukat SW01	77
Gambar 4. 17 Hasil Prediksi Bukan Daun	78



DAFTAR TABEL

Tabel 2. 1 jenis- jenis Citra Daun Alpukat.....	12
Tabel 3. 1 Harga Alpukat	35
Tabel 3. 2 Data <i>Training</i> Sebelum Dan Sesudah Augmentasi.....	39
Tabel 3. 3 Parameter Model MobileNetV3.....	42
Tabel 3. 4 Evaluasi <i>Confusion Matrix</i>	45
Tabel 3. 5 Tabel Predictions.....	52
Tabel 4. 1 Hasil Pengujian Model <i>MobileNetV3Large</i>	58
Tabel 4. 2 Hasil Pengujian Model <i>MobileNetV3Small</i>	60
Tabel 4. 3 Hasil Perbandingan Pengujian	62
Tabel 4. 4 Kesimpulan Pengujian Form Deteksi Jenis Daun Alpukat.....	78

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara yang kaya akan sumber daya alam, dengan iklim tropis yang mendukung pertumbuhan berbagai jenis tanaman sepanjang tahun. Curah hujan yang tinggi serta kesuburan tanah menjadikan Indonesia sebagai salah satu wilayah yang ideal untuk kegiatan pertanian dan perkebunan. Hampir seluruh jenis tanaman pangan dan buah-buahan dapat tumbuh subur di berbagai daerah, seperti alpukat, pisang, durian, salak, jeruk, manggis, dan rambutan. Salah satu komoditas buah tropis yang memiliki nilai ekonomi tinggi serta banyak diekspor adalah alpukat (*Persea americana* Mill), yang berasal dari Amerika Tengah dan kini telah banyak dibudidayakan di berbagai wilayah Indonesia, terutama di Provinsi Riau yang memiliki kondisi tanah dan iklim yang mendukung.

Setiap varietas alpukat memiliki ciri morfologis yang berbeda, termasuk bentuk, warna, dan tekstur daun. Beberapa varietas yang populer di Indonesia antara lain alpukat miki, alligator, dan red vietnam. Namun, perbedaan antar varietas tersebut sering kali sulit dikenali secara visual karena kemiripan bentuk daun, sehingga proses identifikasi secara manual menjadi subjektif dan berpotensi menimbulkan kesalahan. Kondisi ini dapat menimbulkan permasalahan dalam distribusi bibit, seperti ketidaksesuaian antara jenis bibit yang dibeli dengan yang dijanjikan oleh penjual. Masalah ini berdampak pada berkurangnya kepercayaan antara penjual dan pembeli, serta menurunkan kredibilitas pelaku usaha yang jujur dalam perdagangan bibit alpukat. Oleh karena itu, diperlukan suatu solusi yang mampu mengidentifikasi jenis daun alpukat secara cepat, akurat, dan praktis (Sari et al., 2023).

Seiring dengan pesatnya perkembangan teknologi, penerapan kecerdasan buatan (Artificial Intelligence/AI), khususnya dalam bidang pengolahan citra digital, telah memberikan peluang besar bagi sektor pertanian. Salah satu metode yang banyak digunakan untuk klasifikasi citra adalah Convolutional Neural

Network (CNN). Model CNN terbukti efektif dalam mengenali pola visual dan melakukan klasifikasi gambar secara otomatis. Salah satu arsitektur CNN yang efisien untuk perangkat dengan sumber daya terbatas seperti smartphone adalah MobileNetV3, karena memiliki ukuran model yang ringan namun tetap mempertahankan tingkat akurasi yang tinggi.

Dalam konteks serupa, penelitian lain juga telah menunjukkan efektivitas penerapan CNN dalam sektor pertanian. Sebagai contoh, model MobileNetV3 Large telah berhasil digunakan untuk mendeteksi penyakit pada tanaman padi, di mana model dilatih menggunakan dataset citra daun padi berlabel selama 30 epoch dengan batch size 45 dan optimizer Lion. Hasil pengujian menunjukkan akurasi sebesar 75% pada data uji dan 76% pada data latih, menandakan bahwa pendekatan berbasis CNN dapat digunakan secara efektif untuk mendeteksi kondisi kesehatan tanaman. Implementasi tersebut bahkan telah berhasil diintegrasikan ke dalam aplikasi Android yang mampu melakukan klasifikasi penyakit padi secara otomatis (B. Kusuma et al., 2024).

Berdasarkan hasil penelitian sebelumnya yang menunjukkan keberhasilan penerapan model MobileNetV3 pada klasifikasi penyakit tanaman padi, penelitian ini mengadaptasi pendekatan serupa untuk mengklasifikasikan jenis daun alpukat menggunakan arsitektur MobileNetV3. Model yang telah dilatih kemudian dikonversi ke dalam format TensorFlow Lite (.tflite) agar dapat diimplementasikan pada perangkat mobile. Aplikasi ini dikembangkan menggunakan Flutter sebagai antarmuka pengguna (frontend) dan Flask sebagai backend yang menghubungkan aplikasi dengan model klasifikasi. Selain itu, SQLite digunakan untuk menyimpan hasil deteksi dan riwayat klasifikasi sehingga pengguna dapat memantau data identifikasi sebelumnya.

Melalui aplikasi ini, pengguna seperti petani, pembeli, maupun penjual bibit alpukat dapat dengan mudah mengunggah citra daun alpukat dan memperoleh hasil klasifikasi secara real-time dengan tingkat akurasi yang baik. Dengan demikian, sistem ini diharapkan dapat mengurangi kesalahan dalam pemilihan bibit, meningkatkan transparansi antara penjual dan pembeli, serta

memberikan solusi berbasis teknologi kecerdasan buatan yang efisien dan praktis untuk mendukung kemajuan sektor pertanian di Indonesia.

Penelitian ini diangkat dengan judul: “Implementasi Aplikasi Mobile untuk Klasifikasi Jenis Daun Alpukat Menggunakan MobileNetV3”, dengan tujuan untuk menghasilkan aplikasi yang mudah digunakan, efisien, dan bermanfaat, serta mampu memberikan kontribusi positif bagi peningkatan kualitas dan ketepatan pemilihan bibit alpukat di Indonesia.

1.2 Identifikasi Masalah

Berdasarkan penjelasan mengenai latar belakang permasalahan yang telah disampaikan, berikut adalah beberapa isu yang dapat diidentifikasi:

1. Daun dari berbagai jenis tanaman alpukat sering kali memiliki kemiripan bentuk, warna, dan tekstur sehingga menyulitkan proses identifikasi secara manual, terutama oleh petani atau masyarakat awam yang tidak memiliki keahlian khusus di bidang botani.
2. Belum tersedianya sistem yang mampu membedakan jenis tanaman alpukat. Hal ini menyebabkan kesulitan dalam memverifikasi keakuratan jenis bibit yang ditawarkan, sehingga pembeli bisa saja mengira bahwa penjual memberikan bibit yang tidak sesuai dengan jenis yang dijanjikan.
3. Penggunaan model deep learning yang ringan dan efisien seperti MobileNet-V3 sangat dibutuhkan untuk mengatasi keterbatasan sumber daya komputasi, terutama jika sistem akan diimplementasikan pada perangkat mobile atau aplikasi berbasis web.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan diatas, sebuah permasalahan sebagai berikut:

1. Bagaimana mengimplementasikan model MobileNetV3 ke dalam aplikasi Android untuk klasifikasi jenis tanaman alpukat berdasarkan citra daun ?
2. Bagaimana merancang aplikasi mobile yang dapat menyimpan dan menampilkan hasil klasifikasi beserta riwayat deteksi kepada pengguna ?

3. Bagaimana menguji dan mengevaluasi kinerja aplikasi agar dapat berfungsi secara optimal dan memberikan hasil yang akurat ?

1.4 Batasan Masalah

Berdasarkan latar belakang permasalahan yang telah dijelaskan di atas, batasan masalah dapat dirumuskan sebagai berikut:

1. Penelitian ini hanya fokus pada klasifikasi jenis tanaman alpukat berdasarkan citra daun, tanpa memperhatikan bagian tanaman lain seperti batang, buah, atau bunga.
2. Data citra daun alpukat yang digunakan terbatas pada beberapa jenis/varietas alpukat tertentu yang tersedia dalam dataset.
3. Citra daun yang digunakan adalah dalam format digital (JPEG/PNG) dan diperoleh melalui proses pemotretan atau dataset yang sudah tersedia.
4. Penelitian ini menggunakan lima jenis tanaman alpukat, yaitu alpukat Aligator, Kendil, Madu, Super, dan SW01, serta satu kelas bukan daun sebagai pembanding.
5. Penelitian ini menggunakan data primer sebanyak 900 data menggunakan kamera iphone 11 dengan ukuran gambar pada penelitian ini 3000 X 4000 pixel.

1.5 Tujuan Penelitian

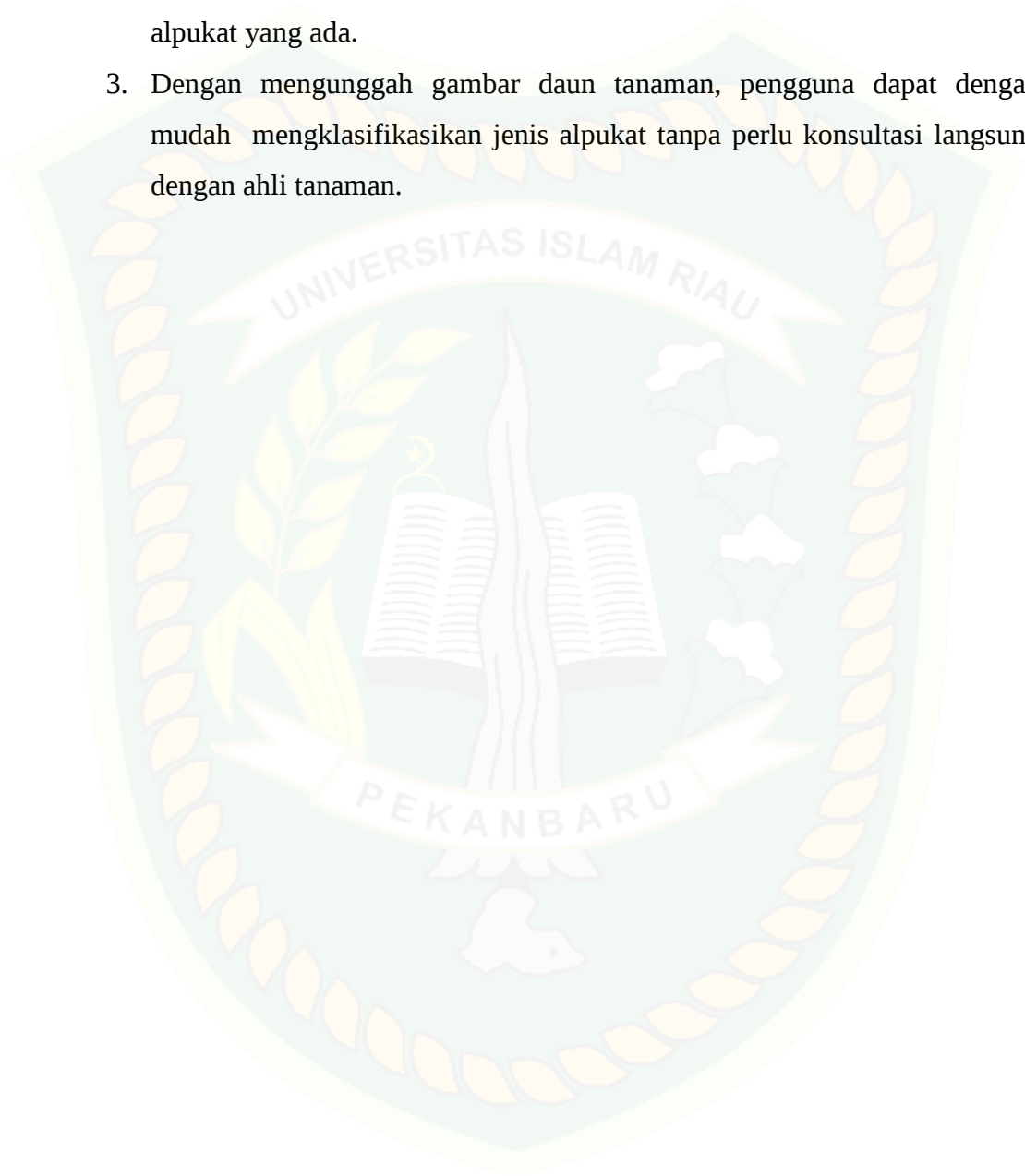
Tujuan penelitian dalam tugas akhir ini adalah sebagai berikut:

1. Mengimplementasikan model MobileNetV3 ke aplikasi Android untuk klasifikasi daun alpukat.
2. Merancang aplikasi mobile dengan fitur penyimpanan hasil klasifikasi dan riwayat.
3. Menguji kinerja aplikasi agar optimal dan akurat.

1.6 Manfaat Penelitian

Berdasarkan tujuan penelitian yang ingin dicapai, diharapkan penelitian ini dapat memberikan berbagai manfaat. Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Membantu dalam mengidentifikasi jenis tanaman alpukat secara cepat dan akurat hanya melalui citra daun, sehingga dapat mempermudah proses budidaya, pemilihan varietas unggul, dan pengelolaan tanaman.
2. Penelitian ini akan menambah pengetahuan masyarakat, petani, dan pelaku pertanian mengenai karakteristik dan nama-nama jenis tanaman alpukat yang ada.
3. Dengan mengunggah gambar daun tanaman, pengguna dapat dengan mudah mengklasifikasikan jenis alpukat tanpa perlu konsultasi langsung dengan ahli tanaman.



BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penyusunan proposal tugas akhir ini, penulis berusaha untuk mengkaji dan merujuk berbagai penelitian sebelumnya yang berkaitan dengan topik skripsi ini. Penelitian ini memberikan dasar teori dan informasi penting dalam pengembangan studi ini. Berikut ini adalah beberapa penelitian yang terkait dengan topik skripsi ini:

Penelitian yang dilakukan (Firdaus et al., 2024) mengenai Sistem Deteksi Penyakit Daun Singkong Menggunakan *Deep Learning* dengan Arsitektur Mobilenet-V3 berbasis Android. Singkong merupakan sumber karbohidrat yang dijadikan makanan utama di beberapa daerah seperti Afrika. Kualitas Singkong dapat dilihat dari daunnya sehingga penelitian ini didasarkan pada 4 penyakit yang mempengaruhi hasil singkong diantaranya, *Cassava Bacterial Blight* (CBB), *Cassava Brown Streak Disease* (CBSD), *Cassava Mosaic Disease* (CMD), dan *Cassava Green Mottle* (CGM). Algoritma yang digunakan adalah CNN dengan Arsitektur MobilenetV3. MobilenetV3 merupakan Arsitektur dari CNN yang cukup baik untuk proses deteksi penyakit pada daun singkong dengan memiliki akurasi 88.78% dan Validasi dengan skor 88.62%, akurasi dengan MobilenetV3 memiliki akurasi yang lebih baik dibandingkan dengan arsitektur EfficientNet-B7 dengan akurasi 71.57% sedangkan dengan ConvNext small memiliki akurasi 81.90%. Aplikasi deteksi penyakit pada daun singkong ini berbasis android dimana aplikasi ini dapat mendeteksi penyakit pada daun singkong secara realtime. Metode yang diterapkan menggunakan *deep learning* yang merupakan subbidang dari *machine learning* dimana tahapan penelitian yang dipakai adalah dataset, *preprocessing*, evaluasi model, *embed to Android App* dan pengujian. Deteksi Penyakit pada daun singkong menggunakan Mobilenetv3 memiliki akurasi yang cukup baik akan tetapi bisa penelitian selanjutnya bisa menggunakan arsitektur yang lebih terbaru seperti Yolov8 dan Yolo-nas.

Kemudian penelitian yang dilakukan (Mohammad Haydir Awaludin Waskito et al., 2024) mengenai Klasifikasi Penyakit Kronis Melalui Mata Menggunakan Algoritma *Convolutional Neural Network* Dengan Model *MobileNet-V3*. Penyakit kronis seperti glaukoma, hipertensi, diabetes, dan lainnya umumnya sulit dideteksi hanya melalui pengamatan visual, sehingga memerlukan pemeriksaan medis lanjutan di fasilitas kesehatan. Oleh karena itu, penelitian ini bertujuan mengintegrasikan pengetahuan medis dengan teknologi komputasi untuk menghasilkan sistem klasifikasi penyakit kronis yang lebih cepat dan akurat. Dalam proses klasifikasi citra mata, diperlukan kemampuan ekstraksi fitur yang optimal agar karakteristik setiap penyakit dapat dikenali dengan baik. Penelitian ini menerapkan metode *deep learning* berbasis CNN dengan arsitektur *MobileNet-V3*, yang dikenal efektif dalam mengekstraksi fitur dari citra beresolusi tinggi. Hasil penelitian menunjukkan bahwa model *MobileNet-V3* dengan pendekatan *transfer learning* mampu mengklasifikasikan tujuh kategori penyakit, yaitu Normal, Diabetes, Glaucoma, Cataract, Age-related macular degeneration, Hypertension, dan Pathological myopia dengan akurasi sebesar 81% dan nilai *loss* sebesar 0.4913. Peneliti merekomendasikan penambahan jumlah dataset dan variasi citra penyakit kronis lainnya untuk meningkatkan performa model serta memperoleh hasil klasifikasi yang lebih akurat dan beragam.

Penelitian yang dilakukan (B. Kusuma et al., 2024) mengenai Optimasi Dan Integrasi *Convolutional Neural Network* Pada Aplikasi Android Untuk Deteksi Penyakit Daun Padi. Padi merupakan komoditas pangan utama di Indonesia yang memiliki peran penting dalam menopang perekonomian nasional serta memenuhi kebutuhan konsumsi masyarakat. Namun, dalam beberapa tahun terakhir, produksi padi cenderung menurun akibat serangan berbagai hama dan penyakit tanaman. Oleh karena itu, deteksi dini dan klasifikasi penyakit padi secara akurat menjadi langkah penting untuk meminimalkan potensi kerugian. Penelitian ini mengembangkan model *convolutional neural network* (CNN) berbasis arsitektur *MobileNetV3 Large* untuk mengidentifikasi kondisi kesehatan daun padi. Pelatihan model dilakukan menggunakan dataset citra daun padi berlabel selama 30 *epoch* dengan ukuran *batch* 45 dan *optimizer* Lion. Hasil evaluasi menunjukkan bahwa model mencapai akurasi 75% pada data uji dengan

nilai *loss* sebesar 59%, serta akurasi 76% pada data latih dengan *loss* 61%. Selain itu, model yang telah dikembangkan berhasil diintegrasikan ke dalam aplikasi mobile berbasis Android. Penelitian ini diharapkan dapat memberikan kontribusi dalam bidang pertanian, khususnya terkait pengembangan sistem deteksi penyakit padi yang lebih efisien, cepat, dan akurat, sehingga dapat mendukung peningkatan produktivitas serta keberlanjutan sektor pertanian di Indonesia.

Penelitian yang dilakukan (Kirom et al., 2025) mengenai Klasifikasi Ras Kelinci Menggunakan *Convolutional Neural Network* (CNN) untuk Optimasi Sistem Identifikasi Visual. Kelinci merupakan mamalia dengan keragaman jenis serta ciri fisik yang unik, sehingga membedakan tiap ras—terutama yang memiliki kemiripan bentuk tubuh maupun pola warna—sering kali menjadi tantangan bagi banyak orang. Penelitian ini bertujuan mengembangkan sistem klasifikasi jenis kelinci berbasis CNN dengan arsitektur *MobileNet-V3*. Dataset yang digunakan terdiri dari 1.500 citra tiga jenis kelinci, yaitu bligon, hyla, dan new zealand white, yang diproses melalui *resizing*, augmentasi, dan normalisasi untuk meningkatkan kualitas data. Model dilatih menggunakan *optimizer* Adam dan menghasilkan akurasi 97% pada data validasi serta 90% pada dataset eksternal, menunjukkan kemampuan generalisasi yang baik. Temuan tersebut menegaskan bahwa CNN lebih efektif dibandingkan metode manual dalam mengenali pola visual, sekaligus mengurangi keterbatasan waktu dan potensi kesalahan manusia. Meski demikian, variasi pencahayaan dan sudut pengambilan gambar pada dataset masih menjadi kendala yang memengaruhi performa model. Penelitian ini tidak hanya mendukung efisiensi dalam pengelolaan peternakan, tetapi juga menunjukkan potensi besar penerapan teknologi kecerdasan buatan di sektor peternakan Indonesia. Pengembangan dataset yang lebih beragam serta eksplorasi arsitektur model lain direkomendasikan untuk peningkatan performa di masa mendatang. Selain itu, pengembangan aplikasi *real-time* berbasis perangkat mobile diperlukan agar proses identifikasi jenis kelinci dapat dilakukan secara praktis dan efisien. Penelitian ini diharapkan menjadi langkah awal dalam penerapan teknologi kecerdasan buatan untuk mendukung sektor peternakan di Indonesia, khususnya dalam meningkatkan akurasi dan efisiensi identifikasi ras kelinci.

Penelitian yang dilakukan (Zahro et al., 2024) mengenai Optimisasi Hyperparameter pada Model Klasifikasi Jenis Bunga dengan arsitektur MobileNetV3-Small. Studi ini bertujuan mengklasifikasikan berbagai jenis bunga melalui pendekatan *deep learning* menggunakan arsitektur *convolutional neural network MobileNetV3-Small*. Tahapan penelitian meliputi proses pengumpulan data, pembagian dataset, pelatihan model dengan arsitektur *MobileNetV3-Small*, serta evaluasi performa menggunakan *confusion matrix*. Delapan model dengan variasi *hyperparameter* diuji untuk memperoleh konfigurasi terbaik dengan akurasi tertinggi. Hasil penelitian menunjukkan bahwa model kelima menghasilkan akurasi validasi tertinggi, yaitu 99,25%. Evaluasi lebih lanjut memperlihatkan bahwa model yang sama mencapai akurasi pengujian sebesar 92%. Temuan ini menegaskan bahwa peningkatan jumlah data serta pengaturan *hyperparameter* yang optimal dapat secara signifikan meningkatkan akurasi model, sehingga performanya dapat mendekati sempurna apabila didukung oleh data pelatihan yang lebih besar dan konfigurasi parameter yang tepat. Dataset yang digunakan dalam penelitian ini berasal dari platform Kaggle dan mencakup lima jenis bunga, yaitu daisy, dandelion, rose, bunga matahari, dan tulip. Penelitian ini belum menerapkan tahapan *preprocessing* mendalam seperti *cropping*, *resize*, maupun augmentasi data. Oleh karena itu, disarankan untuk menerapkan teknik *preprocessing*, menggunakan *learning rate* yang rendah, serta menambah jumlah *epoch* pada penelitian berikutnya guna meningkatkan performa model secara lebih optimal.

