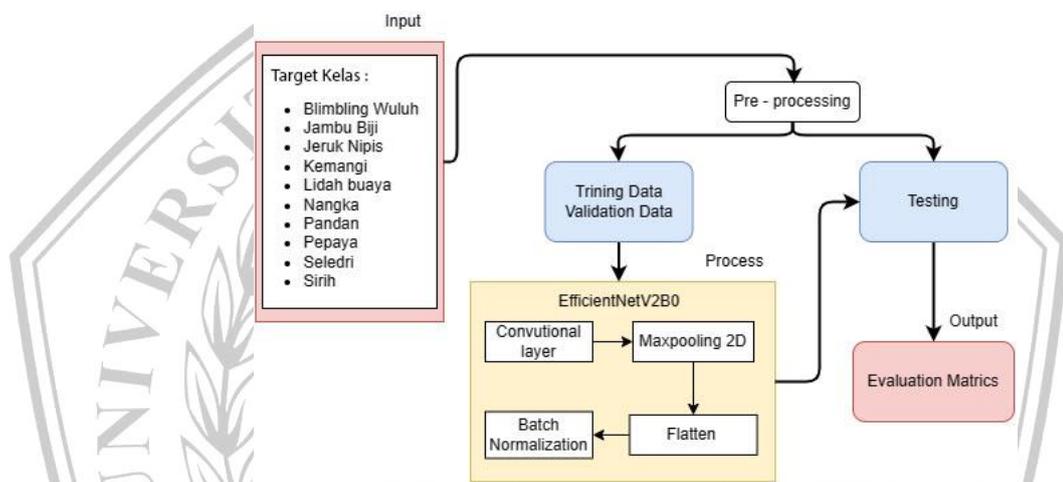


BAB III METODOLOGI PENELITIAN

3.1. Rancangan Penelitian

Data yang digunakan dalam penelitian ini merupakan dataset citra *Indonesian Herbal Leaf* yang tersedia pada Kaggle, Github, dan Mendelay. Terdapat 4 tahap yang dilakukan, yaitu pengumpulan dataset, pre-processing, pemodelan deep learning hingga evaluasi. Berikut merupakan alur penelitian seperti yang disajikan pada Gambar



Gambar 3. 1 Rancangan Penelitian

3.2. Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset yang berjudul *Indonesian Herbal Leaf* yang tersedia pada situ web Kaggle, Github, serta mendelay secara gratis. Dataset ini memiliki 10 kategori terdiri dari sejumlah gambar daun dari beberapa tanaman herbal di Indonesia, seperti; blimbing wuluh, jambu biji, jeruk nipis, kemangi, lidah buaya, nangka, pandan, pepaya, seledri, dan sirih. Resolusi dari setiap gambar nya 1600x1200 piksel. Yang terdiri dari 3500 gambar dengan pembagian 350 gambar daun disetiap kategorinya. Data yang digunakan untuk data latih sebesar 2800, untuk data uji sebesar 350 dan 350 sebagai data validasi. Objek ini telah menjadi salah satu dataset standar dalam komunitas pembelajaran mesin dan digunakan untuk menguji dan mengembangkan

model-model pembelajaran mesin yang efektif dalam mengklasifikasikan.

3.3. Preprocessing Data

Pada tahap ini, data yang berdimensi 1600x1200 piksel dilakukan proses perubahan ukuran gambar untuk memperkecil dimensi data menjadi berukuran 224x224 piksel tanpa menghilangkan berbagai fitur pada gambar. Tujuan lain dari perubahan ukuran gambar adalah untuk mempercepat waktu komputasi selama pelatihan. Ukuran 224x224 piksel dipilih karena sesuai dengan bentuk masukan arsitektur EfficientNet [20]. Setelah pra-pemrosesan, data dipisahkan menjadi set pengujian, validasi, dan pelatihan. Dataset ini dipisahkan menjadi tiga kategori: 80% data adalah data pelatihan, 10% adalah data validasi, dan 10% adalah data pengujian. Jumlah total data pada dataset ini adalah 3500 data. Validasi data digunakan untuk mengkonfirmasi melalui pengujian berdasarkan dataset yang telah dipelajari pada tahap pelatihan. Detail preprocessing data dapat dilihat pada table 3.1

Tabel 3.1 Detail Preprocessing

Data	Deskripsi	Total
Training	Seed = 123 Image_size = 224 x 224 subset = "Training"	2800
Testing	Seed = 123 Image_size = 224 x 224 subset = "Testing"	350
Validation	Seed = 123 Image_size = 224 x 224 subset = "Validation"	350

3.4. Model EfficientNetV2B0

CNN merupakan jaringan yang mempunyai susunan neuron $M \times N \times C$ 3 dimensi (lebar M, tinggi N, kanal C) yang dirancang untuk mengatasi masalah overfitting. Overfitting terjadi ketika sistem kehilangan kemampuan mengenali pola umum dan hanya mampu mengenali data pelatihan. Hal ini sering kali disebabkan oleh arsitektur jaringan yang terlalu kompleks atau terlalu banyak iterasi dalam sistem.

