

**ANALISIS ARSITEKTUR DEEP LEARNING MOBILENET
DALAM MENGLASIFIKASI HAMA
DAUN JAMBU MADU**

SKRIPSI

**FITRIYAN SYAHPUTRA
198160053**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2023**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 16/1/24

Access From (repository.uma.ac.id)16/1/24

**ANALISIS ARSITEKTUR DEEP LEARNING MOBILENET
DALAM MENGLASIFIKASI HAMA
DAUN JAMBU MADU**

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana di Fakultas Teknik
Universitas Medan Area



Oleh:

**FITRIYAN SYAHPUTRA
198160053**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2023**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 16/1/24

Access From (repository.uma.ac.id)16/1/24

HALAMAN PENGESAHAN

Judul Skripsi : Analisis Arsitektur Deep Learning Mobilenet Dalam
Mengklasifikasi Hama Daun Jambu Madu
Nama : Fitriyan Syahputra
NPM : 198160053
Fakultas : Teknik
Prodi : Teknik Informatika

Disetujui Oleh :
Pembimbing



Muhathir, S.T., M.Kom
Diketahui Oleh :

Dekan Fakultas Teknik

Ketua Prodi Teknik Informatika



Dedias Murtanto, ST., MT.
NIDN : 0102027402



Rizki Moliono, S.Kom.M.Kom
NIDN : 0109038902

Tanggal Lulus : 29 September 2023

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian – bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi – sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan 29 September 2023


Fitriansyah Putra
NPM 198160053

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, Saya bertanda tangan dibawah ini:

Nama : Fitriansyah Putra
NPM : 198160053
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

Analisis Arsitektur Deep Learning Mobilenet Dalam Mengklasifikasi Hama Daun Jambu Madu

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (*Database*), merawat, dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan

Pada Tanggal : 29 September 2023

Yang Menyatakan



Fitriansyah Putra

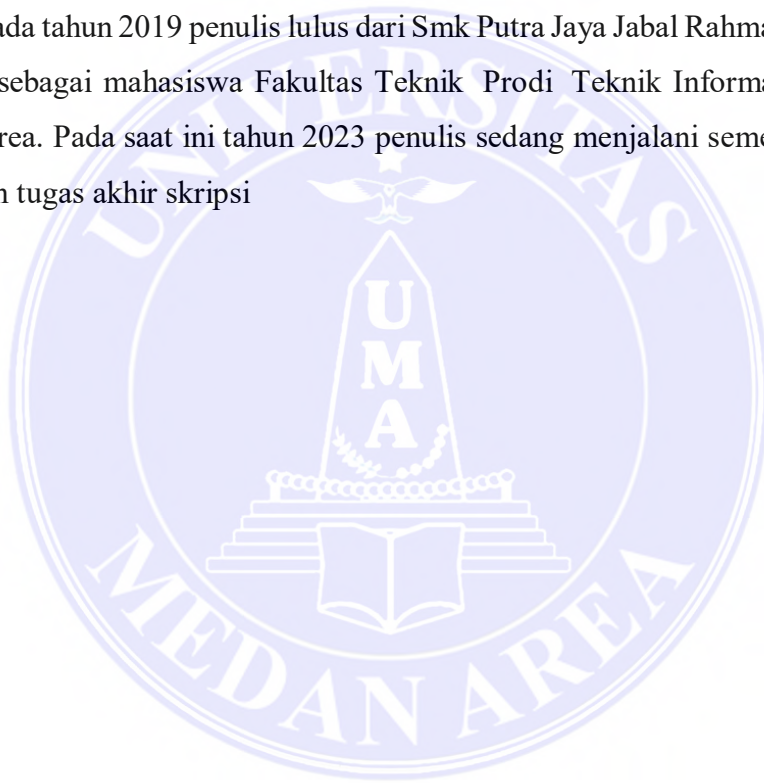
NPM 198160053

RIWAYAT HIDUP

Penulis lahir di Desa Teluk pada tanggal 24 Desember 2000 dari Ayah khairul Anwar dan Ibu Erhamna. Penulis adalah anak pertama dari 2 (dua) bersaudara

Penulis pertama kali mengenyam Pendidikan dibangku SD 056616 Dusun Pasar XII pada tahun 2007-2013, meneruskan Pendidikan di Sekolah Menengah Pertama Negeri 2 Stabat diselesaikan pada tahun 2013-2016, Meneruskan Pendidikan Sekolah Menengah Kejuruan Smk Putra Jaya Jabal Rahmah pada tahun 2016-2019.

Pada tahun 2019 penulis lulus dari Smk Putra Jaya Jabal Rahmah dan pada 2019 terdaftar sebagai mahasiswa Fakultas Teknik Prodi Teknik Informatika Universitas Medan Area. Pada saat ini tahun 2023 penulis sedang menjalani semester 8 (Delapan) menyusun tugas akhir skripsi



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Kuasa atas segala karuniaNya sehingga skripsi ini berhasil diselesaikan. Tema yang dipilih dalam penelitian ini ialah Deep Learning dengan judul “*Analisis Arsitektur Deep Learning Mobilenet Dalam Mengklasifikasi Hama Daun Jambu Madu*”.

Skripsi ini merupakan salah satu syarat untuk menyelesaikan pendidikan untuk mencapai gelar sarjana di Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area. Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc. selaku Rektor Universitas Medan Area.
2. Bapak Dr. Rahmad Syah, S.Kom., M. Kom selaku Dekan Fakultas Teknik Universitas Medan Area.
3. Bapak Rizki Muliono, S.Kom., M.Kom selaku Kepala Program Studi Teknik Informatika Universitas Medan Area.
4. Bapak Muhathir, S.T., M.Kom selaku Dosen pembimbing yang telah membantu penulis dari segi materi dan moril sehingga penulis dapat menyelesaikan skripsi ini.
5. Orang tua penulis yaitu Bapak Khairil Anwar dan Ibu Erhamna yang telah mendoakan tiada henti dan memberikan semangat serta membantu penulis dalam segi materi dan moril sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik baiknya.
6. Seluruh Dosen dan Staf Program Studi Teknik Informatika Universitas Medan Area.
7. Teman dekat penulis, Rahmad Syuhada yang selalu memberikan semangat selama penulisan proposal skripsi ini.
8. Seluruh teman-teman yang sudah memberikan dukungannya selama penulisan proposal skripsi ini, khususnya teman-teman Teknik Informatika angkatan 2019.

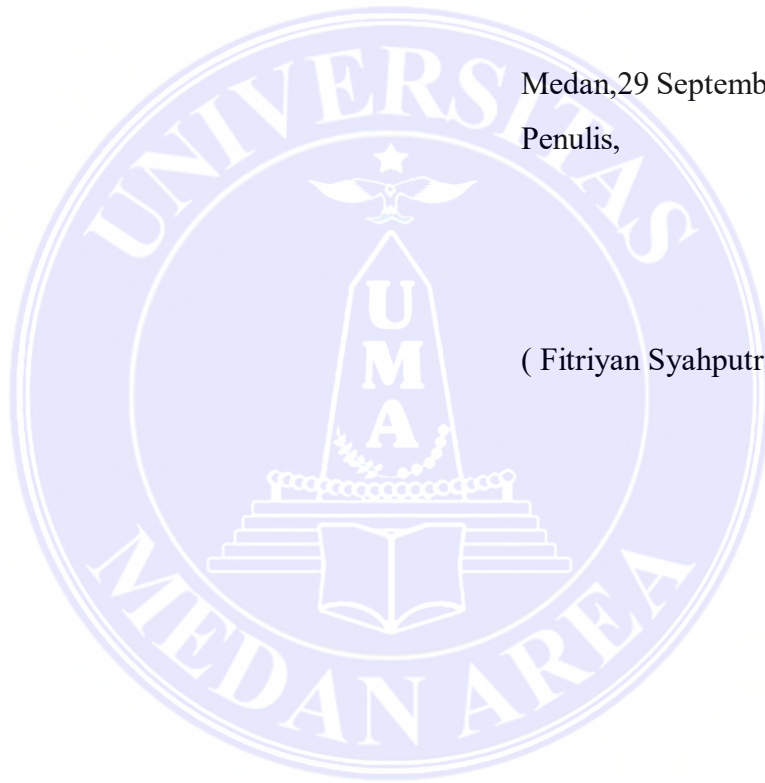
9. Seluruh pihak yang tidak dapat disebutkan satu persatu yang membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan skripsi ini. Penulis berharap tugas skripsi ini dapat bermanfaat baik untuk kalangan pendidikan maupun masyarakat. Akhir kata penulis ucapkan terima kasih.

Medan, 29 September 2023

Penulis,

(Fitriyan Syahputra)



ABSTRAK

Petani yang membudidayakan jambu madu memiliki masalah berupa hama yang menyerang pada daun. Klasifikasi hama dapat dilakukan dengan mengandalkan tenaga profesional atau pengalaman kerja petani, namun membutuhkan waktu dan keahlian yang cukup tinggi, sehingga mempengaruhi efisiensi serta keberhasilan dalam pengendalian hama. Oleh karena itu, diperlukan pendekatan berbasis teknologi digital yang dapat membantu petani dalam mengklasifikasi jenis hama pada daun jambu madu sebagai tindakan pencegahan, sehingga penurunan produktivitas atau gagal panen dapat dihindari. Penelitian ini memanfaatkan metode Transfer Learning dengan arsitektur *MobileNetV2* untuk klasifikasi hama pada daun jambu madu. Hasil dari uji coba pada penelitian ini menunjukkan bahwa metode Transfer Learning *MobileNetV2* memiliki performa klasifikasi yang baik dan menghasilkan nilai akurasi tinggi sebesar 99.72% dengan menggunakan hyperparameter jumlah *epoch* 20, batch size 32, optimizer Adam, dan learning rate 0.01. Hasil akurasi dari *model* setelah di uji menggunakan data testing (40 citra per class) didapatkan hasil akurasi sebesar 100%, nilai presisi (precision) sebesar 100%, recall 100%, dan f1-score 100% pada masing-masing class.

Kata Kunci : Klasifikasi, Hama Daun Jambu Madu, Deep Learning, Convolutional Neural Network, MobileNetV2

ABSTRACT

The farmers who cultivate honey guava have a problem in the form of pests that attack the leaves. Classification of pests can be done by relying on professional staff or farmer work experience, but it requires a high level of time and expertise, thus affecting efficiency and success in pest control. Therefore, a digital technology-based approach is needed that can help farmers classify types of pests on guava leaves as a preventive measure, so that decreased productivity or crop failure can be avoided. This study utilized the Transfer Learning method with the MobileNetV2 architecture for classifying pests on honey guava leaves. The results of the trials in this study indicate that the Transfer Learning MobileNetV2 method has good classification performance and produces a high accuracy value of 99.72% using the hyperparameter epoch number 20, batch size 32, Adam optimizer, and learning rate 0.01. The results of the accuracy of the model after being tested using data testing (40 images per class) obtained results of 100% accuracy, 100% precision, 100% recall, and 100% f1-score for each class.