Penelitian yang dilakukan (B. M. Kusuma et al., 2025) mengenai Klasifikasi Jenis Penyakit Pada Tanaman Padi Menggunakan Algoritma *Convolutional Neural Network*. Penelitian ini mengembangkan serta melatih model CNN untuk mengenali kondisi kesehatan tanaman padi melalui citra daun. Dataset yang digunakan berasal dari situs Kaggle.com dengan total 53.300 citra daun padi yang terbagi ke dalam empat kategori, yaitu Brownspot, Leafblast, Hispa, dan Healthy. Tahapan persiapan data meliputi *resize*, augmentasi, dan *re-scaling* piksel untuk memastikan kualitas data tetap optimal. Model dilatih dengan parameter terbaik, yaitu 30 *epoch*, *batch size* 45, dan *optimizer* Lion. Hasil evaluasi menunjukkan bahwa model memperoleh akurasi 75% pada data uji

dengan nilai *loss* sebesar 59%, serta akurasi 76% pada data latih dengan *loss* sebesar 61%. Selain itu, model yang dihasilkan berhasil diintegrasikan ke dalam aplikasi mobile berbasis Android. Penelitian ini diharapkan dapat memberikan kontribusi bagi sektor pertanian Indonesia dengan menghadirkan alat deteksi penyakit padi yang lebih efisien dan efektif, sehingga dapat mendukung peningkatan produktivitas dan pengelolaan tanaman padi secara lebih optimal.

Berdasarkan hasil telaah dari sejumlah penelitian yang dianalisis, dapat disimpulkan bahwa penggunaan arsitektur CNN seperti *MobileNetV3* memiliki potensi besar dalam mendukung proses klasifikasi citra di berbagai bidang, termasuk pertanian, kesehatan, peternakan, serta pengenalan objek secara umum. Beragam studi tersebut menunjukkan bahwa CNN mampu menghasilkan performa yang akurat dan efisien, terutama ketika diimplementasikan pada aplikasi mobile berbasis Android. Meskipun setiap penelitian memiliki fokus yang berbeda—mulai dari deteksi penyakit tanaman, klasifikasi penyakit kronis pada manusia, identifikasi ras kelinci, klasifikasi jenis bunga, hingga pendeteksian penyakit pada daun padi—seluruhnya memanfaatkan pendekatan serupa dan memberikan hasil yang konsisten positif.

Namun, perbedaan utama dari penelitian-penelitian tersebut terletak pada objek yang diklasifikasi, jumlah dan jenis data yang digunakan. Beberapa penelitian juga sudah mengembangkan aplikasinya ke bentuk nyata seperti website atau aplikasi Android untuk mempermudah pengguna. Penggunaan arsitektur *MobileNetV3* secara umum terbukti efektif, namun penelitian-penelitian ini juga membuka peluang untuk eksplorasi model lain yang lebih baru dan pengembangan sistem yang lebih optimal melalui peningkatan kualitas data dan teknik preprocessing yang lebih baik.

2.2 Dasar Teori

2.2.1 Tanaman Alpukat

Tanaman alpukat merupakan salah satu buah tropis yang banyak dikonsumsi masyarakat karena kandungan lemak dan protein yang bermanfaat bagi tubuh. Alpukat termasuk komoditas utama yang mendapat prioritas dalam

bidang penelitian dan perdagangan, bersama dengan jeruk, pisang, nanas, dan mangga. Produksi alpukat di Indonesia mengalami fluktuasi namun cenderung meningkat, dengan jumlah produksi sebesar 289.893 ton pada tahun 2013, meningkat menjadi 307.318 ton pada tahun 2014, lalu naik lagi menjadi 382.537 ton pada tahun 2015. Pada tahun 2016, produksi menurun menjadi 304.932 ton, namun kembali meningkat pada tahun 2017 mencapai 363.148 ton.

Tanaman alpukat memiliki ukuran pohon yang relatif kecil, dengan tinggi berkisar antara 3 hingga 10 meter. Batangnya bulat dan memiliki banyak cabang. Daunnya lebat, akarnya tunggang, serta batangnya berwarna coklat kusam. Alpukat berasal dari kawasan Amerika Tengah seperti Meksiko, Peru, dan Venezuela, namun kini telah tersebar luas ke berbagai wilayah termasuk Asia Tenggara dan Indonesia. Terdapat tiga jenis utama alpukat berdasarkan bijinya, yaitu dari kelompok Guatemala, India Barat, dan Meksiko. Ketiga jenis ini memiliki perbedaan dalam hal ukuran buah, tekstur kulit, cita rasa, kandungan lemak, ketahanan terhadap penyakit dan penyimpanan, serta kemampuan beradaptasi dengan lingkungan baru (Artikel, 2022). Gambar 2.1 menunjukkan tanaman alpukat



Gambar 2. 1 Tanaman Alpukat

2.2.2 Jenis Alpukat

Alpukat merupakan salah satu buah yang memiliki kandungan nutrisi, serat, vitamin, dan mineral yang cukup tinggi. Meskipun demikian, sebagian orang memilih untuk tidak mengonsumsi alpukat karena merasa khawatir terhadap kandungan lemak yang terdapat di dalamnya..

Dengan banyaknya varietas alpukat yang tersedia, sering kali muncul kesulitan dalam mengidentifikasi masing-masing jenis secara cepat dan akurat. Ketepatan dalam proses identifikasi sangat diperlukan, khususnya dalam kegiatan

pemasaran, penentuan harga, serta pengelolaan kualitas buah agar sesuai dengan kebutuhan konsumen maupun standar produksi. strategi budidaya. Kesalahan dalam identifikasi dapat berdampak pada penurunan nilai ekonomi dan ketidaksesuaian strategi pemasaran bagi petani dan pelaku agribisnis (Hapuarachchi et al., 2024).

Jenis Alpukat yang ada di Indonesia antara lain Alpukat Hass, Alpukat Jumbo, Alpukat Hijau, Alpukat Merah, Alpukat Keprok, Alpukat Mentega, Alpukat Raja, Alpukat Bali, Alpukat Borneo, Alpukat Krim, Alpukat Kendil, Alpukat miki, Alpukat Wina, Alpukat Kendil, Alpukat Pluwang, dan Alpukat Aligator (Putriani et al., 2022) . Melalui penelitian ini terdapat 5 macam citra daun alpukat yang dijadikan sample antara lain; citra daun alpukat aligator, citra daun alpukat kendil, citra daun alpukat madu, dan citra daun alpukat super avocado. Citra setiap daun alpukat memiliki bentuk yang berbeda beda. Contohnya seperti yang disajikan pada Tabel 2.1.

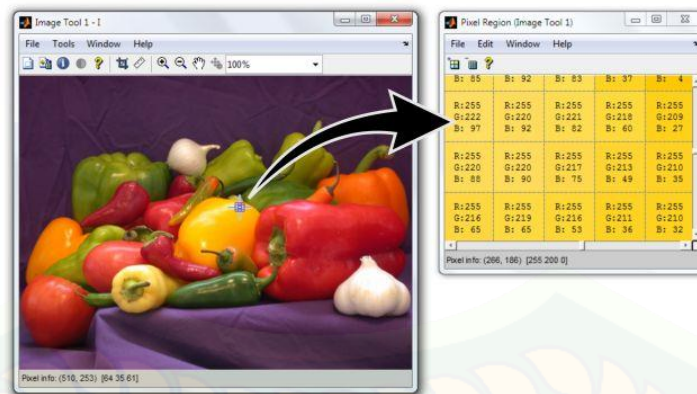
Tabel 2. 1 jenis- jenis Citra Daun Alpukat

No	Citra Daun Alpukat	Jenis Alpukat
1.		Alpukat SW01
2.		Alpukat Super

3.		Alpukat Madu
4.		Alpukat Kendil
5.		Alpukat Aligator

2.2.3 Tahapan Pengolahan Citra Digital

Pengolahan citra digital adalah prosedur untuk memanipulasi gambar dengan bantuan teknologi komputer. Dengan demikian, proses ini memanfaatkan algoritma perangkat lunak untuk menghasilkan gambar yang lebih baik atau untuk memperoleh berbagai jenis informasi yang bermanfaat. Pengolahan citra digital dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Pengolahan Citra Digital

2.2.3.1 Pengolahan Citra Digital

Pengolahan Citra Digital merupakan bidang ilmu yang mempelajari berbagai metode untuk memproses citra, baik citra statis seperti foto maupun citra dinamis seperti video. Istilah *digital* menandakan bahwa seluruh proses pengolahan dilakukan secara elektronik dengan bantuan komputer. Salah satu model warna yang umum digunakan dalam pengolahan citra digital adalah RGB, yang merupakan akronim dari *Red*, *Green*, dan *Blue*, yaitu tiga warna primer yang menjadi dasar pembentukan berbagai warna lainnya. Melalui model RGB, setiap warna direpresentasikan dalam bentuk nilai numerik, sehingga memungkinkan konsistensi tampilan warna pada berbagai perangkat. Sistem komputer telah mengadopsi model warna ini secara luas, sehingga proses pemrosesan warna berbasis RGB dapat berlangsung secara efisien dan seragam. (Ratna, 2020).

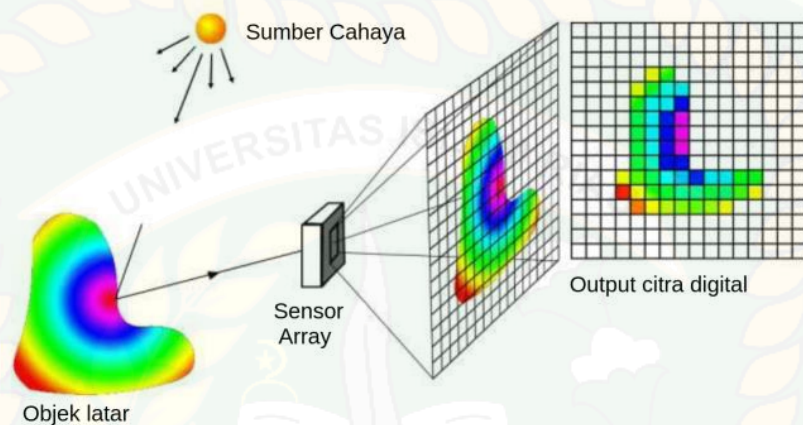
Pengolahan citra memiliki beberapa tujuan utama, antara lain meningkatkan kualitas gambar, melakukan kompresi, menganalisis citra, serta melakukan rekonstruksi dan restorasi gambar. Tujuan-tujuan ini dapat diterapkan dalam kehidupan nyata untuk berbagai keperluan, misalnya meningkatkan kualitas citra agar dapat digunakan dalam proses lanjutan. Salah satu contoh penerapan dari pengolahan citra adalah dalam sistem deteksi atau pengenalan objek.

2.2.3.2 Akuisisi Citra

Akuisisi citra adalah langkah dalam mengambil atau memindai citra analog untuk menghasilkan citra dalam format digital. Ada sejumlah elemen yang harus diperhatikan dalam prosedur akuisisi citra, diantaranya adalah: tipe alat

akuisisi yang digunakan, resolusi dari kamera, metode pencahayaan, ukuran atau zooming, jarak, serta sudut saat citra diambil.

Tujuan utama dari akuisisi citra adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital (Zuhdi & Busyairi, 2023). Tahap yang dilakukan pada proses akuisisi citra digambarkan dengan Gambar 2.3.



Gambar 2. 3 Proses Akuisisi Citra

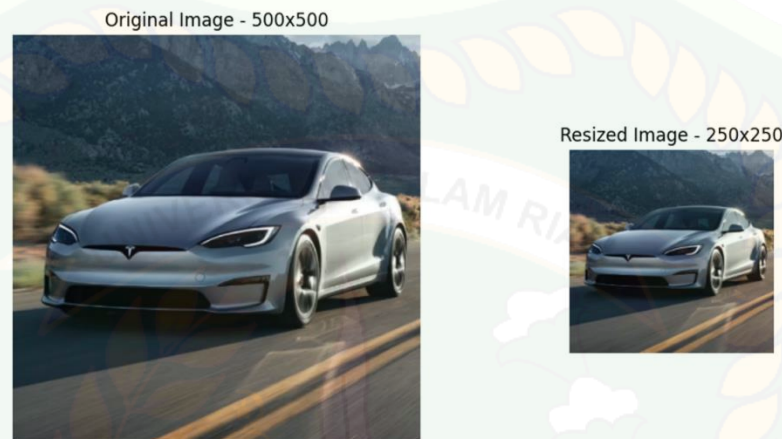
2.2.3.3 *Preprocessing*

Tahapan *preprocessing* dilakukan untuk menyiapkan data agar sesuai dengan kebutuhan model CNN. Proses ini bertujuan mempermudah tahapan pengolahan data serta meningkatkan kualitas data yang akan digunakan dalam pelatihan model (Komputer et al., 2022). Proses ini mencakup pengubahan ukuran gambar dengan ukuran standar, seperti 224x224 piksel, guna memastikan konsistensi input. Selain itu, dilakukan augmentasi dimana menggunakan library *ImageDataGenerator* dari library TensorFlow Keras. TensorFlow sendiri dimanfaatkan dalam berbagai tugas kecerdasan buatan, termasuk pemrosesan citra. Tahap preprocessing yang digunakan dapat dilihat sebagai berikut :

1. *Resize*

Resize Citra atau penskalaan merupakan proses mengubah ukuran citra menjadi lebih kecil. Tahap di mana gambar diubah melalui pemotongan atau perubahan ukuran dari gambar aslinya dengan berbagai dimensi menjadi

gambar sesuai ukuran. Pada penelitian ini, proses *resize* dilakukan tanpa menggunakan metode khusus, yakni dengan mengubah ukuran menjadi 224 X 224 piksel. Langkah ini penting dilakukan agar pemrosesan citra oleh komputer menjadi lebih efisien, menghemat ruang penyimpanan dan mempercepat proses komputasi. Contoh citra yang sedang melakukan proses *resizing* dapat dilihat pada Gambar 2.4.

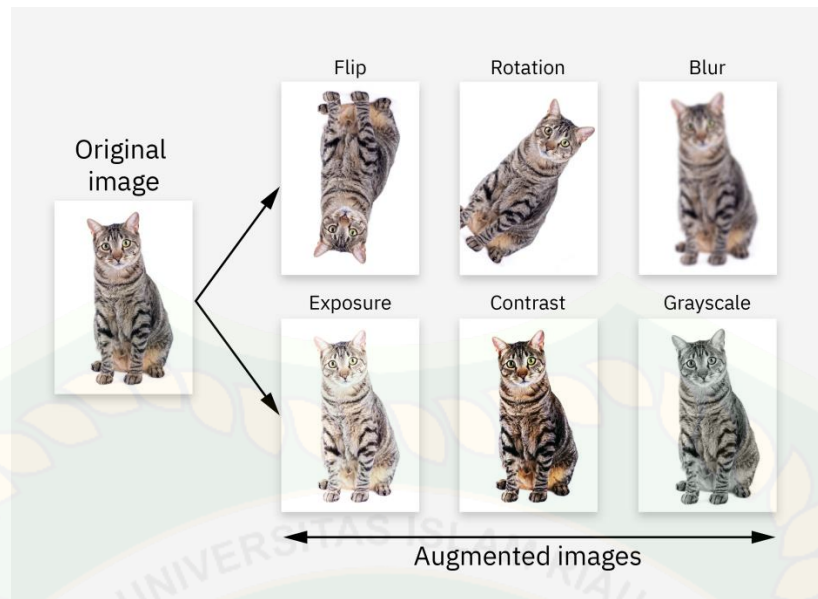


Gambar 2. 4 Perbedaan Sebelum Dan Sesudah *Resizing*

Proses *resize* dilakukan untuk mempercepat perhitungan dengan mengubah ukuran citra menjadi 224x224 piksel. Selanjutnya, dilakukan peningkatan kualitas citra melalui penyesuaian kecerahan atau peningkatan kontras, dengan format output berupa png, jpg, dll.

2. Augmentasi

Augmentasi data merupakan teknik yang digunakan untuk menambah jumlah citra melalui berbagai manipulasi tanpa mengubah citra asli. Proses ini tidak dilakukan secara langsung pada gambar utama, tetapi melalui pembuatan variasi baru secara virtual. Tujuan dari augmentasi data adalah membantu model dalam mempelajari serta mengenali beragam variasi citra, sekaligus meningkatkan jumlah data pelatihan tanpa perlu menambahkan citra asli tambahan (Satria Maheswara et al., 2022). Gambar 2.5 contoh proses dari augmentasi data.



Gambar 2. 5 Contoh Dari Augmentasi Data

Augmentasi data dalam pengenalan citra diterapkan untuk mencegah terjadinya overfitting selama proses pelatihan model. Teknik ini hanya diterapkan pada data latih. Proses augmentasi dilakukan menggunakan data generator dari Keras, namun dari berbagai parameter yang tersedia, hanya sebagian yang dipilih dan digunakan dalam implementasinya (Hakiki; et al., 2023).

Augmentasi data merupakan pendekatan yang memungkinkan penambahan variasi pada data pelatihan tanpa perlu melakukan pengumpulan data baru. Berbagai teknik seperti *cropping*, *padding*, dan *horizontal flipping* umum digunakan dalam proses pelatihan jaringan neural skala besar. Namun, pada praktiknya, sebagian besar proses pelatihan jaringan neural masih mengandalkan teknik augmentasi dasar. Sementara itu, penelitian yang lebih mendalam cenderung berfokus pada pengembangan arsitektur jaringan neural itu sendiri. (Sanjaya & Ayub, 2020).

Augmentasi data terdiri dari berbagai teknik yang bertujuan untuk memperbesar dan memperbaiki kualitas data pelatihan, sehingga memungkinkan pembangunan model deep learning yang lebih optimal. Teknik-teknik peningkatan citra yang digunakan mencakup transformasi bentuk, penyesuaian warna, penggunaan filter kernel, penggabungan gambar, penyegaran secara acak,

perluasan representasi fitur, pelatihan adversarial, jaringan adversarial generatif, transfer gaya neural, serta pembelajaran meta. Dua metode augmentasi data yang paling populer dan efektif dalam pelatihan CNN adalah random flipping dan random cropping.

3. Normalisasi Pixel

Normalisasi Nilai MobileNet mengharapkan input yang dinormalisasi dengan nilai rata rata (mean) 0 dan standar deviasi sebesar 1. Perlu dilakukan normalisasi pada setiap gambar dalam dataset (Ramdani & Rahmatulloh, 2024). Rumus normalisasi nilai pixel dapat dilihat pada persamaan 2.1.

$$X_{\text{norm}} = \frac{x}{127,5} - 1 \quad (2.1)$$

2.2.4 Klasifikasi

Klasifikasi merupakan salah satu teknik yang paling banyak digunakan dalam *data mining*. Proses ini terdiri atas dua tahap utama, yaitu tahap pembelajaran (*learning*) dan tahap pengklasifikasian. Pada tahap pembelajaran, data latih dianalisis menggunakan algoritma klasifikasi tertentu untuk membentuk model atau aturan klasifikasi. Sedangkan pada tahap pengklasifikasian, data baru diuji dengan mengacu pada tingkat akurasi dari aturan yang telah dibangun sebelumnya. Dalam konteks pengolahan citra, klasifikasi gambar adalah proses pemberian label pada sebuah citra berdasarkan kategori yang telah ditentukan sehingga citra tersebut memiliki makna atau informasi yang jelas. Secara umum, klasifikasi citra bertujuan menganalisis gambar yang diberikan sebagai input dan menghasilkan keluaran berupa label yang sesuai dengan kategori yang telah ditetapkan (Rosyani et al., 2021).

Klasifikasi merupakan proses untuk membangun sebuah model atau fungsi yang dapat merepresentasikan serta membedakan berbagai kelas atau konsep data, sehingga model tersebut dapat digunakan untuk memprediksi kelas dari objek yang sebelumnya belum diketahui. Dalam ilmu statistika, metode klasifikasi yang lazim digunakan adalah Analisis Diskriminan dan Regresi Logistik. Namun, dengan munculnya era big data yang ditandai oleh pertumbuhan volume data yang sangat besar, dibutuhkan alat analisis yang lebih kuat dan fleksibel untuk

mengeksktraksi informasi berharga dan mengubahnya menjadi pengetahuan yang terstruktur. Di sisi lain, kemajuan pesat dalam teknologi kecerdasan buatan turut mendorong perkembangan metode *machine learning*, yaitu sistem yang dirancang untuk mampu belajar secara mandiri tanpa instruksi langsung dari pengguna, dengan dasar pengembangan yang berasal dari berbagai disiplin ilmu seperti statistika, matematika, dan data mining.

Dalam Deep Learning, metode klasifikasi yang paling umum digunakan adalah *Convolutional Neural Network* (CNN), karena sangat efektif dalam memproses dan mengenali objek pada gambar. Selain CNN, terdapat pula metode lain yang kerap diterapkan seperti *Deep Neural Network* (DNN), *Artificial Neural Network* (ANN), dan *Recurrent Neural Network* (RNN).

Proses klasifikasi data melibatkan beberapa tahapan, salah satunya adalah tahap pelatihan (training), di mana data latih dianalisis untuk membangun model klasifikasi. Tahapan berikutnya adalah klasifikasi, yaitu proses penerapan model yang telah dibentuk untuk memprediksi kelas dari data uji (testing) (Lahir et al., 2021).

