

Klasifikasi Penyakit Daun Cabai Menggunakan Model *MobileNet*

Liza Puspita Putri^{1*}, Muhammad Rafi Muttaqin², Yudhi Raymond Ramadhan³

^{1,2,3}Teknik Informatika/STT Wastukencana.

¹Email: lizapsptp@gmail.com

²Email: rafi@wastukencana.ac.id

³Email: yudhi.raymond@wastukencana.ac.id

*)Corresponding Author

ABSTRACT

Chilli is the most popular and strategic commodity to reduce inflation in Indonesia. Chili consumption in Indonesia has continued to increase from 2019 to 2023 but chili production in Indonesia has decreased from 2021 to 2023 as well as the chili harvest area which has decreased from 2020 - 2023 which has resulted in obstacles in meeting the needs of chili consumption in Indonesia. The problem that occurs is due to chilli farmers who experience crop failure due to disease in the chilli plant. With these problems, there needs to be a solution to overcome these problems, therefore an application is made that can classify diseases on chili leaves so that chili farmers or the general public who want to plant chilies can detect diseases on chili leaves early. The model made using the CRISP-DM method by applying the concept of supervised learning with the Convolutional Neural Network (CNN) algorithm and MobileNet architecture. From the results of the research that has been done, the model results have good performance and do not show signs of severe overfitting with the results of the training accuracy model evaluation of 94.18%, validation accuracy of 83.66%, training loss of 16% and validation loss of 45%. The confusion matrix results have an accuracy of 84.84% of the total testing data of 132 data.

Keywords: CNN, CRISP-DM, MobileNet, Supervised Learning.

ABSTRAK

Cabai merupakan komoditas yang paling diminati dan strategis untuk mengurangi inflasi di Indonesia. Konsumsi cabai di Indonesia terus meningkat sejak tahun 2019 sampai dengan 2023 akan tetapi produksi cabai di Indonesia mengalami penurunan ditahun 2021 sampai dengan 2023 begitu pun dengan luas panen cabai yang menurun di tahun 2020 – 2023 yang mengakibatkan kendala dalam memenuhi kebutuhan konsumsi cabai di Indonesia. Adapun permasalahan yang terjadi adalah dikarenakan petani cabai yang mengalami gagal panen dikarenakan penyakit pada tumbuhan cabai tersebut. Dengan adanya permasalahan tersebut perlu adanya solusi untuk mengatasi permasalahan tersebut, maka dari itu dibuatkannya aplikasi yang dapat mengklasifikasi penyakit pada daun cabai sehingga para petani cabai atau masyarakat umum yang ingin menanam cabai dapat mendeteksi dini penyakit pada daun cabai. Model dibuat menggunakan metode CRISP-DM dengan menerapkan konsep supervised learning dengan algoritma Convolutional Neural Network (CNN) dan arsitektur MobileNet. Dari hasil penelitian yang telah dilakukan, hasil model tersebut memiliki performa yang baik dan tidak menunjukkan tanda overfitting yang parah dengan hasil evaluasi model training accuracy sebesar 94.18%, validation accuracy sebesar 83.66%, training loss sebesar 16% dan validation loss sebesar 45%. Hasil confusion matrix memiliki akurasi 84.84% dari jumlah data testing sebanyak 132 data.

Keywords: CNN, CRISP-DM, MobileNet, Supervised Learning.

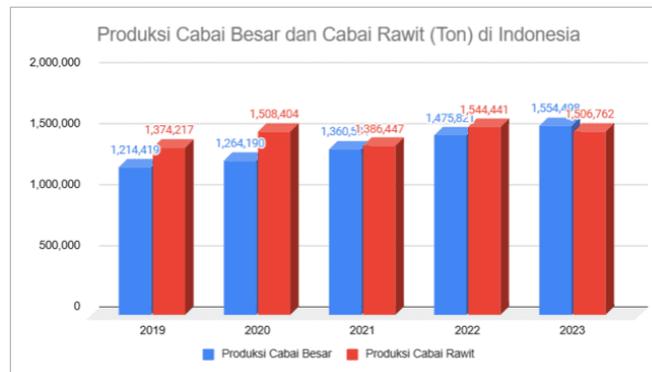
A. PENDAHULUAN

1. Latar Belakang

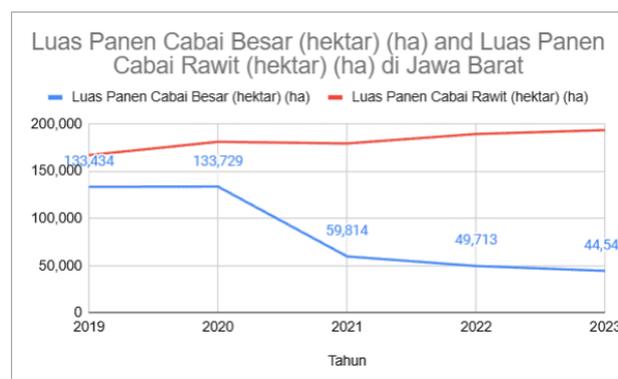
Menurut [1] pertanian adalah kegiatan manusia mengelola tanaman, hewan untuk memenuhi kebutuhan pangan manusia. Salah satu sektor pertanian adalah hortikultura yang meliputi tanaman buah-buahan, tanaman sayuran, tanaman hias, dan tanaman obat-obatan.

Menurut [2] Indonesia merupakan satu di antara negara agraris yang menghasilkan dan mengolah tanaman pangan, perkebunan, hortikultura, dan kehutanan. Sebagian besar dari produksi yang dihasilkan pada bidang-bidang tersebut memasok kebutuhan dalam negeri dengan nilai yang tinggi. Hal ini mengindikasikan bahwa profesi petani di Indonesia memiliki porsi yang cukup besar dalam upaya membantu pemerintah dalam memenuhi kebutuhan pangan penduduk. Satu di antara jenis tanaman yang memiliki tingkat permintaan yang cukup tinggi adalah tanaman cabai yang tergolong ke dalam tanaman hortikultura.