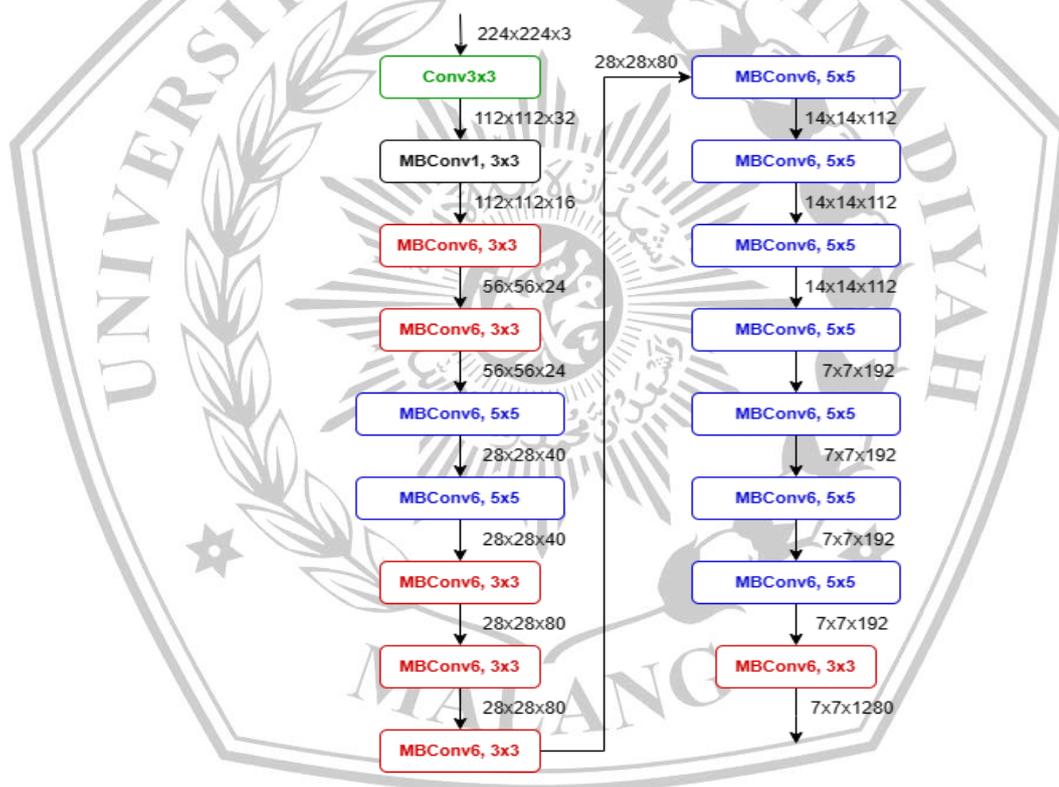
Arsitektur CNN lainnya seringkali memerlukan penggunaan parameter dalam jumlah besar dan juga disertai dengan dataset yang cukup besar. Hal ini memerlukan sumber daya komputasi yang cukup besar dan waktu pelatihan yang lama. Diperlukan eksperimen lebih lanjut dengan menggunakan model CNN terbaru seperti yang dilakukan pada penelitian menggunakan EfficientNetV2B0 yang memperbarui arsitektur sebelumnya, EfficientNet. Model jaringan saraf konvolusional (CNN) baru, EfficientNetV2 mengungguli pendahulunya dalam hal efisiensi parameter dan kecepatan pelatihan. Penelitian ini didasarkan pada penelitian sebelumnya oleh dan bertujuan untuk menggunakan EfficientNetV2B0 untuk mengkategorikan tahapan perkembangan buah Ambarella.

EfficientNetV2B0 adalah jenis arsitektur CNN yang menggunakan teknik penskalaan gabungan untuk memungkinkan kinerja yang lebih baik. Melalui pengurangan jumlah parameter dan Operasi Floating point Per Second (FLOPs), EfficientNetV2B0 berupaya meningkatkan efisiensi komputasi sekaligus meningkatkan kinerja. Tabel 1 mengilustrasikan bagaimana penelitian kontemporer, meskipun mengklaim keuntungan signifikan dalam kinerja pelatihan atau inferensi, sering kali mengungguli EfficientNet dalam hal parameter FLOP dan efisiensi.

