

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal sebagai salah satu negara dengan tingkat keanekaragaman hayati yang sangat tinggi, termasuk beragam jenis tanaman buah. Salah satu buah khas yang berasal dari kekayaan flora tersebut adalah matoa (*Pometia pinnata*). Tanaman ini termasuk ke dalam famili *Sapindaceae* dan tumbuh luas di daerah beriklim tropis. Selain dimanfaatkan sebagai buah konsumsi, matoa juga sering digunakan dalam pengobatan tradisional karena mengandung berbagai senyawa *bioaktif*, seperti *flavonoid*, *tanin*, dan *saponin* [1].

Matoa (*Pometia pinnata*) merupakan buah tropis khas Papua yang tumbuh pada daerah bercurah hujan tinggi hingga ketinggian sekitar 1.400 meter di atas permukaan laut. Buah ini memiliki rasa manis khas perpaduan kelengkeng dan rambutan, dengan daging lembut dan berair, serta bentuk bulat hingga lonjong berukuran 5–6 cm. Matoa terbagi menjadi matoa kelapa (kenyal) dan matoa papeda (lembek), serta memiliki variasi warna kulit seperti hijau (*Emme Anokhong*), kuning (*Emme Khabhelaw*), dan merah (*Emme Bhanggahe*) [2].

Perkembangan teknologi komputer saat ini mendorong pemanfaatan kecerdasan buatan atau *Artificial Intelligence* (AI) dalam berbagai bidang, termasuk pengolahan citra digital. Menurut Andreas Kaplan dan Michael Haenlein, kecerdasan buatan merupakan kemampuan suatu sistem untuk menginterpretasikan data dari lingkungan sekitar, mempelajari data tersebut,

kemudian menggunakan hasil pembelajaran untuk menyelesaikan berbagai tugas secara adaptif dan fleksibel [3]. Salah satu cabang dari AI adalah *machine learning*, yaitu metode yang memungkinkan sistem atau mesin mempelajari data secara otomatis sehingga mampu menjalankan tugas tertentu berdasarkan pola yang telah dipelajari [4].

Seiring perkembangannya, *machine learning* melahirkan metode *deep learning* yang memanfaatkan jaringan saraf tiruan berlapis untuk mengotomatisasi proses prediksi dan klasifikasi data yang kompleks [5]. Dalam bidang pengolahan citra, salah satu metode *deep learning* yang banyak digunakan adalah *Convolutional Neural Network* (CNN), yaitu arsitektur jaringan saraf tiruan yang dirancang khusus untuk mengolah citra dan data visual dengan meniru cara kerja visual cortex manusia dalam mengenali pola [6]. Salah satu pengembangan arsitektur CNN yang cukup populer adalah *MobileNetV2*, yaitu model yang dikembangkan oleh *Google* dengan struktur ringan sehingga dapat dijalankan secara optimal pada perangkat dengan kemampuan komputasi terbatas [7].

Dalam bidang pengolahan citra digital, teknologi komputer dapat digunakan untuk mengenali objek berdasarkan karakteristik visual seperti warna, bentuk, dan tekstur. Identifikasi citra secara otomatis penting dilakukan karena mampu meningkatkan kecepatan, konsistensi, dan efisiensi pengamatan. Pada buah matoa, perbedaan warna kulit matang merupakan ciri visual yang dapat dikenali langsung oleh manusia dengan mata telanjang untuk membedakan jenis buah. Meskipun demikian, proses identifikasi yang dilakukan secara langsung oleh manusia masih memiliki berbagai keterbatasan, terutama karena hasil

pengamatan sangat dipengaruhi oleh tingkat ketelitian, pengalaman, serta persepsi masing-masing pengamat.

Perbedaan persepsi tersebut dapat menyebabkan ketidakkonsistenan dalam menentukan jenis buah matoa, terutama ketika jumlah buah yang diamati cukup banyak dan dilakukan dalam waktu yang lama sehingga pengamat dapat mengalami kelelahan. Selain itu, kesalahan identifikasi juga berpotensi memengaruhi proses sortasi dan pengelompokan hasil panen yang berdampak pada menurunnya kualitas distribusi maupun pemasaran buah. Permasalahan tersebut menunjukkan bahwa proses identifikasi secara manual masih kurang efektif untuk diterapkan secara berkelanjutan, khususnya dalam sistem berbasis digital yang membutuhkan hasil cepat dan akurat. Oleh karena itu, diperlukan sistem berbasis pengolahan citra yang mampu mengenali karakteristik buah matoa secara otomatis agar proses identifikasi dan klasifikasi dapat dilakukan dengan lebih akurat.

Dengan adanya sistem klasifikasi otomatis tersebut diharapkan dapat membantu petani dan pelaku usaha dalam proses identifikasi dan sortasi buah secara lebih cepat, akurat, dan konsisten sehingga meningkatkan efisiensi kerja serta kualitas distribusi hasil panen. Selain mudah diakses melalui sistem berbasis *web*, penelitian ini juga dapat menjadi referensi untuk pengembangan klasifikasi buah tropis lainnya. Tanpa sistem klasifikasi yang baik, identifikasi manual berpotensi menyebabkan kesalahan pengamatan dan sortasi, sehingga teknologi *deep learning* diperlukan untuk meningkatkan akurasi dan standarisasi klasifikasi jenis buah matoa berdasarkan warna kulit matang.

Beberapa penelitian sebelumnya telah dilakukan untuk mengidentifikasi dan mengklasifikasikan tingkat kematangan buah menggunakan pendekatan berbasis pengolahan citra dan deep learning. Penelitian oleh Liandaputra dan Zahra (2024) yang berjudul *“Klasifikasi Tingkat Kematangan Buah Nanas Menggunakan Metode Deep Learning”* menggunakan arsitektur *Convolutional Neural Network* (CNN), khususnya *MobileNetV2*, dengan teknik augmentasi data geometrik dan fotometrik. Penelitian ini mengklasifikasikan tingkat kematangan buah nanas ke dalam empat kelas, yaitu mentah, kurang matang, matang, dan terlalu matang. Hasil penelitian menunjukkan bahwa model *MobileNetV2* mampu mencapai akurasi hingga 93%, serta menunjukkan performa yang lebih baik dibandingkan arsitektur lain seperti *AlexNet* dan *VGG16* [8].

Selanjutnya, penelitian oleh Nugroho dan Suria (2025) dengan judul *“Klasifikasi Antraknosa dan Buah Busuk pada Citra Cabai Rawit Menggunakan Metode MobileNetV2 dengan SVM”* mengkaji pemanfaatan metode *deep learning* yang dikombinasikan dengan *machine learning* dalam klasifikasi kondisi buah cabai. Penelitian ini menggunakan arsitektur *MobileNetV2* sebagai ekstraktor fitur dan *Support Vector Machine* (SVM) sebagai algoritma klasifikasi untuk membedakan tiga kondisi cabai, yaitu sehat, terinfeksi antraknosa, dan busuk. Hasil penelitian menunjukkan bahwa model yang diusulkan mampu mencapai akurasi sangat tinggi hingga 100% pada data uji, dengan nilai *precision*, *recall*, dan *F1-score* yang juga mencapai 100% pada seluruh kelas [9].

Selain itu, penelitian oleh Isnaini dan Pramudwiatmoko (2025) berjudul *“Implementasi MobileNetV2 untuk Klasifikasi Kematangan Buah Kelapa Sawit*

pada Aplikasi Mobile” mengkaji penerapan *MobileNetV2* berbasis *transfer learning* yang dioptimasi dengan *post-training quantization* ke format *TensorFlow Lite*. Model diintegrasikan dalam aplikasi *Flutter* dan mampu berjalan secara *offline* pada perangkat dengan sumber daya terbatas. Hasil penelitian menunjukkan akurasi sebesar 89%, dengan ukuran model berkurang hingga 84,6% serta waktu inferensi rata-rata 120 ms pada perangkat Android kelas menengah [10].

Berdasarkan uraian tersebut, diperlukan suatu penelitian yang mengimplementasikan metode *deep learning* menggunakan arsitektur *MobileNetV2* untuk mengklasifikasikan jenis buah matoa berdasarkan warna kulit matang. Oleh karena itu, judul yang diangkat dalam penelitian ini adalah “Penerapan Arsitektur *MobileNetV2* untuk Klasifikasi Jenis Buah Matoa Berdasarkan Citra Warna Kulit Matang.”

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka dapat dirumuskan suatu masalah pada penelitian ini, yaitu “Bagaimana merancang dan membangun sistem berbasis *website* untuk mengklasifikasikan buah matoa secara otomatis menggunakan arsitektur *MobileNetV2* berdasarkan warna kulit matang?”