Menurut [3] cabai merupakan komoditas yang paling banyak dikonsumsi masyarakat yang berperan besar dalam pembentukan inflasi nasional, selain dari pada itu cabai juga memiliki dampak cukup besar terhadap kebutuhan masyarakat. Karena hal itu, cabai merupakan komoditas yang strategis di Indonesia. Dengan menjadi komoditas yang paling banyak dikonsumsi masyarakat, produksi cabai yang terus meningkat sangat diperlukan untuk memenuhi kebutuhan konsumen masyarakat di Indonesia.



Gambar 1. Produksi Cabai Besar dan Cabai Rawit di Indonesia



Gambar 2. Luas Panen Cabai di Indonesia

Berdasarkan data yang dipublikasikan oleh Kementerian Pertanian RI pada gambar 1 dari Badan Pusat Statistik dan Direktorat Jenderal Hortikultura, produksi cabai rawit dan cabai besar di Indonesia dalam 5 tahun terakhir mengalami kenaikan dari tahun 2019 sampai dengan 2023, akan tetapi terjadi penurunan hasil produksi cabai rawit di tahun 2021 dan 2023 (Badan Pusat Statistik, 2024).

Sementara pada gambar 2 untuk luas lahan produksi tanaman cabai besar mengalami penurunan yang cukup signifikan yang semula 133,729 ha ditahun 2020 menjadi 44,541 ha ditahun 2023. Lain halnya dengan cabai besar, luas lahan produksi cabai rawit mengalami kenaikan yang semula 166,943 ha ditahun 2019 menjadi 193,423 ha ditahun 2023.

Penurunan produksi cabai dapat disebabkan karena adanya gagal panen yang terjadi di kalangan petani. Adapun alasan terjadinya gagal panen yang terjadi disebabkan karena adanya hama yang menyerang daun cabai dan menyebabkan adanya penyakit *yellowish*, *leaf spot*, *leaf curl*, *powdery mildew* dan *whitefly*. Jika hal tersebut tidak segera ditangani, maka petani akan mengalami kerugian yang besar yang menyebabkan terjadinya penurunan produksi cabai dan kendala dalam pemenuhan konsumsi cabai di kalangan masyarakat.

Berdasarkan penjelasan dari permasalahan tersebut, perkembangan teknologi yang terus berlangsung dapat menjadi solusi untuk menanggulangi permasalahan gagal panen petani cabai. Salah satu cara untuk mengatasi masalah ini adalah dengan menerapkan teknologi di bidang pertanian. Pembuatan aplikasi berbasis *mobile* yang dapat mengklasifikasi penyakit pada daun cabai dapat menjadi solusi yang efektif untuk mengatasi permasalahan tersebut, dengan adanya aplikasi yang dapat mendeteksi penyakit pada daun cabai sehingga dapat membantu para petani baru atau masyarakat umum yang ingin menanam cabai di rumahnya dapat menggunakan aplikasi tersebut untuk mendeteksi dini penyakit pada daun cabai sehingga dapat mengurangi kerugian dari potensi gagal panen dikarenakan penyakit pada tanaman cabai.

2. Permasalahan

Berdasarkan latar belakang yang telah diuraikan, terdapat permasalahan dalam memenuhi kebutuhan konsumsi cabai di Indonesia akibat penurunan jumlah produksi dan berkurangnya luas lahan panen cabai. Dengan terbatasnya lahan panen, diperlukan upaya untuk mengoptimalkan lahan yang tersedia guna mengurangi risiko gagal panen akibat serangan penyakit, sehingga kebutuhan konsumsi cabai di Indonesia dapat terpenuhi.

Penelitian ini dilakukan dengan menggunakan algoritma *Convolutional Neural Network* (CNN) dan arsitektur *MobileNet*, yang kemudian diimplementasikan dalam bentuk aplikasi *mobile*. Aplikasi ini dirancang agar dapat digunakan oleh petani pemula maupun masyarakat umum yang ingin menanam cabai, dengan harapan dapat meningkatkan produksi cabai di Indonesia sehingga mampu memenuhi kebutuhan konsumsi cabai nasional.

3. Tinjauan Pustaka

a. Machine Learning

Machine learning (ML) adalah salah satu cabang ilmu komputer yang bertujuan untuk mengembangkan algoritma yang memungkinkan komputer untuk belajar dari data yang ada. Dengan kata lain, komputer diajarkan untuk belajar dari informasi dari data historis sehingga dapat meningkatkan kinerja dalam menganalisis dan mengolah informasi yang diberikan. Hal ini membuat *machine learning* sering disebut sebagai proses “belajar dari data”, di mana komputer memperoleh pengetahuan atau keterampilan baru melalui pengalaman yang diperoleh dari data yang ada. Tujuan utamanya untuk menciptakan model atau program yang dapat mengoptimalkan performa dalam memahami dan menginterpretasikan data yang diberikan [4].

b. Deep Learning

Deep learning adalah bentuk khusus dari pembelajaran mesin yang menggunakan beberapa lapisan pemrosesan untuk mempelajari konsep kompleks dengan membangun hierarki abstraksi, di mana setiap lapisan belajar dari abstraksi yang lebih sederhana dari lapisan sebelumnya. Model *deep learning* mengubah *input* menjadi *output* melalui serangkaian transformasi linier dan non-linier. Proses ini dilakukan dengan banyak lapisan unit komputasi yang mengekstraksi fitur dari data mentah pada berbagai tingkat abstraksi, sehingga model dapat belajar pola langsung dari data tanpa perlu rekayasa fitur manual.