Tabel 3. 2 Perbandingan Kinerja Arsitektur CNN

	Ketepatan(%)	Parameter (M)	FLOP(B)
EfisienNetV2B0	84.6	43	19
ResNet-RS-420	84.4	192	64
NFNet-F1	84.7	193	36

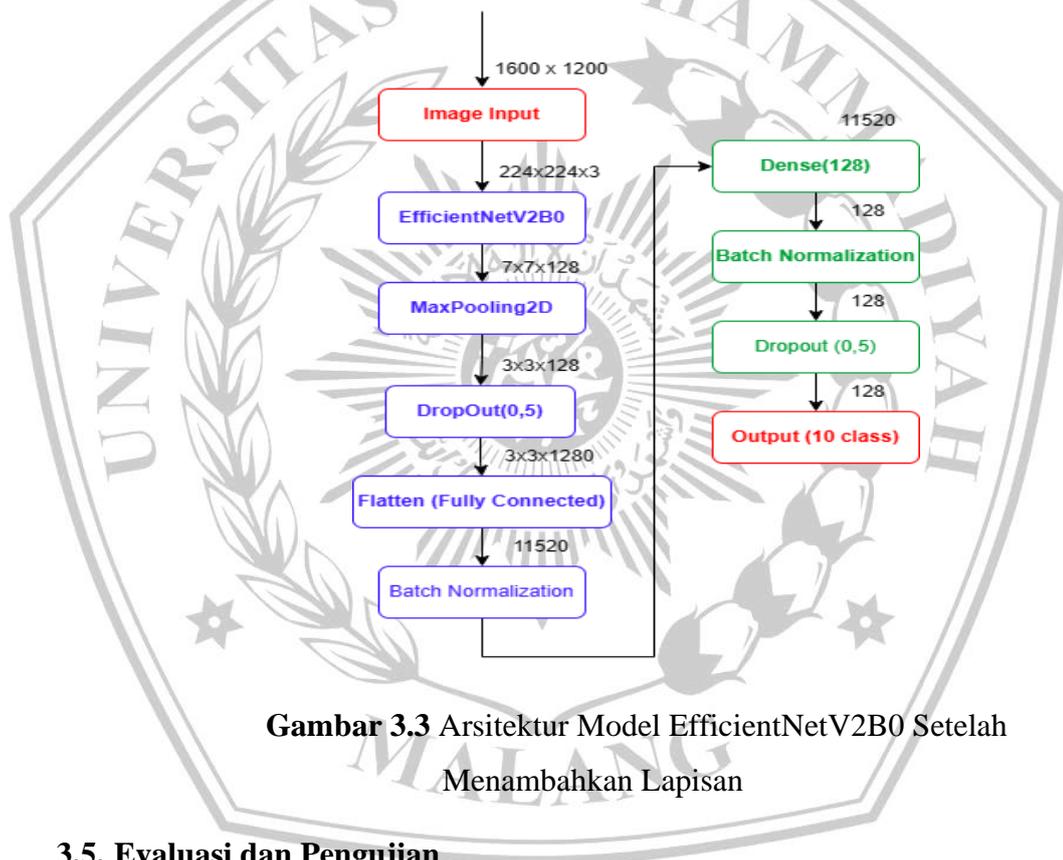
Model EfficientNetV2B0 terbukti mampu mengungguli model lainnya hanya dengan menggunakan parameter yang lebih sedikit [29], Jumlah parameter yang sedikit akan mempercepat proses klasifikasi dan mempersingkat waktu pelatihan model secara keseluruhan. Desain model EfficientNetV2B0 dapat dilihat pada Gambar



Gambar 3.2 Arsitektur model EfficientNetV2B0

Pada model EfficientNetV2B0 akan ditambahkan output layer untuk menyesuaikan jumlah kelas target pada dataset yang digunakan. Sebelum lapisan keluaran juga ditambahkan 2 lapisan untuk mengurangi overfitting sehingga kita bisa mendapatkan hasil yang lebih baik. Serta

menambahkan MaxPooling2D, Flatten, dan Batch Normalization sebagai layer tambahan pada model yang dibuat. Fungsi MaxPooling2D adalah untuk mereduksi dimensi spasial dari representasi fitur yang dihasilkan oleh lapisan sebelumnya [30]. Kemudian Flatten berfungsi untuk mengubah representasi fitur multidimensi menjadi vektor satu dimensi sehingga lapisan-lapisannya menjadi terhubung sepenuhnya [31]. Normalisasi Batch membantu dalam mempercepat pelatihan dan memiliki fungsi yang mirip dengan Regularisasi L2 dalam menangani overfitting [32]. Model keseluruhan dapat dilihat pada Gambar 4.