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah merancang dan membangun sistem berbasis *website* yang mampu mengklasifikasikan buah matoa secara otomatis menggunakan arsitektur *MobileNetV2* berdasarkan warna kulit matang.

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini, yaitu :

1. Menggunakan metode CNN dengan arsitektur *MobileNetV2* berbasis *transfer learning*.
2. Dataset yang digunakan terdiri dari 3600 citra buah matoa format .jpg hasil pengambilan kamera *smartphone*.
3. Klasifikasi dilakukan pada tiga jenis matoa : Hijau, Kuning, dan Merah.
4. Penelitian ini berfokus pada deteksi dan klasifikasi pada citra warna kulit buah matoa.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Memberikan kontribusi dalam pengembangan ilmu *Artificial Intelligence* dan *deep learning*, khususnya penerapan CNN *MobileNetV2* untuk klasifikasi citra berbasis warna, serta menjadi referensi bagi penelitian selanjutnya.
2. Membantu petani, pelaku usaha, pengembang sistem, dan masyarakat dalam mengidentifikasi jenis buah matoa secara cepat dan akurat, melalui sistem klasifikasi berbasis website.
3. Menjadi bahan pembelajaran dan rujukan bagi mahasiswa dalam memahami implementasi *deep learning*, serta sebagai dasar untuk pengembangan penelitian lanjutan, termasuk peningkatan model dan integrasi dengan teknologi seperti *IoT*.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri dari pokok-pokok permasalahan yang dibahas dan diuraikan menjadi beberapa bagian :

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penelitian.

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan pada penelitian ini. Teori-teori yang berhubungan dengan pengembangan sistem yang akan dibangun.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tahapan-tahapan dalam pengumpulan data, perancangan sistem, perumusan masalah dan analisa.

BAB 4 ANALISA DAN PERANCANGAN

Bab ini berisi analisa dan pengembangan sistem yang akan dibangun atau dikembangkan.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari analisa dan perancangan dan pengujian pada sistem yang berhasil dibangun.

BAB 6 PENUTUP

Bab ini berisi rangkuman dari hasil penelitian yang telah dilakukan dan saran-saran untuk pengembangan sistem atau penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan Buatan (AI) adalah kemampuan mesin atau sistem komputer untuk meniru atau melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia, seperti penalaran logis, pembelajaran, dan pemecahan masalah [11]. Kecerdasan buatan mengacu pada pengembangan dan implementasi sistem komputer yang memiliki kemampuan untuk melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Tugas-tugas tersebut termasuk belajar dari pengalaman, pemikiran dari permasalahan yang kompleks, memahami bahasa alami, dan beradaptasi dengan situasi baru [12].

Salah satu arah perkembangan AI di masa kini adalah kemampuan untuk menjadi lebih pintar dan memahami konteks. AI saat ini sudah mampu melakukan tugas-tugas seperti pengenalan suara dan gambar dimasa kini AI diharapkan mampu meningkatkan kemampuan untuk memahami dan membuat keputusan berdasarkan situasi dan konteks [13] .

2.2 *Machine Learning*

Machine learning sendiri merupakan bagian dari AI yang berisi kumpulan algoritma yang membuat mesin (sistem) memiliki pengetahuan dari data train yang diberikan. Pengetahuan terhadap permasalahan yang ada tidak serta merta didapat tanpa adanya proses belajar. Mesin terlebih dahulu di latih (*train*) menggunakan data yang sebelumnya telah melalui serangkaian proses hingga layak menjadi data *train* [14].

Machine learning digunakan dalam berbagai aplikasi seperti deteksi spam, pengenalan suara, sistem rekomendasi, dan pengolahan citra digital, khususnya untuk mengenali objek dalam gambar. Salah satu perkembangannya adalah *deep learning*, yaitu metode berbasis jaringan saraf tiruan yang mampu mempelajari representasi data secara bertingkat untuk memahami informasi kompleks seperti gambar, suara, dan teks [15].

Penelitian oleh Ahmad menjelaskan *Machine learning* atau pembelajaran mesin merupakan cabang dari kecerdasan buatan (AI) yang berfokus pada perancangan dan pengembangan algoritma agar komputer dapat belajar dan membentuk pola perilaku berdasarkan data empiris, seperti data dari sensor maupun basis data [16].

2.3 *Deep Learning*

Deep learning adalah area pembelajaran mesin yang menggunakan jaringan saraf tiruan untuk menyelesaikan masalah dengan kumpulan data besar. Metode *deep learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*. Dengan menambahkan lapisan, memungkinkan model pembelajaran untuk merepresentasikan gambar berlabel dengan lebih baik [17].

Deep learning digunakan untuk mempelajari pola-pola kompleks dalam data seperti gambar, suara, teks dan lain-lain dengan memanfaatkan komputasi yang sangat paralel dan berdaya tinggi [18]. *Deep learning* adalah metode untuk mengotomatisasi proses prediksi dan klasifikasi data menggunakan jaringan saraf tiruan berlapis (*deep neural networks*) guna memodelkan pola dan abstraksi data yang kompleks [5].

2.4 *Transfer Learning*

Transfer learning adalah sebuah teknik machine learning yang menggunakan model yang telah dilatih sebelumnya untuk menyelesaikan sebuah tugas dan kemudian digunakan untuk menyelesaikan tugas lain yang berkaitan dengan tugas awal. Teknik ini biasanya digunakan jika memiliki dataset yang sedikit. Proses *transfer learning* adalah dengan membekukan beberapa convolutional layers di awal dan hanya melatih beberapa lapisan terakhir yang digunakan untuk melakukan klasifikasi [19].

Transfer learning adalah metode yang memanfaatkan model yang telah dilatih sebelumnya untuk mengklasifikasi data baru tanpa perlu melatih dari awal, sehingga lebih efisien waktu dan proses. Teknik ini bekerja dengan mempertahankan lapisan konvolusi dan *pooling*, serta menyesuaikan layer akhir agar sesuai dengan jumlah kelas pada dataset yang digunakan [20]. Metode *transfer learning* memanfaatkan model yang telah dilatih sebelumnya atau pretrained model sebagai dasar untuk mempelajari dan menyelesaikan tugas baru [7].

2.5 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan salah satu bagian dari *Deep Neural Network* yang diimplementasikan untuk *image recognition*. Metode ini merupakan pengembangan dari *Multilayer Perception (MLP)*. Namun MLP menerima input data satu dimensi, sementara CNN data dua dimensi dan salah satunya adalah citra [21].

Penggunaan *Convolutional Neural Network* (CNN) dalam visi komputer terus berkembang, terlihat dari berbagai arsitektur yang dirancang untuk menyelesaikan beragam permasalahan, seperti *AlexNet*, *VGG*, dan *Google Inception*. Secara umum, CNN terdiri dari lapisan tersembunyi yang meliputi *convolutional layer*, *pooling layer*, *normalization layer*, *ReLU*, *fully connected layer*, serta *loss layer* [22].

Convolutional Neural Network (CNN) merupakan jaringan saraf tiruan yang memanfaatkan operasi konvolusi dalam arsitekturnya untuk mengekstraksi fitur dari data. Hasil ekstraksi tersebut kemudian diproses oleh *multilayer perceptron* untuk menghasilkan *output* [6].

2.6 *MobileNetV2*

MobileNetV2 merupakan sebuah model hasil perkembangan dari model sebelumnya yaitu *MobileNetV1*. Struktur *MobileNetV2* secara umum memiliki struktur yang sama seperti pada versi sebelumnya yaitu dengan menerapkan *Depthwise Separable Convolution* sebagai lapisan utamanya. *Depthwise Separable Convolution* digunakan untuk membangun sebuah neural network yang memiliki bobot lebih ringan. *Depthwise Separable Convolution* akan membagi konvolusi menjadi dua yaitu *Depthwise Convolution* dan *Pointwise Convolution*. *Depthwise Convolution* mengaplikasikan satu filter untuk setiap input channel pada model yang kemudian hasilnya akan dilakukan konvolusi 1x1 pada *Pointwise Convolution* [23].

MobileNetV2 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) berbasis perangkat bergerak yang dirancang untuk mengatasi

kebutuhan akan sumber daya komputasi yang terbatas. Arsitektur ini merupakan perbaikan dari arsitektur *MobileNet* sebelumnya. Salah satu perbedaan utama antara *MobileNet* dan arsitektur CNN lainnya terletak pada penggunaan lapisan konvolusi atau *convolution layer* [24].