Berbeda dengan pembelajaran mesin tradisional, *deep learning* tidak memerlukan rekayasa fitur manual karena mampu mempelajari fitur secara hierarkis. Ini membuat *deep learning* efektif untuk menangani data tidak terstruktur, seperti gambar, *video*, *audio*, teks, dan data kesehatan. Proses pembelajaran model *deep learning* biasanya dimulai secara acak, kemudian menggunakan optimasi berbasis gradien untuk menyempurnakan parameter model secara iteratif, meskipun hasil akhirnya tidak selalu mencapai optimal

global. Untuk mencapai hasil yang baik, *deep learning* membutuhkan banyak data dan sumber daya komputasi [5]

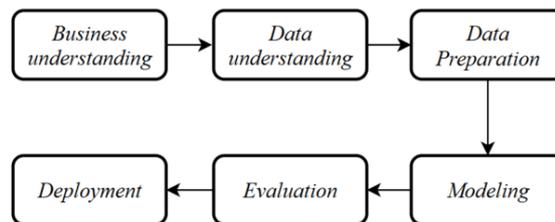
c. *Convolutional Neural Network* (CNN)

Convolutional Neural Networks adalah bagian dari kecerdasan buatan. Jaringan ini melakukan tugas yang disebut Jaringan Syaraf, di mana kita dapat melakukan operasi matematika. Dalam model ini, perkalian matriks dilakukan. Ini terdiri dari lapisan *input*, lapisan tersembunyi, dan lapisan *output*. CNN ini digunakan dalam reorganisasi gambar dalam pemrosesan gambar dengan membuat gambar dipartisi ke dalam sejumlah piksel. CNN memiliki nama singkat yang disebut *ConvNets* [6][7].

d. *MobileNet*

MobileNet adalah model yang dikembangkan oleh Andrew G. Howard dan rekan-rekannya. *MobileNet* dapat digunakan untuk deteksi atau klasifikasi objek. *MobileNet* dikembangkan dengan menggunakan arsitektur *depthwise separable convolution*. *Depthwise separable convolution* dibagi menjadi dua, yaitu *depthwise convolution* dan *1x1 pointwise convolution* [8].

B. METODE



Gambar 3. Tahapan Metode CRISP-DM

Perancangan model *deep learning* pada penelitian ini menggunakan metode *Cross Industry Standard Process for Data Mining* (CRISP-DM) [9]. CRISP-DM terdiri dari 6 tahapan. Tahapan awal adalah *business understanding* berfokus pada pemahaman tujuan penelitian berdasarkan perspektif bisnis yaitu mengklasifikasikan kelompok produk. Pada tahap *data understanding* dimulai dengan pengumpulan data awal kemudian melakukan identifikasi dan eksplorasi data. Tahap ketiga, *data preparation* mencakup aktivitas untuk menyusun data mentah agar dapat digunakan dalam pemodelan dengan melakukan pemilahan, pembersihan, penyusunan, dan pengintegrasian data. Tahap *modeling*, berbagai teknik atau algoritma diterapkan berdasarkan jenis data yang dipilih. Pemodelan meliputi pemilihan teknik, pembuatan, dan penilaian model. Selanjutnya tahap *evaluation*, dalam tahap evaluasi, hasilnya diperiksa sesuai dengan *business understanding* yang telah ditetapkan. Pada tahap ini akan evaluasi dengan melihat *training graph* dan menghitung hasil *confusion matrix*. Keenam, *deployment*. Pada fase *deployment* hasilnya bisa berupa laporan akhir ataupun berbentuk *software* (perangkat lunak / aplikasi) [1].

C. HASIL DAN PEMBAHASAN

Hasil dan pembahasan pada penelitian ini menggunakan metode *Cross Industry Standard Process for Data Mining* (CRISP-DM). CRISP-DM adalah metode yang pas untuk penelitian ini karena, memiliki langkah yang terstruktur yang dimulai dari pemahaman masalah yang ada di dalam tahapan *business understanding*, lalu dilanjutkan ke tahapan pemahaman *dataset* yang dipakai pada tahapan *data understanding*, lalu persiapan data dari *dataset* yang digunakan pada tahapan *data preparation* dan dilanjutkan ke proses pembuatan model pada tahapan *modeling* lalu dievaluasi di tahapan *evaluation* kemudian dilanjutkan ke dalam tahapan implementasi aplikasi pada tahapan *deployment* [10]. Berikut uraian hasil dan pembahasan pada penelitian ini:

1. *Business Understanding*

Penelitian ini dibuat bertujuan untuk membuat aplikasi *mobile* yang dapat melakukan klasifikasi penyakit pada daun cabai dengan metode supervised learning yang menggunakan

algoritma *Convolutional Neural Network* dan arsitektur *MobileNet*. Aplikasi melakukan pengenalan penyakit pada daun cabai dengan cara mengambil gambar dengan menggunakan kamera *handphone* yang kemudian akan menghasilkan *output* jenis penyakit pada daun cabai berdasarkan gambar yang telah diambil.

2. Data Understanding

Data yang digunakan berupa gambar penyakit pada daun cabai yang didapatkan dari *website roboflow*. Gambar dari *dataset* ini memiliki ukuran piksel yang serupa yaitu (640, 640), *dataset* ini memiliki 6 buah *class/label*. Tabel 1 adalah penjelasan dari *class* yang dipakai.