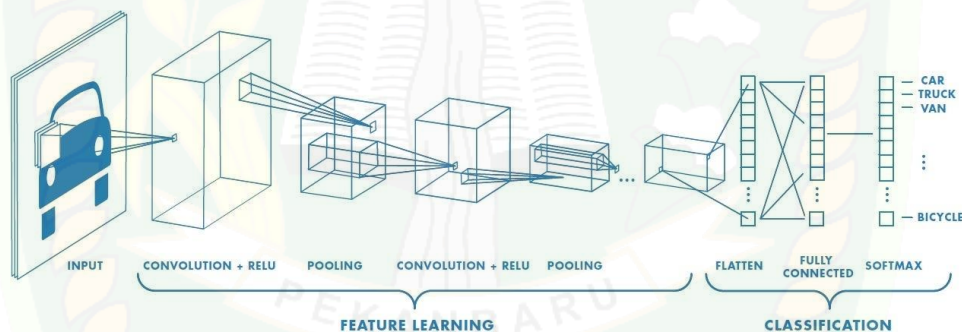
2.2.5 *Convolutional Neural Network* (CNN)

Convolutional Neural Networks (CNN) dikembangkan khusus untuk menangani tugas pengenalan gambar dan pertama kali digunakan dalam permasalahan pengenalan angka tulisan tangan (Fukushima 1980; LeCun 1989). Desain utama CNN bertujuan membentuk jaringan di mana neuron pada lapisan awal berperan dalam mengekstraksi fitur visual lokal, sedangkan neuron pada lapisan-lapisan berikutnya menggabungkan fitur-fitur tersebut menjadi representasi fitur yang lebih kompleks dan tinggi tingkatannya.

CNN merupakan jenis jaringan saraf di mana sinyal masukan diproses melalui serangkaian pasangan lapisan *convolutional* dan *pooling* yang tersusun secara berurutan (lapisan *conv-pool*). Output dari lapisan terakhir kemudian diteruskan ke beberapa lapisan *fully connected* sebelum akhirnya masuk ke lapisan *softmax* sebagai tahap klasifikasi akhir. Arsitektur CNN ditunjukkan pada

Gambar 1. CNN bekerja dengan memanfaatkan operasi konvolusi. Dengan menggeser sebuah *kernel* atau *filter* berukuran tertentu pada citra, komputer dapat mengekstraksi informasi representatif baru melalui hasil perkalian antara bagian citra dan filter yang digunakan. Proses ini memungkinkan model mengenali pola penting seperti tepi, tekstur, maupun bentuk pada gambar.

Convolutional Neural Network (CNN) merupakan bentuk lanjutan dari *Multi Layer Perceptron* (MLP) yang dirancang khusus untuk memproses data berdimensi dua. CNN termasuk dalam kategori *Deep Neural Network* dan sangat umum digunakan dalam pengolahan citra karena kemampuannya menangani struktur data yang kompleks. CNN terdiri dari tiga bagian utama yaitu lapisan input, lapisan output, serta beberapa lapisan tersembunyi. Lapisan tersembunyi ini mencakup *convolutional layer*, *pooling layer*, dan *fully connected layer*. Gambar 2.6 menunjukkan gambar lapisan pada metode *Convolution Neural Network*.

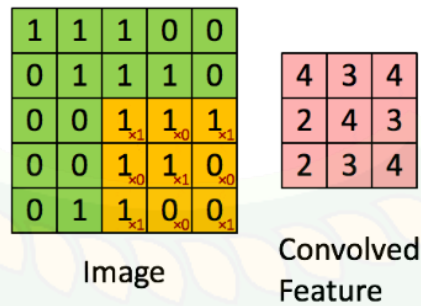


Gambar 2. 6 Proses *Convolution Neural Network*

2.2.5.1 *Convolution Layer*

Convolutional layer terdiri dari sejumlah neuron yang disusun membentuk sebuah *filter* dengan ukuran tertentu, baik pada panjang maupun tinggi dalam satuan piksel. Proses konvolusi dilakukan menggunakan *kernel* dan *stride*, di mana konvolusi merupakan operasi penggabungan antara dua matriks berbeda untuk menghasilkan matriks baru yang merepresentasikan fitur-fitur penting dari citra input. (Azmi et al., 2023). Dalam konteks pengolahan citra, konvolusi berarti menerapkan sebuah kernel pada seluruh bagian citra dengan berbagai

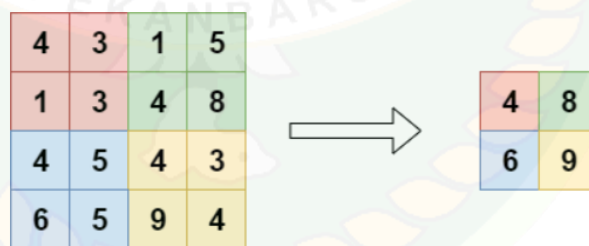
posisi (offset) yang memungkinkan, seperti yang digambarkan pada ilustrasi di Gambar 2.7.



Gambar 2. 7 Gambaran Operasi Konvolusi

2.2.5.2 Pooling Layer

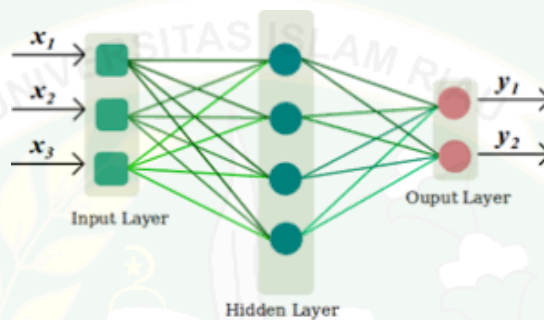
Lapisan *pooling* berfungsi pada setiap kumpulan *feature map* untuk mengurangi ukuran representasinya. Secara umum, lapisan ini menggunakan *filter* berukuran 2×2 dengan *stride* dua, yang diterapkan pada setiap bagian dari data masukan (Azmi et al., 2023) . Pada *max pooling*, nilai yang dipilih adalah nilai tertinggi dari area yang dijangkau oleh *kernel*, sedangkan pada *average pooling*, nilai yang dihitung merupakan rata-rata dari area tersebut. Proses dari *Pooling layer* dapat dilihat pada Gambar 2.8.



Gambar 2. 8 Pooling Layer

2.2.5.3 Fully Connected Layer

Fully Connected Layer merupakan lapisan yang berfungsi mengubah representasi data ke dalam bentuk yang sesuai untuk proses klasifikasi secara linear. Tidak seperti lapisan konvolusional, lapisan ini tidak menggunakan operasi konvolusi, melainkan mengandalkan perkalian matriks yang dilanjutkan dengan penambahan nilai bias. Setiap neuron pada lapisan ini terhubung sepenuhnya dengan seluruh neuron di lapisan sebelumnya, sehingga dinamakan *fully connected layer*. Gambar 2.10 menunjukkan proses dari *Fully Connected Layer*.



Gambar 2. 9 Fully Connected Layer

2.2.6 MobileNet-V3

MobileNet adalah arsitektur *Convolutional Neural Network* (CNN) yang dirancang dengan dua set hyperparameter guna menghasilkan model yang ringan dan memiliki latensi rendah, sehingga cocok untuk aplikasi yang membutuhkan efisiensi tinggi. Arsitektur ini umumnya digunakan dalam perangkat mobile dan sistem tertanam (*embedded systems*) (Santoso et al., n.d.).

MobileNetV3 adalah arsitektur jaringan saraf yang dirancang secara efisien dan ringan, sehingga sangat cocok digunakan pada perangkat mobile dengan keterbatasan daya dan memori (Chu et al., 2020). MobileNet-V3 memiliki 2 jenis model yaitu MobileNet-V3 Small dan MobileNet-V3 Large kedua model tersebut bisa dimanfaatkan untuk penggunaan sumber daya yang rendah ataupun sumberdaya tinggi. MobileNetV3-Small adalah arsitektur Convolutional Neural Network (CNN) yang ringan dan efisien, dikembangkan oleh Google. Arsitektur ini menggabungkan beberapa teknik efisiensi seperti *depthwise separable convolution* dan *squeeze-and-excitation*. Versi “Small”

dirancang khusus untuk digunakan pada perangkat dengan sumber daya terbatas, seperti smartphone dan sistem tertanam (*embedded systems*), namun tetap mampu memberikan performa yang baik tanpa mengorbankan akurasi secara signifikan (Liang et al., 2023). *MobileNetV3-Large* dirancang untuk digunakan pada perangkat dengan sumber daya tinggi maupun rendah. Arsitektur ini dikembangkan menggunakan metode *platform-aware neural architecture search* (NAS) seperti MnasNet, serta teknik *NetAdapt* untuk pencarian dan optimasi struktur jaringan. Model ini mengintegrasikan berbagai peningkatan pada arsitektur jaringan guna mencapai keseimbangan antara efisiensi dan performa (Suyuti, 2023). Gambar 2.10 dan Gambar 2.11 menunjukkan arsitektur *MobileNetV3 Large* dan arsitektur *MobileNetV3 Small*.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Gambar 2. 10 Arsitektur MobileNetV3 Large

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Gambar 2. 11 Arsitektur MobileNetV3 Small

Dibandingkan dengan arsitektur jaringan neural lainnya, MobileNetV3 memberikan perpaduan yang optimal antara performa dan efisiensi. Arsitektur ini dirancang khusus untuk kebutuhan aplikasi pada perangkat seluler, sehingga mampu memberikan hasil unggul dalam tugas klasifikasi, deteksi, dan segmentasi pada platform mobile. Meskipun beberapa arsitektur lain mungkin memiliki tingkat akurasi yang lebih tinggi, hal tersebut biasanya dibarengi dengan beban komputasi yang lebih besar, menjadikannya kurang ideal untuk perangkat dengan keterbatasan sumber daya. Fleksibilitas dan kinerja MobileNetV3 membuatnya menjadi pilihan yang banyak digunakan dalam pengembangan aplikasi pembelajaran mesin berbasis seluler.

MobileNet-V3 telah banyak dimanfaatkan dalam berbagai aplikasi pemrosesan citra di perangkat mobile, seperti deteksi objek, pengenalan wajah, serta klasifikasi gambar. Popularitas model ini di kalangan peneliti dan industri didorong oleh kemampuannya dalam memberikan performa yang efisien dan akurat, meskipun dijalankan pada perangkat dengan keterbatasan sumber daya (Qian, 2021).

2.2.7 Python

Python adalah bahasa pemrograman open-source yang sangat populer dan banyak dimanfaatkan dalam pengembangan kecerdasan buatan, khususnya pada bidang machine learning dan deep learning. Bahasa ini memiliki sintaks yang sederhana dan mudah dipahami, sehingga memudahkan pengembang

dalam menulis serta memahami kode program. Selain itu, Python memiliki komunitas besar dan aktif yang terus berkontribusi dalam pengembangan pustaka, framework, serta berbagai alat pendukung untuk memperluas fungsionalitasnya. Beberapa pustaka terkenal seperti TensorFlow, Keras, OpenCV, dan Scikit-learn menyediakan berbagai fungsi yang memudahkan proses pelatihan model, pengolahan citra, hingga deployment aplikasi kecerdasan buatan secara lebih efisien. Dukungan dokumentasi yang lengkap serta sumber belajar yang melimpah juga mempercepat proses pembelajaran dan pengembangan aplikasi berbasis AI (Ashraf, 2023). Logo Python dapat dilihat pada Gambar 2.12.



Gambar 2. 12 Logo *Python*

Python merupakan salah satu bahasa pemrograman yang bersifat dinamis dan banyak digunakan dalam pengembangan aplikasi. Bahasa ini mudah diimplementasikan karena tidak memerlukan proses kompilasi, sehingga skrip Python dapat langsung dijalankan segera setelah disimpan tanpa perlu langkah tambahan. Hal ini menjadikan *Python* sangat efisien dan praktis untuk pengembangan cepat dalam berbagai bidang aplikasi (Saladdin et al., 2020).

2.2.8 Google Colab

Google Colab merupakan platform berbasis *cloud computing* yang disediakan oleh Google, yang memungkinkan pengguna menjalankan kode *Python* secara langsung di lingkungan cloud tanpa perlu melakukan instalasi atau pengaturan pada perangkat lokal. Platform ini banyak digunakan oleh para peneliti dan ilmuwan data, peneliti, dan pengembang untuk berbagai keperluan, seperti pemrosesan data, analisis data, pengembangan model kecerdasan buatan, hingga pelatihan model machine learning (Yanuar & Andarsyah, 2024). *Google Colab* merupakan platform yang ramah bagi pemula, sehingga cocok digunakan untuk memulai pembelajaran pemrograman *Python* maupun eksplorasi data tanpa harus

melalui proses konfigurasi pada perangkat lokal. Alat ini sangat populer di kalangan komunitas data science dan kecerdasan buatan berkat kemudahan penggunaannya, akses terhadap sumber daya komputasi yang kuat, serta fitur kolaborasi yang memungkinkan pengguna bekerja sama dengan anggota tim secara efisien. Logo Google Colab dapat dilihat pada Gambar 2.13.



Gambar 2. 13 Logo Google Colab

2.2.9 *Flask*

Flask merupakan sebuah framework web yang dikembangkan menggunakan bahasa pemrograman python. Framework ini dikategorikan sebagai micro-framework karena tidak memerlukan depedensi alat atau pustaka khusus, serta menyediakan kemampuan pengelolaan basis data secara integritas dan sederhana (Mamuriyah et al., 2025). Flask tidak menyediakan lapisan abstraksi untuk basis data, validasi formulir, maupun komponen lainnya sebagai fungsi bawaan, karena fungsionalitas tersebut biasanya disediakan oleh pustaka pihak ketiga yang sudah ada. Namun demikian, Flask menyediakan dukungan ekstensi yang memungkinkan penambahan fitur-fitur aplikasi secara fleksibel seolah-olah fitur tersebut memang merupakan bagian integral dari Flask itu sendiri (Saladdin et al., 2020).

Dalam konteks pembelajaran mesin, salah satu tantangan utama setelah model selesai dilatih adalah tahap deployment. Model yang hanya disimpan dalam bentuk file atau notebook tidak akan menghasilkan manfaat praktis tanpa adanya mekanisme distribusi yang memungkinkan pengguna untuk mengakses fungsionalitasnya. *Flask* dapat berperan sebagai perantara antara model dan pengguna akhir dengan menyediakan REST API. Contohnya, sebuah endpoint POST /predict dapat disusun untuk menerima data masukan dalam format JSON, melakukan pra-pemrosesan data, menjalankan model untuk menghasilkan prediksi, dan kemudian mengembalikan hasilnya dalam respons JSON. Dengan

metode ini, integrasi model pembelajaran mesin dengan aplikasi eksternal, sistem akademik, maupun dashboard analitik menjadi lebih mudah dan efisien. Logo Flask dapat dilihat pada Gambar 2.14.



Gambar 2. 14 Logo *Flask*

2.2.10 SQLite

SQLite merupakan sistem manajemen basis data yang menggunakan model penyimpanan berorientasi baris (row-oriented). Sistem ini sangat populer, terutama pada perangkat smartphone berbasis Android, karena memiliki karakteristik yang sangat ringan dan efisien dalam penggunaan memori sekaligus memberikan performa yang cepat. Sebagai embedded database, SQLite tertanam langsung dalam aplikasi, sehingga sangat ideal untuk keperluan penyimpanan data pada level pengguna akhir. Berbagai fitur yang dimiliki SQLite meliputi ukuran yang kecil, tidak memerlukan instalasi terpisah, sifatnya yang bersifat open source, serta kompatibilitas lintas platform yang luas (Marques, n.d.). Logo SQLite dapat dilihat pada Gambar 2.15.



Gambar 2. 15 Logo SQLite

2.2.11 Flutter

Flutter merupakan framework open-source yang dikembangkan oleh Google untuk membangun aplikasi mobile, web, dan desktop dengan menggunakan bahasa pemrograman Dart. Framework ini mengusung konsep widget dalam pembuatan antarmuka pengguna, sehingga mampu menghadirkan tampilan yang konsisten dan responsif di berbagai platform (Gunawan, n.d.).

Flutter dirancang untuk mengembangkan aplikasi dengan performa tinggi yang dapat dijalankan dan dipublikasikan pada platform Android maupun iOS hanya dari satu codebase. Logo Flutter dapat dilihat pada Gambar 2.16.



Gambar 2. 16 Logo Flutter

Flutter bekerja dengan cara yang berbeda dibandingkan framework mobile lainnya. Ketika terjadi perubahan pada sebuah widget, Flutter akan merender ulang seluruh *widget tree* (pohon widget), tetapi hanya melakukan pembaruan pada komponen yang benar-benar mengalami perubahan. Mekanisme ini memungkinkan Flutter bekerja dengan performa yang cepat dan efisien. Selain itu, berkat penggunaan bahasa pemrograman Dart, Flutter mendukung fitur *Just-In-Time (JIT) compilation*, yang memberi keleluasaan bagi pengembang dalam proses pengembangan aplikasi melihat hasil perubahan pada aplikasi secara real-time, sehingga proses pengembangan menjadi lebih cepat dan interaktif.

2.2.12 Pengujian dengan Menggunakan *Confusion Matrix*

Confusion Matrix merupakan teknik yang digunakan untuk mengevaluasi performa suatu metode klasifikasi. Matriks ini berfungsi untuk menggambarkan hasil kerja model ketika diuji pada data yang memiliki label asli (*ground truth*). Secara sederhana, confusion matrix menampilkan tabel berisi perbandingan antara hasil prediksi model dan kelas sebenarnya. Melalui matriks ini, dapat diketahui jumlah prediksi yang tepat maupun yang keliru pada setiap kategori, serta jenis kesalahan yang terjadi. Selama proses pelatihan model, grafik *validation loss* dan *accuracy* dipantau secara berkala untuk mendeteksi kemungkinan munculnya masalah seperti overfitting atau underfitting.

Confusion matrix adalah sebuah tabel yang menyajikan informasi mengenai jumlah data yang diklasifikasikan dengan benar dan jumlah data yang diklasifikasikan secara keliru dari hasil pengujian model (Normawati & Prayogi, 2021).

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 17 Confusion Matrix

Berdasarkan **Gambar 2.17** Dalam confusion matrix terdapat empat istilah utama, yaitu *true positive* (TP), yaitu ketika data positif berhasil diprediksi sebagai data positif; *false positive* (FP), yaitu data negatif yang justru diprediksi sebagai data positif; *false negative* (FN), yaitu data positif yang diprediksi sebagai data negatif; serta *true negative* (TN), yaitu kondisi ketika data negatif diprediksi sebagai data negatif.

Untuk menghitung *performance metrics*, confusion matrix digunakan sebagai dasar penilaian. *Performance metrics* ini berperan sebagai indikator untuk menilai seberapa baik model bekerja. Tingkat akurasi model menunjukkan kemampuan model dalam memberikan prediksi yang benar, khususnya pada data dengan label positif yang sesuai.

Accuracy merupakan ukuran yang dihitung dari perbandingan antara jumlah prediksi yang benar dengan keseluruhan jumlah data yang diuji. Akurasi dapat menggambarkan keakuratan model klasifikasi yang digunakan. Nilai akurasi dapat diperoleh menggunakan persamaan 2.2

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (2.2)$$

Precision adalah ukuran yang menggambarkan seberapa tepat model dalam memprediksi kelas positif, yaitu rasio antara jumlah prediksi positif yang benar dengan seluruh prediksi yang berlabel positif. Nilai presisi dihitung berdasarkan perbandingan antara jumlah prediksi positif yang benar dengan total prediksi positif yang dihasilkan oleh model, dan dapat dirumuskan melalui persamaan 2.3

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2.3)$$

Recall adalah metrik yang mengukur seberapa besar bagian dari data positif aktual yang berhasil dikenali dengan benar oleh model. Nilai ini menunjukkan kemampuan model klasifikasi dalam menangkap seluruh instance positif yang ada pada dataset, sehingga memberikan gambaran mengenai seberapa efektif model dalam menemukan kembali data yang relevan. Nilai Recall dapat diperoleh menggunakan persamaan 2.4

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.4)$$

F1-Score, atau *F-Value*, merupakan metrik yang diperoleh dari penggabungan nilai precision dan recall melalui perhitungan harmonis antara keduanya. F1-Score dapat diperoleh menggunakan persamaan 2.5

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.5)$$

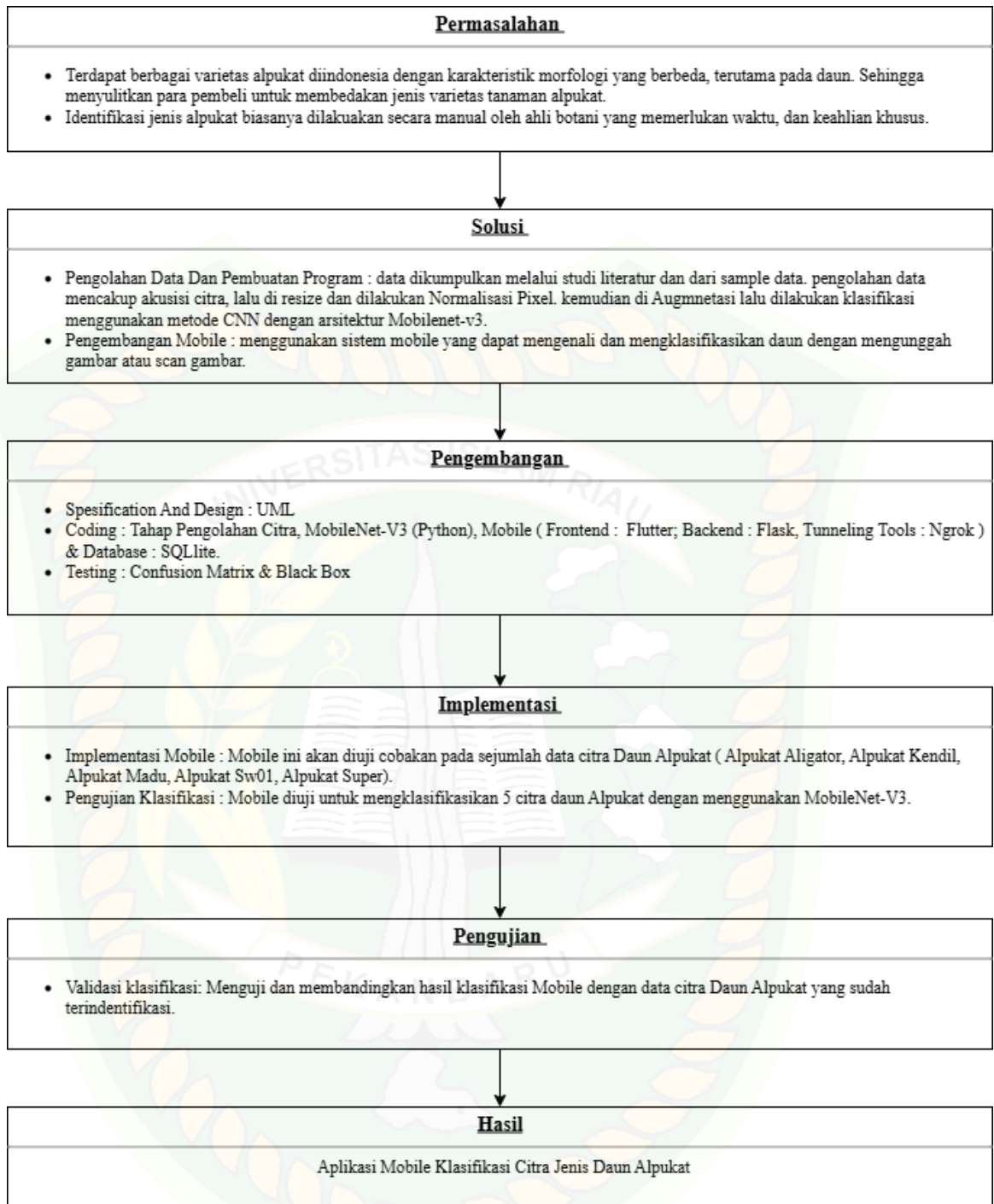
2.3 Kerangka Pemikiran

Indonesia sebagai negara agraris memiliki beragam jenis alpukat yang tersebar di berbagai daerah. Setiap jenis alpukat memiliki karakteristik morfologi daun yang unik, yang dapat digunakan sebagai dasar dalam proses identifikasi dan klasifikasi. Namun, dalam praktiknya Identifikasi jenis tanaman alpukat sering kali masih dilakukan secara manual, terutama oleh petani atau ahli pertanian yang sudah berpengalaman. Padahal, setiap jenis alpukat memiliki bentuk dan karakteristik daun yang berbeda, yang sebenarnya bisa dimanfaatkan sebagai penanda dalam proses klasifikasi. Tidak semua orang bisa membedakan jenis-jenis tersebut hanya dengan melihat daunnya, apalagi masyarakat umum yang belum familiar dengan budidaya tanaman alpukat. Masalah ini bisa menjadi penghambat, khususnya dalam hal pengembangan varietas unggul dan distribusi bibit berkualitas. Selain itu, kesulitan dalam mengenali jenis alpukat juga bisa berdampak pada efektivitas kegiatan pertanian dan pemasaran produk. Di tengah

kemajuan teknologi saat ini, seharusnya ada solusi yang bisa membantu proses identifikasi tersebut menjadi lebih cepat dan mudah diakses.