MobileNetV3 dikembangkan dengan memanfaatkan *Neural Architecture Search* (NAS) dan mekanisme *Squeeze-and-Excitation* (SE) untuk meningkatkan efisiensi serta akurasi model [25]. Adapun *MobileNetV4* (MNv4) merupakan generasi terbaru dari arsitektur *Convolutional Neural Network* (CNN) yang diperkenalkan oleh *Google* pada tahun 2024. Arsitektur ini dikembangkan untuk memberikan efisiensi yang bersifat universal serta performa yang optimal pada berbagai platform perangkat bergerak. *MobileNetV4* menghadirkan dua komponen utama, yaitu *Universal Inverted Bottleneck* (UIB) dan *Mobile Multi-Query Attention* (MMQA), di mana mekanisme MMQA hanya diterapkan pada varian *Hybrid* [26].

Meskipun *MobileNetV3* dan *MobileNetV4* merupakan pengembangan yang lebih baru, penelitian ini menggunakan *MobileNetV2* sebagai arsitektur utama karena memiliki struktur yang lebih sederhana, mudah diimplementasikan, dan efisien dalam penggunaan sumber daya komputasi. Selain itu, *MobileNetV2* menawarkan keseimbangan yang baik antara akurasi, jumlah parameter, dan kompleksitas model sehingga sesuai untuk penelitian dengan dataset yang terbatas. Arsitektur ini juga telah banyak digunakan dan divalidasi pada berbagai penelitian klasifikasi citra, sehingga keandalan dan stabilitas performanya telah

terbukti. Oleh karena itu, Oleh karena itu, *MobileNetV2* dipilih sebagai arsitektur klasifikasi pada penelitian ini.

2.7 *Fine Tuning*

Pada tahap awal, pelatihan hanya dilakukan pada layer *classifier*, sedangkan layer pada model pra-latih tetap di-*freeze* sehingga bobotnya tidak mengalami perubahan. Untuk meningkatkan kinerja model, dilakukan *fine-tuning* dengan membuka (*unfreeze*) beberapa layer teratas pada pre-trained model. Dengan cara ini, proses training tidak hanya memperbarui bobot *classifier*, tetapi juga menyesuaikan bobot pada layer tertentu dari model dasar. *Fine-tuning* dilakukan dengan mengatur atribut *trainable* pada layer model pra-latih menjadi *True* pada layer yang dipilih, terutama layer bagian atas karena lebih spesifik terhadap karakteristik dataset baru [27].

Fine-tuning dilakukan dengan melatih model menggunakan data *training* yang telah tersedia, sementara beberapa layer awal tetap dipertahankan dalam kondisi *frozen* agar bobotnya tidak berubah. Selama proses pelatihan, kinerja model perlu dipantau dan dievaluasi secara berkala melalui metrik seperti akurasi, *loss*, maupun metrik relevan lainnya [28]. Hasil dari proses *fine-tuning* berupa model yang telah mengalami penyesuaian bobot sesuai dengan kebutuhan tugas yang dikerjakan. Keberhasilan proses ini dipengaruhi oleh pengaturan hiperparameter dan keragaman data yang digunakan, sehingga berbagai aspek perlu diperhatikan agar model dapat bekerja secara optimal [29].

2.8 Klasifikasi

Klasifikasi adalah proses penggunaan model untuk menentukan label kelas suatu objek dengan tujuan memprediksi kategori yang tepat. Metode ini menggunakan pendekatan *supervised learning*, yaitu data yang telah memiliki label digunakan untuk melatih model dalam menghasilkan prediksi terhadap variabel target [30].

Klasifikasi merupakan tugas utama dalam *machine learning* yang bertujuan mengelompokkan data ke dalam kategori berdasarkan pola tertentu. Dalam pengolahan citra digital, teknik ini digunakan untuk mengenali objek berdasarkan fitur visual seperti warna, bentuk, dan tekstur, serta diterapkan pada berbagai bidang seperti deteksi objek, pengenalan wajah, dan identifikasi produk. Menurut Nugroho dkk, performa klasifikasi citra dapat ditingkatkan melalui kombinasi ekstraksi fitur dengan algoritma seperti SVM, KNN, dan CNN [31].

Penelitian terbaru oleh Abdul Syawal dkk, menunjukkan bahwa penggunaan teknologi *Teachable Machine* berbasis pembelajaran mesin mampu menghasilkan sistem klasifikasi citra dengan tingkat akurasi yang cukup baik, meskipun tanpa memerlukan pemrograman yang kompleks. Penerapannya mencakup berbagai bidang, seperti deteksi jenis tanah maupun identifikasi produk industri [32].

2.9 Dataset

Dataset adalah kumpulan data yang terorganisir dalam format yang terstruktur. Dataset merupakan sekumpulan data yang dikumpulkan dan disusun secara sistematis untuk digunakan dalam proses analisis maupun pelatihan model

dalam bidang *Machine Learning* dan *Deep Learning*. Data yang terdapat dalam sebuah dataset dapat berupa teks, angka, audio, maupun citra yang digunakan sebagai sumber pembelajaran bagi sistem komputer [33].

Berdasarkan berbagai penelitian terdahulu, jumlah dataset merupakan salah satu faktor krusial yang memengaruhi performa model *deep learning* dalam klasifikasi citra. Penelitian “*Klasifikasi Buah Manggis Lokal dan Ekspor Berdasarkan Warna dengan Arsitektur MobileNetV2*” menggunakan 384 citra dan memperoleh akurasi 95% [34]. Selanjutnya, penelitian “*Klasifikasi Tingkat Kematangan Buah Nanas Menggunakan Metode Deep Learning*” menggunakan 400 citra dengan akurasi validasi 93%, namun performa model masih dapat ditingkatkan melalui penambahan dataset [8]. Penelitian “*Deep Learning untuk Identifikasi Daun Tanaman Obat Menggunakan Transfer Learning MobileNetV2*” menggunakan 600 citra dan menghasilkan akurasi 94,16% [7].

Jumlah dataset yang lebih besar juga menunjukkan hasil yang lebih optimal. Penelitian “*Klasifikasi Tingkat Kematangan Buah Jengkol Menggunakan CNN dengan Arsitektur MobileNetV3*” menggunakan 1.356 citra dan memperoleh akurasi terbaik sebesar 98,53% [35]. Selain itu, penelitian “*Breaking the Data Barrier*” mengelompokkan dataset berdasarkan skala, di mana dataset dengan jumlah lebih dari 1000 citra sudah termasuk kategori menengah hingga besar, sehingga lebih mendukung proses pembelajaran model secara optimal [36].

Mengacu pada temuan tersebut, penggunaan dataset sebanyak 3600 citra yang terbagi ke dalam 3 kelas dengan masing-masing 1200 citra per kelas dalam

penelitian ini telah memenuhi kriteria dataset menengah hingga besar, sehingga dinilai cukup representatif dan layak untuk mendukung proses pelatihan model *MobileNetV2* dalam klasifikasi citra buah matoa.

2.10 Citra Digital (*Digital Image*)

Citra digital adalah representasi visual dari objek dalam bentuk diskret yang tersimpan dalam format digital dan diproses oleh komputer. Setiap citra terdiri atas piksel-piksel yang merepresentasikan informasi visual, seperti warna atau intensitas cahaya, dalam koordinat dua dimensi. Dalam dunia teknologi informasi, citra digital menjadi dasar dalam banyak aplikasi seperti pengenalan wajah, klasifikasi objek, sistem pengawasan, hingga analisis medis. Menurut Devi dan Rosyid (2022), citra digital mampu menyediakan data visual yang presisi dan dapat dianalisis secara matematis melalui berbagai teknik pengolahan untuk mengekstraksi informasi yang tersembunyi di dalamnya [37].

Citra digital merujuk pada proses manipulasi gambar dua dimensi dengan menggunakan komputer, Citra digital harus diwakili dalam bentuk angka numerik dengan nilai intensitas cahaya yang diskrit, sehingga komputer dapat memprosesnya dengan baik. Citra Digital bertujuan memberikan pemahaman dan penguasaan pengetahuan serta keterampilan tentang konsep dasar tipografi, citra *bitmap*, dan citra *vector* [38].

Dalam penelitian ini, akuisisi citra dilakukan menggunakan kamera *smartphone* beresolusi 13 *megapiksel* (MP). Resolusi tersebut dipilih karena mampu menghasilkan citra yang jelas sehingga karakteristik visual kulit buah matoa, seperti warna, tekstur, dan detail permukaan, dapat terekam dengan baik.

Pada penelitian klasifikasi citra, tidak terdapat standar baku mengenai jumlah megapiksel yang harus digunakan. Pencahayaan yang tinggi dapat memberikan kejelasan gambar dan perbedaan warna antar *pixel* yang cukup kontras, sehingga semakin memperlancar proses pemisahan antara objek dan latar. Selain itu, jarak objek terhadap kamera juga mempengaruhi keakuratan dalam proses deteksi [39].