Keyword : *Classification, Leaf Pests of Honey Guava, Deep Learning, Convolutional Neural Network, MobileNetV2*

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
KATA PENGANTAR.....	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Deep Learning	9
2.2 Convolutional Neural Network	10
2.2.1 Input Layer.....	11
2.2.2 Convolutional Layer	12
2.2.3 Activation Function.....	12
2.2.4 Pooling Layer.....	13
2.2.5 Flatten Layer	14
2.2.6 Fully-Connected Layer	15
2.3 Arsitektur MobileNetV2	16
2.4 Hyperparameter	20
2.5 Confusion Matrix.....	21
2.6 Hama Daun Jambu Madu.....	23
2.7 Penelitian Terdahulu	25

BAB III METODOLOGI PENELITIAN.....	30
3.1 Waktu dan Lokasi Penelitian	30
3.2 Alat dan Bahan Penelitian	30
3.3 Tahapan Penelitian.....	31
3.4 Prosedur Kerja	33
BAB IV HASIL DAN PEMBAHASAN.....	38
4.1 Preprocessing Dataset	38
4.1.1 Persiapan Dataset	38
4.1.2 Pembagian Dataset	42
4.1.3 Augmentasi Dataset.....	44
4.2 Hasil	47
4.2.1 Pemodelan Arsitektur MobileNetV2.....	47
4.2.2 Skenario Model	51
4.2.3 Hasil Training Model.....	51
4.2.4 Hasil Evaluasi Model.....	67
4.3 Pembahasan	74
4.3.1 Implementasi Sistem	75
4.3.2 Pengujian Sistem.....	78
BAB V KESIMPULAN DAN SARAN	82
5.1 Kesimpulan.....	82
5.2 Saran	82
DAFTAR PUSTAKA	84

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Convolutional Neural Network	11
Gambar 2. 2 Convolutional Layer.....	12
Gambar 2. 3 Pooling Layer dengan Max Pooling dan Average Pooling	15
Gambar 2. 4 Global Average Pooling	16
Gambar 2. 5 Fully Connected Layer	17
Gambar 2. 6 Arsitektur MobilenetV2.....	19
Gambar 2. 7 Struktur Layer MobilenetV2.....	19
Gambar 2. 8 Model Confusion Matrix	23
Gambar 2. 9 Jenis Hama.....	26
Gambar 3. 1 Sampel Data Penelitian	31
Gambar 3. 2 Diagram Prosedur Kerja Sistem.....	35
Gambar 4. 1 Sampel Dataset Kualitas Rendah	40
Gambar 4. 2 Proses Cropping Dataset.....	41
Gambar 4. 3 Proses Pelabelan Dataset	42
Gambar 4. 4 Sampel Dataset lalat Buah.....	42
Gambar 4. 5 Sampel Dataset Ulat Pagoda.....	43
Gambar 4. 6 Sampel Dataset Ulat Walang Sangit	43
Gambar 4. 7 Susunan Direktori Dataset.....	44
Gambar 4. 8 Kode Program Augmentasi Dataset	45
Gambar 4. 9 Hasil Augmentasi dengan Rotation.....	46
Gambar 4. 10 Hasil Augmentasi dengan Widht Shift	46
Gambar 4. 11 Hasil Augmentasi dengan Height Shift	46
Gambar 4. 12 Hasil Augmentasi dengan Shear	47
Gambar 4. 13 Hasil Augmentasi dengan Zoom.....	47
Gambar 4. 14 Hasil Augmentasi dengan Horizontal Flip.....	47
Gambar 4. 15 Rancangan Model Dengan Arsitektur MobilenetV2.....	53
Gambar 4. 16 Kode program untuk base model Mobilenetv2.....	54
Gambar 4. 17 Kode program untuk Create Model	54
Gambar 4. 18 Model Summary.....	55
Gambar 4. 19 Kode Program Model Skenario 1.....	55
Gambar 4. 20 Proses Fitting Model Skenario 1	56
Gambar 4. 21 Grafik Accuracy dan Loss Model Skenario 1.....	57
Gambar 4. 22 Grafik Accuracy dan Loss Model Skenario 2.....	58
Gambar 4. 23 Grafik Accuracy dan Loss Model Skenario 3.....	59
Gambar 4. 24 Grafik Accuracy dan Loss Model Skenario 4.....	60
Gambar 4. 25 Grafik Accuracy dan Loss Model Skenario 5.....	61
Gambar 4. 26 Grafik Accuracy dan Loss Model Skenario 6.....	62

Gambar 4. 27 Grafik Accuracy dan Loss Model Skenario 7.....	63
Gambar 4. 28 Grafik Accuracy dan Loss Model Skenario 8.....	64
Gambar 4. 29 Grafik Accuracy dan Loss Model Skenario 9.....	65
Gambar 4. 30 Grafik Accuracy dan Loss Model Skenario 10.....	66
Gambar 4. 31 Grafik Accuracy dan Loss Model Skenario 11.....	67
Gambar 4. 32 Grafik Accuracy dan Loss Model Skenario 12.....	68
Gambar 4. 33 Grafik Perbandingan Validation Accuracy Model.....	69
Gambar 4. 34 Grafik Perbandingan Validation Loss Model.....	69
Gambar 4. 35 Confusion Matrix Model Skenario 3.....	71
Gambar 4. 36 Confusion Matrix Model Skenario 6.....	71
Gambar 4. 37 Confusion Matrix Model Skenario 8.....	72
Gambar 4. 38 Confusion Matrix Model Skenario 9.....	73
Gambar 4. 39 Confusion Matrix Model Skenario 10.....	73
Gambar 4. 40 Confusion Matrix Model Skenario 11.....	74
Gambar 4. 41 Kode Program Simpan Model.....	75
Gambar 4. 42 Implementasi Halaman Utama.....	76
Gambar 4. 43 Halaman Klasifikasi	77
Gambar 4. 44 Hasil Klasifikasi Pada hama lalat buah	77
Gambar 4. 45 Hasil Klasifikasi Pada hama ulat pagoda	78
Gambar 4. 46 Hasil Klasifikasi Pada hama Walang Sangit.....	79

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	27
Tabel 3. 1 sampel Data Penelitian.....	31
Tabel 4. 1 Pembagian Dataset.....	44
Tabel 4. 2 Skenario Training Model	51
Tabel 4. 3 Perbandingan Akurasi Skenario Model	70
Tabel 4. 4 Hasil Pengujian Menggunakan Aplikasi.....	80
Tabel 4. 5 Pperbandingan Akurasi Model Dengan Penelitian terdahulu.....	80





BAB I

PENDAHULUAN

1.1 Latar Belakang

Tanaman jambu madu merupakan salah satu komoditas pertanian yang sudah sangat dikenal oleh masyarakat dan telah dimanfaatkan untuk bahan makanan serta memiliki nilai ekonomi tinggi. Produksi hasil panen menjadi tujuan utama petani dalam membudidayakan jambu madu. Namun, tanaman jambu madu sering kali diserang oleh hama daun yang dapat menyebabkan kerugian yang signifikan pada produksi dan kualitas hasil panen. Hama pada daun jambu madu memiliki karakteristik yang berbeda-beda dan dapat dibedakan dari kerusakan pada daunnya. Hama daun pada tanaman jambu madu dapat beragam jenisnya, namun yang umum menyerang tanaman jambu madu, diantaranya adalah hama lalat buah, hama ulat pagoda, dan hama walang sangit.

Serangan hama pada daun jambu madu dapat menurunkan produksi hasil panen dan bahkan dapat menyebabkan gagal panen. Semakin besar persentase serangan hama maka produktivitas lahan akan semakin rendah. Oleh karena itu, klasifikasi hama pada daun jambu madu sangat penting dilakukan untuk mengetahui perawatan yang diperlukan dalam keberlanjutan sistem budidaya sehingga dapat mencegah turunnya hasil panen dan kualitas tanaman.

Klasifikasi hama pada daun jambu madu dapat dilakukan dengan mengandalkan tenaga profesional atau pengalaman kerja petani, namun membutuhkan waktu dan keahlian yang cukup tinggi, sehingga mempengaruhi efisiensi serta keberhasilan dalam pengendalian hama. Selain itu, kesalahan dalam identifikasi hama pada daun jambu madu dapat menyebabkan penggunaan pestisida

yang tidak tepat sasaran, yang pada gilirannya dapat berdampak negatif terhadap tanaman jambu madu. Oleh karena itu, diperlukan pendekatan berbasis teknologi digital yang dapat membantu petani dalam mengklasifikasi jenis hama pada daun jambu madu sebagai tindakan pencegahan, sehingga penurunan produktivitas atau gagal panen dapat dihindari.

Dalam beberapa tahun terakhir, perkembangan teknologi *Deep Learning* telah menunjukkan potensi yang besar dalam mengatasi masalah klasifikasi visual, termasuk dalam bidang pertanian. Salah satu arsitektur *Deep Learning* yang telah terbukti efektif dalam tugas-tugas klasifikasi gambar adalah *MobileNetV2*. Penelitian terdahulu yang berkaitan dengan penerapan arsitektur *MobileNetV2* pada umumnya telah banyak dilakukan oleh peneliti-peneliti sebelumnya.

Dalam penelitian (Desi Ramayanti, 2022) melakukan klasifikasi 50 jenis citra kupu-kupu dengan membandingkan arsitektur *VGG16* dan *MobileNetV2*, diperoleh hasil bahwa *MobileNetV2* lebih unggul dibandingkan *VGG16*, dengan akurasi terbaik diperoleh *MobileNetV2* tanpa *fine-tuning* mencapai 96%. Dalam penelitian (Yusup Miftahuddin, 2022) dengan membandingkan *EfficientnetB3* dan *MobileNetV2* untuk identifikasi 7 jenis buah-buahan. Hasil penelitian menunjukkan bahwa *MobileNetV2* dengan *epoch* 20 mendapat akurasi 99% dan lebih unggul dari *EfficientnetB3* yang hanya mencapai akurasi 23%.

Penelitian lain dilakukan (Gulzar, 2023) melakukan klasifikasi pada 40 jenis gambar buah-buahan. Hasil penelitian menunjukkan bahwa arsitektur *MobileNetV2* menghasilkan akurasi 99% dan lebih unggul dari *AlexNet*, *VGG16*, *InceptionV3*, dan *ResNet*. Dalam penelitian (Dino Hariatma Putra, 2023) dengan membandingkan *VGG16*, *MobileNetV2* dan *InceptionV3* untuk sistem deteksi 3 kelas (menggunakan

masker dengan benar, tidak menggunakan masker, dan menggunakan masker tidak bena) masker secara otomatis. Hasil penelitian menunjukkan bahwa *MobileNetV2* mendapatkan hasil akurasi tertinggi dengan persentase mencapai 100%.

Hasil kajian dari penelitian-penelitian terdahulu membuktikan arsitektur *MobileNetV2* memiliki performa yang sangat baik dalam melakukan tugas klasifikasi pada gambar. Hal ini menjadi rujukan dalam memilih *MobileNetV2* sebagai arsitektur *Deep Learning* dalam klasifikasi hama pada daun jambu madupada penelitian ini. Namun, meskipun arsitektur *MobileNetV2* telah banyak digunakan dalam berbagai aplikasi klasifikasi gambar, penggunaannya dalam mengklasifikasikan hama daun pada tanaman jambu madu masih terbatas. Permasalahan ini muncul karena keterbatasan data pelatihan yang spesifik untuk hama daun jambu madu serta perbedaan karakteristik visual antara hama daun dengan kelas-kelas lain dalam *dataset* gambar. Oleh karena itu, perlu dilakukan analisis arsitektur *Deep Learning MobileNet* dalam konteks klasifikasi hama daun jambu madu. Analisis ini diharapkan dapat memberikan pemahaman lebih dalam tentang keunggulan dan keterbatasan *MobileNetV2* dalam mengenali dan mengklasifikasikan hama daun pada tanaman jambu madu.

Berdasarkan latar belakang masalah yang telah dijelaskan sebelumnya, maka dilakukan penelitian dengan judul “Analisis Arsitektur *Deep Learning Mobilenet* Dalam Mengklasifikasi Hama Daun Jambu Madu”. Dengan menggunakan arsitektur *MobileNetV2*, klasifikasi jenis hama pada daun jambu madu dapat dilakukan secara otomatis dan akurat sehingga dapat membantu petani dalam mengidentifikasi jenis hama untuk dilakukan tindakan lebih lanjut.

1.2 Perumusan Masalah

Merujuk pada latar belakang masalah yang telah diuraikan serta hasil dari penelitian sebelumnya, maka dapat dirumuskan pokok permasalahan yang akan dibahas dalam penelitian ini, yaitu sebagai berikut:

1. Bagaimana menerapkan arsitektur *Deep Learning Mobilenet* dalam mengklasifikasi hama daun pada tanaman jambu madu.
2. Bagaimana performa arsitektur *Deep Learning Mobilenet* dalam mengklasifikasikan hama daun pada tanaman jambu madu.

1.3 Batasan Masalah

Permasalahan yang dibahas pada penelitian ini memiliki ruang lingkup yang luas, sehingga diberikan batasan-batasan agar penelitian lebih terarah dan sesuai dengan tujuan yang ingin dicapai, yaitu sebagai berikut:

1. Objek penelitian yang diklasifikasi adalah hama pada daun jambu air jenis Jambu air Madu Deli Hijau.
2. Jumlah data eksperimen yang digunakan yaitu 2400 *dataset* gambar hama daun jambu madu dengan format *.jpg/jpeg* yang diambil dengan menggunakan kamera *smartphone*.
3. *Dataset* terbagi menjadi 3 yaitu data *training* dengan persentase 80%, data *validation* 15%, dan data *testing* 5%.
4. Hasil klasifikasi hama daun jambu madu terbagi menjadi tiga kelas, yaitu (a) hama lalat buah (b) hama ulat pagoda (c) hama walang sangit. Hasil pengelompokan hanya didasarkan pada sistem klasifikasi dengan komputer tanpa bantuan dari ahli penyakit tanaman.