Gambar 3.3 Arsitektur Model EfficientNetV2B0 Setelah Menambahkan Lapisan

3.5. Evaluasi dan Pengujian

Persamaan (1) menggambarkan nilai presisi; persamaan (2) nilai recall; persamaan (3) nilai akurasi; dan persamaan (4) nilai F1-Score. Confusion Matrix adalah alat analisis prediktif yang menunjukkan dan membandingkan nilai aktual dengan nilai model prediksi. Hal ini dapat digunakan untuk menghasilkan evaluasimetrik seperti ini. Tabel matriks

konfusi menghasilkan empat nilai: False Positive (FP), True Positive (TP), False Negative (FN), dan True Negative (TN).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (3)$$

$$\text{F1 - Score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \quad (4)$$

3.5.1. classification report

Classification report merupakan salah satu metrik evaluasi yang sering digunakan untuk mengukur kinerja model klasifikasi. Classification report memberikan gambaran yang lebih rinci dan komprehensif mengenai performa model klasifikasi dengan menguraikan beberapa parameter penting seperti precision, recall, f1-score, dan support untuk setiap kelas. Precision mengukur seberapa banyak prediksi positif yang benar-benar positif, sedangkan recall mengukur seberapa banyak instance positif yang berhasil diidentifikasi dengan benar oleh model. F1-score merupakan rata-rata harmonis antara precision dan recall, yang memberikan gambaran keseimbangan antara kedua metrik tersebut. Support menunjukkan jumlah instance dari masing-masing kelas dalam data. Selain itu, classification report juga menampilkan nilai rata-rata presisi, rata-rata recall, rata-rata f1-score, dan jumlah total support untuk memberikan gambaran keseluruhan kinerja model.

3.5.2. grafik akurasi model

Grafik akurasi model adalah representasi visual yang digunakan untuk menggambarkan kinerja atau performa suatu model machine learning dalam melakukan tugas klasifikasi atau prediksi. Grafik ini menampilkan hubungan antara akurasi prediksi model dengan beberapa parameter atau faktor yang mempengaruhi akurasi tersebut.

grafik akurasi model menampilkan akurasi (dalam bentuk persentase atau nilai desimal) pada sumbu vertikal, sementara sumbu horizontal menggambarkan parameter atau faktor yang diuji pengaruhnya terhadap akurasi. Parameter ini dapat berupa jumlah data latih, nilai hyperparameter, jenis algoritma yang digunakan, atau variabel lain yang relevan dengan model yang dibangun. grafik akurasi model juga dapat digunakan untuk membandingkan kinerja beberapa model atau algoritma yang berbeda pada parameter yang sama.

3.5.3. grafik loss model

Grafik loss model adalah sebuah representasi visual yang digunakan untuk memantau dan menganalisis proses pelatihan (training) model machine learning, terutama dalam konteks deep learning. Grafik ini menggambarkan nilai fungsi loss atau error yang terjadi selama proses pelatihan model terhadap jumlah iterasi atau epoch yang dilakukan. Pada sumbu vertikal, grafik loss model menampilkan nilai loss yang merupakan ukuran seberapa jauh prediksi model menyimpang dari target atau nilai sebenarnya. Semakin kecil nilai loss, semakin baik kinerja model dalam memprediksi target dengan akurat. Sumbu horizontal menampilkan jumlah iterasi atau epoch yang telah dilalui selama proses pelatihan. Grafik loss model memberikan informasi penting tentang bagaimana model belajar dan menyesuaikan parameter-parameternya selama pelatihan. Pada awal pelatihan, nilai loss biasanya tinggi karena model belum terlatih dengan baik. Namun, dalam beberapa kasus, grafik loss dapat menunjukkan pola yang tidak diinginkan, seperti overfitting (nilai loss terus menurun pada data latih tetapi meningkat pada data validasi) atau underfitting (nilai loss tidak menurun secara signifikan baik pada data latih maupun validasi).

3.5.4. confusion matrix

Confusion matrix adalah sebuah tabel yang digunakan untuk

mengevaluasi kinerja model klasifikasi dalam memprediksi kelas-kelas yang berbeda. Confusion matrix terdiri dari baris yang mewakili kelas sebenarnya (ground truth) dan kolom yang mewakili kelas yang diprediksi oleh model. Setiap sel dalam matriks menunjukkan jumlah instance yang termasuk dalam kombinasi kelas sebenarnya dan kelas prediksi tertentu. Dengan demikian, confusion matrix memberikan gambaran jelas tentang jenis kesalahan yang dilakukan oleh model, apakah False Positive (memprediksi positif saat seharusnya negatif) atau False Negative (memprediksi negatif saat seharusnya positif). Informasi dalam confusion matrix dapat digunakan untuk menghitung berbagai metrik evaluasi kinerja model, seperti akurasi, presisi, recall, F1-score, dan lainnya. Confusion matrix sangat berguna dalam kasus klasifikasi multi-kelas, di mana terdapat lebih dari dua kelas yang harus diprediksi.