Penggunaan kamera dengan resolusi lebih rendah tidak selalu menurunkan akurasi selama fitur-fitur penting objek masih dapat direkam dengan jelas. Namun, resolusi yang terlalu rendah berpotensi mengurangi detail citra sehingga dapat memengaruhi kemampuan model dalam mengekstraksi fitur. Oleh karena itu, penggunaan kamera 13 MP pada penelitian ini dinilai telah memadai untuk menghasilkan dataset yang berkualitas dan mendukung proses pelatihan model *MobileNetV2* secara optimal..

2.11 Evaluasi Model

Evaluasi performa model dilakukan untuk mengukur tingkat akurasi serta menilai sejauh mana kemampuan model dalam bekerja pada penelitian ini melalui penggunaan metrik evaluasi. Pada *Confusion Matrix* terdapat empat komponen utama, yaitu TP (*True Positive*), TN (*True Negative*), FN (*False Negative*), dan FP (*False Positive*). Masing-masing komponen tersebut menunjukkan jumlah data dengan kondisi prediksi dan nilai aktual sama-sama positif, prediksi dan nilai aktual sama-sama negatif, prediksi bernilai negatif tetapi nilai aktual positif, serta prediksi bernilai positif namun nilai aktual negatif [40].

Nilai pada metrik evaluasi diperoleh melalui perhitungan berdasarkan nilai-nilai yang terdapat pada *Confusion Matrix*. Metrik evaluasi yang digunakan

meliputi *accuracy*, *precision*, *recall*, dan *F1-score*, yang dijelaskan sebagai berikut [41]:

1. *Accuracy* (Akurasi)

Accuracy menunjukkan tingkat kemampuan model dalam menghasilkan prediksi yang benar dibandingkan dengan seluruh data yang diprediksi.

Rumus perhitungan *accuracy* adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. *Precision* (Presisi)

Precision digunakan untuk mengukur seberapa baik model dalam memprediksi kelas positif secara benar dari seluruh data yang diprediksi positif. Rumus *precision* yaitu:

$$Precision = \frac{TP}{TP + FP}$$

3. *Recall* (Sensitivitas)

Recall menunjukkan kemampuan model dalam mengenali data positif dengan tepat dari seluruh data yang sebenarnya positif. Rumus *recall* adalah sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

4. *F1-Score*

F1-score merupakan rata-rata harmonis antara nilai *precision* dan *recall* yang digunakan untuk mengetahui keseimbangan performa model dalam klasifikasi. Persamaan *F1-score* dituliskan sebagai berikut:

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

Keterangan:

- a. TP (*True Positive*) = data positif yang diprediksi positif
- b. TN (*True Negative*) = data negatif yang diprediksi negatif
- c. FP (*False Positive*) = data negatif yang diprediksi positif
- d. FN (*False Negative*) = data positif yang diprediksi negatif

2.12 Buah Matoa (*Pometia pinnata*)

Matoa (*Pometia pinnata*) adalah tanaman khas Papua Barat yang tumbuh subur di daerah dataran rendah hingga ketinggian sekitar 1.400 meter di atas permukaan laut dengan kebutuhan curah hujan yang tinggi. Buah tropis ini dapat dijumpai di beberapa wilayah Indonesia, dengan variasi warna kulit yang menarik, mulai dari hijau, merah, ungu, hingga cokelat. Dikenal memiliki rasa manis yang unik, perpaduan antara kelengkeng dan rambutan, matoa sering menjadi favorit bagi pecinta buah eksotis. Menurut beberapa penelitian tanaman matoa diketahui memiliki senyawa golongan *alkaloid*, *saponin*, *tannin*, *triterpenoid*, *flavonoid*, *fenolik* dan vitamin A, C, E yang dapat meningkatkan sistem kekebalan tubuh. Buah Matoa juga memiliki aktivitas antibakteri dan antioksidan, yang dapat digunakan sebagai komponen senyawa bioaktif. Bagian biji matoa belum dimanfaatkan secara optimal [42].

Buah matoa berbentuk bulat atau lonjong panjang 5–6 cm, dengan kulit keras namun agak tipis, dan empulur keputihan. Daging buah berwarna putih bening seperti leci dan memiliki tekstur lembut seperti kelengkeng. Berdasarkan tekstur daging buah, matoa dibedakan menjadi dua jenis, yaitu matoa kelapa dan matoa papeda. Matoa kelapa memiliki daging buah yang kenyal seperti rambutan Aceh, dengan diameter buah 2,2–2,9 cm dan diameter biji 1,25–1,40 cm. Sementara itu, matoa papeda memiliki daging buah yang agak lembek dan lengket, dengan diameter buah 1,4–2,0 cm [43].



Gambar 2. 1 Matoa Hijau



Gambar 2. 2 Matoa Kuning



Gambar 2. 3 Matoa Merah

Tanaman matoa termasuk dalam keluarga rambutan (*Sapindaceae*). Pohon ini dapat tumbuh menjulang hingga 20–40 meter dengan diameter batang mencapai sekitar 1,8 meter. Matoa memiliki daun majemuk yang tersusun berseling, terdiri atas 4 hingga 12 pasang anak daun. Saat muda, daun matoa berwarna merah cerah, kemudian berubah menjadi hijau seiring pertumbuhannya. Berdasarkan warna kulit buah, matoa dibedakan menjadi tiga jenis yaitu *Emme Anokhong* (Matoa Kulit Hijau), dan *Emme Khabhelaw* (Matoa Kulit Kuning) *Emme Bhanggahe* (Matoa Kulit Merah) [2].

2.13 UML (Unified Modeling Language)

Menurut Sulastri dkk., (2021) *Unified Modeling Language* (UML) merupakan sebuah alat yang digunakan dalam pengembangan sistem berbasis objek maupun sistem yang bersifat berkelanjutan. UML mempermudah proses perancangan sistem karena disajikan dalam bentuk visual, serta telah menjadi bahasa standar dalam penyusunan blueprint perangkat lunak. Dalam perancangan

sistem berorientasi objek, UML memiliki beberapa jenis diagram yang umum digunakan, antara lain *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Deployment Diagram* [44].

UML digunakan untuk menggambarkan desain awal dari sistem yang akan dikembangkan. Meskipun UML memiliki berbagai jenis pemodelan, hanya beberapa yang dianggap relevan untuk dibahas. Sebagai alat yang efektif dalam pengembangan sistem berbasis objek, UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang merepresentasikan konsep sistem dalam bentuk yang terstandarisasi dan mudah dipahami. Selain itu, UML juga memiliki mekanisme yang mendukung proses berbagi serta komunikasi desain dengan pihak lain secara lebih efisien [45].

Sedangkan menurut arif (2011), UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML juga didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software*. UML diharapkan mampu mempermudah pengembangan perangkat lunak serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap dan tepat [46].

2.13.1 Use Case Diagram

Use case diagram merupakan salah satu bentuk pemodelan yang digunakan dalam perancangan sistem informasi. Diagram ini menggambarkan hubungan atau interaksi antara satu maupun beberapa aktor dengan sistem yang dikembangkan. Selain itu, *use case* juga berfungsi untuk menunjukkan fitur atau

fungsi yang tersedia pada sistem serta pihak-pihak yang memiliki hak untuk menggunakan fungsi tersebut [47].

Use case diagram merupakan pemodelan untuk kelakuan sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu [48]. Menurut Y. Sugiarti *Use Case Diagram* adalah model untuk mendeskripsikan perilaku sistem yang ingin dibuat. *Use Case Diagram* menggambarkan sebuah interaksi antara satu atau lebih aktor dan sistem yang ingin dibuat [49].

2.13.2 Activity Diagram

Activity Diagram merupakan metode yang digunakan untuk menggambarkan logika prosedural dalam proses bisnis maupun alur kerja pada suatu sistem. Diagram ini memiliki fungsi yang hampir serupa dengan *flowchart*, namun keduanya mempunyai perbedaan. *Activity diagram* menampilkan rangkaian aktivitas secara menyeluruh mulai dari proses awal hingga akhir, sedangkan *flowchart* lebih berfokus pada alur logika dan algoritma suatu proses. Dengan kata lain, *activity diagram* digunakan untuk menjelaskan berbagai aliran aktivitas dalam sistem yang dirancang, termasuk bagaimana proses dimulai, keputusan yang mungkin muncul, hingga proses tersebut selesai [47].

Activity Diagram merupakan salah satu jenis diagram perilaku yang sebagian besar statusnya berisi aksi atau aktivitas, serta transisinya dipicu oleh selesainya proses sebelumnya (*internal processing*). Oleh karena itu, *activity*

diagram tidak digunakan untuk menjelaskan perilaku internal sistem maupun interaksi antar subsistem secara rinci, melainkan untuk menggambarkan alur proses dan jalannya aktivitas secara umum pada tingkat yang lebih tinggi [49].