5. Algoritma *Deep Learning* yang digunakan adalah *Convolutional Neural Network* dengan menggunakan arsitektur *MobileNetV2*.
6. Bahasa pemrograman yang digunakan untuk melakukan *training model* adalah *Python 3.10.11* menggunakan bantuan *Google Collaboratory Pro*.
7. Pengujian *model* berdasarkan perbandingan jumlah *epoch* (15 dan 20), *optimizer* (*adam* dan *sgd*) serta *learning rate* (0.01 dan 0.03). Sedangkan parameter dari *input shape* : 224x224x3, dan *batch size* : 32 akan dijadikan statis pada setiap skenario *model* yang diuji.
8. Pengujian *model* dengan data baru (*data testing*) akan dievaluasi dengan menggunakan *confusion matrix* dan *classification report* untuk memilih *model* terbaik.
9. *Model* terbaik yang dipilih akan di *deploy* kedalam aplikasi berbasis *web* menggunakan bahasa pemrograman *Python* dengan *Flask* sebagai *library* utama.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah maka tujuan yang ingin dicapai dalam penelitian ini, yaitu sebagai berikut:

1. Membangun sebuah *model* yang dapat mengklasifikasi hama pada daun jambu madu dengan arsitektur *Deep Learning Mobilenet* sehingga dapat membantu petani dalam mengidentifikasi jenis hama untuk dilakukan tindakan lebih lanjut.
2. Mengevaluasi performa tingkat akurasi *model* dari arsitektur *Deep Learning Mobilenet* sehingga dapat menghasilkan klasifikasi hama pada daun jambu madu secara akurat.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini, baik secara teoritis maupun secara praktis antara lain yaitu sebagai berikut:

1. Memberikan pemahaman yang lebih dalam tentang keefektifan dan keunggulan arsitektur *MobileNetV2* dalam mengklasifikasikan hama daun pada tanaman jambu madu. Hasil penelitian ini dapat memberikan informasi penting bagi petani dan praktisi pertanian dalam memilih dan menerapkan teknologi yang tepat untuk mengendalikan hama pada tanaman jambu madu.
2. Membantu meningkatkan efisiensi dan akurasi dalam identifikasi dan klasifikasi hama daun pada tanaman jambu madu. Dengan memanfaatkan arsitektur *MobileNetV2*, penelitian ini dapat memberikan solusi yang efektif dalam mengenali jenis hama daun secara otomatis, sehingga memungkinkan adopsi teknologi yang lebih cepat dan efisien dalam pengendalian hama pada tanaman jambu madu.
2. Berpotensi mendukung pengembangan teknologi pertanian yang lebih efisien dan berkelanjutan. Dengan meningkatnya efisiensi dan akurasi dalam klasifikasi hama daun jambu madu, petani dapat mengurangi penggunaan pestisida yang berlebihan dan mengarahkan tindakan pengendalian hama secara tepat sasaran, sehingga berkontribusi pada keberlanjutan lingkungan dan kesehatan manusia.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan skripsi ini dikelompokkan menjadi lima bab, yaitu sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini dijelaskan terkait penyampaian masalah yang dikemas melalui latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini menguraikan rangkuman dasar teori yang dihimpun dari berbagai pustaka yang relevan dengan topik yang menjadi objek kajian.

BAB III : METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang tahapan-tahapan yang akan dilakukan dalam penelitian ini. Setiap rencana dari tahapan penelitian dideskripsikan secara rinci.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian serta pembahasan dari penelitian yang telah dilaksanakan. Dalam bab ini disajikan gambar, tabel, serta grafik dari hasil penelitian yang telah dilakukan.

BAB V : KESIMPULAN DAN SARAN

Bab ini merupakan kesimpulan dari keseluruhan pembahasan terutama hasil penelitian yang telah dijelaskan pada bab sebelumnya. Bab ini juga berisi saran untuk penelitian selanjutnya yang akan mengangkat tema yang sama dengan penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 *Deep Learning*

Deep Neural Network atau *Deep Learning* adalah *neural network* yang memiliki banyak *layer*. Pada umumnya, *Deep Neural Network* memiliki 3 *layers* (*input layer*, $N \geq 2$ *hidden layers*, *output layers*), dengan kata lain adalah *Multi layer Perceptron* (MLP) dengan lebih banyak *layer* (Dino & Sulistyowati, , 2022) disebut sebagai *deep* karena banyaknya *hidden layer* maupun *neuron* didalamnya bisa banyak sekali. Algoritma *Deep Learning* dilatih untuk mengidentifikasi pola dan mengklasifikasikan berbagai jenis informasi untuk memberikan *output* yang diinginkan ketika menerima *input* baru (Intan & Ino , 2022).

Deep Learning merupakan salah satu bidang dalam *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan *dataset* besar (Dwi , Imsak , & Andriyan , 2022). *Deep Learning* terus berkembang dan sering digunakan oleh peneliti serta industri dalam membantu memecahkan banyak masalah dalam data besar seperti misalnya *Computer Vision*, *Speech Recognition*, dan *Natural Language Processing* (TIKA , 2021)

Perbedaan utama dari *Machine Learning* dan *Deep Learning*, yaitu di *Machine Learning*, *feature extraction* perlu dirancang secara manual, yang mana memakan banyak sekali waktu dan tenaga. Sedangkan *Deep Learning* akan secara otomatis mengekstrak *features* untuk klasifikasi. Namun di sisi lain, *Deep Learning* menuntut sejumlah data yang besar untuk melatih algoritmanya (Fauzan , Intan, & Ino , 2022) *Convolutional Neural Network* (CNN) adalah salah satu algoritma *Deep Learning* yang dapat diterapkan untuk klasifikasi citra.

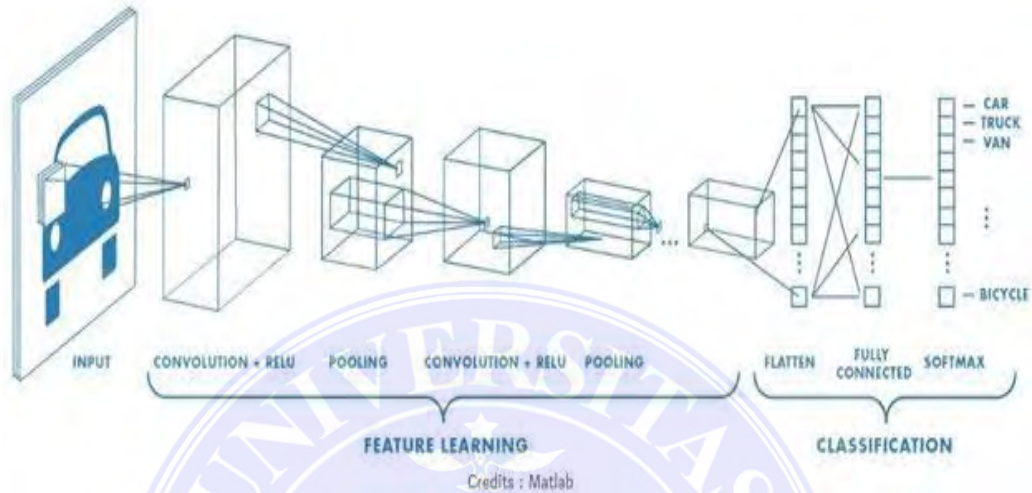
2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu algoritma *Deep Learning* yang dirancang untuk mengolah data dalam bentuk data dua dimensi, misalnya gambar atau suara (Fauzan , Intan, & Ino , 2022). CNN sangat cocok dalam pemrosesan *input* berupa citra (Fakhri , Fasliah , Muhamad , & Toni , 2022) CNN adalah pengembangan dari *Multi layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi (Dino & Sulistyowati, Perbandingan Tingkat Akurasi Arsitektur Convolutional Neural Network untuk Model Deteksi Penggunaan Masker Secara Otomatis, 2023) .Metode kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap *neuron* dipresentasikan dalam bentuk dua dimensi, tidak semacam MLP yang tiap *neuron* hanya berukuran satu dimensi (Ferry , Made , & Duman , 2021)

Tingkat popularitas algoritma CNN semakin meningkat dalam teknik *Deep Learning* karena beberapa faktor penting, seperti tidak perlu ekstraksi fitur secara manual, kemampuan untuk melatih ulang untuk tugas mengenali objek baru sehingga dapat dibangun di atas jaringan yang sudah ada (Dwiretno , Mohammad , & Muhamad , 2023) Sebagai algoritma *Deep Learning*, CNN telah mengungguli banyak metode konvensional dalam hal akurasi klasifikasi (Tsabitah , Vizza, & Agus , 2021).

Arsitektur CNN terdiri dari beberapa komponen diantaranya, satu lapisan masukan (*input layer*), suatu lapisan keluaran (*output layer*) dan sejumlah lapisan tersembunyi (*hidden layer*). *Hidden layer* umumnya berisi *convolutional layers*, *pooling layers*, *normalization layers*, *ReLU layer*, dan *fully connected layers* (wharyanto, 2021). Dimana komponen-komponen tersebut yang membangun

arsitektur dari CNN, sehingga pemrosesan dengan *input* berupa citra sangat cocok menggunakan CNN (Dwi , Arif , Putri, & Chaerur , 2022). Adapun arsitektur dari CNN dapat ditunjukkan pada Gambar 2.1.



Gambar 2.1 Arsitektur Convolutional Neural Network
Sumber : (Susi , Aradea, & Andi , 2022)

Secara umum tipe lapisan CNN pada Gambar 2.1 dibagi menjadi dua bagian, yaitu *feature extraction layer (feature learning)* dan *classification layer*. Pada bagian *feature learning* terdiri dari *convolutional layer*, fungsi aktivasi *ReLU*, dan *pooling layer*. Sedangkan pada bagian *classification layer* terdiri dari *flatten layer*, *fully connected layer*, dan fungsi aktivasi *Softmax*. Adapun penjelasan dari masing-masing *layer* pada arsitektur CNN dapat diuraikan pada sub bab selanjutnya.

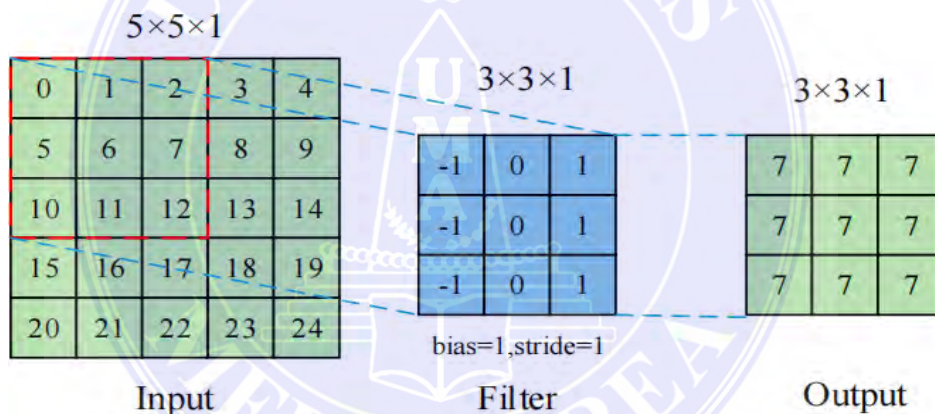
2.2.1 Input Layer

Input layer pada CNN merepresentasikan gambar atau citra yang dimasukkan ke dalam *model*. Jika citra yang dimasukkan memiliki ukuran 224x224 piksel dan menggunakan format RGB (*Red, Green, Blue*), maka citra *input* tersebut akan berupa *array* multidimensi dengan ukuran 224x224x3 (angka 3

merepresentasikan jumlah *channel* warna RGB). Tahap *input layer* ini akan menjadi masukan pada proses *convolutional layer*.

2.2.2 Convolutional Layer

Convolutional layer (lapisan konvolusi) adalah blok utama CNN dan terdiri dari berbagai *filter* yang di inisialisasi secara acak untuk melakukan operasi konvolusi yang bertindak sebagai ekstraktor fitur untuk memeriksa representasi fitur dari gambar *input* (Arif , Arief , Moch, & Abdur , 2022) Fungsi *convolution layer* yaitu mengambil fitur pada citra yang akan digunakan untuk melatih *model*. Keluaran dari *convolutional layer* biasa disebut *feature map* (Julianto, 2021). Adapun ilustrasi proses konvolusi dapat dijelaskan pada Gambar 2.2.



Gambar 2.2 Ilustrasi Pada Convolution Layer
Sumber : (Jinzhu , Lijuan , & Huanyu , 2021)

Konvolusi merupakan istilah matematis untuk pengaplikasian sebuah fungsi pada *output* fungsi yang lain secara berulang. Operasi ini menerapkan fungsi *output* sebagai *feature map* pada *input* citra. Gambar *input* diperkecil ke ukuran yang lebih kecil menggunakan *filter*. *Filter* digeser selangkah demi selangkah mulai dari sisi kiri atas gambar. Pada setiap langkah, nilai pada gambar dikalikan dengan nilai

filter dan hasilnya dijumlahkan. Operasi konvolusi pada citra dengan lebih dari satu dimensi, dapat digunakan rumus pada persamaan (2.1).

$$\begin{aligned} S_{(i,j)} &= (K * I)_{(i,j)} \\ &= \sum_{i=1}^m \sum_{j=1}^n (I_{(i-m,j-n)} * K)_{(m,n)} \end{aligned} \quad (2.1)$$

Persamaan 2.1 merupakan rumus untuk menghitung operasi konvolusi dengan i dan j sebagai piksel dari citra. Rumus ini bersifat komutatif dan hanya terjadi saat kernel K dan *input* I dapat dibalik relatif terhadap satu sama lain.