Tabel 1. *Class Dataset*

Class	Definisi	Sumber
<i>Healthy</i>	Gambar daun cabai yang sehat	https://universe.roboflow.com/chili-leaves-disease-classification/chili-futhd/dataset/1
<i>Whitefly</i>	Gambar daun cabai yang terkena penyakit whitefly atau kutu kebul	
<i>Yellowish</i>	Gambar daun cabai yang terkena penyakit yellowish atau penyakit kuning	
<i>Leaf spot</i>	Gambar daun cabai yang terkena penyakit leaf spot atau daun berbintik	
<i>Leaf curl</i>	Gambar daun cabai yang terkena penyakit leaf curl atau daun keriting	
<i>Powdery mildew</i>	Gambar daun cabai yang terkena penyakit powdery mildew atau penyakit embun tepung	

3. Data Preparation

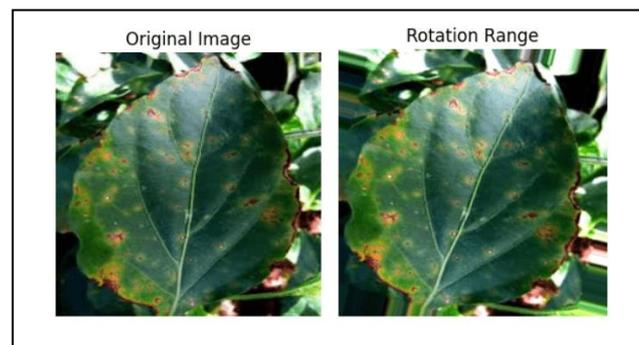
Sebelum dilakukan *training* model *dataset*, perlu untuk dilakukannya pemisahan *dataset* menjadi *train*, *val* dan *test*, adapun komposisi dari pembagian *dataset* tersebut adalah *train* 70%, *val* 20% dan *test* 10%. Lalu dilanjutkan ketahapan penggunaan *data augmentation* dengan menggunakan *library ImageDataGenerator* dari *tensorflow* guna meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru. Teknik augmentasi data seperti *Rescale*, *rotation_range*, *width_shift_range*, *height_shift_range*, *shear_range*, *zoom_range*, *horizontal_flip*, *fill_mode*. Berikut *data augmentation* yang diterapkan dalam penelitian ini:

a. *Rescale*

Pada penelitian ini gambar akan dilakukan *rescale* sehingga gambar memiliki ukuran 255 piksel.

b. *rotation_range*

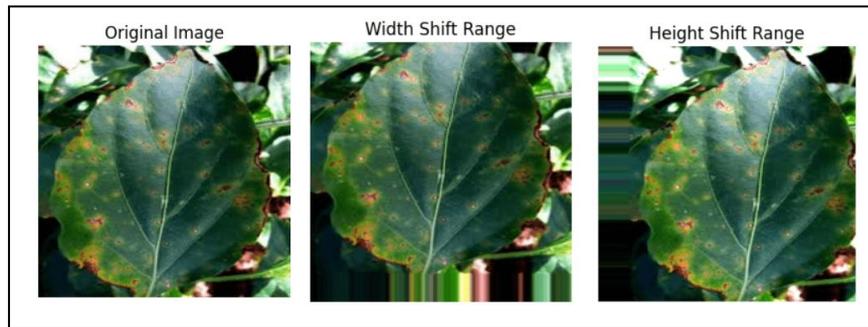
Pada penelitian ini dilakukan rotasi sebesar 20 derajat. Gambar 4 menampilkan hasil dari penggunaan *rotation range*.



Gambar 4. Gambar Hasil Rotasi

c. *Width dan Height Shift Range*

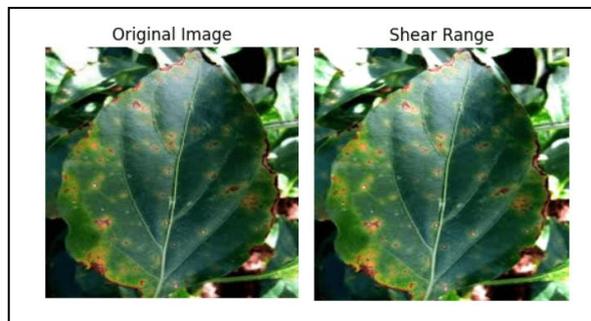
Menggeser gambar secara horizontal (*width*) dan vertikal (*height*) hingga 20% dari lebar dan tinggi gambar. Gambar 5 menampilkan hasil dari *width* dan *height range*.



Gambar 5. Gambar Hasil *Width* dan *Height Shift Range*

d. *shear_range*

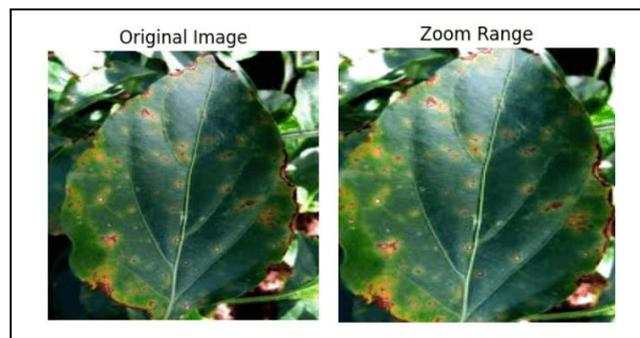
Menerapkan transformasi *shear* (geser miring) hingga 20 derajat. Pada gambar 6 menampilkan hasil dari penggunaan *shear range*.



Gambar 6. Gambar Hasil *Shear Range*

e. *zoom_range*

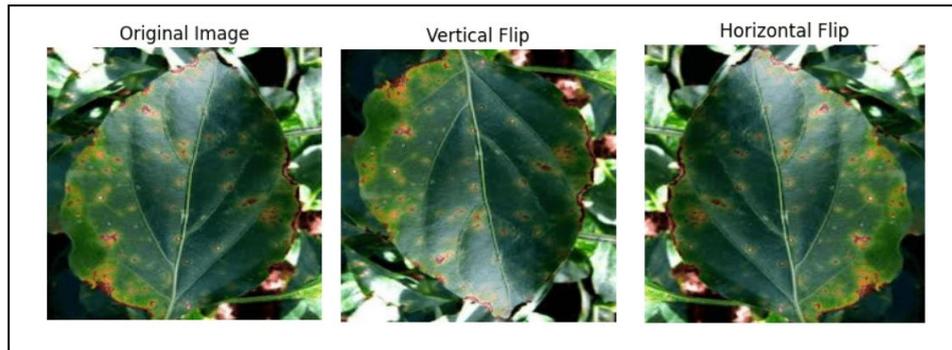
Pada penelitian ini dilakukan *zoom* sebesar 20%, pada gambar 7 menampilkan hasil dari penggunaan *zoom 20%*



Gambar 7. Gambar Hasil *Zoom Range*

f. *Vertical dan Horizontal Flip*

Berikut ini hasil gambar yang telah dilakukan *vertical* dan *horizontal flip* pada gambar 8 menampilkan hasil dari penggunaan *vertical* dan *horizontal flip*.