Activity diagram menggambarkan *work flow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem [48].

2.13.3 *Sequence Diagram*

Sequence Diagram merupakan diagram yang digunakan untuk menggambarkan perilaku objek pada sebuah *Use Case* melalui interaksi berupa pesan yang dikirim dan diterima antar objek. Oleh karena itu, dalam pembuatan *sequence diagram* perlu diketahui objek-objek yang terlibat pada *Use Case* beserta class yang memiliki metode-metode yang kemudian diinstansiasi menjadi objek tersebut [49].

Sequence diagram digunakan untuk menggambarkan perilaku objek dalam suatu *use case* melalui interaksi antarobjek berupa pengiriman dan penerimaan pesan. Oleh karena itu, sebelum membuat *sequence diagram*, perlu diketahui objek-objek yang terlibat dalam *use case* tersebut beserta metode yang dimiliki oleh *class* yang diimplementasikan menjadi objek [47].

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *Sequence diagram* maka harus

diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. *Membuat Sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case* [48].

2.13.4 Class Diagram

Class Diagram digunakan untuk menggambarkan kelas-kelas beserta paket yang terdapat dalam suatu sistem. Diagram ini menyajikan struktur sistem secara statis serta hubungan antar kelas yang ada di dalamnya. Dalam pengembangan sebuah sistem, biasanya dibuat lebih dari satu class diagram untuk memperlihatkan bagian tertentu dari kelas-kelas beserta relasinya secara lebih rinci [47].

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [48].

Class Diagram merupakan diagram yang digunakan untuk menggambarkan objek-objek yang terdapat dalam suatu sistem beserta hubungan statis antarobjek tersebut. Selain itu, diagram ini juga menunjukkan batasan pada atribut, operasi kelas, serta relasi yang dimiliki oleh setiap objek dalam sistem [49].

2.14 Flowchart

Flowchart adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program [50]. *Flowchart*

merupakan diagram yang menggambarkan urutan langkah-langkah dalam menyelesaikan suatu permasalahan. Pembuatan *flowchart* bertujuan untuk memvisualisasikan proses penyelesaian masalah menggunakan simbol-simbol standar secara jelas, sistematis, terstruktur, dan mudah dipahami tanpa menimbulkan ambiguitas [51].

Menurut Wibawanto (2017), *flowchart* merupakan bagan yang menggunakan simbol-simbol tertentu untuk menggambarkan urutan proses secara rinci serta hubungan antarproses atau instruksi dalam suatu program. Diagram alur ini digunakan untuk menunjukkan aliran kendali suatu algoritma, yaitu bagaimana rangkaian kegiatan dijalankan secara logis dan sistematis [52].

2.15 Bahasa Pemrograman

2.15.1 Python

Python adalah bahasa pemrograman yang mempunyai fungsi yang begitu banyak. Mulai dari pembuatan *website*, program berbasis GUI (*Grafik User Interface*), Aplikasi Android, serta pemanfaatan dalam bidang pengembangan machine learning dan AI (*Artificial Intelligence*) yang dimanfaatkan dari segala bidang. Mulai dari bidang militer, kesehatan, pertanian, serta pendidikan [53].

Bahasa *Python* dirancang untuk membantu programmer membangun kode yang mudah dibaca, mudah dimengerti, dan mudah dirawat. *Python* telah digunakan untuk membangun berbagai jenis aplikasi, mulai dari program desktop hingga situs *web* dan aplikasi *mobile*. Selain itu, *Python* juga sangat fleksibel dan dapat digunakan dalam berbagai macam proyek dan aplikasi, termasuk pemrosesan data, kecerdasan buatan, pembelajaran mesin, dan pengembangan

game. Dalam hal penggunaannya, *Python* juga memiliki sintaks yang mudah dipelajari, sehingga membuatnya menjadi bahasa pemrograman yang ideal bagi pemula maupun pengembang berpengalaman [54].

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer, dikenal karena sintaksnya yang sederhana dan kemampuannya dalam menangani berbagai tugas, mulai dari pengembangan situs *web* hingga analisis data dan pembelajaran mesin [55].

Ada banyak *library* dan *library-library* ini memiliki fitur yang memadai untuk mempermudah proses implementasi dan pengembangan proyek CNN [56]. yaitu sebagai berikut :

1. *Tensorflow* adalah platform *open source end-to-end* yang dibuat dan dikembangkan oleh *Google Brain* untuk mendukung *Machine Learning*. *Tensorflow* memiliki alat, pustaka, dan sumber daya yang komprehensif dan fleksibel yang memungkinkan penelitian menggunakan *Machine Learning* yang mutakhir dan *developer* dengan mudah membangun dan menerapkan aplikasi yang didukung *Machine Learning*. *Tensorflow* dapat digunakan dengan bahasa pemrograman *Python* yang stabil. *Tensorflow* menawarkan pemrograman yang melakukan berbagai macam tugas ML/DL (regresi dan klasifikasi).
2. Keras adalah sebuah API (*Application Programming Interface*) tingkat tinggi dari *Tensorflow* yang dibuat oleh François Chollet, seorang *engineer* di *Google*. Keras dirancang agar mudah dipelajari dan digunakan, terutama bagi para peneliti, sehingga mereka dapat fokus pada eksperimen dan

bukan pada penggunaan *library*. Keras dapat digunakan untuk membangun CNN, seperti melakukan *one-hotencoding* pada label citra, membuat arsitektur CNN, menggunakan arsitektur CNN populer yang tersedia, melakukan *plotting model* CNN, memuat dataset citra ke dalam arsitektur CNN, serta melakukan *training* dan *testing model* CNN untuk menghasilkan evaluasi seperti akurasi dan *loss*. Selain itu, Keras juga dapat digunakan untuk menyimpan dan mengakses model CNN dalam format h5.

3. *Sklearn* Sebuah tools untuk *Machine Learning* yang digunakan untuk berbagai model statistik seperti *classification*, *regression*, *clustering*, dan *dimensionality reduction*. Dalam implementasi CNN, *Sklearn* dapat digunakan sebagai alat untuk menerapkan teknik *K-Fold Cross Validation* pada dataset, membuat *Confusion Matrix*, dan *Classification Report*.
4. *Numpy* Sebuah *library* yang berfungsi untuk memanipulasi dan mengolah array serta matriks. Salah satu contoh penggunaannya dalam CNN adalah dengan melakukan reshape pada dimensi citra, menghitung rata-rata dan standar deviasi akurasi model CNN, dan melakukan pengolahan data.
5. *Pandas* Berguna untuk melakukan manipulasi dan analisis data. Dalam konteks penggunaan pada CNN, contohnya adalah menyimpan riwayat pelatihan model (dalam bentuk akurasi dan loss) ke dalam sebuah tabel CSV.
6. *Matplotlib* Digunakan untuk membuat grafik dan visualisasi data. Contoh penggunaannya pada CNN adalah memplot dataset beserta labelnya dan

memplot hasil pelatihan dan pengujian model CNN ke dalam grafik untuk memudahkan pemahaman dan analisis.

2.15.2 HTML (*HyperText Markup Language*)

Hyper Text Markup Language (HTML) merupakan bahasa yang paling banyak dipakai untuk membangun halaman *web*. HTML dibuat oleh Berners-Lee pada akhir 1991, namun HTML 2.0 merupakan HTML pertama dengan spesifikasi standar yang kemudian dipublikasikan pada 1995. Saat ini kita menggunakan HTML versi 5 yang merupakan pengembangan dari HTML 4.01, dan versi ini dipublikasikan pada 2012. Awalnya, HTML dikembangkan untuk mendefinisikan struktur dokumen, seperti *heading*, *paragraph*, *list*, dan sebagainya untuk memfasilitasi informasi keilmuan di antara para peneliti. Namun, sekarang HTML banyak digunakan untuk mengatur format halaman web [57].

HTML merupakan teknologi dasar dan keterampilan penting bagi siapa saja yang tertarik dalam pengembangan *web* atau komunikasi digital. Makalah ini mengeksplorasi peran HTML dalam meningkatkan keterampilan komunikasi dengan memungkinkan individu membuat dan menerbitkan konten di *web* [58]. Selain itu, penggunaan HTML juga dapat membantu meningkatkan kemampuan seseorang dalam mempresentasikan informasi secara jelas dan efektif melalui *website* [59].