2.2.3 Activation Function

Fungsi aktivasi (*activation function*) adalah node yang diletakkan di akhir atau di antara *neural networks* supaya hasil yang dikeluarkan lebih akurat lagi (Tsabitah Ayu, Vizza Dwi, & Agus Eko Minarno , 2021). Fungsi dari *activation function* yaitu menentukan apakah suatu *neuron* harus aktif atau tidak berdasarkan dari *weighted sum* dari *input* (ADILAH , 2021). Fungsi aktivasi yang umum digunakan dalam metode CNN adalah fungsi aktivasi *ReLU* dan fungsi aktivasi *Softmax*.

1. Fungsi aktivasi *ReLU*

Fungsi aktivasi *ReLU* (*Rectified Linear Unit*) digunakan setelah melakukan proses konvolusi atau sebelum melakukan *pooling* (di tahap *feature extractor*) dan dipakai juga di setiap *node* pada *hidden layer* (di tahap *fully connected*) (WINNARTO , 2021). *ReLU* digunakan untuk mengubah nilai piksel menjadi 0 jika nilai piksel negatif. Jika nilainya tidak kurang dari 0, berlaku sebaliknya. Tujuan *ReLU* ialah mengurangi linearitas yang terjadi dari proses konvolusi sehingga CNN lebih mudah mencapai nilai optimum.

ReLU juga dapat menutupi kelemahan yang dimiliki oleh *Sigmoid* dan *Tanh* (Tika, 2021) Fungsi aktivasi *ReLU* dengan menggunakan persamaan (2.2).

$$f(x_i) = \max(0, x_i) \quad (2.2)$$

2. Fungsi aktivasi *Softmax*

Aktivasi *softmax* merupakan penerapan pada lapisan *layer* terakhir (*output layer*). *Softmax* lebih umum digunakan dari pada *Sigmoid*, *ReLU*, ataupun *Tanh* (Faizin, 2022). *Softmax* merupakan fungsi yang digunakan untuk klasifikasi lebih dari dua kelas. Keuntungan menggunakan *Softmax* yaitu nilai rentang probabilitas yang dihasilkan dari 0 hingga 1, dan jumlah semua kemungkinan sama dengan satu (Winarto, 2021). Fungsi aktivasi *Softmax* dengan menggunakan persamaan (2.3).

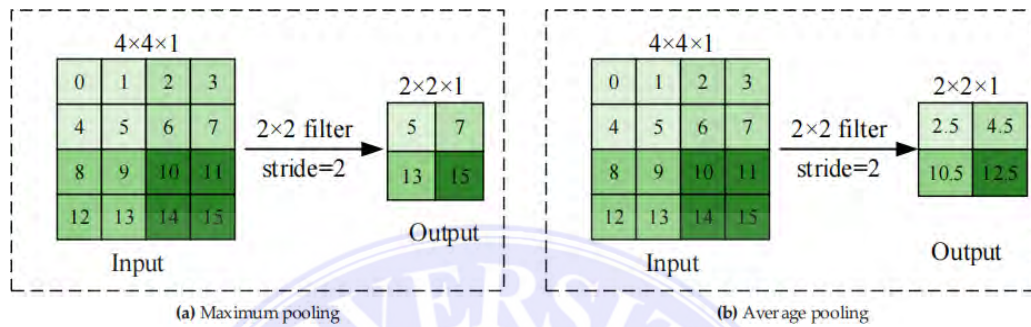
$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (2.3)$$

Notasi f_j menunjukkan hasil fungsi untuk setiap elemen ke- j pada vektor keluaran kelas. Argumen z adalah hipotesis yang diberikan oleh *model* pelatihan agar dapat diklasifikasi oleh fungsi *Softmax*.

2.2.4 *Pooling Layer*

Pooling adalah pengurangan ukuran pada matriks dengan menggunakan operasi *pooling*. *Pooling layer* biasanya terletak setelah *convolutional layer*. Pada dasarnya *pooling layer* terdiri dari sebuah *filter* yang memiliki ukuran dan langkah tertentu yang akan bergeser pada seluruh area pada *feature map*. *Pooling* yang umum digunakan adalah *maximum pooling* dan *average pooling*. Tujuannya yaitu mengurangi dimensi peta fitur (*down sampling*) sehingga mempercepat perhitungan, karena sedikit parameter yang diperbaharui dan *over-fitting* (Muhammad , Aji , & Muhammad , 2021).

Average pooling yaitu mengambil nilai rata-rata dari area yang dipilih, sedangkan *max pooling* yaitu mengambil nilai terbesar dari area yang dipilih (Hariz et al, 2022). Adapun ilustrasi *pooling layer* dengan operasi *max* atau biasa disebut dengan *max-pooling* dan *average pooling* dapat dilihat pada Gambar 2.3.

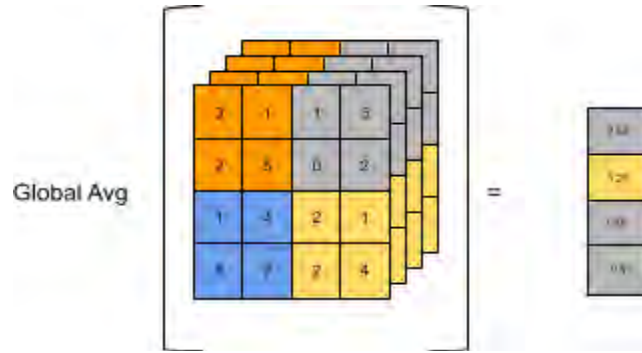


Gambar 2.3 Ilustrasi Pada Pooling Layer
Sumber : (Lu , Lijuan , & Huanyu , 2021)

2.2.5 Flatten Layer

Flatten layer merupakan langkah yang dilakukan sebelum masuk ke *fully connected layer*. *Feature map* dari tahap *feature extractor* masih berbentuk *array multidimensi*, sehingga perlu dilakukan proses *flatten* untuk mengubah ulang fitur (*reshape feature map*) menjadi vektor satu dimensi agar dapat digunakan sebagai *input* untuk tahap klasifikasi atau *fully connected*. Pada penelitian ini *flatten layer* akan diganti dengan *Global Average Pooling*.

Global Average Pooling adalah proses konvolusi yang dapat menggantikan *Flatten* sebagai pengekstrak fitur (SYARIF , 2021). *Global Average Pooling* memiliki keunggulan yaitu, komputasi biaya dapat dikurangi dengan mengurangi jumlah parameter yang akan ditangani (*hyperparameters*) serta beberapa parameter *model* dapat dihilangkan untuk mengurangi *overfitting*. Oleh karena itu, tidak perlu bergantung pada *dropout* (ADILAH , 2021). Konsep dari *Global Average Pooling* dapat ditunjukkan pada Gambar 2.4.

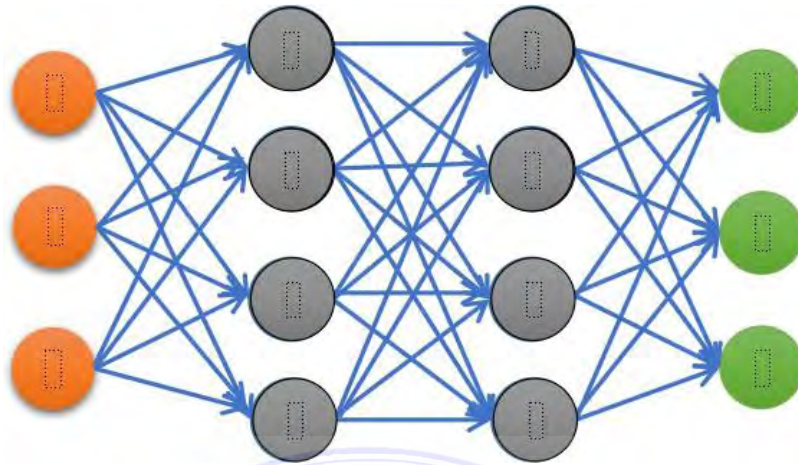


Gambar 2.4 Ilustrasi Pada Global Average Pooling
Sumber : (SYARIF , 2021)

Global Average Pooling merupakan operasi yang mereduksi suatu ukuran matriks menjadi matriks yang kecil. Contohnya matriks 4x4 diubah menjadi matriks 1x1 dengan cara mengambil nilai rata-rata pada setiap matriks tersebut seperti yang ditunjukkan oleh Gambar 2.4.

2.2.6 Fully-Connected Layer

Fully-Connected layer merupakan lapisan dimana setiap *neuron* yang telah diaktivasi dari lapisan sebelumnya terhubung pada semua *neuron* di lapisan selanjutnya. Nilai matriks yang diperoleh setelah proses *convolution*, *pooling*, dan *activation* dimasukkan kedalam *fully connected layer* sebagai pengenalan *input* dan kalsifikasi di *layer* ini (Ayu, Vizza, & Agus, 2021) *Fully connected layer* merupakan lapisan dimana dimensi seluruh data akan diubah menjadi satu dimensi, proses perubahan ukuran dimensi disebut *Flatten*. Lapisan ini memiliki *node* yang saling berhubungan, memiliki bobot, dan memiliki fungsi aktivasi. Lapisan ini mengeluarkan sebuah prediksi berdasarkan data masukan (Hariz, Yulita, & Suryana, 2022). Gambar 2.5 merupakan ilustrasi dari *Fully Connected layer* serta *output layer*



Gambar 2.5 Ilustrasi Proses Fully Connected Layer

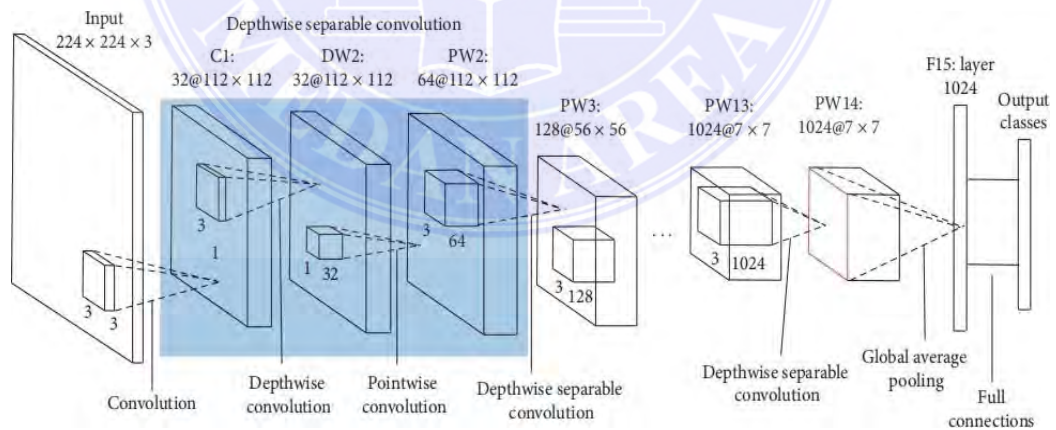
Fully-Connected Layer terdiri dari *input*, *hidden* dan *output layer*, jika diperhatikan *layer-layer* ini sama seperti di *Artificial Neural Network* (ANN). Oleh karena itu beberapa literatur menyebut *fully connected layer* ini sebagai ANN. Disetiap *hidden layer* terdapat fungsi aktivasi, yang umum digunakan yaitu *ReLU*, dan begitu pula di *output layer*, juga terdapat fungsi aktivasi, dalam kasus klasifikasi (yang lebih dari 2 label/kelas/kategori) yang umum digunakan yaitu *Softmax*. Tujuan utama dari *fully connected layer* ialah mengolah data sehingga bisa dilakukan klasifikasi. *output* dari *fully connected layer* yaitu probabilitas terhadap kategori (jika menggunakan *Softmax*) (WINNARTO , 2021).

2.3 Arsitektur MobileNetV2

MobileNet merupakan salah satu arsitektur yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* yang berlebih (Fuadi & Suharso, 2022). *MobileNetV2* adalah salah satu arsitektur CNN yang merupakan *model* pengembangan dari *MobileNetV1*. *MobileNetV2* menunjukkan hasil akurasi yang

lebih baik dibandingkan *MobileNetV1* dengan jumlah parameter yang lebih sedikit (Hariz, Yulita, & Suryana, 2022). *MobileNetV2* terbukti efektif dalam berbagai tugas *Machine Learning*, termasuk klasifikasi gambar dan pengenalan objek (Syahputra, Siregar, & Insani, 2023). Perbedaan antara arsitektur *MobileNetV2* dan CNN pada umumnya adalah penggunaan *convolution layer* dengan ketebalan *filter* yang sesuai dengan ketebalan dari *input image* (Miftahuddin & Zaelani S, 2022).

