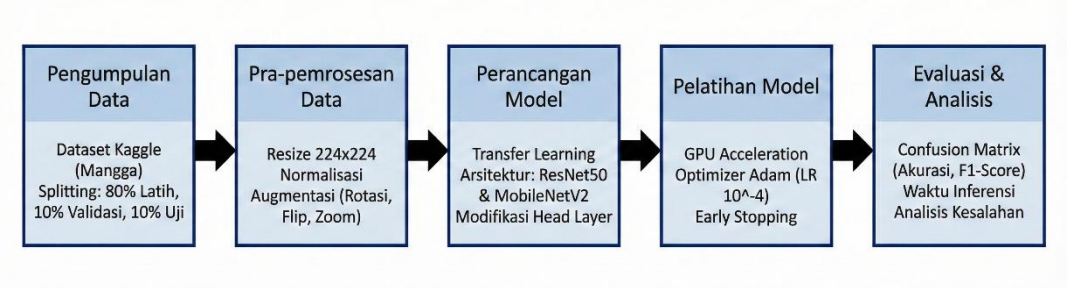
BAB III
METODOLOGI PENELITIAN

Bab ini mendeskripsikan secara komprehensif seluruh tahapan penelitian yang telah dilaksanakan, mencakup desain penelitian, strategi pengumpulan dan pemrosesan data, perancangan arsitektur model, konfigurasi pelatihan beserta justifikasi teknis mendalam atas setiap *hyperparameter* kritis, hingga mekanisme evaluasi komparatif yang diterapkan.

3.1 Desain, Strategi, dan Tahapan Penelitian

Penelitian ini menggunakan desain penelitian komparatif dengan pendekatan kuantitatif *in-silico*. Pendekatan ini memungkinkan pengukuran perbedaan performa antara ResNet50 dan MobileNetV2 secara objektif dan terukur. Strategi penelitian dirancang untuk menemukan model dengan *trade-off* paling optimal antara akurasi dan efisiensi komputasi. Tahapan yang telah dilaksanakan dalam penelitian ini divisualisasikan secara sistematis pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

Berdasarkan Gambar 3.1, tahapan penelitian ini secara berurutan meliputi: Pengumpulan Data, Prapemrosesan Data, Perancangan Model, Pelatihan Model (Training), serta Pengujian dan Evaluasi Kinerja.

3.2 Tahap Pengumpulan Data

3.2.1 Spesifikasi Dataset

Penelitian ini menggunakan dataset citra buah mangga dari platform repositori publik Kaggle (URL: https://www.kaggle.com/datasets/warcoder/mangofruitdds). Dataset riil yang diolah berjumlah total 1.700 citra digital buah mangga, diklasifikasikan ke dalam

dua kelas: Layak (*Fresh/Good*) dan Tidak Layak (*Rotten/Defective*). Seluruh citra memiliki variabilitas latar belakang tinggi (*cluttered background*) yang merepresentasikan lingkungan penyortiran nyata.

3.2.2 Pembagian Data (Data Splitting)

Menggunakan skema *Holdout Validation* dengan stratifikasi proporsional untuk menjaga distribusi kelas di ketiga subset:

Tabel 3.1 Pembagian Data (Holdout Validation)

Subset Data	Proporsi	Jumlah Citra	Fungsi
Data Latih (Training Set)	80%	±1.360 citra	Pembaruan bobot model selama pelatihan.
Data Validasi (Validation Set)	10,6%	±180 citra	Pemantauan konvergensi dan pemicu callback.
Data Uji (Testing Set)	9,4%	±160 citra	Evaluasi akhir pada unseen data.

3.3 Tahap Prapemrosesan Data (Preprocessing)

3.3.1 Resizing ke Dimensi 224x224 Piksel

Seluruh citra diubah ukurannya menjadi dimensi standar 224×224 piksel. Penetapan dimensi ini memiliki landasan teknis yang kuat: ukuran tersebut merupakan dimensi *input* spasial standar baku untuk ResNet50 maupun MobileNetV2 dengan bobot *pretrained* ImageNet. Penggunaan dimensi berbeda akan menciptakan ketidaksesuaian (*mismatch*) dengan representasi fitur internal bobot tersebut. Selain kompatibilitas teknis, 224×224 piksel merepresentasikan titik keseimbangan optimal: cukup tinggi untuk melestarikan fitur patologis halus (bercak antraknosa awal, jejak *oviposition* lalat buah), sekaligus efisien secara komputasional untuk inferensi *real-time* pada perangkat terbatas (He, Zhang, Ren, & Sun, 2016) (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018).

3.3.2 Normalisasi Piksel

Nilai piksel dinormalisasi ke rentang [0, 1] melalui pembagian dengan 255. Normalisasi ini mempercepat konvergensi optimasi gradien dan menyesuaikan distribusi masukan dengan asumsi bobot *pretrained* ImageNet.