2.15.3 CSS (*Cascading Style Sheet*)

CSS merupakan *Cascading Style Sheet*, berguna untuk mempercantik tampilan HTML atau menentukan bagaimana elemen HTML ditampilkan, seperti

menentukan posisi, merubah warna teks atau background dan lain sebagainya [60]. Secara fungsional, CSS digunakan untuk mengatur estetika dan tata letak halaman *web* dengan memilih elemen tertentu, kemudian memberikan properti dan nilai sesuai kebutuhan tampilan. Aturannya terdiri dari *selector*, *property*, dan *value* sebagai komponen utama dalam proses styling [61].

Cascading Style Sheets (CSS) digunakan untuk mengontrol gaya dari dokumen *web* dengan sederhana dan mudah. CSS merupakan bahasa desain yang bertujuan untuk mempermudah proses pembuatan halaman *web* yang menarik. CSS bertanggung jawab terhadap tampilan halaman *web*. Dengan CSS, kita dapat mengatur warna teks, gaya font, jarak antar paragraf, dan berbagai elemen lainnya [57].

2.15.4 JavaScript

JavaScript merupakan bahasa pemrograman tingkat tinggi yang digunakan bersama HTML dan CSS untuk menghadirkan fitur interaktif dan dinamis pada halaman *web* [62]. Bahasa ini awalnya dikembangkan oleh Brendan Eich pada *browser Netscape* dengan nama *Mocha*, kemudian berubah menjadi *LiveScript*, sebelum akhirnya dikenal sebagai *JavaScript*. Sebagai bahasa *client-side*, *JavaScript* dijalankan langsung oleh *browser* sehingga memungkinkan penambahan fungsi yang tidak dapat dicapai hanya dengan HTML [61].

Dalam praktik pengembangan *web* modern, hampir seluruh situs memanfaatkan *JavaScript* karena dukungannya yang luas di berbagai perangkat dan *browser*. Sifatnya yang dinamis serta kemampuannya mendukung paradigma berorientasi objek dan fungsional membuat *JavaScript* fleksibel untuk beragam

kebutuhan, baik di sisi client maupun server, sehingga mampu meningkatkan interaktivitas dan responsivitas halaman web [63].

2.16 Draw.Io

Menurut Arianto dkk., (2021) *Draw io* adalah website dan software yang digunakan untuk membuat *flowchat*, *draw io* berguna untuk merancang *Use Case diagram* maupun *activity diagram* [64].

Draw io merupakan alat diagram berbasis web yang digunakan untuk membuat berbagai macam jenis diagram seperti *Flowchart*, *usecase*, *ERD*, dan lain-lain. *Draw io* menyediakan berbagai macam template yang dapat digunakan untuk membuat diagram, sehingga memudahkan dalam proses pembuatan *Flowchart* dan juga *usecase* [65].

Draw.io adalah sebuah website yang didesain khusus untuk menggambarkan diagram UML secara *online*. Dimana punya tampilan yang sangat *responsive* dan terintegrasi dengan layanan penyimpanan file milik *google* yaitu *Google Drive* sehingga *draw.io* menjadi alternatif dalam pembuatan diagram UML dengan waktu yang lebih singkat [66].

2.17 Alat Bantu Pemrograman

2.17.1 OpenCV (Open Source Computer Vision Library)

Open Source Computer Vision Library (OpenCV) adalah sebuah software yang bersifat *open-source* yang berisikan *library* yang mendukung pendeteksian objek yang mudah diimpor kedalam bahasa pemrograman *Java* [67]. Menurut Randy Moh Yusup dkk , *OpenCV* adalah perpustakaan *computer vision* sumber

terbuka secara luas yang dikenal dengan koleksi fungsi dan algoritma yang luas [68].

Menurut D. Arniady, *OpenCV* adalah sebuah *library* (perpustakaan) yang digunakan untuk mengolah gambar dan video hingga kita mampu mengekstrak informasi di dalamnya. *OpenCV* dapat berjalan di berbagai bahasa pemrograman, seperti *C*, *C++*, *Java*, *Python*, dan juga didukung di berbagai platform seperti *Windows*, *Linux*, *Mac OS*, *iOS* dan *Android* [69].

2.17.2 *Visual Studio Code*

Visual Studio Code (VS Code) adalah editor teks lintas platform yang dikembangkan oleh Microsoft, memiliki fitur lengkap dan handal untuk mendukung pengembangan perangkat lunak. Aplikasi ini dapat dijalankan di berbagai sistem operasi seperti *Windows*, *Linux*, dan *macOS*. *VS Code* mendukung banyak bahasa pemrograman, seperti *JavaScript*, *TypeScript*, dan *Node.js*, serta dilengkapi dengan fitur-fitur seperti penyorotan sintaks (*syntax highlighting*), *IntelliSense*, *debugging*, dan integrasi dengan sistem *version control*, sehingga menjadikannya salah satu editor kode yang paling populer di kalangan *programmer* [70].

Visual Studio Code adalah editor kode ringan lintas platform dari *Microsoft* yang berjalan di *Windows*, *Linux*, dan *macOS*. Editor ini mendukung *JavaScript*, *TypeScript*, dan *Node.js* secara bawaan, serta bahasa lain melalui ekstensi. Dibangun dengan *Electron*, *VS Code* dilengkapi fitur seperti *Command Palette*, *Terminal*, dan *Extensions*, serta terintegrasi dengan berbagai tools

pengembangan. Keunggulannya meliputi gratis, fleksibel, kaya ekstensi, dan mendukung berbagai bahasa pemrograman [71].

Visual Studio Code adalah perangkat lunak editor kode sumber yang dikembangkan oleh *Microsoft* dan dapat digunakan pada berbagai sistem operasi seperti *Linux*, *macOS*, dan *Windows*. Editor ini mendukung pengeditan berbagai bahasa pemrograman, termasuk *JavaScript*, *HTML*, *CSS*, dan lainnya [72].

2.17.3 *Google Colaboratory*

Menurut Carniero et al., (2018) dalam penelitian “*Google Colab and Virtual Simulations: Practical e-Learning Tools to Support the Teaching of Thermodynamics and to Introduce Coding to Students*” *Google Colab*, terkadang dikenal sebagai *Google Colaboratory*, adalah layanan pembelajaran mesin gratis yang ditawarkan oleh *Google*. Platform ini memungkinkan pengguna untuk menelusuri dan menjalankan kode *Python*, mengunggah dan menyimpan kode bersama data, dan memungkinkan kolaborasi waktu nyata dengan pengguna lain. Selain itu, *Google Colab* menyediakan akses ke mesin virtual dengan CPU, GPU, dan TPU (*Unit Pemrosesan Tensor*) yang dapat digunakan untuk menjalankan kode *Python* [73].

Google Collaboratory, juga dikenal sebagai *Google Colab*, adalah alat gratis yang berbasis *cloud* yang digunakan untuk tujuan pendidikan dan penelitian [74]. Alat ini dapat diakses secara gratis melalui internet. *Google Colab* dapat digunakan oleh dosen dan mahasiswa sebagai catatan penelitian atau catatan kuliah. *Google Colab* menarik karena memberikan layanan GPU gratis kepada

penggunanya sebagai backend komputasi dan memungkinkan untuk menggunakannya selama 12 jam sekaligus [75].

2.17.4 Web Browser

Web browser adalah aplikasi perangkat lunak yang digunakan untuk mengakses dan menampilkan sumber informasi. *Web browser* merupakan perangkat lunak yang dapat memproses paket HTTP dan menampilkannya kembali kepada *user* dengan format HTML [76]. Menurut Wardah & Fitriana (2018), web browser merupakan perangkat atau media yang digunakan untuk membaca kode HTML serta menerjemahkannya ke dalam tampilan visual yang dapat dilihat oleh pengguna [77].

Web Browser adalah *software/aplikasi* yang digunakan untuk mengakses halaman *web* yang ditampilkan. Lebih khusus lagi, *browser web* adalah aplikasi perangkat lunak yang digunakan untuk mencari, mengambil, dan menampilkan informasi di *World Wide Web*, termasuk halaman *web*, foto, video, dan file lainnya. *Browser* juga memiliki kemampuan untuk menampilkan kode semantik seperti *HTML*, *JavaScript*, *CSS* dan bahasa pemrograman *website* pada halaman yang mudah dipahami semua orang. Ada beberapa jenis *browser* yang tersedia untuk pengguna Internet. Berikut adalah contoh *browsers*nya [61] :

1. *Google Chrome*

Google Chrome adalah penjelajah *web* sumber terbuka yang dikembangkan oleh perusahaan *Google* dengan menggunakan mesin rendering *WebKit*. Proyek sumber terbukanya *chrome* dinamakan dengan *Chromium*.