Pada arsitektur *MobileNetV2* ada 2 fitur yaitu *Linear Bottleneck* dan *Shortcut Connection* antar *Bottleneck*. Kedua fitur ini menjadikan proses *training* pada *neural network* dapat berjalan dengan lebih efisien dan menghasilkan akurasi yang lebih baik (Annur, Jumhurul, Annafii, Trisnaningrum, & Putra, 2023). *MobileNetV2* adalah *pre-trained model* (model yang sudah dilatih) dalam mengklasifikasikan gambar dengan menggunakan *dataset* yang besar. Dengan menggunakan *pre-trained model*, maka pengembang tidak perlu membangun atau melatih *model* dari awal sehingga dapat menghemat waktu (Al-Rammahi, 2022).



Gambar 2.6 Arsitektur MobileNetV2
Sumber : (Wang & oLi, 2020)

Terdapat dua operasi di dalam *MobileNet*, yaitu *Depthwise Convolution* (DW) dan *Pointwise Convolution* (PW) (Hastomo, Sugiyanto, & Sudjiran, 2021),

yang pertama digunakan untuk *residual block* dan yang kedua digunakan untuk *down sizing* (Julianto, 2021). Pada intinya *model MobileNetV2* pada setiap *block* terdapat *layer*, jika ditotal kan terdapat ± 150 *layer* (Miftahuddin & Zaelani S, 2022). Gambar 2.7 menampilkan *layer* pada arsitektur *MobileNetV2*.

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 2.7 Struktur Layer MobileNetV2
 Sumber : (Julianto, 2021)

Struktur dari *block MobileNetV2* dapat dilihat pada Gambar 2.7. Disetiap baris mendeskripsikan *sequence* dari 1 atau lebih modul *stride layers*, beserta jumlah berapa kali pengulangan *n*. Setiap *layer* di *sequence* yang sama memiliki nilai *c* sebagai *output channel*. Di *layer* pertama setiap *sequence* memiliki nilai *stride s* dan *stride* lainnya menggunakan *stride* 1. Seluruh *spatial convolution* menggunakan 3×3 *kernels* (Julianto, 2021). Dalam penelitian ini, arsitektur *MobileNetV2* digunakan untuk mengklasifikasi hama pada daun jambu madu.

2.4 *Hyperparameter*

model yang baik adalah *model* yang mampu menjelaskan data secara akurat tanpa terpengaruh oleh data *noise*. *model* yang baik akan memiliki *loss* rendah dan akurasi tinggi. Namun, dalam penggunaannya di CNN, seringkali terjadi *overfitting* sehingga diperlukan *fine-tuning* atau penyesuaian *model*. Salah satu cara penyesuaian yang dapat dilakukan adalah dengan menyesuaikan *hyperparameter*, yaitu parameter yang digunakan untuk mengontrol proses pelatihan. Dengan menyesuaikan *hyperparameter*, hasil akurasi prediksi dapat ditingkatkan. Berikut beberapa *hyperparameter* yang bisa disesuaikan oleh perancang *model* CNN, yaitu sebagai berikut:

1. *Input Shape*

Ukuran citra yang terlalu kecil dapat menyebabkan *model* CNN tidak dapat mempelajari fitur-fitur di dalam citra dengan baik. Oleh karena itu, ukuran citra yang digunakan sebagai *input* perlu diperhatikan. Beberapa arsitektur CNN yang terkenal menyarankan penggunaan ukuran citra (*shape*) sebesar 224x224x3 piksel (3 melambangkan *channel* RGB) (Winarto, 2021).

2. *Batch Size*

Pelatihan (*training*) data keseluruhan pada *dataset* yang besar, terutama pada data citra, akan menjadi sangat berat secara komputasi. Oleh karena itu, penggunaan *batch size* dapat membantu memecah *dataset* menjadi bagian-bagian kecil untuk mempercepat proses pelatihan. Terdapat beberapa nilai yang sering digunakan sebagai *batch size*, seperti 16, 32, 64, dan 128.

3. *epoch*

epoch adalah *hyperparameter* yang menentukan berapa kali algoritma pembelajaran akan bekerja mengolah *data training* keseluruhan (Nurani, Imsak , & Andriyan , 2022), Satu *epoch* merupakan satu putaran penuh pelatihan (*training*) dimana seluruh *dataset* telah digunakan. Tidak ada jawaban pasti terkait jumlah *epoch* yang optimal karena bergantung pada *dataset* yang digunakan. Terdapat beberapa jumlah *epoch* yang umum digunakan dalam beberapa literatur, yaitu jumlah *epoch* 10, 100, 500, 1000, dan lebih besar lagi.

4. *Optimizer*

Optimizer merupakan algoritma atau metode yang berfungsi untuk melakukan *update* pada bobot dengan tujuan menurunkan nilai *loss* dan meningkatkan *score* akurasi. Salah satu *optimizer* yang paling sering digunakan dan dianggap efektif oleh para peneliti adalah *Adam Optimizer*, karena dianggap cepat dalam mencapai nilai *loss minimum* (*converge*).

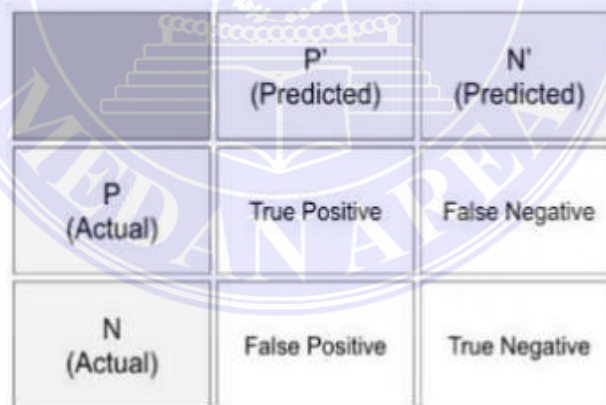
5. *Learning Rate*

Learning rate merupakan parameter yang mengatur kecepatan *model* mempelajari masalah pada saat pelatihan (*training*). Untuk nilai *learning rate* itu sendiri berada pada *range* nol (0) sampai satu (1) (Nurani, Imsak , & Andriyan , 2022) Jika *learning rate* terlalu rendah, *model* memerlukan waktu yang lama untuk mencapai titik optimal. Namun, jika terlalu tinggi, *model* mungkin melewati titik optimal tersebut. Terdapat beberapa nilai *learning rate* umum yang digunakan, seperti 0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, dan 0.3.

2.5 Confusion Matrix

Dalam menentukan baik tidaknya performa dari suatu *model* klasifikasi yang sudah dilatih (*training*) dapat dilihat dari pengukuran performa parameternya, yaitu *accuracy*, *recall*, *precision*, dan *f1-score*. Untuk menghitung faktor-faktor tersebut diperlukan sebuah matriks yang biasa disebut dengan *confusion matrix*. Proses evaluasi *model* dalam penelitian ini menggunakan *confusion matrix* untuk mengetahui persebaran hasil prediksi dibandingkan dengan label yang sebenarnya.

Confusion matrix adalah matriks perbandingan antara hasil prediksi dan nilai aktual, serta dapat dengan jelas menampilkan prediksi dan detail prediksi untuk setiap kategori klasifikasi (Faizin, 2022) Dalam *confusion matrix* terdapat beberapa jenis parameter yaitu *precision*, *recall*, dan *f1-score* (Ridhovan & Aries, 2022) Model *confusion matrix* untuk kasus klasifikasi dua kelas (*binary classification*) dapat dilihat pada Gambar 2.8.



	P' (Predicted)	N' (Predicted)
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

Gambar 2.8 Model Confusion Matrix
Sumber : (Hariz, Yulita, & Suryana, 2022)

Berdasarkan Gambar 2.8 terdapat beberapa nilai yang ada pada matriks yaitu “*True Positive (TP)*”, “*True Negative (TN)*”, “*False Positive (FP)*”, dan “*False Negative (FN)*”, seluruh kemungkinan kejadian sebenarnya positif (P) dan seluruh

kemungkinan kejadian sebenarnya negatif (N). Berdasarkan kombinasi dari nilai yang ada pada tabel *confusion matrix*, dapat dihitung beberapa nilai evaluasi *model* (*performance metrics*) yang berbeda seperti akurasi (*accuracy*), *recall*, presisi (*precision*), dan *f1-score*. Parameter-parameter tersebut dapat dirumuskan sebagai berikut:

1. *Accuracy*

Accuracy adalah rasio dari prediksi yang benar dari keseluruhan data (Nisa, Eva , & Hendra, 2020) Akurasi digunakan sebagai parameter untuk mengetahui akurat atau tidaknya suatu *model* dalam melakukan klasifikasi. Nilai akurasi dapat dihitung dengan persamaan (2.4).

$$\text{Accuracy} = \frac{\text{TP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (2.4)$$

2. *Precision*

Precision atau *positive predictive value* adalah rasio sampel yang benar diprediksi sebagai positif terhadap seluruh sampel yang diprediksi positif (Hariz, Yulita, & Suryana, 2022) jadi *precision* digunakan untuk menggambarkan seberapa tepat suatu *model* memprediksi kejadian positif dalam serangkaian kegiatan prediksi. Nilai presisi dapat dihitung dengan persamaan (2.5).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.5)$$

3. *Recall*

Recall yang juga dikenal dengan *sensitivity* atau *true positive rate* adalah rasio sampel yang benar diprediksi sebagai positif terhadap seluruh sampel

positif (Hariz, Yulita, & Suryana, 2022) Recall dapat dihitung dengan persamaan (2.6).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.6)$$

4. *F1-score*

F1-score merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan (Shintaa, Muhammad, Febi, & Suwanto, 2023). *F1-score* memperhitungkan nilai *precision* dan *recall* sekaligus, sehingga dapat mengukur performa *model* lebih akurat untuk kasus *imbalance class* dibandingkan dengan *accuracy* (Hariz, Yulita, & Suryana, 2022). Nilai *F1-score* dapat dihitung dengan persamaan (2.7).

$$\text{F1score} = 2 \frac{\text{Recall} * \text{Precison}}{\text{Recall} + \text{Precison}} \quad (2.7)$$

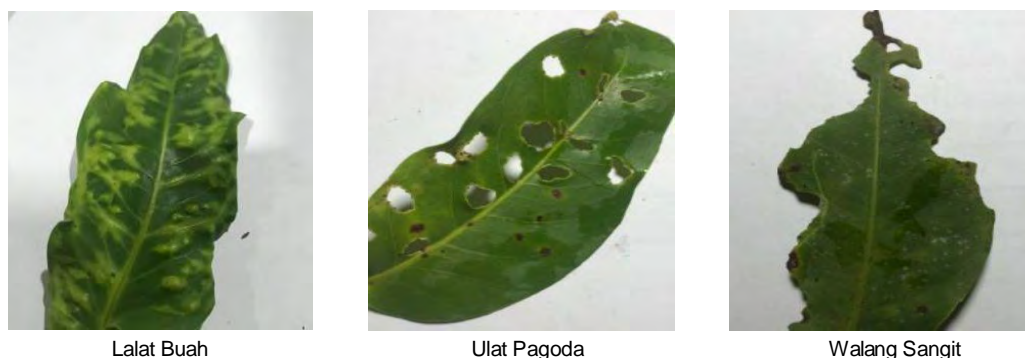
2.6 Hama Daun Jambu Madu

Jambu air merupakan salah satu tanaman yang memiliki banyak jenis, seperti jambu air madu deli, jambu air jamaika, dan jambu air citra (Daniel, 2022). Jambu air Madu Deli Hijau adalah salah satu kultivar unggul yang di introduksi dari Taiwan bernama *Jade Rose Apple* (Erfan, Tengku, Koko, & Martin, 2020) Jambu air madu Deli Hijau menjadi salah satu komoditi unggulan terbaru yang mulai banyak dikembangkan oleh petani hortikultura di daerah kota Binjai. Jambu ini berasal dari kelurahan Paya Roba, Kecamatan Binjai Barat, Kota Binjai, Provinsi Sumatera Utara (Dewi Br Siagian, 2023)

Dalam proses budidaya tanaman jambu air, terkadang ditemui hambatan dalam perawatannya. Salah satunya adanya serangan hama pada daun. Keberadaan hama dapat menyebabkan kerugian karena dapat merusak tanaman dan menurunkan kualitas panen. Hama pada daun jambu madu memiliki karakteristik yang berbeda-beda dan dapat dibedakan dari kerusakan pada daunnya. Serangan hama pada daun jambu madu dapat menurunkan produksi hasil panen dan bahkan dapat menyebabkan gagal panen. Semakin besar persentase serangan hama maka produktivitas lahan akan semakin rendah.