Gambar 8. Gambar Hasil *Vertical* dan *Horizontal Flip*g. *fill_mode*

Pada penelitian ini digunakan *fill mode nearest* di mana nilai piksel yang kosong diisi oleh nilai piksel terdekatnya.

Setelah menerapkan augmentasi data, selanjutnya akan dilakukan *split dataset* untuk memisahkan antara *train*, *val* dan *test*. Pada penelitian ini sebanyak 70% data akan dijadikan data *training*, 20% untuk data validasi dan 10% untuk data *testing*. Berikut ini tabel 2 menyatakan hasil *split data* perkelasnya serta total citra untuk setiap *directory train*, *val* dan *test*.

Tabel 2. Pembagian Dataset

<i>Sample</i>	<i>Data Training</i> 70%	<i>Data Validation</i> 20%	<i>Data Testing</i> 10%	<i>Total</i> 100%
<i>Healthy</i>	150	43	22	215
<i>Leaf curl</i>	155	44	23	222
<i>Leaf spot</i>	166	48	24	238
<i>Whitefly</i>	124	36	18	178
<i>Yellowish</i>	138	40	20	198
<i>Powdery Mildew</i>	163	47	25	235
Total	896	258	132	1.286

4. *Modeling*

Modeling terbagi menjadi dua bagian yaitu, proses *modeling* yang merupakan tahapan dalam pembuatan model dan penentuan parameter *modeling* yang merupakan tahap menentukan parameter *epochs* dan akan membandingkan parameter mana yang lebih baik performanya. Tahapan yang pertama adalah proses membuat *base model* yang menggunakan arsitektur *MobileNet* dan menambahkan beberapa parameter pada *base model*. Tabel 3 adalah uraian penjelasan parameter yang digunakan di dalam *base model*:

Tabel 3. Penjelasan Parameter Base Model

Parameter	Penjelasan
<i>Weights = 'imagenet'</i>	Berfungsi untuk bobot dari masing-masing <i>layer</i> yang sudah di <i>training</i> berdasarkan bobot yang ada pada <i>library imagenet</i> , sehingga proses <i>training</i> akan lebih ringan.
<i>Include_top = False</i>	Berfungsi untuk tidak perlu membawa yang merupakan bagian <i>prediction node</i> dari <i>mobilenet</i> yang memiliki nilai 1000 kelas, sedangkan penelitian ini tidak mencapai 1000 kelas
<i>input_shape = 224,224,3</i>	Ukuran piksel 224x224 dengan 3 <i>channel</i> warna RGB (<i>Red, Green, Blue</i>).

Selanjutnya adalah menambahkan lapisan tambahan di atas *MobileNet*. Pada penelitian ini menambahkan *GlobalAveragePooling2D* layer, *Dense* layer dan *Dropout* layer. Tabel 4 adalah penjelasan terkait parameter pada lapisan tambahan.

Tabel 4. Parameter

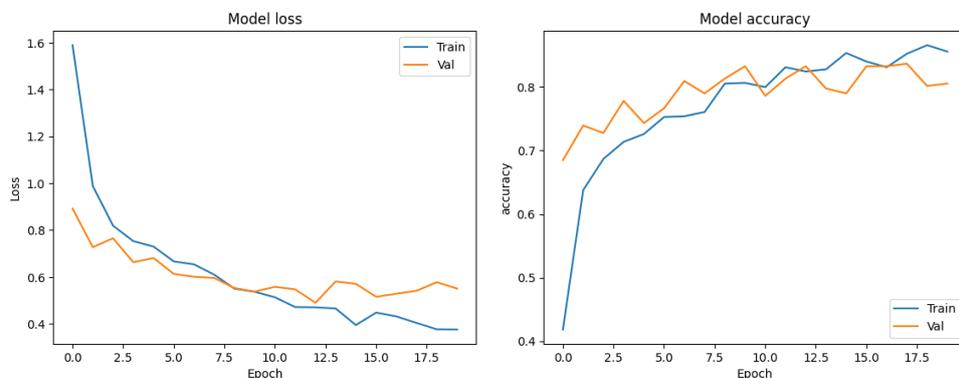
Parameter	Penjelasan
Base_model	Variabel <i>base model</i> yang berisi arsitektur <i>MobileNet</i>
<i>GlobalAveragePooling2D</i>	Untuk mereduksi dimensi fitur dari <i>output MobileNet</i> , menghasilkan vektor yang lebih ringkas.
<i>Dense</i>	Berfungsi untuk menambahkan <i>fully-connected layer</i> yang memiliki 6 kelas dengan 256 <i>neuron</i> dan fungsi aktivasi <i>ReLU</i> untuk memberikan kemampuan pembelajaran non-linear.
<i>Dropout</i>	Dengan jumlah <i>neuron</i> sesuai dengan jumlah kelas dalam data pelatihan (<i>train_data.num_classes</i>), menggunakan fungsi aktivasi <i>softmax</i> untuk menghasilkan probabilitas untuk setiap kelas.