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi jenis tanaman alpukat berdasarkan daun menggunakan MobileNet-V3. Dalam hal ini, digunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur MobileNet-V3, yang dikenal ringan dan efisien untuk pengolahan citra di perangkat dengan spesifikasi terbatas. Sistem ini diharapkan mampu mengenali jenis-jenis alpukat hanya dari foto daun, sehingga bisa dimanfaatkan oleh petani, pelaku usaha, atau bahkan masyarakat umum yang ingin mengetahui jenis alpukat tertentu. Data dalam penelitian ini dikumpulkan dari berbagai jenis daun alpukat yang ada di lapangan. Setelah itu, citra daun akan diproses dan dilatih menggunakan model MobileNetV3 untuk membentuk sistem klasifikasi otomatis. Hasil akhir dari penelitian ini diharapkan dapat menjadi alat bantu yang praktis dan akurat dalam mengenali jenis tanaman alpukat, serta mendorong pemanfaatan teknologi digital dalam bidang pertanian secara lebih luas.

Kerangka pemikiran ini menjelaskan tahapan pengembangan sistem klasifikasi jenis tanaman alpukat berdasarkan daun. Permasalahan utama yang diangkat adalah keterbatasan sarana untuk mengenali jenis tanaman alpukat secara cepat dan akurat. Untuk mengatasi permasalahan tersebut, solusi yang ditawarkan mencakup pengolahan data melalui akuisisi citra, augmentasi data, serta pemodelan menggunakan arsitektur MobileNetV3 untuk proses klasifikasi. Pengembangan sistem dilakukan melalui pemodelan arsitektur dan implementasi berbasis *Python* dengan framework *Flask* untuk backend, *Flutter* untuk frontend, serta penggunaan basis data *SQLite*. Pada tahap implementasi, dilakukan pengujian terhadap data daun alpukat dan validasi hasil klasifikasi. Evaluasi performa dilakukan menggunakan confusion matrix dan pengujian black-box untuk memastikan tingkat akurasi yang optimal. Hasil akhir dari penelitian ini adalah aplikasi berbasis *Flutter* yang mampu mengidentifikasi lima jenis tanaman alpukat secara otomatis.



Gambar 2. 18 Kerangka Pemikiran

BAB III

METODOLOGI PENELITIAN

3.1 Analisa Sistem yang Sedang Berjalan

Salah satu tahapan esensial dalam pengembangan system klasifikasi jenis alpukat berbasis citra daun menggunakan MobileNet-V3 adalah melakukan analisis mendalam terhadap system eksiting yang telah berjalan. Analisis ini bertujuan untuk memperoleh pemahaman komprehensif mengenai prosedur-prosedur yang saat ini diimplementasikan, sehingga informasi yang diperoleh dapat dijadikan landasan dalam perancangan system klasifikasi yang lebih optimal dan akurat. Selain itu, dengan mengidentifikasi kebutuhan pengguna secara menyeluruh, perumusan permasalahan yang akan diselesaikan dapat dilakukan secara lebih terstruktur dan jelas.



Gambar 3. 1 Jenis Alpukat Kendil

Gambar 3.1 menggambarkan kondisi daun alpukat dari jenis kendil yang merupakan hasil pertanian lokal di wilayah Sungai Pagar. Daun-daun ini memiliki karakteristik morfologi yang beragam, sehingga memperkaya keanekaragaman hayati tanaman alpukat. Namun demikian, hingga saat ini, identifikasi jenis

alpukat berdasarkan ciri daun masih belum dikenal luas di kalangan masyarakat. Hal ini disebabkan oleh keterbatasan informasi yang tersedia serta belum adanya sistem yang efektif untuk mengklasifikasikan jenis alpukat berdasarkan morfologi daun. Akibatnya, pengetahuan masyarakat mengenai perbedaan jenis alpukat hanya sebatas pada ciri fisik buah, tanpa memahami detail karakteristik daunnya.

Kurangnya pemahaman masyarakat terhadap jenis-jenis alpukat berdasarkan daun sering kali menyebabkan rendahnya kesadaran akan pentingnya identifikasi varietas secara akurat. Tanpa adanya sistem klasifikasi berbasis citra daun, masyarakat akan mengalami kesulitan dalam membedakan jenis alpukat hanya melalui pengamatan visual daun. Kondisi ini juga menyulitkan petani dan pelaku agribisnis dalam memberikan informasi yang tepat mengenai varietas alpukat kepada konsumen atau pihak terkait. Akibatnya, mereka seringkali menghadapi tantangan dalam menjawab berbagai pertanyaan yang muncul mengenai identifikasi jenis alpukat.

Pengembangan sistem klasifikasi jenis alpukat berbasis citra daun menggunakan metode MobileNetV3 diharapkan dapat menjadi solusi atas permasalahan tersebut. Sistem berbasis mobile ini memungkinkan analisis data citra daun alpukat secara sistematis dan otomatis, sehingga memudahkan proses identifikasi dan klasifikasi jenis alpukat. Dengan menyediakan informasi yang akurat dan mudah diakses, sistem ini dapat membantu petani, konsumen, dan masyarakat umum dalam memahami perbedaan antar jenis alpukat. Selain itu, sistem ini juga meningkatkan aksesibilitas informasi mengenai varietas alpukat bagi masyarakat lokal maupun pihak yang berkepentingan, sehingga mereka dapat mengeksplorasi berbagai jenis alpukat secara digital tanpa harus bergantung pada sumber informasi konvensional.

Selain permasalahan identifikasi varietas, hasil pengamatan di lapangan juga menunjukkan bahwa harga jual alpukat di wilayah Sungai Pagar tidak ditentukan berdasarkan jenis atau varietas tanaman, melainkan berdasarkan ukuran dan kualitas fisik (grade) buah. Petani setempat membedakan alpukat menjadi beberapa kategori ukuran, di mana buah berukuran besar dan berkualitas baik dihargai lebih tinggi dibanding buah kecil atau yang memiliki cacat fisik. Sistem

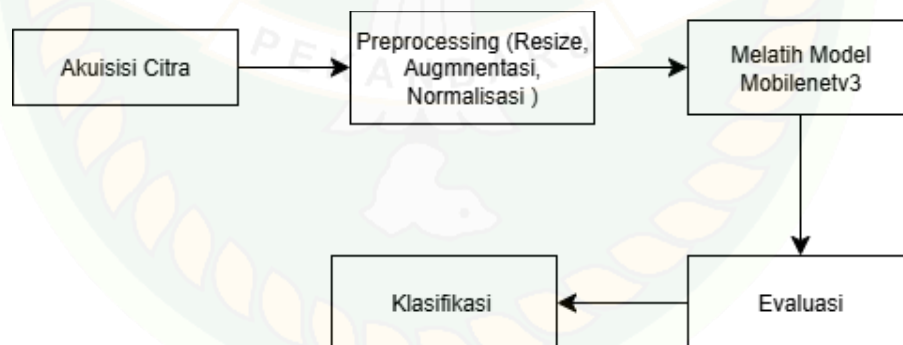
penentuan harga ini dilakukan secara visual berdasarkan pengalaman petani. Tabel 3.1 menunjukkan kisaran harga alpukat berdasarkan ukuran buah.

Tabel 3. 1 Harga Alpukat

Kategori / Grade	Ukuran Buah /Kg	Harga
Grade AA	0,8kg - 1kg	Rp. 50.000
Grade A	0,7 kg – 0.8 kg	Rp. 45.000
Grade B	0,5 kg – 0,6 kg	Rp. 40.000
Grade C	0,4kg	Rp. 35.000

3.2 Metode Penelitian

Penelitian ini bertujuan untuk merancang sebuah aplikasi mobile yang dapat membantu pengguna dalam mengidentifikasi berbagai jenis alpukat berdasarkan citra daun. Sistem ini dilengkapi dengan fitur yang dapat mendeteksi dan mengenali beberapa jenis varietas alpukat secara otomatis.



Gambar 3. 2 Proses Pengolahan Citra

Berikut adalah penjelasan mengenai langkah-langkah dalam proses pengolahan citra daun Alpukat seperti yang ditunjukkan pada Gambar 3.2.

3.2.1 Akuisisi Citra

Proses pengolahan data dalam penelitian klasifikasi jenis alpukat berbasis daun menggunakan model MobileNetV3 diawali dengan pengumpulan citra primer sebanyak 900 gambar daun alpukat yang diperoleh secara langsung melalui kamera handphone. Seluruh citra kemudian diubah ukurannya menjadi 224 x 224 piksel dengan format berwarna RGB agar sesuai dengan kebutuhan input model. Setelah tahap praproses awal, dataset dibagi menjadi tiga subset utama dengan proporsi 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Dengan demikian diperoleh 720 citra untuk pelatihan, 90 citra untuk validasi, dan 90 citra untuk pengujian.

Selanjutnya, proses augmentasi citra dilakukan hanya pada subset data pelatihan guna memperbanyak variasi data serta memperkuat kemampuan generalisasi model. Teknik augmentasi yang digunakan mengacu pada enam transformasi, yaitu rotasi acak hingga 20 derajat, pergeseran lebar (*Width Shift*) hingga 15%, pergeseran tinggi (*Height Shift*) hingga 15%, variasi kecerahan dalam rentang 0,85–1,15, perbesaran acak (*Zoom*) hingga 15%, serta pembalikan horizontal. Seluruh transformasi ini diterapkan dengan mode pengisian nearest untuk menjaga konsistensi piksel pada area hasil augmentasi.

Pendekatan ini dirancang agar model memperoleh data pelatihan yang lebih beragam secara virtual, sementara data validasi dan pengujian tetap dibiarkan dalam kondisi asli sehingga dapat memberikan evaluasi yang objektif terhadap kinerja model. Strategi tersebut tidak hanya meminimalkan risiko overfitting, tetapi juga meningkatkan kemampuan generalisasi MobileNetV3 dalam membedakan jenis alpukat berdasarkan karakteristik daun. Gambar 3.3 menunjukkan Citra daun Alpukat Aligator.



Gambar 3. 3 Citra Daun Alpukat Kendil

3.2.2 Citra Resize

Pada tahap *preprocessing*, citra mengalami proses awal setelah berhasil diakuisisi. Proses ini meliputi penyesuaian ukuran gambar (*resize*) untuk menyamakan dimensi citra agar dapat diproses secara konsisten dalam penelitian. Setelah pengambilan citra daun alpukat selesai, seluruh gambar yang telah dikumpulkan dan diklasifikasikan siap digunakan sebagai dataset penelitian. Meskipun citra yang diperoleh memiliki ukuran asli yang beragam dan relatif besar, proses *resize* dilakukan untuk mengubah dimensi citra menjadi seragam sehingga memudahkan pengolahan dan kompatibilitas dengan model MobileNetV3. Dalam penelitian ini, citra daun alpukat diubah ukurannya menjadi 224 x 224 piksel sesuai dengan kebutuhan input model. Citra dengan ukuran yang telah disesuaikan ini kemudian digunakan sebagai dataset utama dalam proses pelatihan dan pengujian model klasifikasi. Gambar awal (Gambar 3.4) menunjukkan dimensi asli citra, sedangkan (Gambar 3.5) gambar hasil *resize* memperlihatkan citra dengan dimensi yang telah disesuaikan untuk mendukung proses ekstraksi fitur dan klasifikasi secara optimal.



Image ID	
Dimensions	3000 x 4000
Width	3000 pixels
Height	4000 pixels

Gambar 3. 4 Citra Dengan Dimensi Asli

Resize 224x224



Gambar 3. 5 Citra Hasil *Resize*

3.2.3 Augmentasi

Pada penelitian ini digunakan 900 citra daun alpukat sebagai data awal. Seluruh citra terlebih dahulu dibagi menjadi tiga subset dengan proporsi 80% untuk pelatihan (720 citra), 10% untuk validasi (90 citra), dan 10% untuk pengujian (90 citra). Setelah proses pembagian, augmentasi citra dilakukan hanya pada subset pelatihan sebanyak enam kali menggunakan berbagai transformasi, sehingga jumlah data pada subset pelatihan bertambah signifikan menjadi 3600 citra. Sementara itu, subset validasi dan pengujian tetap menggunakan citra asli

tanpa augmentasi, masing-masing berjumlah 90 dan 90 citra. Pada Table 3.2 menunjukkan jumlah data *training* sebelum dan sesudah di augmentasi.

Tabel 3. 2 Data *Training* Sebelum Dan Sesudah Augmentasi

No	Jenis Alpukat	Citra Asli	Augmentasi
1	Alpukat Aligator	120	600
2	Alpukat Madu	120	600
3	Alpukat Kendil	120	600
4	Alpukat Sw01	120	600
5	Alpukat Super	120	600
6	Bukan Daun	120	600
Total		720	3600

Jenis augmentasi yang dilakukan pada penelitian ini adalah sebagai berikut

1. *Flip Horizontal*

Pembalikan gambar secara horizontal dilakukan secara acak untuk menciptakan variasi orientasi kiri-kanan pada daun. Teknik ini membantu model mengenali pola daun yang simetris atau berlawanan arah tanpa mengubah identitas kelas, sehingga memperkaya variasi data pelatihan.

2. *Rotasi Acak*

Rotasi gambar hingga ± 20 derajat memberikan keragaman orientasi citra daun. Dengan adanya variasi rotasi, model belajar untuk mengenali ciri morfologi daun tanpa bergantung pada posisi tertentu dalam citra, sehingga meningkatkan kemampuan generalisasi dalam kondisi pengambilan gambar yang beragam.

3. *Zoom Range*

Perubahan skala citra dengan rentang hingga 15% digunakan untuk mensimulasikan perbedaan jarak pengambilan gambar di lapangan. Teknik ini memungkinkan model beradaptasi dengan ukuran objek daun yang bervariasi, baik yang tampak lebih dekat maupun lebih jauh.

4. Geser *Horizontal* dan *Vertikal*

Pergeseran posisi citra secara *horizontal* maupun *vertikal* hingga 15% dari ukuran gambar dilakukan untuk memberikan variasi letak objek daun di dalam frame. Dengan cara ini, model dilatih agar lebih robust terhadap perubahan posisi daun, sehingga tidak hanya mengandalkan keberadaan objek pada titik tertentu.

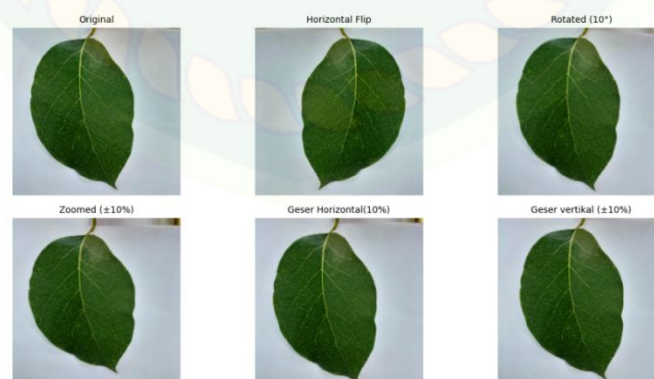
5. *Brightness Range*

Penyesuaian tingkat kecerahan citra dalam rentang 0,85–1,15 dilakukan untuk mensimulasikan kondisi pencahayaan yang berbeda, mulai dari agak redup hingga lebih terang. Variasi ini membantu model agar lebih tahan terhadap perubahan intensitas cahaya yang sering terjadi di lapangan saat pengambilan gambar.

6. *Fill Mode (Nearest)*

Pada saat dilakukan transformasi seperti rotasi, pergeseran, atau zoom, terdapat bagian piksel kosong pada citra. Dengan pengaturan `fill_mode='nearest'`, piksel kosong tersebut diisi menggunakan nilai dari piksel terdekat. Hal ini bertujuan untuk menjaga kontinuitas bentuk dan tekstur daun sehingga tetap realistis bagi model dalam proses pembelajaran.

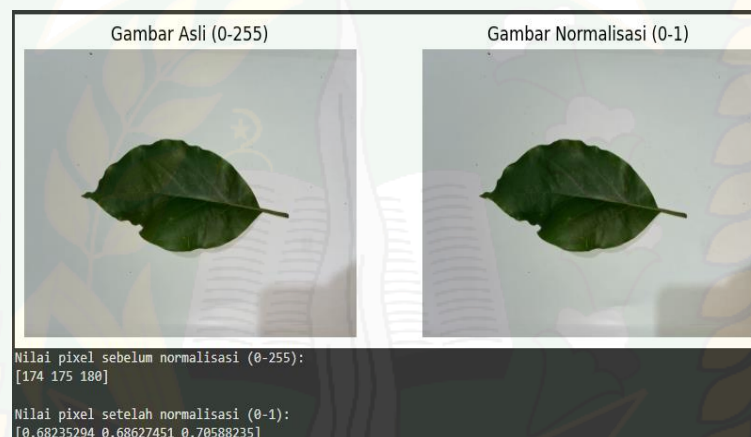
Adanya proses augmentasi ini, variasi citra yang tersedia menjadi lebih beragam, sehingga diharapkan dapat memperkaya representasi data dan meningkatkan kinerja model dalam melakukan klasifikasi. Gambar 3.6 menunjukkan tahapan augmentasi yang digunakan pada penelitian ini.



Gambar 3. 6 Citra Yang Dilakukan Augmentasi

3.2.4 Normalisasi Pixel

Setelah proses *resize*, nilai piksel pada citra diubah ke rentang 0 hingga 1 dengan cara membagi nilai asli piksel (0–255) dengan 255. Proses normalisasi ini dilakukan untuk menjaga kestabilan pelatihan model, sehingga terhindar dari nilai input yang terlalu besar yang dapat memicu ketidakstabilan numerik dan memperlambat proses konvergensi. Dengan nilai piksel yang seragam, model CNN dapat belajar lebih optimal dan menghasilkan bobot yang lebih baik. Pendekatan ini juga telah digunakan pada penelitian klasifikasi tingkat kematangan buah jambu kristal berbasis CNN dan terbukti mampu memberikan akurasi yang tinggi. Perbandingan antara nilai sebelum dan sesudah dilakukan Normalisasi Pixel dapat dilihat pada Gambar 3.7.



Gambar 3. 7 Hasil Rescale

3.2.5 Melatih Model *MobileNetV3*

Pada tahap ini, seperti yang ditunjukkan pada **Tabel 3.2**, dikembangkan metode klasifikasi jenis daun alpukat menggunakan arsitektur *MobileNetV3*, yang dirancang khusus untuk mengekstraksi fitur visual dari citra daun secara efektif dan efisien. Model *MobileNetV3* ini dioptimalkan untuk mengenali pola-pola kompleks pada bentuk daun yang berkaitan dengan berbagai jenis varietas alpukat. Selanjutnya, model yang telah dikembangkan diintegrasikan ke dalam aplikasi berbasis *Flutter*, sehingga pengguna dapat melakukan deteksi secara langsung melalui kamera ponsel atau unggahan gambar. Integrasi ini juga mendukung penggunaan secara real-time dan memudahkan para pengguna, seperti petani maupun peneliti, dalam mengidentifikasi jenis alpukat secara cepat dan

praktis. Dengan demikian, sistem ini diharapkan mampu meningkatkan kecepatan, akurasi, dan efisiensi dalam proses identifikasi jenis daun alpukat.