Klasifikasi hama pada daun jambu madu sangat penting dilakukan untuk mengetahui perawatan yang diperlukan dalam keberlanjutan sistem budidaya sehingga dapat mencegah turunnya hasil panen dan kualitas tanaman. Klasifikasi merupakan suatu proses untuk menemukan sekumpulan *model* maupun fungsi yang menjelaskan dan membedakan data ke dalam kelas-kelas tertentu, dengan tujuan menggunakan *model* tersebut dalam menentukan kelas dari suatu objek yang belum diketahui kelasnya. Ada 2 proses dalam klasifikasi, yaitu proses learning/training. Melakukan pembangunan *model* menggunakan data training. Proses testing melakukan tes terhadap data testing menggunakan *model* yang telah diperoleh dari proses training (Qudsiah , 2023).

Ada beberapa jenis hama yang umum menyerang tanaman jambu madu, diantaranya adalah hama lalat buah, hama ulat pagoda, dan hama walang sangit.



Gambar 2.9 Jenis Hama Daun Jambu Madu

Dalam beberapa tahun terakhir, perkembangan teknologi *Deep Learning* telah menunjukkan potensi yang besar dalam mengatasi masalah klasifikasi visual, termasuk dalam bidang pertanian. Salah satu arsitektur *Deep Learning* yang telah terbukti efektif dalam tugas-tugas klasifikasi gambar adalah *MobileNetV2*. Dengan menggunakan arsitektur *MobileNetV2*, klasifikasi jenis hama pada daun jambu madu dapat dilakukan secara otomatis dan akurat sehingga dapat membantu petani dalam mengidentifikasi jenis hama untuk dilakukan tindakan lebih lanjut.

2.7 Penelitian Terdahulu

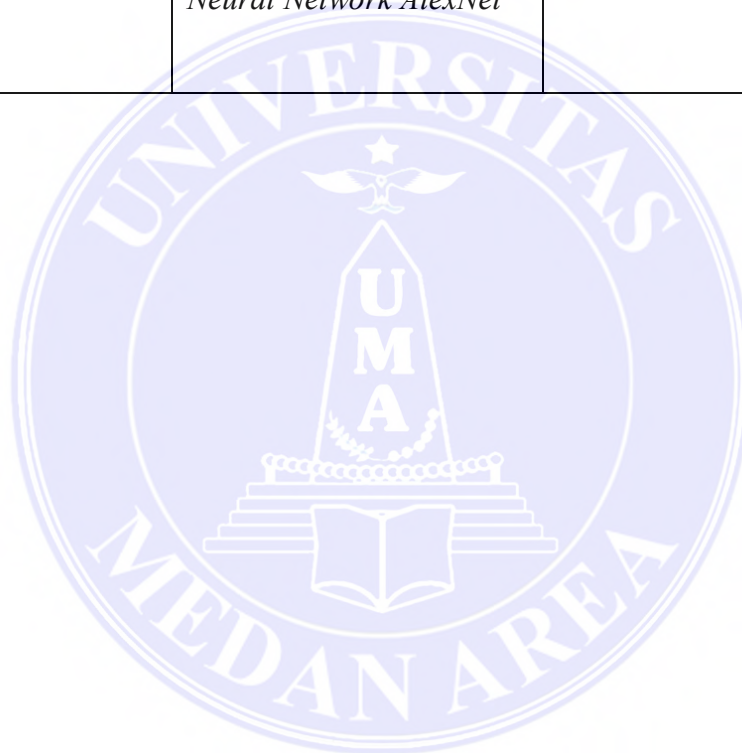
Dalam penelitian ini, penulis mengambil inspirasi dan merujuk pada beberapa penelitian sebelumnya yang erat kaitannya dengan latar belakang masalah. Referensi dari penelitian sebelumnya sangatlah penting untuk menghindari plagiat atau pengulangan penelitian yang sudah ada sebelumnya. Berikut adalah beberapa ulasan tentang penelitian sebelumnya yang pernah dilakukan dan menjadi rujukan untuk menyusun penelitian ini. Secara garis besar hasil penelitian terdahulu dapat disajikan pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No.	Penulis	Judul Penelitian	Persamaan	Perbedaan
1.	Muhammad Ryan Syah Erwin, Aji Gautama Putrada, Muhammad Agus Triawan, 2021	Deteksi Hama Ulat Pada Tanaman Selada Berbasis <i>Aquaponic</i> Menggunakan CNN (<i>Convolutional Neural Network</i>)	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan
2.	V. Malathi & M. P. Gopinath, 2021	<i>Classification of Pest Detection in Paddy Crop Based on Transfer Learning Approach</i>	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan
3.	Susi Yuliany, Aradea, Andi Nur Rachman, 2022	Implementasi <i>Deep Learning</i> pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode <i>Convolutional Neural Network (CNN)</i>	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan
4.	Andreanov Ridhovan, Aries Suharso, 2022	Penerapan Metode <i>Residual Network (ResNet)</i> Dalam Klasifikasi Penyakit Pada Daun Gandum	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan
5.	Afis Julianto, Andi	Optimasi <i>Hyperparameter Convolutional Neural</i>	Arsitektur CNN yang	Objek yang diklasifikasi

	Sunyoto, Ferry Wahyu Wibowo, 2022	<i>Network Untuk Klasifikasi Penyakit Tanaman Padi</i>	digunakan	dan arsitektur CNN yang digunakan
6.	Joelyan Vicky, Frisca Ayu, Bagas Julianto, 2023	Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan
7.	Angga Prima Syahputra, Alda Cendekia Siregar, Rachmat Wahid Saleh Insani, 2023	<i>Comparison of CNN models With Transfer Learning in the Classification of Insect Pests</i>	Arsitektur CNN yang digunakan	Objek yang diklasifikasi
8.	Dwiretno Istiyadi Swasono, Mohammad Abuemas Rizq Wijaya, Muhamad Arief Hidayat, 2023	Klasifikasi Penyakit pada Citra Buah Jeruk Menggunakan <i>Convolutional Neural Networks (CNN) dengan Arsitektur AlexNet</i>	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan
9.	Rahma Shintaa, Jasrilb, Muhammad	Klasifikasi Citra Penyakit Daun Tanaman Padi	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur

	Irsyadc, Febi Yantod, Suwanto Sanjaya, 2023	Menggunakan CNN dengan Arsitektur <i>VGG-19</i>		CNN yang digunakan
10.	Qudsiah Nur Azizah, 2023	Klasifikasi Penyakit Daun Jagung Menggunakan Metode <i>Convolutional Neural Network AlexNet</i>	Metode CNN yang digunakan	Objek yang diklasifikasi dan arsitektur CNN yang digunakan



BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Lokasi Penelitian

Penelitian ini dilaksanakan pada semester genap tahun akademik 2022/2023. Lokasi penelitian berada di salah satu petani jambu madu di daerah kota Binjai kelurahan Paya Roba, Kecamatan Binjai Barat, Kota Binjai, Provinsi Sumatera Utara.

3.2 Alat dan Bahan Penelitian

Pada bagian sub-bab ini menguraikan secara ringkas mengenai alat dan bahan yang digunakan selama penelitian berlangsung. Alat yang digunakan dalam penelitian ini adalah satu unit smartphone dan seperangkat laptop yang dilengkapi dengan sistem operasi, perangkat keras (*hardware*), dan perangkat lunak (*software*). Adapun spesifikasi kebutuhan perangkat keras (*hardware*) yang digunakan dalam penelitian ini yaitu:

1. *Processor Intel(R) Core(TM) i3-6006U.*
2. *CPU @ 2.00GHz 1.99 GHz.*
3. *Random Access Memory (RAM) 4 GB DDR3.*
4. *Hard Disk Internal 1 TB.*
5. *Monitor 15.6 LED.*
6. *Mouse dan Keyboard.*

Sedangkan spesifikasi kebutuhan perangkat lunak (*software*) yang digunakan dalam penelitian ini, yaitu:



1. *Microsoft Windows 10 tipe 64-bit* sebagai sistem operasi.
2. *Python* versi 3.10.11 sebagai bahasa pemrograman.

3. *Sublime Text3* sebagai teks editor untuk menulis coding program.
4. *Git Bush* sebagai terminal atau *Command-Line Interface (CLI)* untuk menjalankan *coding* program.
5. *Microsoft Office 2010* untuk membuat dokumentasi laporan penelitian.


Adapun bahan yang digunakan dalam penelitian ini mencakup *dataset* berupa gambar daun jambu madu yang terserang hama lalat buah, ulat pagoda, dan walang sangit. Jenis data yang digunakan dalam penelitian ini adalah merupakan data sekunder dalam bentuk citra yang diperoleh dari salah satu petani jambu madu di daerah kota Binjai kelurahan Paya Roba. *dataset* yang dikumpulkan sebanyak 2400 gambar hama daun jambu madu dengan format .jpg/jpeg yang diambil dengan menggunakan kamera smartphone. Sebanyak 2400 data citra daun jambu madu tersebut terbagi kedalam tiga kategori berdasarkan jenis hama, yaitu masingmasing sebanyak 800 citra per kategori.

Adapun contoh dari sampel data yang digunakan pada penelitian ini dapat disajikan pada Tabel 3.1.

Tabel 3.1 Sampel Data Penelitian

Citra Hama	Nama Label	Jumlah Data
	Lalat Buah	800
	Ulat Pagoda	800

Tabel 3.1 Sampel Data Penelitian (lanjutan)

Citra Hama	Nama Label	Jumlah Data
	Walang Sangit	800

Selain itu bahan pendukung lainnya yang digunakan dalam penelitian ini meliputi literatur pendukung dan hasil penelitian terdahulu yang relevan dengan pokok permasalahan dalam penelitian ini.

3.3 Tahapan Penelitian

Tahapan penelitian ini bertujuan untuk memberikan ketentuan bentuk masalah dan tujuan sebagai pedoman dalam penelitian agar lebih terarah dan sesuai dengan tujuan yang ingin dicapai. Secara garis besar tahapan-tahapan yang dilakukan dalam penelitian ini dapat ditampilkan pada Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Berdasarkan diagram alur tahapan penelitian pada Gambar 3.1, maka dapat diuraikan penjabaran dari setiap proses pada masing-masing tahapan yaitu:

1. Identifikasi Permasalahan

Pada tahap ini dilakukan identifikasi permasalahan yang akan dijadikan sebagai objek penelitian. Identifikasi permasalahan dilakukan untuk menentukan permasalahan apa yang akan diangkat dalam penelitian, lalu dilakukan perumusan terhadap masalah yang dihadapi untuk menentukan tujuan, manfaat dan batasan dari penelitian.

2. Studi Literatur

Pada tahap ini dilakukan untuk memahami dasar teori mengenai informasi yang berkaitan dengan permasalahan dalam penelitian yang akan dilakukan dengan mencari berbagai sumber-sumber pendukung, berupa buku, paper ilmiah, dan jurnal hasil penelitian terdahulu. Beberapa hal yang dipelajari terkait penelitian ini diantaranya adalah mengenai hama daun jambu madu, *Deep Learning*, metode CNN, dan arsitektur *MobileNetV2*.

3. Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan data sekunder berupa data dalam bentuk citra yang diperoleh dari salah satu petani jambu madu di daerah kota Binjai kelurahan Paya Roba. *dataset* yang dikumpulkan sebanyak 2400 gambar hama daun jambu madu dengan format *.jpg/jpeg* yang diambil dengan menggunakan kamera smartphone.

4. Pengolahan Data

Setelah data yang dikumpulkan sudah mencukupi maka tahap selanjutnya akan dilakukan pengolahan data. Pengolahan data yang dilakukan mencakup

data cleaning untuk menghilangkan data yang berkualitas rendah (citra *blur* dan citra *noise*), melakukan *cropping*, pelabelan data, pembagian data, dan terakhir augmentasi data.