3.3.3 Augmentasi Data (On-the-fly)

Augmentasi data dinamis diterapkan secara on-the-fly menggunakan lapisan sekuensial TensorFlow (TensorFlow Sequential Layer) secara acak pada setiap batch data latih:

Tabel 3.2 Konfigurasi Augmentasi Data

Teknik Augmentasi	Parameter	Rasionalisasi
Random Rotation	±20%	Merotasikan citra secara acak hingga rentang ±20% dari sudut penuh (360°).
Horizontal Flip	Aktif	Mensimulasikan variasi posisi kiri-kanan buah saat difoto.
Vertical Flip	Aktif	Mensimulasikan variasi posisi atas-bawah buah di lapangan.
Zoom Range	±20%	Mensimulasikan variasi jarak kamera terhadap buah.
Brightness Shift	±20%	Mensimulasikan variasi kondisi pencahayaan di lapangan.

3.4 Tahap Perancangan Model

3.4.1 Konfigurasi Backbone (Ekstraktor Fitur)

Bagian *backbone* kedua model seluruh lapisan konvolusi dibekukan (*frozen*) pada fase pelatihan awal agar representasi fitur visual umum dari ImageNet tetap utuh. ResNet50 menggunakan 48 lapisan konvolusi dalam 16 *Bottleneck*

Residual Block, sementara MobileNetV2 menggunakan 17 *Inverted Residual Block*.

3.4.2 Konfigurasi Lapisan Head (Klasifikasi)

Lapisan *head* baru yang identik ditambahkan pada kedua arsitektur:

Tabel 3.3 Konfigurasi Lapisan Head Klasifikasi

Lapisan	Konfigurasi	Fungsi
Global Average Pooling 2D	-	Mereduksi feature map 3D menjadi vektor 1D.
Dense (*Dropout*)	ReLU, *Dropout Rate* = 0,5	Transformasi fitur non-linear dan regularisasi eksplisit (menonaktifkan 50% neuron acak selama pelatihan).
Dense (*Output*)	1 neuron, *Sigmoid*	Menghasilkan probabilitas [0,1] untuk klasifikasi biner.

3.5 Tahap Pelatihan Model (Training)

3.5.1 Fungsi Kerugian dan Algoritma Optimasi

Proses pelatihan menggunakan fungsi kerugian Binary CrossEntropy:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (3.1)$$

Algoritma optimasi Adam dipilih karena menggabungkan keunggulan Momentum dan RMSProp untuk pembaruan bobot adaptif perparameter yang lebih cepat konvergen (Kingma & Ba, 2015).

3.5.2 Justifikasi Teknis Learning Rate Fase Transfer Learning (10^{-4})

Learning rate awal ditetapkan pada 10^{-4} untuk fase *Transfer Learning* dasar. Penetapan ini didasarkan pada dua kondisi yang harus dipenuhi secara simultan:

1. Cukup kecil untuk menjaga stabilitas pelatihan: Nilai 10^{-4} memastikan pembaruan bobot lapisan *head* berlangsung dalam skala yang proporsional

dan terkendali, mencegah gradien yang tidak stabil yang dapat mengganggu konvergensi keseluruhan.

2. Cukup besar untuk konvergensi efisien: Lapisan *head* diinisialisasi dengan bobot acak dan membutuhkan pembaruan signifikan untuk beradaptasi pada distribusi fitur domain mangga. Nilai 10^{-4} memberikan laju perubahan substansial untuk mencapai konvergensi tanpa risiko divergensi (Kingma & Ba, 2015) (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018).

3.5.3 Justifikasi Teknis Learning Rate Fase Fine-Tuning (10^{-5})

Pada eksperimen *Fine-Tuning* ResNet50, nilai *learning rate* diturunkan drastis menjadi 10^{-5} satu per sepuluh dari fase sebelumnya. Penurunan ekstrim ini memiliki dua justifikasi teknis fundamental:

1. Pencegahan Catastrophic Forgetting: Ketika lapisan *backbone* dibuka kuncinya (*unfreeze*), terdapat risiko serius *catastrophic forgetting* pembaruan bobot agresif yang menghancurkan representasi fitur visual umum yang dipelajari dari jutaan citra ImageNet. Nilai 10^{-5} memastikan setiap pembaruan sangat inkremental, memodifikasi bobot secara marginal untuk adaptasi domain tanpa menghapusnya secara destruktif.
2. Pembaruan Bobot Domain-spesifik yang Stabil: Tujuan *Fine-Tuning* adalah mengadaptasi lapisan konvolusi dalam agar responsif terhadap fitur domain-spesifik mangga gradien warna lesi antraknosa, mikrotekstur *oviposition*. Pembaruan fitur ini menuntut langkah sangat kecil dan presisi: terlalu besar menyebabkan *catastrophic forgetting*, terlalu kecil menyebabkan stagnasi. Nilai 10^{-5} merupakan batas bawah yang andal untuk *deep layer Fine-Tuning* (He, Zhang, Ren, & Sun, 2016).