2. *Mozilla Firefox Browser*

Mozilla Firefox mulanya bernama *Phoenix*, kemudian sekarang ini dikenal dengan nama *Mozilla Firebird*. *Mozilla Firefox* adalah penjelajah *web* antar platform *free* (gratis) yang dikembangkan oleh Yayasan *Mozilla* dan ratusan sukarelawan.

3. *Internet Explorer*

Internet Explorer atau yang dikenal dengan disingkat *IE* atau *MSIE* adalah *browser web proprieter* yang secara gratis dari *Microsoft*.

2.18 ***Blackbox Testing***

Black box testing adalah metode pengujian yang berfokus pada fungsi perangkat lunak, untuk memastikan proses input berjalan sesuai harapan serta data yang disimpan secara eksternal tetap akurat dan mutakhir [78].

Pada tahap ini, sistem akan menjalani proses pengujian secara menyeluruh setelah seluruh komponennya selesai dikembangkan. Setiap bagian sistem diperiksa menggunakan metode *Black Box Testing* guna mengurangi kemungkinan terjadinya kesalahan pada sistem. *Black Box Testing* merupakan metode pengujian perangkat lunak yang berfokus pada fungsi dan keluaran sistem tanpa meninjau struktur atau proses internal di dalamnya [79].

Teknik *Black Box Testing* memungkinkan penguji memperoleh berbagai masukan dari pengguna untuk mengidentifikasi kesalahan yang terdapat pada perangkat lunak. Metode ini berfokus pada pencarian beberapa jenis kesalahan, antara lain [80]:

1. Fungsi sistem yang tidak berjalan dengan benar.
2. Kesalahan pada antarmuka (*interface*).
3. Kesalahan pada struktur data.
4. Masalah pada kinerja sistem (*performance*).
5. Kesalahan pada proses inisialisasi dan terminasi.

2.19 Penelitian Terkait

Berikut ini merupakan beberapa penelitian terdahulu yang relevan dan berkaitan dengan skripsi Penerapan *Deep Learning* Arsitektur *MobileNetV2* untuk Klasifikasi Jenis Buah Matoa Berdasarkan Citra Warna Kulit Matang, yang dijadikan sebagai landasan dalam mendukung serta memperkuat penelitian yang dilakukan.

Tabel 2. 1 Penelitian Terkait

No	Nama	Tahun	Judul	Hasil
1	Tedi Hadi Mursid, M. Rafi Muttaqin, Agus Sunandar	2025	Klasifikasi Buah Manggis Lokal dan Ekspor Berdasarkan Warna dengan Arsitektur <i>MobileNetV2</i>	Penelitian ini menerapkan metode CNN dengan arsitektur <i>MobileNetV2</i> untuk klasifikasi buah manggis berdasarkan warna kulit menggunakan 384 citra. Model yang dihasilkan mencapai akurasi 95%, menunjukkan bahwa <i>MobileNetV2</i> efektif dan efisien untuk klasifikasi citra pada

				perangkat dengan sumber daya terbatas.
2	Sephia Ceren	2026	Sistem Klasifikasi Penyakit pada Buah Jeruk Menggunakan <i>MobileNetV2</i>	Penelitian ini menerapkan CNN dengan arsitektur <i>MobileNetV2</i> untuk klasifikasi penyakit buah jeruk menggunakan pembagian data 80% latih dan 20% uji serta <i>preprocessing resizing</i> dan normalisasi. Model mencapai akurasi 96,67% dan terbukti efektif untuk identifikasi penyakit secara otomatis dan akurat.
3	Dede Wulan Rahayu, David Setiadi, Dwi Yuniarto	2024	Penerapan <i>Deep Learning</i> untuk Klasifikasi Buah Berdasarkan Citra dengan Metode <i>Convolutional Neural Networks</i>	Model CNN yang dikembangkan mampu mengklasifikasikan berbagai jenis buah dengan akurasi tinggi di atas 96% pada data pelatihan dan validasi. Selain itu, nilai <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> mendekati 1.0, menunjukkan performa model yang sangat baik dan mampu menggeneralisasi data dengan baik tanpa overfitting yang signifikan. Penelitian ini membuktikan bahwa CNN efektif untuk meningkatkan kecepatan dan akurasi klasifikasi buah serta berpotensi diterapkan dalam sektor pertanian dan industri pangan.
4	Hadi Rizky Dwi Via Prahesta, Danar Putra Pamungkas, Patmi Kasih	2025	Analisis Performa CNN Arsitektur <i>MobileNet</i> untuk Pengenalan Penyakit pada Tanaman	Penelitian ini mengembangkan sistem klasifikasi penyakit bawang merah berbasis citra menggunakan CNN <i>MobileNetV1</i> dengan 250 data dalam 4 kelas serta

			Bawang Merah	teknik augmentasi, <i>EarlyStopping</i> , dan <i>ModelCheckpoint</i> . Model mencapai akurasi 92% dengan performa yang baik, meskipun terdapat sedikit overfitting akibat keterbatasan data.
5	Novia Amanda, Resty Wulanningrum, Julian Sahertian	2025	Penerapan <i>MobileNetV2</i> untuk Klasifikasi Kondisi Buah Alpukat Mentega	Penelitian ini menerapkan CNN <i>MobileNetV2</i> untuk mengklasifikasikan alpukat mentega ke dalam tiga kelas menggunakan 1.500 citra yang telah dipreproses. Model dilatih 10 epoch (<i>batch size</i> 32) dan mencapai akurasi 98% serta 99,33% pada pengujian, menunjukkan kinerja klasifikasi yang sangat baik.
6	Herwina, Darmatasia, Ashabul Kahfi Ash Shiddiq, Theddy Dzikrullah Syahputra	2022	Deteksi Penyakit pada Tanaman Padi Menggunakan <i>MobileNet Transfer Learning</i> Berbasis Android	Penelitian ini mengembangkan sistem deteksi penyakit daun padi berbasis CNN dengan <i>transfer learning</i> menggunakan <i>MobileNetV1</i> dan <i>MobileNetV2</i> , dengan akurasi masing-masing 94% dan 95%. Model terbaik diimplementasikan pada aplikasi <i>Android (TensorFlow Lite)</i> untuk mendeteksi tiga penyakit secara otomatis, sehingga membantu deteksi dini dan mengurangi risiko gagal panen.
7	Bima Mahardhika, Ghovin Suraju	2025	Implementasi Pendeteksi Penyakit pada Buah Alpukat Menggunakan Metode CNN	Penelitian ini menggunakan CNN <i>MobileNetV2</i> untuk mendeteksi penyakit buah alpukat dengan data yang dipreproses (<i>resize</i> 224×224 dan augmentasi)

				serta pembagian 70:30 untuk pelatihan dan validasi. Model menghasilkan akurasi tinggi dan diimplementasikan dalam aplikasi mobile untuk membantu identifikasi penyakit secara cepat dan efisien.
8	Nuryanto, Danar Putra Pamungkas, Danang Wahyu Widodo	2024	Penerapan Metode <i>Convolutional Neural Network</i> Pada Tanaman Bawang Merah	Penelitian ini menggunakan CNN untuk mengklasifikasikan penyakit bawang merah dari citra daun dengan 480 data (400 training, 40 testing, 40 validasi). Model dilatih 20 epoch (<i>batch size</i> 32) dan mencapai akurasi 97%, menunjukkan efektivitas dalam deteksi dini penyakit pada tanaman.
9	Bima Mahardhika, Umi Mahdiah, Resty Wulanningrum	2025	Deteksi Penyakit pada Buah Alpukat Menggunakan Metode CNN	Penelitian ini mengembangkan sistem deteksi penyakit alpukat berbasis CNN <i>MobileNetV2</i> dengan empat kelas melalui preprocessing (segmentasi, grayscale, augmentasi) dan pembagian data 40:20:40. Model mencapai akurasi 96,5% dengan metrik evaluasi tinggi, mampu mengklasifikasikan sebagian besar kelas dengan baik, serta berpotensi diterapkan pada aplikasi mobile untuk identifikasi cepat dan akurat.
10	Mohamad Ansori Pratama	2025	Implementasi Arsitektur <i>MobileNetV2</i> untuk Deteksi	Penelitian ini menerapkan CNN <i>MobileNetV2</i> untuk mendeteksi antraknosa dan busuk buah pada cabai