Selanjutnya melakukan *compile model* menggunakan *compile model* menggunakan *adam optimizer*, untuk *loss function* menggunakan *categorical_crossentropy* karena memiliki lebih dari dua kelas dan *metrics* evaluasi yang dipilih adalah *accuracy*. Langkah selanjutnya adalah proses *training* dengan menggunakan *model fit*. Pada penelitian ini akan menggunakan *dataset train* dan *val* yang sebelumnya telah disimpan pada variabel *train_data* dan *val_data*, kemudian menentukan jumlah iterasi *training* atau *epochs*.

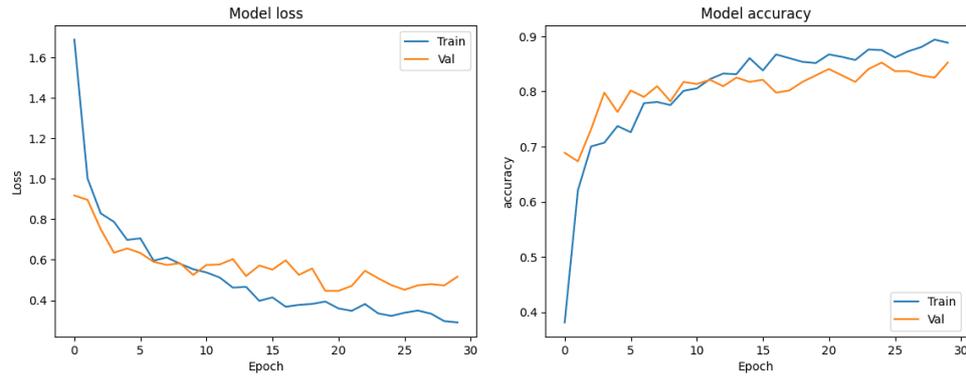
Tahapan menentukan parameter *modeling* pada saat melakukan *training* model berfungsi untuk menentukan seberapa baik model tersebut berdasarkan perbandingan parameter yang ditentukan sebelumnya. Parameter yang dimaksud adalah penentuan jumlah *epochs*. Tabel 4 adalah tabel hasil perbandingan *epoch* dalam proses *training* model:

Tabel 5. Perbandingan *Epochs* Proses *Training* Model

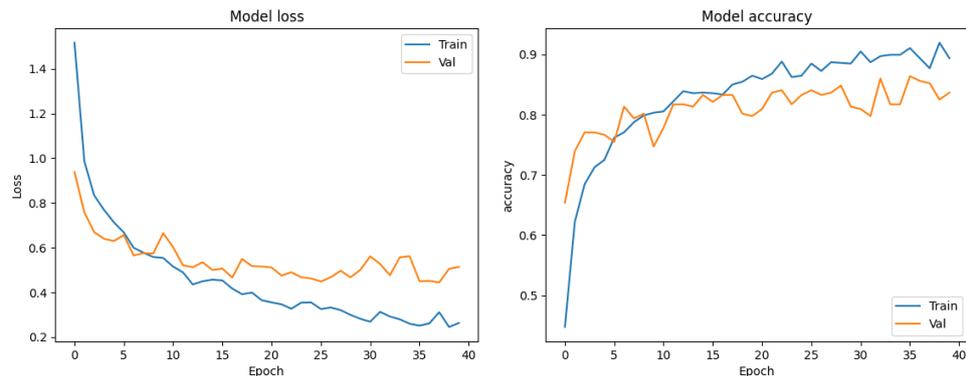
<i>Epoch</i>	<i>Train Acc</i>	<i>Train Loss</i>	<i>Val Acc</i>	<i>Val Loss</i>	<i>Test Acc</i>	<i>Avrg Acc</i>
20	90.49%	0.265	82.88%	0.548	83.33%	85.57%
30	93.40%	0.183	84.82%	0.462	83.33%	87.18%
40	94.18%	0.162	83.66%	0.448	84.85%	87.56%
50	94.52%	0.149	85.21%	0.482	83.33%	87.69%



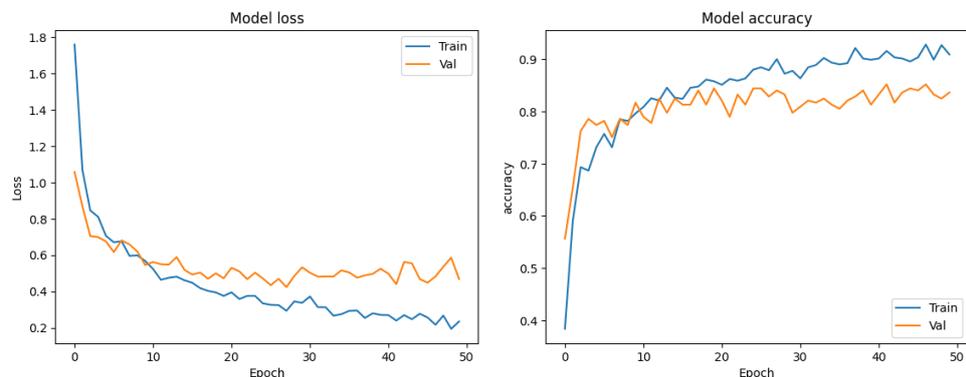
Gambar 9. Grafik *Loss* dan *Accuracy* 20 *Epochs*



Gambar 10. Grafik Loss dan Accuracy 30 Epochs



Gambar 11. Grafik Loss dan Accuracy 40 Epochs



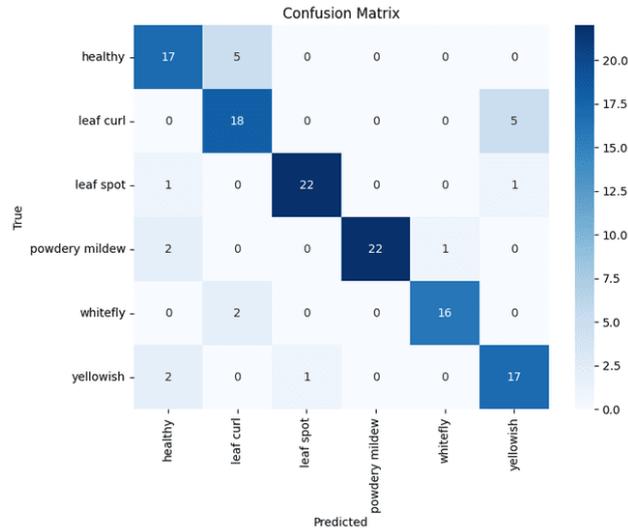
Gambar 12. Grafik Loss dan Accuracy 50 Epochs