Tabel 3. 3 Parameter Model MobileNetV3

No	Parameter	Nilai
1.	<i>Optimizer</i>	<i>Adam</i>
2.	<i>Learning Rate</i>	0,0001
3.	<i>Epochs</i>	50
4.	<i>Batch Size</i>	32
5.	<i>Callback</i>	<i>EarlyStopping</i>

3.2.5.1 Tahapan Pelatihan

Pelatihan model *MobileNetV3* pada penelitian ini menggunakan pendekatan *transfer learning* dengan metode *feature extraction* tanpa melakukan tahap *fine-tuning*. Pendekatan *transfer learning* dipilih karena model pra-latih seperti *MobileNetV3* telah memiliki kemampuan representasi fitur yang baik hasil pelatihan pada dataset berskala besar seperti ImageNet. Dengan demikian, proses pelatihan dapat dilakukan lebih efisien dan tetap menghasilkan akurasi yang optimal meskipun jumlah data terbatas.

Pada tahap pelatihan, seluruh lapisan konvolusional pada *base model* dibekukan (*frozen*) agar bobot pra-latih tidak berubah. Hanya bagian akhir dari arsitektur model atau *classifier head* yang dilatih untuk menyesuaikan dengan karakteristik data citra daun alpukat. Tahapan ini memungkinkan model memanfaatkan pengetahuan visual umum dari *MobileNetV3* dan menyesuaikannya secara spesifik terhadap kelas-kelas daun yang digunakan. Strategi ini bertujuan menjaga kestabilan fitur dasar yang telah terbentuk sekaligus mengoptimalkan kemampuan klasifikasi pada domain baru tanpa risiko *overfitting* akibat pembaruan bobot secara menyeluruh.

Tahap *fine-tuning* tidak diterapkan pada penelitian ini karena tujuan utama pelatihan adalah memperoleh model yang efisien dan stabil dengan risiko *overfitting* yang minimal. Mengingat jumlah data citra daun alpukat yang relatif

terbatas, pembukaan kembali seluruh atau sebagian lapisan pada *base model* berpotensi menyebabkan model menyesuaikan bobot secara berlebihan terhadap data pelatihan, sehingga menurunkan kemampuan generalisasi. Selain itu, proses *fine-tuning* memerlukan sumber daya komputasi yang lebih besar serta waktu pelatihan yang lebih panjang. Oleh karena itu, seluruh lapisan konvolusional pada *MobileNetV3* dibiarkan dalam kondisi beku (*frozen*), dan pelatihan difokuskan pada lapisan klasifikasi untuk memperoleh performa optimal dengan efisiensi komputasi yang tinggi.

3.2.5.2 *Optimizer*

Optimizer merupakan algoritma yang digunakan untuk menyesuaikan bobot model berdasarkan gradien yang diperoleh dari fungsi kerugian. Pada penelitian ini digunakan *Adam (Adaptive Moment Estimation)*, yaitu algoritma yang mengombinasikan kelebihan dua metode optimasi sebelumnya, yaitu *AdaGrad* dan *RMSProp*. Pendekatan ini memanfaatkan momentum gradien serta nilai kuadrat gradien guna mempercepat proses konvergensi selama pelatihan model. Adam dikenal efisien karena dapat menyesuaikan laju pembelajaran (*learning rate*) secara adaptif untuk setiap parameter, sehingga mampu bekerja secara stabil pada dataset berukuran besar dan kompleks.

3.2.5.3 *Learning rate*

Learning rate adalah parameter kunci dalam proses pelatihan yang mengatur seberapa besar penyesuaian bobot jaringan pada setiap iterasi. Jika nilai *learning rate* terlalu tinggi, model dapat gagal mencapai konvergensi, sementara nilai yang terlalu rendah dapat menyebabkan proses pelatihan berlangsung sangat lambat. Dalam penelitian ini digunakan *learning rate* sebesar 0.0001 ($1e-4$) selama proses pelatihan bagian klasifikasi (*training head*). Nilai ini dipilih karena dianggap memberikan keseimbangan antara kecepatan pembelajaran dan kestabilan model.

3.2.5.4 *Batch Size*

Batch size menunjukkan jumlah sampel data yang diproses sekaligus sebelum bobot model diperbarui. Nilai *batch size* yang terlalu besar dapat meningkatkan kecepatan pelatihan, namun berisiko menurunkan kemampuan

generalisasi model. Sebaliknya, *batch size* yang terlalu kecil dapat memperlambat proses pelatihan. Dalam penelitian ini digunakan *batch size* sebesar 32, karena ukuran tersebut dianggap seimbang antara efisiensi komputasi dan kestabilan gradien. Ukuran batch ini juga sesuai dengan kapasitas GPU pada platform Google Colab yang digunakan dalam eksperimen.

3.2.5.5 Jumlah Epoch

Epoch merupakan jumlah siklus penuh di mana seluruh data pelatihan digunakan untuk memperbarui bobot model. Pada penelitian ini, proses pelatihan dilakukan sebanyak 50 *epoch* dengan fokus pada bagian klasifikasi (*training head*), karena lapisan konvolusional utama pada *MobileNetV3* dibekukan. Selain itu, digunakan mekanisme *callback* berupa *EarlyStopping* untuk menghentikan proses pelatihan secara otomatis apabila nilai kehilangan (*validation loss*) tidak mengalami perbaikan dalam beberapa *epoch* berturut-turut. Pendekatan ini bertujuan menjaga efisiensi waktu pelatihan sekaligus mencegah terjadinya *overfitting* pada data pelatihan.

3.2.5.6 Callback

Selama proses pelatihan, digunakan *callback EarlyStopping* untuk mengoptimalkan efisiensi dan mencegah *overfitting*. *Callback* ini secara otomatis menghentikan proses pelatihan ketika nilai *validation loss* tidak menunjukkan perbaikan selama beberapa *epoch* berturut-turut. Dengan demikian, model dapat mempertahankan bobot terbaik yang telah dicapai tanpa melanjutkan pelatihan yang tidak memberikan peningkatan performa, sekaligus menghemat waktu komputasi.

3.2.6 Evaluasi

Untuk menilai kualitas suatu sistem, dapat digunakan *confusion matrix*, yang berfungsi untuk menganalisis tingkat akurasi, presisi, recall, dan *F1-Score*. Pengujian menggunakan *confusion matrix* ini dilakukan untuk mengklasifikasikan lima jenis tanaman alpukat berdasarkan daun, yaitu Alpukat Aligator, alpukat Kendil, alpukat madu, alpukat Super, dan alpukat SW01. Matriks ini memiliki peran penting dalam mengevaluasi hasil klasifikasi secara menyeluruh menggunakan metode *MobileNetV3*. Dengan menggunakan *confusion matrix*,

kinerja model dalam mengenali dan mengklasifikasikan jenis daun alpukat dapat diukur secara lebih detail.

Tabel 3. 4 Evaluasi *Confusion Matrix*

Hasil Klasifikasi Kelas Sebelumnya	Positif	Negatif
Positif	TP	FP
Negatif	FN	TN

Pada hasil evaluasi yang ditampilkan pada **Tabel 3. 3** diatas, terdapat empat istilah yang digunakan untuk mempresentasikan keluaran proses klasifikasi. Istilah tersebut meliputi *True Positive* (TP), yaitu ketika suatu citra berhasil diprediksi dengan benar sesuai kelasnya, misalnya citra daun Alpukat Aligator dikenali sebagai kelas Alpukat Aligator. *True Negatif* (TN) menunjukkan kondisi dimana citra bukan Alpukat Aligator dan tidak diprediksi sebagai Alpukat Aligator, *False Positive* (FP) yang dalam penelitian ini contohnya citra bukan Alpukat Aligator diprediksi sebagai Alpukat Aligator, dan *False Negative* (FN) yang dalam penelitian ini contohnya citra daun alpukat alligator diprediksi sebagai kelas lain.

3.3 Analisa Kebutuhan Sistem

Analisa kebutuhan sistem adalah tahapan mendalam yang bertujuan untuk mengenali serta mendeskripsikan secara detail komponen-komponen yang dibutuhkan dalam proses pengembangan suatu sistem. Dalam penelitian ini, penulis menggunakan berbagai alat dan bahan sebagai berikut:

3.3.1 Spesifikasi Kebutuhan *Hardware* Dan *Software*

Penelitian ini memerlukan seperangkat alat untuk membangun serta mengoperasikan system yang direncanakan. Alat tersebut terbagi menjadi dua kategori, yaitu *hardware* dan *software*.

A. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini disesuaikan agar mampu menjalankan aplikasi secara optimal, dengan spesifikasi yang diterapkan dalam proses penelitian sebagai berikut:

1. Processor : Intel(R) Core(TM) i3-6006U
2. RAM : 8 GB
3. Startup Disk : Windows
4. System Tipe : 64-bit operating system

B. Perangkat Lunak (*Software*)

Berikut adalah spesifikasi teknis terperinci perangkat lunak (software) yang dibutuhkan untuk pengembangan aplikasi ini:

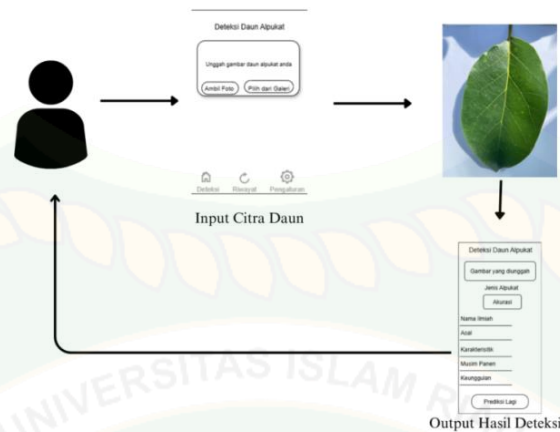
1. Google Colab : Python 3.10.8
2. Browser : Google Chrome
3. Frontend : Flutter
4. Database : SQLite
5. Backend API : Flask
6. Visual Studio Code

3.4 Pengembangan Sistem

Penelitian ini akan mengembangkan sebuah aplikasi mobile yang bertujuan untuk membantu pengguna dalam mengenali berbagai jenis daun alpukat. Aplikasi ini diharapkan dapat menjadi sarana yang bermanfaat untuk mempermudah masyarakat dalam mengidentifikasi dan memahami perbedaan jenis daun alpukat, sehingga dapat meningkatkan pengetahuan tentang karakteristik masing-masing jenis.

Proses penggunaan aplikasi ini dapat dilihat pada Gambar 3.8. Pertama, buka aplikasi pada perangkat mobile. Setelah aplikasi terbuka, pengguna akan diarahkan untuk mengambil foto atau memilih gambar daun alpukat dari galeri yang ingin diklasifikasikan. Setelah proses ini selesai, sistem secara otomatis akan menampilkan jenis daun alpukat yang sesuai. Dengan demikian, pengguna tidak

perlu lagi menebak atau bertanya kepada orang lain untuk mengetahui jenis daun alpukat tersebut.



Gambar 3. 8 Pengembangan Sistem

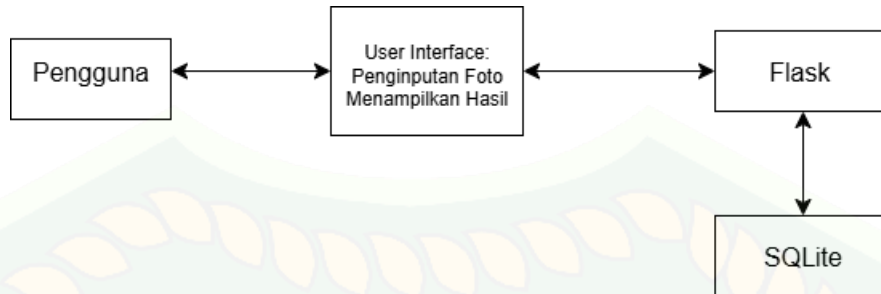
3.5 Perancangan Sistem

Pada bagian perancangan system ini, akan diberikan penjelasan rinci tentang tahapan dan proses perancangan sistem yang akan dikembangkan. Proses ini dapat ditunjukkan dalam paparan berikut:

3.5.1 Arsitektur Sistem

Pada tahap ini, sistem dirancang untuk menghubungkan aplikasi frontend Flutter dengan backend Flask yang menjalankan model machine learning TensorFlow Lite (.tflite). Pengguna memasukkan foto melalui aplikasi Flutter, yang kemudian dikirim ke backend Flask melalui API HTTP. Backend Flask menerima foto, memprosesnya dengan model .tflite untuk melakukan prediksi, dan menyimpan hasil prediksi beserta data gambar ke dalam database SQLite. Hasil prediksi dan informasi terkait kemudian dikirim kembali ke Flutter dalam format JSON untuk ditampilkan pada antarmuka pengguna. Database SQLite hanya diakses oleh backend sebagai penyimpanan, sedangkan Flutter hanya menerima data hasil proses dari backend, sehingga tidak terhubung langsung dengan database. Proses ini memastikan pengolahan model dilakukan di backend, komunikasi data menggunakan format JSON, dan antarmuka Flutter berfungsi sebagai media input dan tampilan hasil yang mudah digunakan. Sistem ini juga menyediakan riwayat prediksi yang dapat diakses oleh pengguna untuk

melihat hasil-hasil sebelumnya. Tahap arsitektur sistem ini dapat dilihat pada Gambar 3.9.



Gambar 3. 9 Arsitektur Sistem

3.5.2 Diagram Konteks

Dalam penelitian ini, diagram konteks digunakan untuk memperlihatkan hubungan aliran data antara pengguna sebagai entitas eksternal dan system klasifikasi jenis alpukat berbasis mobileNetV3 melalui aplikasi mobile. Diagram ini secara spesifik menggambarkan interaksi utama antar pengguna yang menginput foto daun alpukat melalui aplikasi dan system yang memproses serta memberikan hasil klasifikasi. Dengan demikian, diagram konteks pada penelitian ini menyoroti proses inti system dan interaksi langsung antara pengguna dan aplikasi klasifikasi jenis daun alpukat, seperti yang ditunjukkan pada Gambar 3.10.

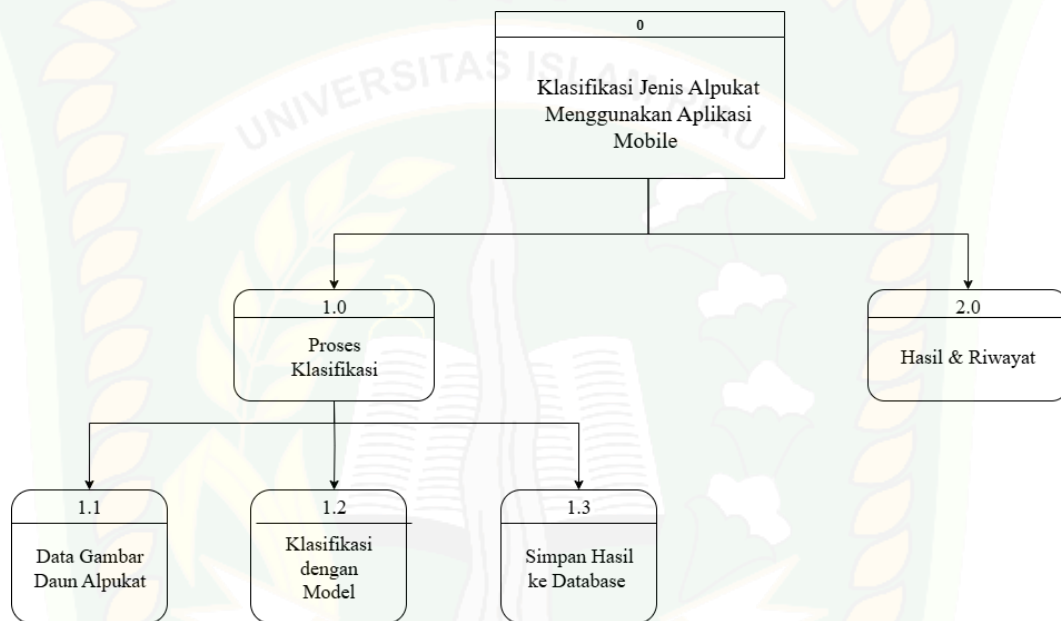


Gambar 3. 10 Diagram Konteks

3.5.3 Hierarchy Chart

Hierarchy chart adalah representasi visual yang memecah sistem klarifikasi daun alpukat menjadi komponen-komponen yang lebih sederhana dan terstruktur. Diagram ini menjelaskan bagaimana setiap proses, mulai dari

pengolahan data hingga klarifikasi, saling terhubung dan mendukung fungsi utama sistem. Meskipun seluruh bagian bekerja sama untuk menghasilkan prediksi, masing-masing memiliki peran dan tingkat tanggung jawab yang berbeda. Dengan *hierarchy chart*, alur kerja sistem dapat dipahami secara lebih jelas, mulai dari input data gambar daun, proses *preprocessing*, klasifikasi menggunakan MobileNetV3, hingga penyajian hasil kepada pengguna. Gambar 3.11 memberikan ilustrasi visual yang menunjukkan bagaimana sistem ini dirancang.



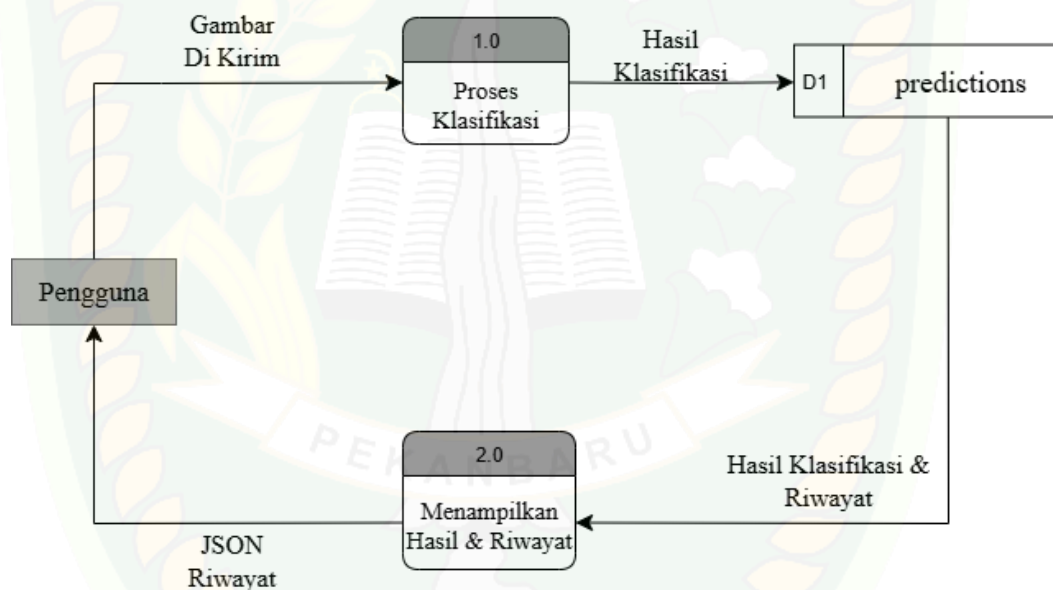
Gambar 3. 11 Hierarchy Chart

3.5.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan model logis yang digunakan untuk menggambarkan sumber, alur, dan tujuan data dalam suatu sistem. DFD menampilkan bagaimana data disimpan serta bagaimana proses mengubah data tersebut. Selain itu, DFD juga memperlihatkan keterkaitan antara data dan proses yang berlangsung di dalam sistem. Pada aplikasi klasifikasi jenis tanaman alpukat berdasarkan citra daun, terdapat beberapa tingkatan proses yang menggambarkan tahapan pengolahan data, mulai dari penerimaan gambar, proses klasifikasi menggunakan model MobileNetV3, hingga penyajian hasil prediksi kepada pengguna.

3.5.4.1 DFD level 0

Diagram ini menggambarkan alur sistem klasifikasi daun alpukat. Pada sistem ini, pengguna mengunggah gambar daun melalui aplikasi. Gambar tersebut diproses pada tahap Proses Klasifikasi dengan memanfaatkan model machine learning (Model_Comp.tflite). setelah dilakukan klasifikasi, hasil klasifikasi jenis daun alpukat disimpan ke dalam database SQLite (predictions). Selanjutnya, melalui proses Hasil & Riwayat, hasil klasifikasi tersebut diambil kembali dan disajikan kepada pengguna dalam bentuk hasil deteksi dan riwayat klasifikasi. Dengan demikian, pengguna dapat mengetahui jenis alpukat dari gambar yang diunggah serta melihat riwayat deteksi yang telah dilakukan sebelumnya. Gambar 3.12 menunjukkan proses DFD Level 0.