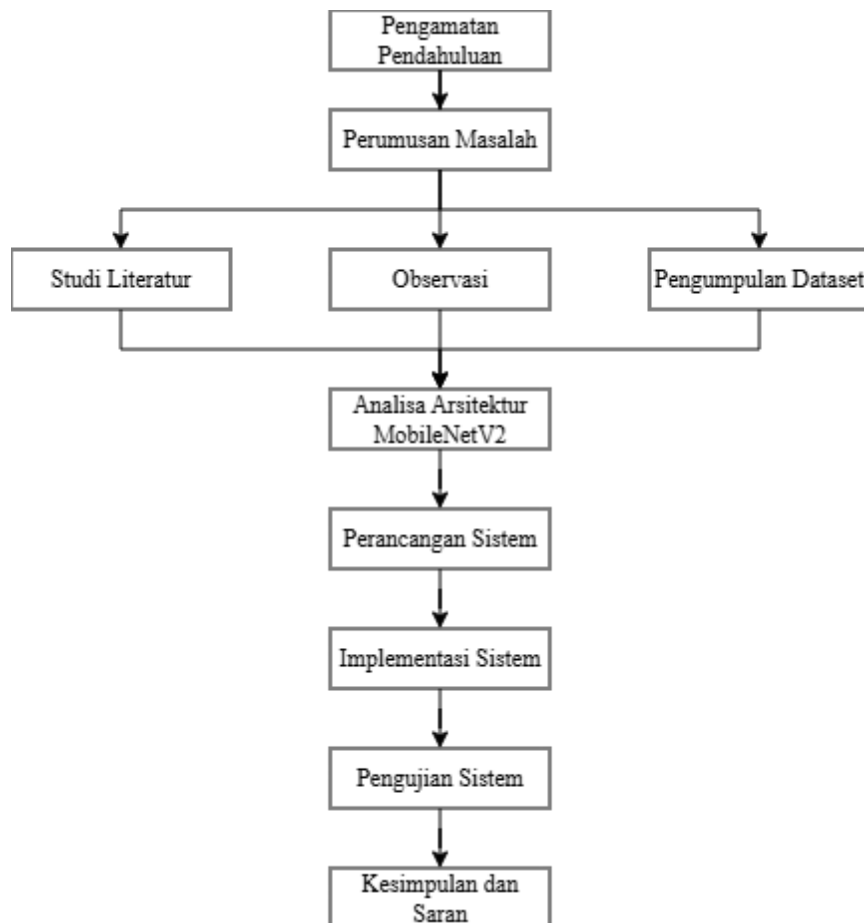
			Penyakit Antraknosa dan Busuk Buah pada Cabai Rawit	rawit dengan data hasil preprocessing dan augmentasi. Model dilatih 100 epoch dan mencapai akurasi 0,9736 (training) dan 1,000 (validasi) dengan loss rendah, menunjukkan kinerja baik meski masih terdapat sedikit kesalahan sehingga memerlukan penambahan data.
--	--	--	---	--

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian merupakan tahapan atau langkah-langkah yang dilakukan dalam penelitian. Tujuan dari metodologi penelitian ini agar penelitian yang dibuat sesuai dengan tujuan dari penelitian itu sendiri. Pada penelitian ini terdapat beberapa tahapan yang akan dilakukan seperti yang akan dilakukan seperti pada gambar berikut :



Gambar 3. 1 Metodologi Penelitian

3.2 Pengamatan Pendahuluan

Tahap pengamatan pendahuluan dilakukan untuk mengidentifikasi fenomena dan permasalahan yang terjadi dalam proses klasifikasi jenis buah matoa berdasarkan warna kulit matang. Peneliti mengamati bahwa proses identifikasi secara manual memiliki keterbatasan, seperti subjektivitas penilaian, ketidakkonsistenan hasil, serta kurang efisien ketika jumlah data semakin besar.

Berdasarkan kondisi tersebut, diperlukan suatu pendekatan berbasis teknologi yang mampu melakukan klasifikasi secara otomatis dan konsisten. Oleh karena itu, penelitian ini mengusulkan penggunaan metode *deep learning* untuk meningkatkan akurasi dan efisiensi dalam proses klasifikasi citra buah matoa.

3.3 Perumusan Masalah

Tahap perumusan masalah bertujuan untuk mendefinisikan permasalahan penelitian secara spesifik dan terarah. Permasalahan utama dalam penelitian ini adalah bagaimana merancang dan mengimplementasikan sistem klasifikasi jenis buah matoa berdasarkan citra warna kulit matang menggunakan pendekatan *deep learning*. Secara lebih rinci, penelitian ini berfokus pada penerapan arsitektur *MobileNetV2* untuk menghasilkan model yang mampu mengklasifikasikan citra secara akurat, serta mengevaluasi performa model yang dihasilkan dalam kondisi data yang bervariasi.

3.3.1 Studi Literatur

Studi Literatur dilakukan dengan cara mencari referensi teori-teori dan penelitian terkait yang sesuai dengan kasus atau permasalahan dalam pembuatan

sistem ini. Referensi didapat dari buku, jurnal, artikel dan berbagai sumber lainnya.

3.3.2 Observasi (Pengamatan)

Observasi atau pengamatan langsung merupakan metode pengumpulan data yang dilakukan dengan cara mengamati secara langsung objek penelitian menggunakan indera penglihatan untuk memperoleh informasi yang relevan dengan kebutuhan penelitian, yaitu buah matoa. Peneliti mengamati karakteristik visual buah, khususnya warna kulit pada kondisi matang, yang menjadi parameter utama dalam proses klasifikasi.

3.3.3 Pengumpulan Dataset

Pada tahap ini, peneliti mengumpulkan data berupa citra buah matoa yang akan digunakan sebagai dataset penelitian. Data diperoleh melalui pengambilan gambar secara langsung maupun dari sumber lain yang relevan, dengan memperhatikan variasi kondisi untuk meningkatkan kualitas dataset. Dataset yang telah dikumpulkan kemudian dikelompokkan berdasarkan kategori tertentu sesuai kebutuhan penelitian. Pengelompokan ini bertujuan untuk mempermudah proses pelatihan model serta meningkatkan kemampuan sistem dalam mengenali pola pada citra.

3.4 Analisa Arsitektur *MobileNetV2*

Tahap analisa dilakukan untuk memahami struktur dan karakteristik arsitektur *MobileNetV2* yang digunakan dalam penelitian. *MobileNetV2* dipilih karena memiliki keunggulan dalam efisiensi komputasi dan ukuran model yang ringan, sehingga cocok untuk implementasi pada perangkat dengan keterbatasan

sumber daya. Analisis juga mencakup cara kerja setiap *layer*, mekanisme *feature extraction*, serta bagaimana model ini dapat diadaptasi untuk tugas klasifikasi citra buah matoa. Hasil analisis ini menjadi dasar dalam proses perancangan sistem.

3.5 Perancangan Sistem

Perancangan sistem dilakukan untuk menentukan alur kerja sistem klasifikasi yang akan dibangun. Tahap ini mencakup desain proses mulai dari input berupa citra, pengolahan data, hingga menghasilkan *output* berupa hasil klasifikasi. Selain itu, perancangan juga meliputi pemilihan perangkat lunak, tools, serta arsitektur sistem yang akan digunakan. Tujuan dari tahap ini adalah memastikan bahwa sistem dapat berjalan secara terstruktur dan sesuai dengan kebutuhan penelitian.

3.6 Implementasi Sistem

Tahap implementasi merupakan proses penerapan rancangan sistem ke dalam bentuk nyata. Pada tahap ini dilakukan pembangunan model *deep learning*, integrasi dataset, serta pengkodean menggunakan bahasa pemrograman dan *framework* yang sesuai. Implementasi juga mencakup proses pelatihan model menggunakan dataset yang telah disiapkan, sehingga sistem dapat belajar mengenali pola dan karakteristik citra buah matoa untuk keperluan klasifikasi.

3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi kinerja model dalam mengklasifikasikan jenis buah matoa berdasarkan citra warna kulit matang. Tahap ini bertujuan untuk mengetahui tingkat akurasi serta keandalan sistem yang telah

dibangun. Selain itu, pengujian juga dilakukan untuk memastikan bahwa sistem mampu bekerja dengan baik pada berbagai kondisi data. Hasil pengujian akan digunakan sebagai dasar dalam analisis performa dan penarikan kesimpulan penelitian.

3.8 Kesimpulan dan Saran

Tahapan terakhir dalam penelitian ini adalah penarikan kesimpulan berdasarkan hasil yang diperoleh dari proses pengembangan dan pengujian model klasifikasi menggunakan arsitektur *MobileNetV2* terhadap citra buah matoa. Pada tahap ini, peneliti merangkum hasil kinerja model dalam mengklasifikasikan jenis buah matoa berdasarkan warna kulit matang. Selain itu, pada bagian ini juga disampaikan saran yang bertujuan untuk memberikan masukan bagi pengembangan penelitian selanjutnya, baik dari segi peningkatan kualitas dataset, metode yang digunakan, maupun pengembangan sistem agar lebih optimal dan bermanfaat.