3.5.4 Konfigurasi Batch Size dan Mekanisme Callback

Tabel 3.4 Konfigurasi Hyperparameter dan Mekanisme Callback

Hyperparameter	Nilai	Justifikasi Pemilihan
Batch Size	32	Kompromi optimal efisiensi komputasi dan kualitas estimasi gradien.

Learning Rate (Transfer Learning)	10^{-4}	Cukup kecil untuk stabilitas pelatihan head, cukup besar untuk konvergensi efisien.
Learning Rate (Fine-Tuning)	10^{-5}	Mencegah catastrophic forgetting dan memastikan pembaruan bobot domain-spesifik yang stabil.
Early Stopping Patience	10 epoch	Menghentikan pelatihan jika validation loss stagnan 10 epoch berturut-turut.
ReduceLROnPlateau Factor	0,5 (Fase Awal) / 0,2 (*Fine-Tuning*)	Menurunkan *learning rate* secara adaptif saat validasi stagnan untuk keluar dari *plateau*.
ReduceLROnPlateau Patience	5 epoch	Penyesuaian learning rate sebelum Early Stopping aktif.

3.6 Tahap Evaluasi dan Analisis

3.6.1 Landasan Matematis Confusion Matrix dan Justifikasi Prioritas Recall

Evaluasi akhir dilakukan terhadap Data Uji yang sepenuhnya independen. *Confusion Matrix* adalah tabel kontingensi 2×2 yang memetakan hasil prediksi terhadap label sebenarnya:

1. TP (True Positive): Buah *Layak* diprediksi benar sebagai *Layak*.
2. TN (True Negative): Buah *Tidak Layak* diprediksi benar sebagai *Tidak Layak*.
3. FP (False Positive): Buah *Tidak Layak* lolos dan salah diprediksi sebagai *Layak*. Kesalahan Tipe I (kegagalan kritis rantai pasok).
4. FN (False Negative): Buah *Layak* salah diprediksi sebagai *Tidak Layak*. Kesalahan Tipe II (pembuangan produk layak).

Dari komponen dasar ini, empat metrik diturunkan:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3.2)$$

$$Precision = \frac{(TP)}{(TP+FP)} \quad (3.3)$$

$$Recall = \frac{(TP)}{(TP+FN)} \quad (3.4)$$

$$F_1 = 2 \times \frac{(Precision \times Recall)}{(Precision+Recall)} \quad (3.5)$$

Justifikasi Prioritas Recall Kelas Tidak Layak (serta minimasi False Positive) dalam konteks penelitian ini bersifat operasional: rendahnya sensitivitas deteksi pada kelas Tidak Layak berarti buah busuk yang sesungguhnya mengandung antraknosa aktif atau larva lalat buah diloloskan oleh sistem sebagai buah Layak. Konsekuensinya: (1) kerugian finansial masif akibat penolakan kontainer oleh otoritas fitosanitasi di pasar ekspor premium; (2) kerusakan reputasi jangka panjang eksportir; (3) risiko penyebaran hama karantina *Bactrocera dorsalis* yang berimplikasi diplomatik serius. Oleh karena itu, memaksimalkan Recall adalah prioritas operasional utama dalam sistem ini.

3.6.2 Evaluasi Efisiensi Komputasi

1. Training Time: Total waktu dari inisialisasi hingga konvergensi, diukur dalam detik.
2. Inference Time: Waktu rata-rata per citra 224×224 piksel, diukur dalam milidetik (ms). Metrik ini mencerminkan kelayakan implementasi real-time.

3.7 Instrumen Penelitian

3.7.1 Perangkat Keras (*Hardware*)

Tabel 3.5 Spesifikasi Perangkat Keras (*Hardware*)

Komponen	Spesifikasi
Platform Komputasi	Google Colab Pro (cloud-based)
Akselerator GPU	NVIDIA L4 Tensor Core
VRAM GPU	24 GB GDDR6 (22,5 GB usable)
RAM Sistem	53,0 GB

Kapasitas Disk	112,6 GB
Penyimpanan Dataset	Google Drive (terintegrasi)

3.7.2 Perangkat Lunak (Software)

Tabel 3.6 Spesifikasi Perangkat Lunak(Software)

Komponen	Keterangan
Bahasa Pemrograman	Python 3.10+
Framework Deep Learning	TensorFlow & Keras API (versi terbaru)
Metrik Evaluasi	Scikit-Learn
Visualisasi	Matplotlib & Seaborn
Manajemen Dataset	Pandas & NumPy
Lingkungan Eksekusi	Jupyter Notebook (Google Colab)