Berdasarkan gambar 9, 10, 11 dan 12 menyatakan bahwa model dengan menggunakan 40 epoch memiliki hasil nilai *training accuracy* sebesar 94.18%, *validation accuracy* sebesar 83.66%, *training loss* sebesar 0.162, *validation loss* sebesar 0.448, *testing accuracy* sebesar 84.85%, rata-rata akurasi sebesar 87.56%. Selain dari pada itu, berdasarkan hasil analisis grafik *loss* untuk setiap parameter *epochs* nya, model dengan 40 epochs memiliki performa yang lebih baik jika dibandingkan dengan parameter *epochs* lainnya.

Model dengan 40 epochs memiliki penurunan *loss* yang baik dan fluktuasi yang lebih terkendali dibandingkan model lainnya. Model memiliki performa yang cukup baik, menunjukkan tidak ada *overfitting* atau *underfitting* yang signifikan. Model ini tetap menunjukkan generalisasi yang lebih baik di antara keempat model.

5. Evaluation

Setelah menentukan jumlah *epochs* yang akan digunakan yaitu 40 epochs maka dilanjutkan ke dalam tahapan evaluasi. Pada tahap evaluasi ini akan dilakukan evaluasi model dengan melihat hasil *confusion matrix* dan *classification report*.

Gambar 13. Hasil *Confusion Matrix*

Berdasarkan hasil *confusion matrix* pada gambar 13, maka hasil akurasi model penyakit pada daun cabai dari jumlah data *testing* sebanyak 132 data adalah 84.84%. Lalu dilanjutkan ke dalam tahapan evaluasi selanjutnya yaitu *classification report*.

	precision	recall	f1-score	support
healthy	0.95	0.82	0.88	22
leaf curl	0.88	0.61	0.72	23
leaf spot	0.96	0.92	0.94	24
powdery mildew	0.96	0.92	0.94	25
whitefly	0.89	0.94	0.92	18
yellowish	0.58	0.90	0.71	20
accuracy			0.85	132
macro avg	0.87	0.85	0.85	132
weighted avg	0.88	0.85	0.85	132

Gambar 14. Hasil *Classification Report*

Pada Gambar 14 menampilkan hasil dari penggunaan *classification report* yang mana menjelaskan bahwa:

1. Metrics per Kelas:

- a. *Precision*: Proporsi prediksi benar dari total prediksi untuk kelas tersebut.
 1. Nilai *precision* tertinggi: *leaf spot* (0.96), *powdery mildew* (0.96).
 2. *Precision* terendah: *yellowish* (0.58), menunjukkan banyak kesalahan prediksi pada kelas ini.
- b. *Recall*: Proporsi prediksi benar dari total jumlah data aktual untuk kelas tersebut.
 1. *Recall* tertinggi: *yellowish* (0.90), menunjukkan hampir semua sampel kelas ini dikenali.
 2. *Recall* terendah: *leaf curl* (0.61), menunjukkan banyak sampel dari kelas ini yang salah diklasifikasikan sebagai kelas lain.
- c. *F1-Score*: Rata-rata harmonis antara *precision* dan *recall*, memberikan keseimbangan antara keduanya.
 1. *F1-score* tertinggi: *leaf spot* (0.94) dan *powdery mildew* (0.94), menunjukkan prediksi untuk kelas ini sangat baik.

2. *F1-score* terendah: *leaf curl* (0.72) dan *yellowish* (0.71), menunjukkan performa model untuk kelas ini kurang memuaskan.
- d. *Support*: Jumlah sampel untuk masing-masing kelas dalam *dataset*. Jumlahnya seimbang, sehingga tidak ada kelas yang mendominasi.

2. Metric Global:

- a. *Accuracy*: Proporsi prediksi yang benar dari seluruh data (85%).
 1. Model ini memiliki akurasi keseluruhan yang baik, namun masih terdapat ruang untuk peningkatan.
- b. *Macro Avg*: Rata-rata *precision*, *recall*, dan *F1-score* dihitung tanpa mempertimbangkan jumlah sampel di tiap kelas.
 2. *Macro avg* menunjukkan performa rata-rata antar kelas adalah 85% untuk *precision*, *recall*, dan *F1-score*.
- c. *Weighted Avg*: Rata-rata *precision*, *recall*, dan *F1-score* dihitung dengan mempertimbangkan jumlah sampel di tiap kelas.
 3. Nilai *weighted avg* serupa dengan akurasi keseluruhan, yaitu 85%, menunjukkan model mampu menangani distribusi data kelas yang seimbang.

Kesimpulan:

1. Model ini memiliki performa yang baik secara umum, dengan akurasi keseluruhan 85%.
 2. Performa untuk kelas *leaf curl* dan *yellowish* memerlukan perbaikan karena memiliki nilai *F1-score* terendah.
 3. Model bekerja sangat baik untuk kelas seperti *leaf spot* dan *powdery mildew*, dengan *F1-score* hampir sempurna (0.94).
6. *Deployment*

Melakukan penerapan dan *deployment* model yang telah dibangun ke dalam aplikasi *mobile*, sehingga aplikasi dapat diunduh dan dapat digunakan oleh siapa pun yang menggunakan *smartphone* dengan sistem operasi *android*.