5. Penerapan Metode CNN

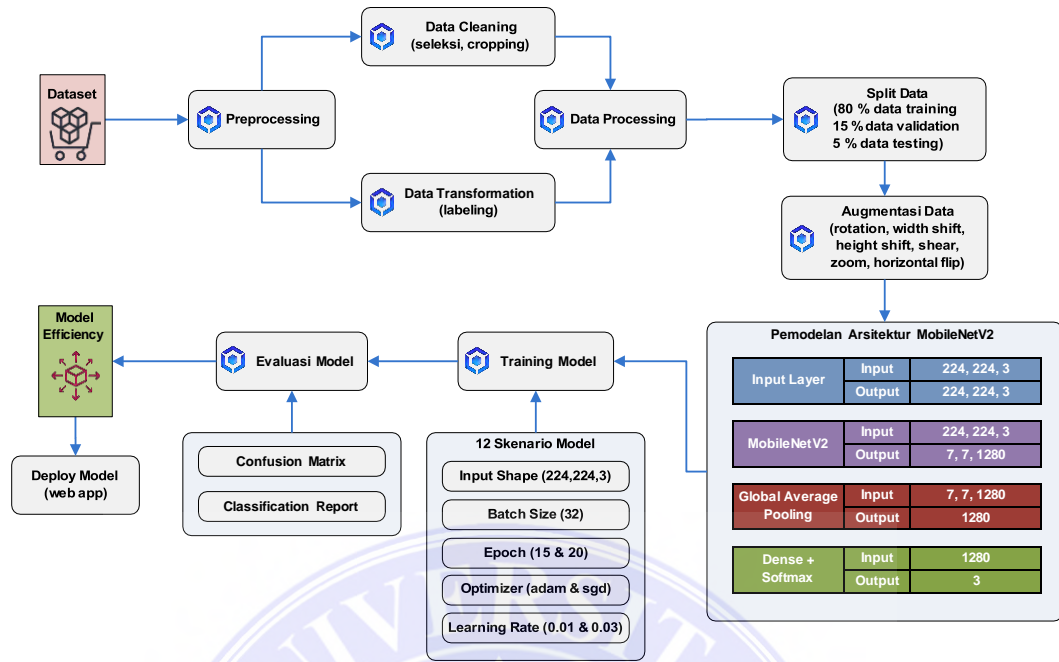
Pada tahap ini akan dilakukan penerapan metode CNN *transfer learning* dengan arsitektur *MobileNetV2* dalam pembuatan *model* yang siap pakai (*pre-trained model*) untuk melakukan klasifikasi jenis hama pada daun jambu madu. Terdapat 12 skenario *model* yang akan diuji untuk mengetahui dan mendapatkan *model* dengan performa terbaik. *model* yang terpilih kemudian akan dievaluasi dengan menggunakan *confusion matrix* untuk mendapatkan nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

6. Penyusunan Laporan

Tahapan terakhir adalah pembuatan laporan sebagai bentuk dokumentasi atas terlaksananya penelitian ini.

3.4 Prosedur Kerja

Penelitian ini menggunakan metode CNN dengan arsitektur *MobileNetV2* untuk mengklasifikasi jenis hama pada daun jambu madu, dengan menerapkan sejumlah pengujian skenario *model* untuk mengetahui dan mendapatkan *model* dengan performa terbaik. Secara garis besar prosedur kerja yang dilakukan dalam mengklasifikasi jenis hama pada daun jambu madu menggunakan metode CNN dengan arsitektur *MobileNetV2* dapat ditampilkan dalam bentuk diagram seperti ditampilkan pada Gambar 3.2.



Gambar 3.2 Diagram Umum Prosedur Kerja Sistem

Secara keseluruhan, prosedur kerja dalam mengklasifikasi hama daun jambu madu dengan menerapkan metode CNN menggunakan arsitektur *MobileNetV2* dapat dijelaskan, yaitu sebagai berikut:

1. *Dataset*

Seperti yang sudah dijelaskan sebelumnya, *dataset* yang digunakan dalam penelitian ini sebanyak 2400 citra daun jambu madu yang terserang hama, terdiri dari 800 citra daun hama lalat buah, 800 citra daun hama ulat pagoda, dan 800 citra daun hama walang sangit.

2. *Preprocessing*

Preprocessing atau pra-pemrosesan pada *dataset* bertujuan agar pada pembuatan *model* CNN dengan arsitektur *MobileNetV2* memiliki performa yang terbaik. Pada *preprocessing* terdapat dua tahapan yang dilakukan, yaitu *data cleaning* dan *data transformation*.

a. *Data Cleaning*

Pada tahap *data cleaning*, *dataset* akan diseleksi antara lain dengan menghilangkan data yang memiliki kualitas rendah (terdapat *blur* dan *noise*). *dataset* yang sudah diseleksi kemudian akan di *cropping*. Proses *cropping* menjadi hal yang cukup penting, karena jika tidak dilakukan maka semua objek yang ada pada citra akan dianalisis oleh komputer, padahal fokus dari objek yang akan diteliti adalah adalah berupa fitur daun yang ada pada citra hama daun jambu.

b. *Data Transformation*

dataset kemudian akan melalui proses transformasi dengan cara menamai ulang (*labeling*) sesuai dengan nama kelas (hama lalat buah, hama ulat pagoda, dan hama walang sangit) dengan pola yang sudah ditentukan.

3. *Split Data*

Dataset yang sudah melalui proses *cleaning* dan *transformation* selanjutnya akan dibagi (*split data*). Data akan dibagi menjadi 3 kategori, yaitu data latih (*data training*), data validasi (*data validation*), dan data uji (*data testing*). Data latih akan digunakan untuk melatih *model*, data validasi digunakan untuk mengoptimasi saat melatih *model*. Sedangkan data uji digunakan untuk menguji kinerja *model* yang dihasilkan untuk dievaluasi. Jumlah data latih adalah sebesar 80% dari keseluruhan *dataset*, data validasi berjumlah 15% sedangkan sisanya 5% digunakan untuk data uji.

4. Augmentasi Data

Augmentasi data dilakukan dengan cara normalisasi nilai piksel citra, pemutaran (*rotation*) pada citra, pergeseran pada sudut citra (*width shift* dan *height shift*), memiringkan posisi citra (*shear*), memperbesar (*zoom*) pada citra, dan pencerminan (*flipping*) secara horizontal. Tujuan dari augmentasi data dalam penelitian ini dilakukan untuk meningkatkan keberagaman data pelatihan dan mengurangi *overfitting* (memiliki kinerja baik selama pelatihan, tetapi buruk pada data baru atau data pengujian).

5. Membangun Model

Pada tahap ini dilakukan perancangan model CNN dengan arsitektur *MobileNetV2* yang dimana menerapkan metode *transfer learning* dalam pembuatan model yang siap pakai (*pre-trained model*) untuk melakukan klasifikasi jenis hama pada daun jambu madu.

6. Training Model

Melakukan proses pembelajaran untuk melatih model menggunakan data latih (*data training*). Terdapat 12 skenario model yang akan di uji pada penelitian ini, dimana nantinya hasil dari masing-masing model akan di bandingkan dengan kondisi aktual. Dengan melihat keakurasian klasifikasi, maka model tersebut yang nantinya akan digunakan dalam klasifikasi hama pada daun jambu madu. Pengujian model menggunakan perbandingan *hyperparameter* berupa jumlah *epoch* (15 dan 20), *optimizer* (*adam* dan *sgd*) serta *learning rate* (0.01 dan 0.03). Sedangkan parameter dari *input shape* : 224x224x3, dan *batch size* : 32 akan dijadikan statis pada setiap skenario model yang diuji.

7. Evaluasi *Model*

Mengevaluasi kinerja *model* menggunakan data uji yang independen. Mengukur metrik evaluasi seperti akurasi, *precision*, *recall*, dan *f1-score* dengan menggunakan *confusion matrix*, untuk mengevaluasi seberapa baik *model* dalam mengklasifikasi hama pada daun jambu madu. *Model* yang dihasilkan akan diuji menggunakan data testing untuk mengevaluasi hasilnya lalu memilih *model* terbaik dengan akurasi yang paling tinggi, dan tahap yang terakhir adalah men-*deploy model* kedalam aplikasi siap pakai.

8. *Deploy Model*

Pada tahap *deploy* dilakukan pengembangan *model* kedalam sebuah aplikasi, guna memudahkan pengguna (*user*) dalam melakukan klasifikasi hama daun jambu madu. *model* yang akan di *deploy* adalah *model* dengan hasil performa terbaik dalam proses *training* sebelumnya sebagai *engine* dalam menentukan hasil klasifikasi hama pada daun jambu madu. *model* akan di *deploy* kedalam aplikasi berbasis *web* dengan menggunakan *Flask* sebagai *library* utama.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil analisis dan pengujian sistem dalam melakukan klasifikasi hama pada daun jambu madu dengan menggunakan arsitektur *MobileNetV2* adalah sebagai berikut:

1. Penerapan metode *Convolutional Neural Network* dengan arsitektur *MobileNetV2* menghasilkan performa yang baik terhadap semua *dataset* sehingga dapat menghasilkan klasifikasi jenis hama pada daun jambu madu secara akurat.
2. Berdasarkan hasil *training* pada 12 skenario *model* yang diuji, diperoleh *model* terbaik pada skenario *model* 9 dengan tingkat akurasi 0.9972% dan loss 0.0169. Skenario *model* menggunakan *hyperparameter* dengan jumlah *epoch* 20, *batch size* 32, *optimizer Adam*, dan *learning rate* 0.01.
3. Hasil akurasi dari *model* setelah diuji menggunakan *data testing* (40 citra per *class*) didapatkan hasil akurasi sebesar 100%, nilai presisi (*precision*) sebesar 100%, *recall* 100%, dan *f1-score* 100% pada masing-masing *class*.

5.2 Saran

Saran yang dapat penulis berikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Melakukan *fine-tuning* atau melakukan eksperimen dengan memodifikasi *hyperparameter* selain jumlah *epoch*, *batch size*, *optimizer* dan nilai *learning rate* terhadap arsitektur yang digunakan.

2. Penelitian selanjutnya dapat mengembangkan (*deploy*) *model* kedalam aplikasi berbasis *mobile* dan proses klasifikasi dapat dilakukan secara *real time*.



DAFTAR PUSTAKA

- ADILAH , T. (2021). IMPLEMENTASI TRANSFER LEARNING ARSITEKTUR MOBILENETV2 UNTUK IDENTIFIKASI PENYAKIT DAUN KENTANG.
- Al-Rammahi, I. A. (2022). Face mask recognition system using MobileNetV2 with optimization function.
- Ayu, T., Vizza, D., & Agus, E. M. (2021). PENDIAGNOSA DAUN MANGGA DENGAN MODEL CONVOLUTIONAL NEURAL NETWORK.
- Faizin, A. (2022). DEEP PRE-TRAINED MODEL MENGGUNAKAN ARSITEKTUR DENSENET UNTUK IDENTIFIKASI PENYAKIT DAUN PADI .
- Fuadi, A., & Suharso, A. (2022). PERBANDINGAN ARSITEKTUR MOBILENET DAN NASNETMOBILE UNTUK KLASIFIKASI PENYAKIT PADA CITRA DAUN KENTANG.
- Hastomo, W., Sugiyanto , & Sudjiran . (2021). CONVOLUTION NEURAL NETWORK ARSITEKTUR MOBILENET-V2 UNTUK MENDETEKSI TUMOR OTAK.
- Intan , N. Y., & Ino , S. (2022). Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet.
- Lu , J., Lijuan , T., & Huanyu , J. (2021). Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification.
- Miftahuddin, Y., & Zaelani S, F. (2022). Perbandingan Metode Efficientnet-B3 dan Mobilenet-V2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun.
- Nisa, C., Eva , Y. P., & Hendra, M. (2020). Penerapan Metode Convolutional Neural Network untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data.
- Putra , S. V. (2023). Perbandingan Tingkat Akurasi Arsitektur Convolutional Neural Network untuk Model Deteksi Penggunaan Masker Secara Otomatis.
- Ridhovan, A., & Aries , S. (2022). PENERAPAN METODE RESIDUAL NETWORK (RESNET) DALAM KLASIFIKASI PENYAKIT PADA DAUN GANDUM.
- Syahputra, P. A., Siregar, C. A., & Insani , S. R. (2023). Comparison of CNN Models With Transfer Learning in the Classification of Insect Pests.
- Annur , F. I., Jumhurul , J., Annafii, N. M., Trisnaningrum, N., & Putra , V. O. (2023). Klasifikasi Tingkat Keparahan Penyakit Leafblast Tanaman Padi Menggunakan MobileNetv2.