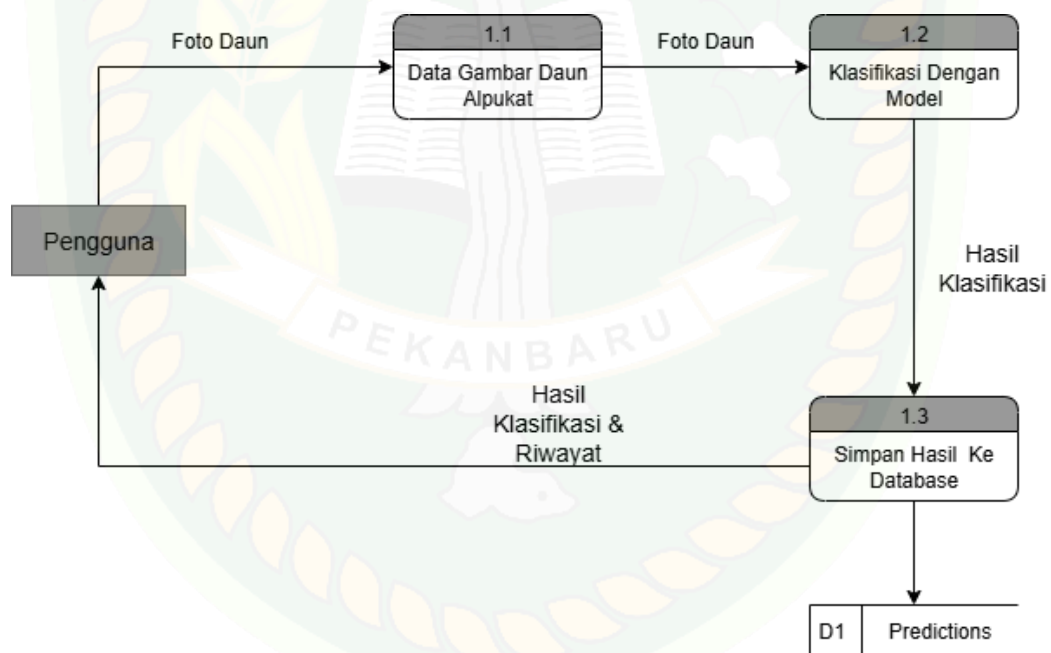


Gambar 3. 12 DFD Level 0

3.5.4.2 DFD Level 1

DFD Level 1 pada sistem klasifikasi daun alpukat ini menggambarkan alur data yang terjadi mulai dari pengguna memberikan input hingga hasil klasifikasi ditampilkan. Proses dimulai ketika pengguna mengambil atau mengunggah foto daun alpukat melalui aplikasi. Foto ini kemudian masuk ke Proses 1.1 – Data Gambar Daun Alpukat, di mana gambar diterima dan dipastikan formatnya sesuai sehingga siap untuk

diproses lebih lanjut. Selanjutnya, foto daun yang sudah valid dikirim ke Proses 1.2 – Klasifikasi dengan Model, yang menggunakan model MobileNetV3 untuk menganalisis citra dan mengklasifikasikan jenis daun alpukat. Hasil dari proses ini berupa label jenis daun, beserta nilai confidence yang menunjukkan tingkat keyakinan model terhadap prediksi tersebut. Hasil klasifikasi kemudian diteruskan ke Proses 1.3 – Simpan Hasil ke Database, di mana data tersebut disimpan dalam database lokal SQLite. Setelah data tersimpan, aplikasi akan menampilkan hasil klasifikasi terkini kepada pengguna sekaligus memberikan akses ke riwayat klasifikasi yang sudah pernah dilakukan sebelumnya. Dengan alur ini, pengguna tidak hanya dapat mengetahui hasil deteksi daun yang baru saja dilakukan, tetapi juga dapat melihat catatan hasil klasifikasi sebelumnya, sehingga proses identifikasi menjadi lebih terstruktur, akurat, dan mudah diakses. Gambar 3.13 menunjukkan proses DFD Level 1.



Gambar 3. 13 DFD Level 1

3.5.5 Desain Database

Desain database yang dibangun mengacu pada Tabel 3.6 Tabel Predictions. Nama Database = database.db.

Tabel 3. 5 Tabel Predictions

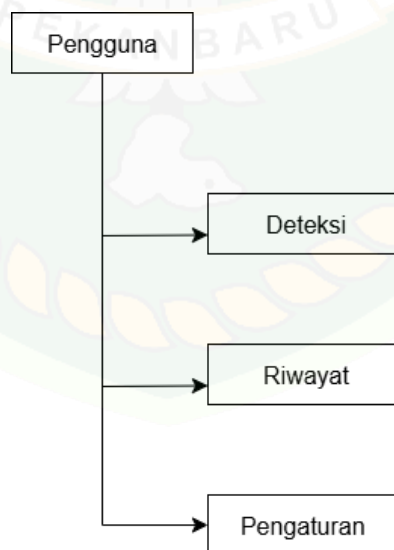
No	Field	Data Type	Ket.
1.	Id	Integer	Primary key
2.	Timestamp	Text Not Null	
3.	Predicted_label	Text Not Null	
4.	Confidence	Real Not Null	
5.	Image_filename	text	

3.5.6 Perancangan Antarmuka

Perancangan antarmuka memiliki peran penting dalam pengembangan aplikasi klasifikasi daun alpukat. Antarmuka berfungsi sebagai penghubung visual antar pengguna dan sistem. Dalam sistem ini, antarmuka dirancang dengan menyusun menu yang jelas dan sederhana agar pengguna dapat dengan mudah mengunggah gambar, melihat hasil klasifikasi, serta mengakses riwayat klasifikasi daun alpukat dengan nyaman.

3.5.6.1 Struktur Menu Antarmuka Program

Desain antarmuka sistem pengenalan citra daun alpukat, bisa dilihat dari struktur menu yang terdapat pada Gambar 3.14.

**Gambar 3. 14** Struktur Menu Antarmuka Program

Pada Gambar 3.13 diatas menampilkan menu- menu dari aplikasi deteksi jeis daun alpukat. Penjelasan rinci mengenai Gambar 3.13 diuraikan sebagai berikut :

- a. Pengguna : Halaman utama menjadi titik awal interaksi pengguna dengan aplikasi. Di sini, pengguna dapat mengakses berbagai fitur utama aplikasi seperti deteksi daun alpukat, melihat riwayat klaisifikasi, dan melakukan pengaturan. Antarmuka pada bagian ini dirancang agar mudah digunakan dan intuitif, memudahkan navigasi antar menu.
- b. Menu Deteksi : fitur utama yang memungkinkan pengguna melakukan proses klasifikasi jenis daun alpukat. Pengguna dapat memilih gambar dari kamera atau galeri, kemudian system akan memproses gambar tersebut menggunakan model klasifikasi dan menampilkan hasilnya.
- c. Menu Riwayat : berisi catatan hasil klasifikasi yang telah dilakukan sebelumnya. Pengguna dapat melihat kembali waktu deteksi, gambar yan digunakan, dan hasil klasifikasi jenis daun alpukat. Fitur ini membantu dalam pelacakan dan dokumentasi hasil deteksi.
- d. Menu Pengaturan : bagian yang memungkinkan pengguna menyesuaikan preferensi aplikasi, seperti mode gelap (*dark mode*) atau melihat informasi terkait aplikasi.

3.5.6.2 Desain Sistem

Desain system pada aplikasi ini dirancang untuk memberikan pengalaman pengguna yang optimal dalam melakukan klasifikasi jenis daun alpukat. Dengan memanfaatkan teknologi kexerdasan buatan, aplikasi ini mampu menganalisis berbagai jenis daun alpukat secara akurat. Hasil analisi kemudian disajikan dalam bentuk visual sederhana dan informative, sehingga pengguna dapat dengan cepat memperoleh informasi terkait jenis daun alpukat yang terdeteksi. Tampilan aplikasi klasifikasi daun alpukat ini ditunjukkan pada Gambar 3.15 dan Gambar 3.16.

Deteksi Daun Alpukat

Unggah gambar daun alpukat anda

Ambil Foto Pilih dari Galeri

Deteksi Riwayat Pengaturan

Gambar 3. 15 Desain Input

Deteksi Daun Alpukat

Gambar yang diunggah

Jenis Alpukat

Akurasi 99.9%

Nama Ilmiah	X (30)
Asal	X (50)
Karakteristik	X (200)
Musim Panen	X (50)
Keunggulan	X (80)

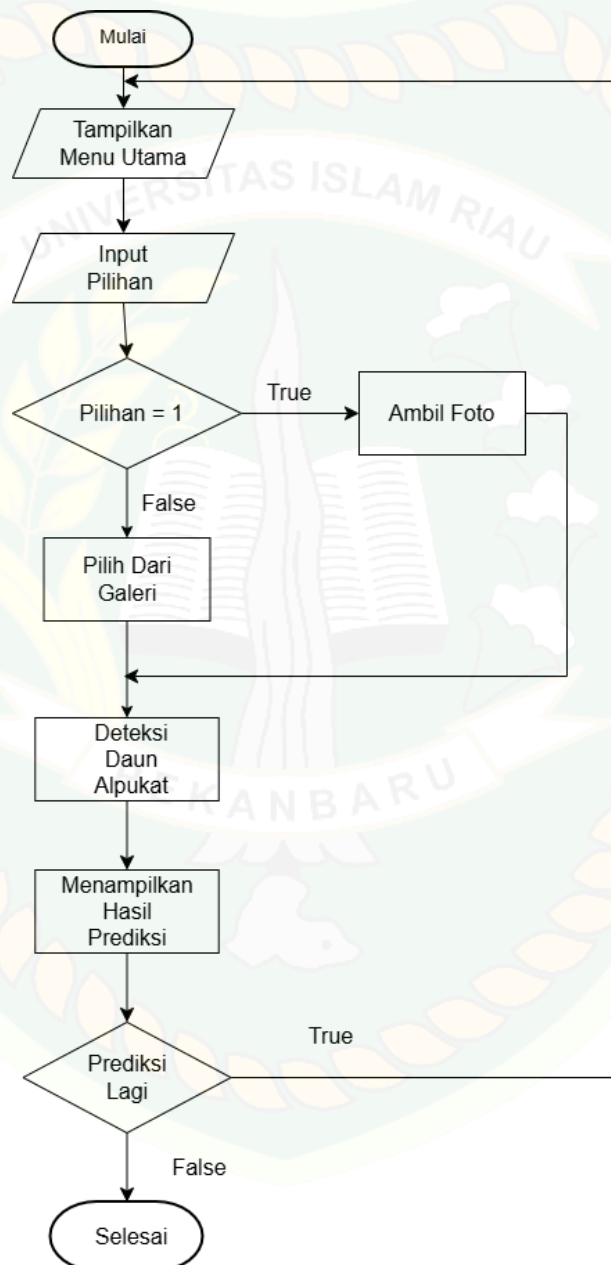
Prediksi Lagi

Gambar 3. 16 Desain Output

3.5.7 Desain Logika Program

Desain logika program menggambarkan alur kerja yang akan diterapkan dalam pengembangan aplikasi dengan menggunakan flowchart sebagai alat bantu. Perancangan ini bertujuan untuk memvisualisasikan tahapan proses, mulai dari input data hingga menghasilkan output, sehingga memudahkan pengguna dalam memahami dan memanfaatkan sistem. Pada tahap ini, alur proses klasifikasi daun alpukat digambarkan melalui sebuah flowchart, sebagaimana ditunjukkan pada Gambar 3.17.

Ketika aplikasi mobile klasifikasi daun alpukat dijalankan untuk pertama kalinya, sistem akan memproses data citra daun alpukat beserta informasi jenis-jenisnya. Pengguna kemudian dapat mengunggah gambar daun alpukat yang ingin diidentifikasi melalui aplikasi mobile ini. Setelah proses analisis selesai, jenis daun alpukat akan ditampilkan, sehingga pengguna dapat dengan mudah mengetahui jenis alpukat tersebut hanya dengan menggunakan perangkat mereka.



Gambar 3. 17 FlowChart Deteksi Alpukat

BAB IV

HASIL DAN PEMBAHASAN

4.1 Akuisisi Citra

Tahap awal dalam pengembangan aplikasi klasifikasi jenis daun alpukat adalah proses pengumpulan dataset yang memadai serta representatif terhadap kondisi sebenarnya di lapangan. Dalam penelitian ini, data primer diperoleh melalui pengambilan citra daun alpukat secara langsung menggunakan kamera ponsel. Pengambilan gambar dilakukan secara manual dengan memperhatikan variasi kondisi pencahayaan alami, sudut pengambilan gambar, serta perbedaan kondisi fisik daun pada setiap varietas. Pendekatan ini bertujuan untuk memperoleh citra yang mencerminkan keragaman visual nyata dari setiap jenis alpukat yang diteliti. Dataset awal pada penelitian ini terdiri atas lima kelas varietas daun alpukat, yaitu alpukat Madu, Kendil, SW01, Aligator, dan Super, serta satu kelas tambahan berupa bukan daun alpukat. Masing-masing kelas terdiri dari 150 citra asli, mencakup alpukat Madu, Kendil, SW01, Aligator, Super, dan bukan daun. Dengan demikian, total keseluruhan data primer yang berhasil dikumpulkan berjumlah 900 citra daun alpukat. Gambar 4.1 menunjukkan total citra daun alpukat. Setiap citra diambil secara langsung di lapangan dengan variasi pencahayaan dan sudut pengambilan untuk memastikan keberagaman visual data serta meningkatkan kemampuan model dalam mengenali karakteristik daun dari setiap varietas alpukat secara akurat.

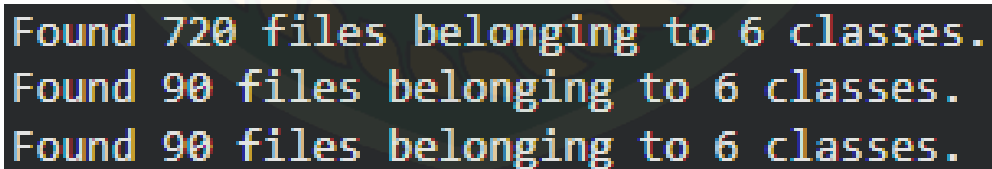
```
734 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
735 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
736 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
737 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
738 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
739 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
740 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
741 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
742 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
743 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
744 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
745 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
746 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
747 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
748 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
749 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/  
750 /content/drive/MyDrive/Colab Notebooks/Daun_Alpukat/Alpukat_Aligator/
```

Gambar 4. 1 Hasil Pengumpulan Data Primer

Seluruh citra daun alpukat yang diperoleh dari proses pengumpulan data kemudian diorganisasikan ke dalam struktur folder yang sistematis, di mana setiap kelas daun ditempatkan pada subfolder tersendiri untuk memudahkan proses pelabelan dan pengolahan data selanjutnya. Dataset tersebut kemudian dibagi menjadi tiga bagian, yaitu 80% untuk data pelatihan (*training*), 10% untuk data validasi (*validation*), dan 10% untuk data pengujian (*testing*).

Untuk meningkatkan jumlah data pelatihan sekaligus memperkaya variasi visual yang tersedia bagi model, dilakukan proses augmentasi citra secara *offline* hanya pada data pelatihan. Setiap citra asli dalam data pelatihan diaugmentasi sebanyak lima kali dengan menerapkan berbagai transformasi, seperti rotasi hingga ± 20 derajat, pergeseran posisi (*width* dan *height shift*), perubahan tingkat kecerahan (*brightness*), pembesaran (*zoom*), serta pembalikan horizontal (*horizontal flip*). Proses ini menghasilkan beragam variasi citra baru yang tetap mempertahankan label kelas aslinya.

Sementara itu, data validasi dan data uji tidak mengalami proses augmentasi untuk menjaga keaslian dan konsistensi dalam evaluasi kinerja model. Setiap citra pada kedua subset tersebut hanya melalui proses penyesuaian ukuran (*resize*) agar sesuai dengan dimensi input model, yaitu 224×224 piksel. Dengan demikian, jumlah total dataset setelah proses augmentasi bertambah secara signifikan pada bagian data pelatihan, yang berperan penting dalam meningkatkan kemampuan model MobileNetV3 untuk mengenali pola visual daun alpukat pada berbagai kondisi pencahayaan, orientasi, dan posisi citra. Gambar 4.2 menunjukkan hasil data split, yang dimana hanya data pelatihan saja yang dilakukan augmentasi.



```
Found 720 files belonging to 6 classes.  
Found 90 files belonging to 6 classes.  
Found 90 files belonging to 6 classes.
```

Gambar 4. 2 Hasil Data Split

4.2 Hasil Pelatihan Model

Model yang digunakan dalam penelitian ini berbasis arsitektur MobileNetV3 yang berfungsi sebagai *feature extractor* untuk mengenali pola visual pada citra daun alpukat. Lapisan akhir model dimodifikasi dengan *fully connected layer* beraktivasi *softmax* guna melakukan klasifikasi antar kelas. Proses pelatihan menggunakan optimizer Adam dengan *learning rate* 0,0001 serta diterapkan *fine-tuning* pada beberapa lapisan akhir untuk menyesuaikan bobot terhadap karakteristik data. Penggunaan *callback functions* seperti *EarlyStopping* dan *ReduceLROnPlateau* membantu mencegah *overfitting*. Hasil pelatihan menunjukkan bahwa konfigurasi ini menghasilkan model dengan performa dan kemampuan generalisasi yang baik terhadap variasi citra daun alpukat.

4.2.1 Hasil Pelatihan Model *MobileNetV3Large*

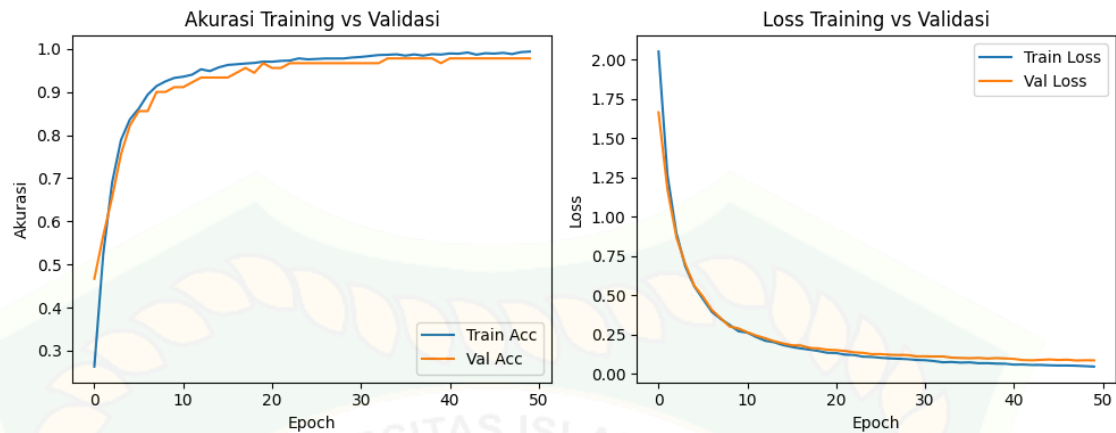
Pelatihan model dengan arsitektur *MobileNetV3Large* dilakukan untuk mengukur efektivitas konfigurasi model dalam mengklasifikasikan citra daun alpukat secara efisien dengan kompleksitas yang rendah. Proses pelatihan dilaksanakan tanpa tahap *fine-tuning*, sehingga bobot dari model dasar (base model) yang telah dilatih pada dataset ImageNet tetap dibekukan dan hanya lapisan klasifikasi bagian atas yang dilatih ulang. Pendekatan ini bertujuan untuk memanfaatkan kemampuan ekstraksi fitur dari model pra-latih sekaligus menjaga efisiensi komputasi. Selama proses pelatihan, digunakan *optimizer* Adam dengan *learning rate* sebesar 0,0001 agar pembaruan bobot berlangsung stabil. Selain itu, diterapkan *callback EarlyStopping* untuk menghentikan pelatihan secara otomatis apabila performa validasi tidak menunjukkan peningkatan dalam beberapa epoch. Strategi ini bertujuan untuk mencegah *overfitting* serta memastikan model mencapai konvergensi optimal dengan efisiensi waktu yang baik. Hasil pengujian menunjukkan bahwa konfigurasi ini mampu mencapai keseimbangan antara efisiensi pelatihan dan akurasi klasifikasi yang tinggi, sebagaimana ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Hasil Pengujian Model *MobileNetV3Large*

<i>Optimizer</i>	<i>Learning Rate</i>	Akurasi	Loss	Akurasi Validasi	Loss Validasi	Best Epoch
<i>Adam</i>	0,0001	96,67%	0,1139	97,78%	0,0841	48

Berdasarkan hasil pelatihan yang ditunjukkan pada Tabel 4.2, model *MobileNetV3Large* dengan penggunaan *optimizer* Adam dan nilai *learning rate* sebesar 0,0001 memberikan performa terbaik. Model ini mencapai akurasi pelatihan sebesar 96,67% dan akurasi validasi tertinggi sebesar 97,78% dengan nilai *loss* validasi terendah, yaitu 0,0841 pada epoch ke-48. Hasil tersebut menunjukkan bahwa *optimizer* Adam mampu menghasilkan proses pembelajaran yang stabil dan konvergen dengan baik pada model ini. Kemampuan Adam dalam menyesuaikan laju pembelajaran (*adaptive learning rate*) serta mengombinasikan momentum menjadikannya lebih efektif dalam mempercepat proses konvergensi sekaligus menjaga kestabilan nilai *loss*. Dengan demikian, kombinasi antara *optimizer* Adam dan *learning rate* 0,0001 terbukti optimal dalam menghasilkan model *MobileNetV3Large* yang memiliki tingkat akurasi tinggi serta generalisasi yang baik terhadap data validasi.

Gambar 4.3 menampilkan grafik perkembangan akurasi dan *loss* pada proses pelatihan serta validasi model *MobileNetV3Large* yang menggunakan *optimizer* Adam dengan *learning rate* 0,0001. Berdasarkan grafik tersebut, terlihat adanya peningkatan akurasi yang signifikan serta penurunan nilai *loss* secara konsisten seiring bertambahnya jumlah *epoch*. Model menunjukkan performa terbaik pada epoch ke-48 dengan akurasi validasi mencapai 96,67% dan *loss* validasi sebesar 0,0841. Pada tahap awal pelatihan, kurva akurasi mengalami kenaikan tajam dan *loss* menurun drastis akibat proses penyesuaian bobot yang masih berlangsung. Seiring waktu, kedua kurva menunjukkan stabilitas yang baik dengan perbedaan yang kecil antara data *training* dan *validasi*, menandakan bahwa model mampu mempelajari pola data dengan efektif tanpa mengalami *overfitting*. Pola ini juga memperlihatkan bahwa mekanisme *adaptive learning rate* pada *optimizer* Adam berhasil menjaga kestabilan pembaruan bobot dan mempercepat konvergensi model. Secara keseluruhan, grafik menunjukkan bahwa model *MobileNetV3Small* mampu mencapai keseimbangan antara akurasi tinggi dan generalisasi yang baik terhadap data validasi.