1. Halaman *onboarding*

Halaman *onboarding* adalah halaman pertama yang muncul ketika aplikasi baru dibuka sebelum masuk ke halaman utama aplikasi. Berikut gambar 15 adalah tampilan dari halaman *onboarding*.

2. Halaman utama

Halaman utama menampilkan semua jenis klasifikasi daun cabai dan terdapat tombol 3 tombol pada bagian *bottom navigation* yaitu media untuk unggah gambar dari galeri, kamera untuk mengambil gambar menggunakan kamera dan *profile* yang berisi identitas *developer* dan detail aplikasi. Berikut gambar 16 adalah tampilan dari halaman utama.

3. Halaman kamera

Halaman kamera berfungsi untuk mengambil citra daun cabai yang akan dideteksi dan diklasifikasi. Berikut gambar 17 adalah tampilan dari halaman kamera.

4. Halaman klasifikasi penyakit

Halaman klasifikasi adalah fitur utama dalam aplikasi ini. Halaman ini berfungsi untuk menerima *input*-an citra dari halaman kamera lalu diklasifikasi jenis penyakitnya dan ditampilkan *output* dari hasil klasifikasi dan terdapat tombol panah yang berfungsi untuk pindah ke halaman detail penyakit. Berikut gambar 18 adalah tampilan dari halaman klasifikasi penyakit.

5. Halaman detail penyakit

Halaman ini menampilkan detail penyakit yang sudah diklasifikasi dari halaman klasifikasi, berisi mengenai penyebab, gejala dan cara pengendalian dari penyakitnya. Berikut gambar 19 adalah tampilan dari halaman detail penyakit.

6. Halaman *profile*

Halaman ini menampilkan detail dari identitas *Android Developer* dan sedikit penjelasan tentang aplikasi yang sudah dibuat. Berikut gambar 20 adalah tampilan dari halaman *profile*.



Gambar 15. Halaman Onboarding



Gambar 16. Halaman Klasifikasi Penyakit



Gambar 17. Halaman Utama



Gambar 18. Halaman Detail Penyakit



Gambar 19. Halaman Kamera



Gambar 20. Halaman Profil

D. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa metode supervised learning dengan menggunakan algoritma *Convolutional Neural Network* (CNN) dan arsitektur *MobileNet* dapat diimplementasikan dengan baik dalam melakukan klasifikasi penyakit pada daun cabai. Hasil model tersebut memiliki performa yang baik dan tidak menunjukkan tanda *overfitting* dengan hasil evaluasi model *training accuracy* sebesar 94.18%, *validation accuracy* sebesar 83.66%, *training*

loss sebesar 16% dan *validation loss* sebesar 45%. Hasil *confusion matrix* memiliki akurasi 84.84% dari jumlah data *testing* sebanyak 132 data. Peneliti berhasil menerapkan model *deep learning* yang telah dibuat menjadi sebuah aplikasi berbasis *mobile* menggunakan bahasa pemrograman *kotlin*.

E. DAFTAR PUSTAKA

- [1] F. S. Anam, M. R. Muttaqin, dan Y. R. Ramadhan, “Klasifikasi Penyakit Pada Daun dan Buah Jambu Menggunakan Convolutional Neural Network,” *Journal of Information Technology and Computer Science*, vol. 7, no. 1, hlm. 115–126, Sep 2023.
- [2] S. F. Nazila, Y. Arman, D. Wahyuni, N. Nurhasanah, dan Y. S. Putra, “Deteksi Dini Serangan Hama Penyakit pada Cabai Rawit Menggunakan Metode Image Recognition,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 2, Agu 2023, doi: 10.28932/jutisi.v9i2.6342.
- [3] D. A. Puspitasari, N. Malahyati, dan Z. N. Fadillah, “Distribusi Perdagangan Komoditas Cabai Merah di Indonesia 2022,” *Badan Pusat Statistik*, 2022.
- [4] A. D. Sidik dan A. Ansawarman, “Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning,” *Formosa Journal of Multidisciplinary Research (FJMR)*, vol. 1, no. 3, hlm. 559–568, 2022, doi: 10.55927.
- [5] S. Misra dan H. Li, “Deep Neural Network Architectures To Approximate The Fluid-Filled Pore Size Distributions Of Subsurface Geological Formations,” <https://www.sciencedirect.com/topics/engineering/deep-learning>.
- [6] L. Utari dan A. Zulfikar, “Penerapan Convolutional Neural Networks Menggunakan Edge Detection Untuk Identifikasi Motif Jenis Batik,” *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 13, no. 1, hlm. 110–123, Feb 2023, doi: 10.36350/jbs.v13i1.184.
- [7] Y. S. Hidayat, A. G. Herdipriyansah, I. A. Agustin, J. Jaenudin, dan D. Desrizal, “Pemodelan Data Mining Menggunakan Neural Network Untuk Seleksi Mahasiswa Penerima Beasiswa Bantuan,” *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 14, no. 1, hlm. 9–20, Jan 2024, doi: 10.36350/jbs.v14i1.222.
- [8] E. Suharto, Suhartono, A. P. Widodo, dan E. A. Sarwoko, “The use of mobilenet v1 for identifying various types of freshwater fish,” *J Phys Conf Ser*, vol. 1524, no. 1, Jun 2020, doi: 10.1088/1742-6596/1524/1/012105.
- [9] M. Irmayansyah, “Fuzzy Inference System Tsukamoto Untuk Optimalisasi Jumlah Media Promosi Brosur di Sekolah Menengah Kejuruan,” *Teknois: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 6, no. 2, hlm. 12–24, Agu 2019, doi: 10.36350/jbs.v6i2.37.
- [10] H. Hudori, “Resampling Neural Network Untuk Penanganan Class Imbalance Pada Prediksi Klaim Asuransi,” *Teknois: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 10, no. 1, hlm. 57–64, Mei 2020, doi: 10.36350/jbs.v10i1.78.