Gambar 4. 3 Hasil Grafik *MobileNetV3Large*

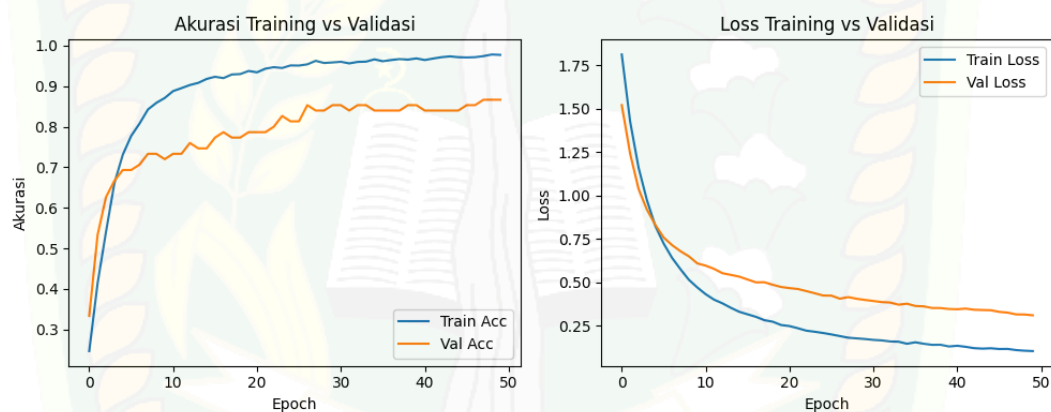
4.2.2 Hasil Pelatihan Model *MobileNetV3Small*

Model *Convolutional Neural Network (CNN)* dengan arsitektur *MobileNetV3* dirancang untuk mengoptimalkan proses ekstraksi fitur secara efisien dan mendalam melalui kombinasi blok konvolusi serta mekanisme *bottleneck* berlapis. Peningkatan kompleksitas jaringan ini memungkinkan model untuk mempelajari representasi fitur yang lebih abstrak dan kontekstual dari citra daun alpukat. Dengan demikian, model mampu mengenali pola visual yang halus, seperti bentuk tepi, tekstur permukaan, dan variasi warna pada daun, yang berkontribusi terhadap peningkatan akurasi dalam proses klasifikasi jenis tanaman alpukat. Selain itu, penggunaan arsitektur *MobileNetV3* juga memberikan keseimbangan antara performa dan efisiensi komputasi, sehingga model tetap ringan namun memiliki kemampuan generalisasi yang tinggi terhadap data baru. Hasil pengujian menunjukkan bahwa konfigurasi model yang digunakan mampu mencapai kinerja pelatihan yang optimal. Rincian mengenai hasil pelatihan dan evaluasi model disajikan pada Tabel 4.2.

Tabel 4. 2 Hasil Pengujian Model *MobileNetV3Small*

<i>Optimizer</i>	<i>Learning Rate</i>	Akurasi	Loss	Akurasi Validasi	Loss Validasi	Best Epoch
<i>Adam</i>	0,0001	90%	0,2582	88,89%	0,3007	50

Berdasarkan hasil pelatihan yang disajikan pada Tabel 4.2, model *MobileNetV3Small* dengan penggunaan *optimizer* Adam dan nilai *learning rate* sebesar 0,0001 menunjukkan performa yang cukup baik. Model ini mencapai akurasi pelatihan sebesar 90% dan akurasi validasi sebesar 88,89% pada *epoch* ke-50, dengan nilai *loss* validasi terendah sebesar 0,3007. Hasil tersebut mengindikasikan bahwa *optimizer* Adam mampu menjaga kestabilan proses pembelajaran dan membantu model beradaptasi dengan pola data secara efektif. Mekanisme penyesuaian *adaptive learning rate* yang dimiliki Adam memungkinkan proses pelatihan berlangsung konvergen dengan baik, sehingga konfigurasi ini dinilai cukup optimal dalam menghasilkan model *MobileNetV3Small* yang memiliki keseimbangan antara akurasi dan generalisasi terhadap data validasi.



Gambar 4. 4 Hasil Grafik *MobileNetV3Small*

Gambar 4. 4 memperlihatkan grafik perkembangan akurasi dan *loss* selama proses pelatihan serta validasi model *MobileNetV3Small* yang menggunakan *optimizer* Adam dengan *learning rate* sebesar 0,0001. Berdasarkan grafik tersebut, terlihat adanya peningkatan akurasi yang signifikan pada tahap awal pelatihan, diikuti dengan penurunan *loss* secara konsisten hingga akhir *epoch*. Model mencapai kestabilan performa pada *epoch* ke-50, di mana kurva akurasi menunjukkan kecenderungan mendekati nilai maksimum dan *loss* semakin menurun dengan laju yang stabil. Perbedaan antara akurasi pelatihan dan validasi relatif kecil, menandakan bahwa model mampu mempelajari pola data dengan baik tanpa mengalami *overfitting*. Selain itu, tren penurunan *loss* yang

konsisten menunjukkan bahwa mekanisme *adaptive learning rate* pada *optimizer Adam* bekerja efektif dalam menjaga konvergensi dan kestabilan pembaruan bobot selama proses pelatihan. Secara keseluruhan, grafik ini menggambarkan bahwa model *MobileNetV3Small* mampu mencapai keseimbangan antara akurasi dan generalisasi yang baik terhadap data validasi.

4.2.3 Analisis Pelatihan Model

Berdasarkan hasil pengujian pada Tabel 4.3, performa model *MobileNetV3* sangat dipengaruhi oleh kombinasi antara arsitektur jaringan, jenis *optimizer*, serta nilai *learning rate* yang digunakan selama proses pelatihan. Kedua varian arsitektur, yaitu *MobileNetV3Small* dan *MobileNetV3Large*, menunjukkan karakteristik yang berbeda dalam hal stabilitas pelatihan, kemampuan konvergensi, serta tingkat generalisasi terhadap data validasi.

Pada konfigurasi *MobileNetV3Small* dengan *optimizer Adam* dan *learning rate* sebesar 0,0001, model menunjukkan keseimbangan yang baik antara efisiensi pelatihan dan akurasi klasifikasi. Model ini mencapai akurasi pelatihan sebesar 90% dan akurasi validasi sebesar 88,89% pada *epoch* terbaik ke-50 dengan nilai *loss* yang relatif rendah. Pola peningkatan akurasi yang stabil serta penurunan nilai *loss* pada data pelatihan dan validasi menunjukkan bahwa proses pembelajaran berlangsung secara konvergen dan terkendali. Selisih yang relatif kecil antara akurasi pelatihan dan validasi menunjukkan bahwa model memiliki kemampuan *generalization* yang baik dan tidak menunjukkan indikasi *overfitting* yang signifikan. Hal ini menandakan bahwa *MobileNetV3Small* mampu mengenali pola visual pada citra daun alpukat secara efektif dengan kompleksitas arsitektur yang ringan dan efisien untuk implementasi pada perangkat mobile.

Sementara itu, *MobileNetV3Large* dengan konfigurasi *optimizer Adam* dan *learning rate* yang sama menghasilkan performa yang lebih tinggi, dengan akurasi pelatihan sebesar 96,67% dan akurasi validasi sebesar 97,78% pada *epoch* terbaik ke-48. Nilai akurasi validasi yang sebanding, bahkan sedikit lebih tinggi dibandingkan akurasi pelatihan, menunjukkan bahwa model memiliki kemampuan *generalization* yang sangat baik dan tidak mengalami *overfitting*. Penurunan nilai

loss yang konsisten pada data pelatihan dan validasi juga menunjukkan bahwa proses pembelajaran berjalan secara stabil. Arsitektur *MobileNetV3Large* dengan jumlah parameter yang lebih besar memungkinkan ekstraksi fitur yang lebih representatif terhadap karakteristik visual daun alpukat sehingga meningkatkan kinerja klasifikasi secara keseluruhan. Namun, kompleksitas model yang lebih tinggi tetap memerlukan pengendalian selama proses pelatihan untuk menjaga kestabilan model.

Berdasarkan perbandingan kedua arsitektur, *MobileNetV3Large* unggul dalam hal akurasi dan kemampuan representasi fitur, sedangkan *MobileNetV3Small* lebih efisien dari sisi komputasi dan waktu pelatihan. Kedua model tidak menunjukkan perbedaan yang mencolok antara akurasi pelatihan dan validasi, sehingga dapat disimpulkan bahwa keduanya memiliki kemampuan *generalization* yang baik. Perbedaan performa ini menunjukkan adanya kompromi antara tingkat akurasi dan efisiensi dalam pemilihan arsitektur model.

Secara keseluruhan, penggunaan *optimizer Adam* dengan *learning rate* sebesar 0,0001 terbukti efektif dalam melatih kedua varian *MobileNetV3*. Konfigurasi ini mampu menghasilkan proses pelatihan yang stabil, akurasi yang tinggi, serta kemampuan *generalization* yang baik terhadap data validasi dan data uji. Dengan demikian, baik *MobileNetV3Small* maupun *MobileNetV3Large* dapat digunakan secara efektif dalam klasifikasi citra daun alpukat, dengan pemilihan model disesuaikan dengan kebutuhan antara efisiensi komputasi dan ketepatan hasil klasifikasi.

Tabel 4. 3 Hasil Perbandingan Pengujian

Model	Optimizer	Learning Rate	Akurasi	Loss	Akurasi Validasi	Loss Validasi	Best Epoch
<i>MobileNetV3Large</i>	<i>Adam</i>	0,0001	96,67%	0,1139	97,78%	0,0841	48
<i>MobileNetV3Small</i>	<i>Adam</i>	0,0001	90%	0,2582	88,89%	0,3007	50

Untuk memperoleh model dengan performa yang optimal, dilakukan serangkaian pengujian terhadap beberapa kombinasi *hyperparameter*. Pada penelitian ini digunakan *optimizer Adam* dengan nilai *learning rate* sebesar 0,0001. Berdasarkan hasil evaluasi yang ditunjukkan pada Tabel 4.3, model *MobileNetV3Large* dengan *optimizer Adam* dan *learning rate* 0,0001 memberikan performa terbaik dibandingkan konfigurasi lainnya. Model ini mencapai akurasi pelatihan sebesar 96,67% dan akurasi validasi sebesar 97,78% pada *epoch* ke-48, dengan nilai *loss* masing-masing sebesar 0,1139 pada data pelatihan dan 0,0841 pada data validasi. Nilai akurasi validasi yang sebanding, bahkan sedikit lebih tinggi dibandingkan akurasi pelatihan, menunjukkan bahwa model memiliki kemampuan *generalization* yang sangat baik. Selain itu, penurunan nilai *loss* yang konsisten pada data pelatihan dan validasi mengindikasikan bahwa proses pembelajaran berlangsung secara stabil dan konvergen. Perbedaan akurasi yang relatif kecil antara data pelatihan dan validasi menunjukkan bahwa model tidak mengalami gejala *overfitting* maupun *underfitting* secara signifikan.

Sementara itu, model *MobileNetV3Small* dengan konfigurasi *optimizer Adam* dan *learning rate* yang sama menghasilkan akurasi pelatihan sebesar 90% dan akurasi validasi sebesar 88,89% pada *epoch* ke-50, dengan nilai *loss* masing-masing sebesar 0,2582 pada data pelatihan dan 0,3007 pada data validasi. Hasil ini menunjukkan bahwa model mampu melakukan klasifikasi dengan cukup baik, namun tingkat akurasi dan kestabilan pembelajarannya masih berada di bawah performa *MobileNetV3Large*. Selisih akurasi antara data pelatihan dan validasi pada *MobileNetV3Small* tetap berada dalam batas wajar, sehingga model ini juga tidak menunjukkan indikasi *overfitting*, meskipun kemampuan representasi fiturnya lebih terbatas. Berdasarkan perbandingan hasil tersebut, dapat disimpulkan bahwa *MobileNetV3Large* mampu mencapai keseimbangan terbaik antara akurasi yang tinggi dan kestabilan proses pembelajaran. Oleh karena itu, model *MobileNetV3Large* dipilih sebagai model terbaik dan digunakan pada tahap implementasi dalam penelitian ini.

4.2.4 Evaluasi Metrix

Pada Gambar 4.5 ditunjukkan hasil evaluasi model *MobileNetV3Large* pada data uji. Berdasarkan *classification report*, model ini mencapai akurasi keseluruhan sebesar 97%. Nilai *precision* berada pada rentang 0,88 hingga 1,00, sedangkan *recall* berkisar antara 0,80 hingga 1,00 untuk setiap kelas. Nilai *f1-score* juga menunjukkan performa yang sangat baik, yaitu antara 0,89 hingga 1,00. Rata-rata *macro* dan *weighted* masing-masing bernilai 0,97, yang menunjukkan bahwa model memiliki kemampuan klasifikasi yang konsisten dan seimbang pada seluruh kelas, termasuk kelas *Bukan_Daun*. Hasil ini mengindikasikan bahwa arsitektur *MobileNetV3Large* mampu mengekstraksi fitur visual daun alpukat secara optimal dan memiliki kemampuan generalisasi yang sangat baik pada data uji.

Sementara itu, pada Gambar 4.6 ditampilkan hasil evaluasi model *MobileNetV3Small* pada dataset uji. Model ini memperoleh akurasi keseluruhan sebesar 90%. Nilai *precision* berada pada rentang 0,79 hingga 1,00, sedangkan *recall* berkisar antara 0,73 hingga 1,00. Nilai *f1-score* yang dihasilkan berada pada kisaran 0,81 hingga 1,00. Rata-rata *macro* dan *weighted* masing-masing mencapai nilai 0,90. Meskipun performanya berada di bawah *MobileNetV3Large*, model *MobileNetV3Small* tetap menunjukkan kemampuan klasifikasi yang cukup baik dan stabil dalam mengenali berbagai jenis daun alpukat serta membedakannya dari kelas *Bukan_Daun*.

Secara keseluruhan, perbandingan hasil evaluasi pada data uji menunjukkan bahwa *MobileNetV3Large* memiliki performa yang lebih unggul dibandingkan *MobileNetV3Small*, terutama dari sisi akurasi dan kestabilan prediksi antar kelas. Namun demikian, *MobileNetV3Small* masih layak digunakan pada perangkat dengan keterbatasan sumber daya komputasi karena ukuran model yang lebih ringan dan performa yang tetap kompetitif.

```

== Classification Report (Test Set) ==

```

	precision	recall	f1-score	support
lpukat_Aligator	0.94	1.00	0.97	15
Alpukat_Kendil	1.00	1.00	1.00	15
Alpukat_Madu	1.00	1.00	1.00	15
Alpukat_SW01	1.00	0.80	0.89	15
Alpukat_Super	0.88	1.00	0.94	15
Bukan_Daun	1.00	1.00	1.00	15
accuracy			0.97	90
macro avg	0.97	0.97	0.97	90
weighted avg	0.97	0.97	0.97	90

Gambar 4. 5 Hasil Metrix *MobileNetV3Large*

```

=== Classification Report (Test Set) ===

```

	precision	recall	f1-score	support
Alpukat_Aligator	0.79	1.00	0.88	15
Alpukat_Kendil	0.79	1.00	0.88	15
Alpukat_Madu	1.00	0.93	0.97	15
Alpukat_SW01	1.00	0.73	0.85	15
Alpukat_Super	0.92	0.73	0.81	15
Bukan_Daun	1.00	1.00	1.00	15
accuracy			0.90	90
macro avg	0.92	0.90	0.90	90
weighted avg	0.92	0.90	0.90	90

Gambar 4. 6 Hasil Metrix *MobileNetV3Small*

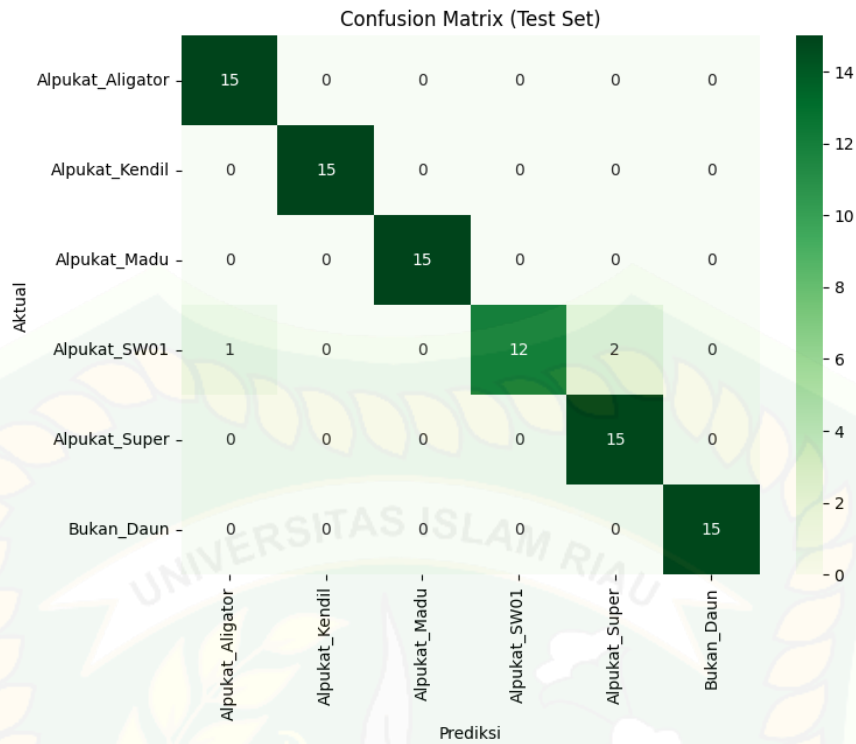
4.2.5 Analisis Confusion Matrix

Pada Gambar 4.7 ditampilkan confusion matrix hasil pengujian model MobileNetV3Large pada data uji. Berdasarkan matriks tersebut, terlihat bahwa sebagian besar data uji berhasil diklasifikasikan dengan sangat baik. Kelas Alpukat_Aligator dan Alpukat_Kendil masing-masing memiliki 15 data yang seluruhnya terklasifikasi dengan benar tanpa kesalahan. Kelas Alpukat_Madu juga menunjukkan performa yang sangat baik dengan 15 data yang seluruhnya diprediksi secara tepat. Pada kelas Alpukat_SW01 terdapat sedikit kesalahan klasifikasi, di mana dari 15 data uji, sebanyak 12 data berhasil diklasifikasikan dengan benar, sementara 1 data salah diklasifikasikan sebagai Alpukat_Aligator dan 2 data salah diklasifikasikan sebagai Alpukat_Super. Kelas Alpukat_Super berhasil diklasifikasikan secara sempurna dengan 15 data yang seluruhnya benar.

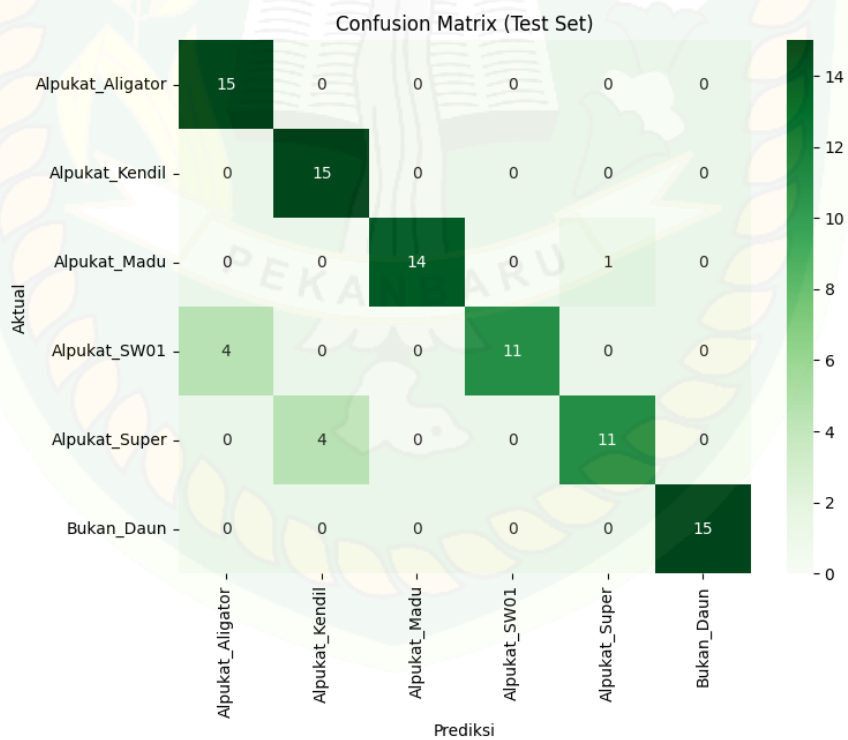
Selain itu, kelas Bukan_Daun juga menunjukkan hasil yang sangat baik dengan seluruh 15 data berhasil dikenali secara tepat tanpa kesalahan. Secara keseluruhan, confusion matrix ini menunjukkan bahwa model MobileNetV3Large memiliki kemampuan klasifikasi yang sangat tinggi dengan tingkat kesalahan yang rendah dan konsentrasi kesalahan hanya pada kelas tertentu yang memiliki kemiripan karakteristik visual.

Berdasarkan confusion matrix pada Gambar 4.8, model MobileNetV3Small menunjukkan performa klasifikasi yang cukup baik meskipun masih terdapat beberapa kesalahan prediksi pada beberapa kelas. Kelas Alpukat_Aligator dan Alpukat_Kendil masing-masing memiliki 15 data uji yang seluruhnya berhasil diklasifikasikan dengan benar tanpa kesalahan. Kelas Alpukat_Madu menunjukkan hasil hampir sempurna dengan 14 data yang terklasifikasi dengan benar dan 1 data yang salah diklasifikasikan sebagai Alpukat_Super. Pada kelas Alpukat_SW01 terlihat adanya kesalahan klasifikasi yang lebih besar dibandingkan kelas lainnya, di mana dari 15 data uji, sebanyak 11 data berhasil diprediksi dengan benar dan 4 data salah diklasifikasikan sebagai Alpukat_Aligator. Kelas Alpukat_Super memiliki 11 data yang terklasifikasi dengan benar, sementara 4 data lainnya salah diklasifikasikan sebagai Alpukat_Kendil. Kelas Bukan_Daun kembali menunjukkan performa yang sangat baik dengan seluruh 15 data berhasil dikenali secara tepat.

Secara keseluruhan, hasil confusion matrix ini menunjukkan bahwa model MobileNetV3Small masih mampu melakukan klasifikasi dengan baik, namun tingkat kesalahan pada beberapa kelas daun yang memiliki kemiripan bentuk relatif lebih tinggi dibandingkan MobileNetV3Large.



Gambar 4. 7 Confusion Matrix *MobileNetV3Large*



Gambar 4. 8 Confusion Matrix *MobileNetV3Small*

4.3 Pengujian Black Box

Blackbox testing merupakan salah satu metode pengujian perangkat lunak yang berfokus pada pemeriksaan fungsi dan keluaran sistem tanpa memperhatikan proses internal atau kode program. Dalam konteks penelitian ini, pengujian *blackbox* diterapkan untuk memastikan bahwa sistem klasifikasi jenis daun alpukat menggunakan model *MobileNetV3* dapat berjalan sesuai dengan fungsinya. Melalui pengujian ini, setiap fitur utama dalam aplikasi—seperti proses unggah citra daun, deteksi jenis alpukat, serta penyimpanan hasil ke riwayat—diperiksa untuk memastikan bahwa sistem memberikan keluaran yang benar dan sesuai dengan kebutuhan pengguna.

4.3.1 Tampilan Halaman Utama Aplikasi

Tampilan halaman utama pada aplikasi Deteksi Daun Alpukat yang ditunjukkan Gambar 4. 5 merupakan antarmuka utama yang dirancang untuk memudahkan pengguna dalam melakukan proses identifikasi jenis daun alpukat. Pada halaman ini, pengguna disajikan dengan dua pilihan utama, yaitu “Ambil Foto” dan “Pilih dari Galeri”. Fitur “Ambil Foto” memungkinkan pengguna untuk mengambil gambar daun alpukat secara langsung melalui kamera perangkat, sedangkan fitur “Pilih dari Galeri” memberikan opsi untuk mengunggah gambar yang telah tersimpan sebelumnya.

Desain antarmuka menggunakan pendekatan sederhana dan intuitif dengan tata letak yang terpusat agar mudah dipahami oleh pengguna dari berbagai kalangan. Komponen utama berupa ikon unggahan dan instruksi “Unggah gambar daun alpukat Anda” ditempatkan di bagian tengah layar untuk menekankan fungsi utama aplikasi, yaitu proses pendeteksian jenis daun alpukat.



Gambar 4. 9 Tampilan Halaman Utama Aplikasi

4.3.2 Tampilan Halaman Riwayat

Tampilan pada Gambar 4.6 tersebut merupakan halaman riwayat prediksi pada aplikasi Deteksi Daun Alpukat. Halaman ini berfungsi untuk menampilkan daftar hasil deteksi daun alpukat yang telah dilakukan oleh pengguna sebelumnya. Setiap hasil prediksi menampilkan nama jenis alpukat, tingkat akurasi dalam bentuk persentase, waktu pelaksanaan deteksi, serta citra daun yang digunakan. Tampilan ini disusun secara kronologis dengan hasil terbaru berada di bagian atas, sehingga memudahkan pengguna dalam meninjau kembali hasil deteksi tanpa perlu melakukan klasifikasi ulang.



Gambar 4. 10 Tampilan Halaman Riwayat

4.3.3 Tampilan Halaman Pengaturan

Tampilan pada Gambar 4.7 tersebut merupakan halaman pengaturan pada aplikasi Deteksi Daun Alpukat. Pada halaman ini terdapat dua fitur utama, yaitu *Mode Gelap* dan *Tentang Aplikasi*. Fitur *Mode Gelap* berfungsi untuk mengubah tampilan aplikasi menjadi tema gelap guna meningkatkan kenyamanan pengguna, terutama saat digunakan dalam kondisi pencahayaan rendah. Sementara itu, bagian *Tentang Aplikasi* menampilkan informasi penting terkait aplikasi, seperti versi aplikasi, nama aplikasi yaitu “Deteksi Daun Alpukat”, serta keterangan bahwa aplikasi ini dikembangkan oleh tim pengembang. Halaman ini dirancang untuk memberikan kemudahan bagi pengguna dalam menyesuaikan tampilan dan mengetahui informasi dasar mengenai aplikasi.



Gambar 4. 11 Tampilan Halaman Pengaturan

4.4 Hasil Prediksi Sistem

4.4.1 Prediksi Alpukat Alligator

Tampilan pada Gambar 4.8 menunjukkan hasil prediksi sistem terhadap citra daun alpukat yang diunggah pengguna. Berdasarkan hasil klasifikasi, model *MobileNetV3Large* mengidentifikasi daun tersebut sebagai jenis *Alpukat Aligator* dengan tingkat akurasi sebesar 41,2%. Informasi yang ditampilkan tidak hanya berupa hasil prediksi dan nilai akurasi, tetapi juga dilengkapi dengan keterangan ilmiah seperti nama ilmiah (*Persea americana 'Alligator'*), asal tanaman, karakteristik buah, serta musim panennya. Fitur ini memudahkan pengguna untuk memahami hasil deteksi secara informatif dan edukatif. Meskipun nilai akurasi pada hasil ini tergolong sedang, sistem tetap memberikan keluaran yang representatif sesuai dengan pola visual daun yang terdeteksi oleh model.



Gambar 4. 12 Hasil Prediksi Alpukat Aligator

4.4.2 Prediksi Alpukat Kendil

Tampilan pada Gambar 4.9 menunjukkan hasil prediksi sistem terhadap citra daun alpukat yang diunggah pengguna. Berdasarkan hasil klasifikasi, model *MobileNetV3Large* mengidentifikasi daun tersebut sebagai jenis *Alpukat Aligator* dengan tingkat akurasi sebesar 41,2%. Informasi yang ditampilkan tidak hanya berupa hasil prediksi dan nilai akurasi, tetapi juga dilengkapi dengan keterangan ilmiah seperti nama ilmiah (*Persea americana 'Alligator'*), asal tanaman, karakteristik buah, serta musim panennya. Fitur ini memudahkan pengguna untuk memahami hasil deteksi secara informatif dan edukatif. Meskipun nilai akurasi pada hasil ini tergolong sedang, sistem tetap memberikan keluaran yang representatif sesuai dengan pola visual daun yang terdeteksi oleh model.



Gambar 4. 13 Hasil Prediksi Alpukat Kendil

4.4.3 Prediksi Alpukat Madu

Gambar 4.10 memperlihatkan tampilan hasil deteksi dari aplikasi klasifikasi daun alpukat. Sistem berhasil mengidentifikasi citra daun sebagai *Alpukat Madu* dengan tingkat akurasi sebesar 85,2%. Selain menampilkan hasil prediksi, aplikasi juga memberikan informasi pendukung seperti nama ilmiah (*Persea americana 'Madu'*), asal varietas, karakteristik morfologi, musim panen, dan keunggulan buah. Desain antarmuka dibuat sederhana dan informatif agar pengguna mudah memahami hasil deteksi serta mengetahui ciri khas dari varietas alpukat yang teridentifikasi.



Gambar 4. 14 Hasil Prediksi Alpukat Madu

4.4.4 Prediksi Alpukat Super

Gambar 4.11 menunjukkan hasil identifikasi aplikasi yang mengenali daun sebagai varietas *Alpukat Super* dengan akurasi 86,1%. Tampilan aplikasi menyajikan informasi secara terstruktur mulai dari nama ilmiah, asal varietas, karakteristik fisik buah, musim panen, hingga keunggulannya. Berdasarkan deskripsi, varietas ini termasuk jenis unggul dengan ukuran buah besar, kulit hijau gelap, serta daging tebal bertekstur lembut.

Deteksi Daun Alpukat



Gambar 4. 15 Hasil Prediksi Alpukat Super

4.4.5 Prediksi Alpukat SW01

Gambar 4.12 menunjukkan hasil prediksi sistem pada aplikasi deteksi daun alpukat, di mana model berhasil mengidentifikasi citra daun sebagai varietas *Alpukat SW01* dengan tingkat akurasi sebesar 93,8%. Nilai akurasi yang tinggi ini menunjukkan bahwa model memiliki tingkat keyakinan yang kuat terhadap hasil klasifikasinya. Hal ini menandakan bahwa ciri visual daun yang dianalisis, seperti bentuk, warna, dan tekstur, sangat sesuai dengan pola yang dimiliki oleh varietas *SW01* dalam data pelatihan. Dengan demikian, prediksi ini dapat dianggap sangat akurat dan menggambarkan kemampuan sistem dalam mengenali jenis daun alpukat secara tepat melalui model MobileNetV3 yang digunakan.



Gambar 4. 16 Hasil Prediksi Alpukat SW01

4.4.6 Prediksi Bukan Daun

Berdasarkan hasil deteksi pada Gambar 4.17 , sistem mengklasifikasikan objek sebagai bukan_daun dengan tingkat kepercayaan sebesar 45,7%. Hal ini menunjukkan bahwa gambar yang diunggah tidak dikenali sebagai daun alpukat maupun daun tanaman lain, melainkan merupakan objek lain yang tidak termasuk dalam kategori daun. Nilai kepercayaan yang relatif rendah menandakan bahwa citra tersebut tidak memiliki ciri visual yang sesuai dengan dataset pelatihan daun. Akibatnya, sistem tidak dapat menampilkan informasi terkait nama ilmiah, asal, karakteristik, musim panen, maupun keunggulan tanaman. Dengan demikian, dapat disimpulkan bahwa kesesuaian objek pada gambar dengan dataset sangat memengaruhi hasil klasifikasi, sehingga diperlukan gambar daun yang jelas dan relevan agar sistem dapat memberikan hasil identifikasi yang lebih akurat.



Gambar 4. 17 Hasil Prediksi Bukan Daun

Hasil proses klasifikasi citra terhadap lima jenis daun alpukat menggunakan model MobileNetV3 disajikan pada Tabel 4.4.

Tabel 4. 4 Kesimpulan Pengujian Form Deteksi Jenis Daun Alpukat

Komponen Yang Diuji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Proses Identifikasi	Proses pengenalan citra daun alpukat berdasarkan jenis (Aligator, Madu, Kendil, Super, SW01, Bukan Daun)	Aplikasi dapat mengenali dan menampilkan nama jenis daun alpukat sesuai hasil klasifikasi model MobileNetV3	[√] Sesuai Harapan [] Tidak Sesuai Harapan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan keseluruhan tahapan penelitian yang meliputi pengambilan data citra daun alpukat, proses pra-pemrosesan dan augmentasi, pelatihan model MobileNetV3, evaluasi performa, serta pengujian pada aplikasi mobile, maka dapat dirumuskan beberapa kesimpulan sebagai berikut:

1. Proses pengambilan citra daun alpukat dilakukan menggunakan kamera smartphone, sehingga diperoleh 900 citra primer dari enam kelas daun alpukat, yaitu Aligator, Madu, Kendil, Super, SW01, dan Bukan Daun. Dataset tersebut kemudian dibagi menjadi 80% data latih, 10% data validasi, dan 10% data uji. Proses augmentasi diterapkan hanya pada data latih, dengan penambahan enam variasi augmentasi untuk setiap citra pada masing-masing kelas. Setelah augmentasi, jumlah citra pada setiap kelas meningkat menjadi 720 data, sehingga total data latih bertambah menjadi 3.600 citra
2. Pengolahan data citra diawali dengan proses akuisisi, kemudian seluruh gambar diubah ukurannya menjadi 224×224 piksel agar sesuai dengan kebutuhan input MobileNetV3. Setelah itu, citra dinormalisasi ke rentang piksel 0–1 dan dilakukan augmentasi berupa rotasi, flipping, zooming, shifting, dan penyesuaian kecerahan untuk memperkaya variasi data. Tahapan ini mendukung proses pelatihan agar model lebih mampu mengenali beragam bentuk dan pola daun alpukat.
3. Penelitian ini menggunakan arsitektur MobileNetV3 dengan pendekatan *transfer learning* sebagai metode klasifikasi. Model dilatih pada dataset daun alpukat yang telah melalui proses pra-pemrosesan dan augmentasi sehingga mampu mempelajari karakteristik visual setiap kelas secara optimal. Pemilihan MobileNetV3 dilakukan karena arsitektur ini efisien, berukuran ringan, dan sangat cocok untuk implementasi pada perangkat mobile.
4. Pada proses pelatihan, model menggunakan *optimizer* Adam, *learning rate* 0.0001, fungsi aktivasi Softmax pada lapisan output, serta *loss function*

Categorical Crossentropy. Hasil pelatihan menunjukkan bahwa MobileNetV3 mampu memberikan performa yang sangat baik dengan akurasi keseluruhan sebesar 97%.

5. Dilakukan perbandingan antara dua varian model, yaitu *MobileNetV3Large* dan *MobileNetV3Small*. *MobileNetV3Large* memperoleh akurasi tertinggi sebesar 97%, sedangkan *MobileNetV3Small* mencapai akurasi sebesar 90%. Perbedaan ini menunjukkan bahwa meskipun kedua model sama-sama efektif, *MobileNetV3Large* memiliki kemampuan klasifikasi yang lebih baik pada dataset ini. Namun, *MobileNetV3Small* tetap menjadi pilihan alternatif untuk implementasi di perangkat yang membutuhkan efisiensi komputasi lebih tinggi.
6. Secara keseluruhan, hasil evaluasi memperlihatkan bahwa model yang dibangun mampu mengklasifikasikan jenis daun alpukat secara efektif, stabil, dan andal pada data pelatihan, validasi, maupun pengujian.

5.2 Saran

Saran dari penulis untuk pengembangan penelitian klasifikasi citra pada tahap selanjutnya adalah sebagai berikut:

1. Pengembangan berikutnya disarankan untuk menambah variasi jenis daun alpukat serta memperbanyak jumlah data. Peningkatan keberagaman dan jumlah citra ini akan memperkaya representasi data, sehingga model mampu melakukan generalisasi dengan lebih baik dan menghasilkan akurasi klasifikasi yang lebih tinggi.
2. Pengembangan lebih lanjut dapat difokuskan pada peningkatan kualitas pemisahan antar kelas melalui penyesuaian threshold kepercayaan (confidence threshold) dan optimalisasi proses prapemrosesan citra, seperti pencahayaan, fokus, dan latar belakang. Dengan demikian, aplikasi dapat memberikan hasil deteksi yang lebih stabil, mengurangi kesalahan prediksi, serta meningkatkan keandalan sistem dalam penggunaan nyata.

DAFTAR PUSTAKA

- Ashraf, M. I. (2023). Advanced Techniques for Artificial Intelligence with Python. *Software Engineering*, 2023(1), 7–10. <https://doi.org/10.5923/j.se.20231001.02>
- Azmi, K., Defit, S., & Sumijan, S. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 28–40. <https://doi.org/10.52072/unitek.v16i1.504>
- Chu, X., Zhang, B., & Xu, R. (2020). MoGA: Searching beyond Mobilenetv3. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May, 4042–4046. <https://doi.org/10.1109/ICASSP40776.2020.9054428>
- Firdaus, D., Sumardi, I., & Aziz, R. R. (2024). Sistem Deteksi Penyakit Daun Singkong Menggunakan Deep Learning dengan Arsitektur MobilenetV3 berbasis Android. 3(2), 71–80.
- Gunawan, A. (n.d.). *Mobile Programming Menggunakan Flutter dan Visual Studio Code Untuk Pemula*.
- Hakiki, N. T., Sthevanie, F., & Ramadhani, K. N. (2023). Analisa Metode Augmentasi Data Untuk Klasifikasi Objek Pada Dataset CIFAR-10 (pp. 1–8). <https://repositori.telkomuniversity.ac.id/pustaka/199712/analisis-metode-augmentasi-data-untuk-klasifikasi-objek-pada-dataset-cifar-10-dalam-bentuk-pengganti-sidang-artikel-jurnal.html>
- Hapuarachchi, N. S., Kämper, W., Hosseini Bai, S., Ogbourne, S. M., Nichols, J., Wallace, H. M., & Trueman, S. J. (2024). Selective Retention of Cross-Fertilised Fruitlets during Premature Fruit Drop of Hass Avocado. *Horticulturae*, 10(6). <https://doi.org/10.3390/horticulturae10060591>
- Kirom, M., Rojul, M., & Witanti, A. (2025). Klasifikasi Ras Kelinci Menggunakan Convolutional Neural Network (CNN) untuk Optimasi Sistem Identifikasi Visual. 6(2). <https://doi.org/10.47065/josh.v6i2.6627>
- Komputer, J. S., Hidayat, A. R., Lusiana, V., Informatika, S. T., & Stikubank, U. (2022). Deteksi Jenis Sayuran dengan Tensorflow Dengan Metode Convolutional Neural Network. 6(September), 1032–1040.
- Kusuma, B. M., Hermanto, T. I., & Lestari, D. (2025). Klasifikasi Jenis Penyakit Pada Tanaman Padi Menggunakan Algoritma Convolutional Neural Network Classification of Paddy Rice Plant Diseases Using Convolutional Neural Network Algorithms. 1, 40–52. <https://doi.org/10.26798/jiko.v9i1.1395>
- Kusuma, B., Teguh Iman Hermanto, & Candra Dewi Lestari. (2024). Optimasi Dan Integrasi Convolutional Neural Network Pada Aplikasi Android Untuk

- Deteksi Penyakit Daun Padi. *Jurnal INSTEK (Informatika Sains Dan Teknologi)*, 9(2), 251–261. <https://doi.org/10.24252/instek.v9i2.49657>
- Lahir, B., Yuliati, I. F., & Sihombing, P. R. (2021). *Penerapan Metode Machine Learning dalam Klasifikasi Risiko Kejadian Berat Penerapan Metode Machine Learning dalam Klasifikasi Risiko Kejadian Berat Badan Lahir Rendah di Indonesia Implementation of Machine Learning Method in Risk Classification on Low Bir. May*. <https://doi.org/10.30812/matrik.v20i2.1174>
- Liang, X., Liang, J., Yin, T., & Tang, X. (2023). A lightweight method for face expression recognition based on improved MobileNetV3. *IET Image Processing*, 17(8), 2375–2384. <https://doi.org/10.1049/ipr2.12798>
- Mamuriyah, N., Haeruddin, H., & Jackson, J. (2025). Developing a Chatbot System for PT. NG Tech Supplies based on the Python Flask Framework. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 7(1), 143–149. <https://doi.org/10.47233/jteksis.v7i1.1821>
- Marques, B. (n.d.). IoT Smart Door Lock with Wireless Key Sharing for Tourism Applications. In *Proceedings of ACM Conference (Conference'17)* (Vol. 1, Issue 1). Association for Computing Machinery.
- Mohammad Haydir Awaludin Waskito, Andreas Nugroho Sihananto, & Achmad Junaidi. (2024). Klasifikasi Penyakit Kronis Melalui Mata Menggunakan Algoritma Convolutional Neural Network Dengan Model MobileNet-V3. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains Dan Informatika*, 2(2), 48–60. <https://doi.org/10.61132/uranus.v2i2.120>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2), 697–711.
- Putriani, K., Permata Dewi, A., Lestari, R., & Ade Syamsuri, N. (2022). Uji Aktivitas Antibakteri Daging Buah Alpukat Dan Ekstrak Etanol Daging Buah Alpukat (Persea Americana Mill) Terhadap Escherichia coli. *Jurnal Ilmiah Farmasi Simplisia*, 2(1), 23–29. <https://doi.org/10.30867/jifs.v2i1.13>
- Qian, S. (2021). *MobileNetV3 for Image Classification*. *Icbaie*, 490–497.
- Ratna, S. (2020). Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm. *Technologia: Jurnal Ilmiah*, 11(3), 181. <https://doi.org/10.31602/tji.v11i3.3294>
- Rosyani, P., Saprudin, S., & Amalia, R. (2021). Klasifikasi Citra Menggunakan Metode Random Forest dan Sequential Minimal Optimization (SMO). *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 9(2), 132. <https://doi.org/10.26418/justin.v9i2.44120>
- Saladdin, S., Arwani, I., & Rahayudi, B. (2020). Pengembangan Chatbot Yanies Cookies Untuk Pemesanan Kue Kering Berbasis Dialogflow. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(7), 2121–2129.

<http://j-ptiik.ub.ac.id>

- Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(2), 311–323. <https://doi.org/10.28932/jutisi.v6i2.2688>
- Santoso, A., Ariyanto, G., Learning, F., Network, C. N., & Pendahuluan, I. (n.d.). *IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK*. 18(01), 15–21.
- Sari, F. A. F., Wulanningrum, R., & Wahyuniar, L. S. (2023). Penggunaan Metode CNN (Convolutional Neural Network) untuk Klasifikasi Jenis Tanaman Alpukat Berdasarkan Pola Daun. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 7, 1275–1284. <https://proceeding.unpkediri.ac.id/index.php/inotek/>
- Satria Maheswara, E., Zuhri, A. B., & Iskandar Maulana, D. (2022). Optimation Image Classification Pada Ikan Hiu Dengan Metode Convolutional Neural Network Dan Data Augmentasi. *Jurnal Tika*, 7(1), 1–11. <https://doi.org/10.51179/tika.v7i1.993>
- Suyuti, M. (2023). Pengembangan Model Klasifikasi Mata Tertutup dan Terbuka Dalam Identifikasi Kelelahan Menggunakan Arsitektur Mobile CNN. *Universitas Islam Indonesia*, 1–61. <https://dspace.uui.ac.id/handle/123456789/42508%0Ahttps://dspace.uui.ac.id/bitstream/handle/123456789/42508/18917213.pdf?sequence=1&isAllowed=y>
- Yanuar, R., & Andarsyah, A. (2024). Sentimen Analisis Aplikasi Posaja Pada Google Playstore untuk Peningkatan Pospay Superapp menggunakan Support Vector Machine. *Jurnal Teknik Informatika*, 16(2), 1–7. <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/3533>
- Zahro, L., Syaputra, H., Oktarina, S., Roseno, M. T., & Supratman, E. (2024). Optimisasi Hyperparameter pada Model Klasifikasi Jenis Bunga dengan arsitektur MobileNetV3-Small. *Jurnal Nasional Ilmu Komputer*, 5(3), 2746–1343.
- Zuhdi, M., & Busyairi, A. (2023). Program Dekonvolusi Blind Berbasis Matlab untuk Mempertajam Citra Akibat Simulasi Efek Buram Lensa. *Kappa Journal*, 7(1), 170–183. <https://doi.org/10.29408/kpj.v7i1.20281>