- Arif , F., Arief , T. A., Moch, L., & Abdur , R. M. (2022). DEEP PRE-TRAINED MODEL MENGGUNAKAN ARSITEKTUR DENSENET UNTUK IDENTIFIKASI PENYAKIT DAUN PADI.
- Arif Faizin, Arief , T., & Lutfi, A. R. (2022). DEEP PRE-TRAINED MODEL MENGGUNAKAN ARSITEKTUR DENSENET UNTUK IDENTIFIKASI PENYAKIT DAUN PADI.
- Arif Faizin, Arief Tri Arsanto, & Lutfi, Abdur Rochim Musa . (2022). DEEP PRE-TRAINED MODEL MENGGUNAKAN ARSITEKTUR DENSENET UNTUK IDENTIFIKASI PENYAKIT DAUN PADI.
- Daniel , W. (2022). Identifikasi Jenis Jambu Air Berdasarkan Bentuk Daun Menggunakan Jaringan Syaraf Tiruan Backpropagation dengan Fitur Histogram of Oriented Gradient.
- Desi Ramayanti, S. L. (2022). Implementasi Model Arsitektur VGG16 dan MobileNetV2 Untuk Klasifikasi Citra Kupu-Kupu .
- Dewi Br Siagian, R. A. (2023). Respon pertumbuhan tanaman jambu air madu (*Syzygium aqueum*) dengan beberapa taraf pemberian air dan Pupuk kompos kotoran ayam pada tanah Ultisol.
- Dino , H. P., & Sulistyowati, V. Y. (2023). Perbandingan Tingkat Akurasi Arsitektur Convolutional Neural Network untuk Model Deteksi Penggunaan Masker Secara Otomatis.
- Dino , H. P., & Sulistyowati, , V. Y. (2022). Perbandingan Tingkat Akurasi Arsitektur Convolutional Neural Network untuk Model Deteksi Penggunaan Masker Secara Otomatis.
- Dino Hariatma Putra, S. V. (2023). Perbandingan Tingkat Akurasi Arsitektur Convolutional Neural Network untuk Model Deteksi Penggunaan Masker Secara Otomatis.
- Dwi , N., Imsak , L. Y., & Andriyan , D. P. (2022). KLASIFIKASI JENIS PENYAKIT PADA CITRA DAUN PADI MENGGUNAKAN ALGORITMA CONVOLUTION NEURAL NETWORK.
- Dwi , S. A., Arif , W., Putri, D. R., & Chaerur , R. (2022). METODE ALGORITMA CONVOLUTIONAL NEURAL NETWORK PADA KLASIFIKASI PENYAKIT TANAMAN CABAI.
- Dwiretno , I. S., Mohammad , A. R., & Muhamad , A. H. (2023). Klasifikasi Penyakit pada Citra Buah Jeruk Menggunakan Convolutional Neural Networks (CNN) dengan Arsitektur Alexnet.
- Erfan , W., Tengku , B. H., Koko , T. R., & Martin , H. P. (2020). KARAKTER PERTUMBUHAN DAN PRODUKSI TANAMAN JAMBU AIR MADU DELI

HIJAU (*Syzygium samarangense*) PADA APLIKASI PUPUK KANDANG
KAMBING DAN PUPUK ORGANIK CAIR NASA.

- Fakhri , H. H., Faslah , F., Muhamad , R. A., & Toni , A. (2022). KLASIFIKASI PENYAKIT PADI MENGGUNAKAN ALGORITMA CNN (CONVOLUTIONAL NEURAL NETWORK).
- Fauzan , A. H., Intan , N. Y., & Ino Suryana. (2022). Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet.
- Fauzan , A. H., Intan, N. Y., & Ino , S. (2022).
- Fauzan , A. H., Intan, N. Y., & Ino , S. (2022).
- Ferry , A. I., Made , S., & Duman , C. (2021). RANCANG BANGUN APLIKASI IDENTIFIKASI PENYAKIT TANAMAN PEPAYA CALIFORNIA BERBASIS ANDROID MENGGUNAKAN METODE CNN MODEL ARSITEKTUR SQUEEZENET.
- Gulzar. (2023). Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique.
- Hariz, A. F., Yulita, N. I., & Suryana, I. (2022). Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet.
- Jin Zhu , L., Lijuan , T., & Huanyu , J. (2021). Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification.
- Julianto, A. (2021). OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI.
- Muhammad , R. E., Aji , G. P., & Muhammad , A. T. (2021). Deteksi Hama Ulat Pada Tanaman Selada Berbasis Aquaponic Menggunakan CNN (Convolutional Neural Network).
- Nurani, D., Imsak , L. Y., & Andriyan , D. (2022). KLASIFIKASI JENIS PENYAKIT PADA CITRA DAUN PADI MENGGUNAKAN ALGORITMA CONVOLUTION NEURAL NETWORK.
- Qudsiah , N. A. (2023). Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet.
- Shintaa, R. J., Muhammad, I., Febi , Y., & Suwanto, S. (2023). Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19.
- Susi , Y., Aradea, & Andi , N. R. (2022). Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN).

- SYARIF , K. A. (2021). SISTEM KLASIFIKASI PENYAKIT TANAMAN CABAI.
- TIKA , A. (2021). IMPLEMENTASI TRANSFER LEARNING ARSITEKTUR MOBILENETV2 UNTUK IDENTIFIKASI PENYAKIT DAUN KENTANG.
- Tika, A. m. (2021). IMPLEMENTASI TRANSFER LEARNING ARSITEKTUR MOBILENETV2 UNTUK IDENTIFIKASI PENYAKIT DAUN KENTANG.
- Tsabitah , A., Vizza, D., & Agus , E. M. (2021). PENDIAGNOSA DAUN MANGGA DENGAN MODEL CONVOLUTIONAL NEURAL NETWORK.
- Tsabitah Ayu, Vizza Dwi, & Agus Eko Minarno . (2021). PENDIAGNOSA DAUN MANGGA DENGAN MODEL CONVOLUTIONAL NEURAL NETWORK.
- Tsabitah, A., Vizza, D., & Agus , E. M. (2021). PENDIAGNOSA DAUN MANGGA DENGAN MODEL CONVOLUTIONAL NEURAL NETWORK.
- Tsabitah, A., Vizza, D., & Agus, E. M. (2021). PENDIAGNOSA DAUN MANGGA DENGAN MODEL CONVOLUTIONAL NEURAL NETWORK.
- Wang, W., & oLi , Y. (2020). A Novel Image Classification Approach via Dense-MobileNet Models.
- wharyanto. (2021). A Novel Image Classification Approach via Dense-MobileNet Models.
- Winarto, N. T. (2021). PENERAPAN ARSITEKTUR MOBILENETV2 PADA KLASIFIKASI PENYAKIT DAUN TEH.
- WINNARTO , M. (2021). PENERAPAN ARSITEKTUR MOBILENETV2 PADA KLASIFIKASI PENYAKIT DAUN TEH.
- Yusup Miftahuddin, F. (2022). Perbandingan Metode Efficientnet-B3 dan Mobilenet-V2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun.

Lampiran 1. Hasil Plagiat

Similarity Report ID: oid:29477:46353478

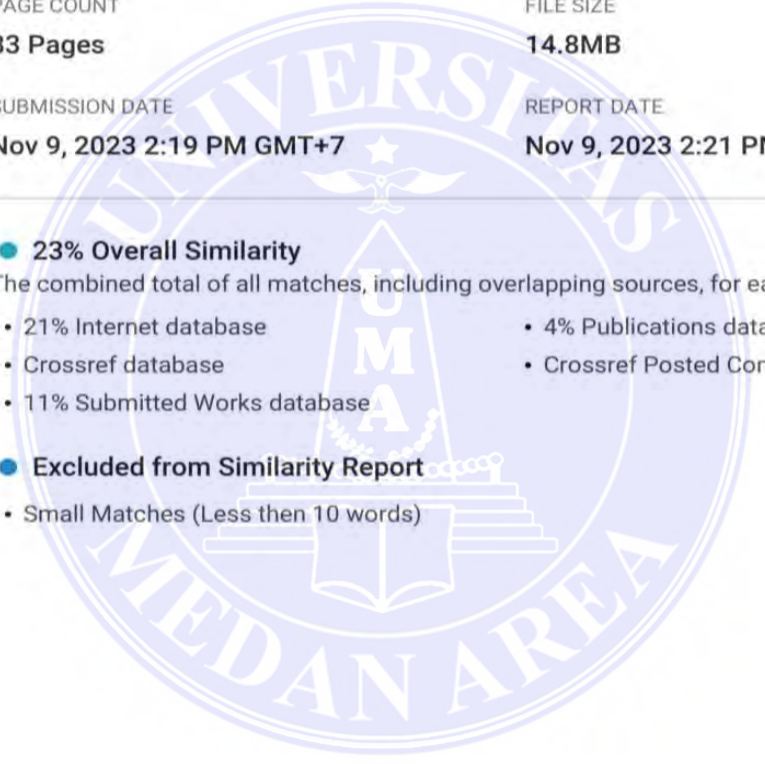
PAPER NAME	AUTHOR
Fitriyan syahputra turnitin 1.docx	Fitriansah Putra

WORD COUNT	CHARACTER COUNT
12422 Words	77245 Characters

PAGE COUNT	FILE SIZE
83 Pages	14.8MB

SUBMISSION DATE	REPORT DATE
Nov 9, 2023 2:19 PM GMT+7	Nov 9, 2023 2:21 PM GMT+7

- **23% Overall Similarity**
The combined total of all matches, including overlapping sources, for each database.
 - 21% Internet database
 - Crossref database
 - 11% Submitted Works database
 - 4% Publications database
 - Crossref Posted Content database
- **Excluded from Similarity Report**
 - Small Matches (Less then 10 words)

1Summary

Lampiran 2,Source Code

SOURCE CODE ANALISIS ARSITEKTUR DEEP LEARNING MOBILENET DALAM MENGLASIFIKASI HAMA DAUN JAMBU MADU.

#1. IMPORT LIBRARY

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import tensorflow as tf
```

```
from tensorflow.keras import Sequential
```

```
from tensorflow.keras.layers import Flatten, Dense, GlobalAveragePooling2D
```

```
from tensorflow.keras.optimizers import Adam, SGD
```

```
from tensorflow.keras.preprocessing import image_dataset_from_directory
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
from tensorflow.keras.callbacks import EarlyStopping
```

```
from tensorflow.keras import metrics
```

```
from tensorflow.keras.models import load_model
```

```
from sklearn.metrics import confusion_matrix, classification_report,  
accuracy_score
```

```
import seaborn as sns
```

```
from IPython.display import clear_output
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
print('Library Loaded...!')  
print('Versi Tensorflow :', tf.__version__)
```

#2. PRAPROSES DATASET

```
# Load Dataset dari Google Drive  
from google.colab import drive  
drive.mount('/content/drive')  
# menentukan direktori dataset (data training, data validation, dan data testing)  
train_dir = '/content/drive/MyDrive/My Dataset/hama-daun-jambu/train'  
val_dir = '/content/drive/MyDrive/My Dataset/hama-daun-jambu/val'  
test_dir = '/content/drive/MyDrive/My Dataset/hama-daun-jambu/test'  
# menampilkan jumlah dataset  
data_train = image_dataset_from_directory(train_dir)  
data_val = image_dataset_from_directory(val_dir)  
data_test = image_dataset_from_directory(test_dir)  
  
# menampilkan nama class  
nama_classes = data_train.class_names  
print("Nama Class :", nama_classes)  
# menampilkan visualisasi dataset  
classes_train = data_train.class_names  
plt.figure(figsize = (10,10))  
for img, label in data_train.take(1):
```

```
for i in range(10):  
    ax = plt.subplot(4,5,i+1)  
    plt.imshow(img[i].numpy().astype('uint8'))  
    plt.title(classes_train[int(label[i])])  
    plt.axis('off')  
  
clear_output()  
  
# Training & Validation generator + Augmentation  
train_datagen = ImageDataGenerator(rescale=1./255,           # Scaling pixel  
value  
rotation_range=20,           # Rotate image from 0 to 20  
degrees  
width_shift_range=0.2,  
height_shift_range=0.2,  
shear_range=0.2,  
zoom_range=0.2,           # Zoom in or zoom out range  
horizontal_flip=True,           # Flip image horizontally  
fill_mode='nearest')  
  
train_generator = train_datagen.flow_from_directory(train_dir,  
target_size=(224,224),  
color_mode='rgb',  
shuffle=True,  
class_mode='categorical')  
  
val_generator = train_datagen.flow_from_directory(val_dir,  
target_size=(224,224),
```

```
color_mode = 'rgb',  
shuffle = True,  
class_mode = 'categorical')
```

```
# Test Generator - Augmentation
```

```
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
test_generator = test_datagen.flow_from_directory(test_dir,  
target_size=(224,224),  
color_mode='rgb',  
batch_size=32,  
# shuffle=True,  
class_mode='categorical')
```

#3. TRANSFER LEARNING MOBILENETV2

3.1. Base Model MobileNetV2

```
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2
```

```
base_model_mobilenet =  
MobileNetV2(weights='imagenet', include_top=False, input_shape=(224,224,3))
```

```
# pre trained model MobileNetV2 dibuat jadi non trainable
```

```
for layer in base_model_mobilenet.layers
```

```
    layer.trainable = False
```


#3.2. Build Model MobileNetV2 Original

```
# Create model

# Inisialisai model jadi sequential

model_mobilenetV2_ori = Sequential()

# Feature Extractor Layer

model_mobilenetV2_ori.add(base_model_mobilenet) # add the convolutional
base model MobileNetV2

model_mobilenetV2_ori.add(GlobalAveragePooling2D())

# output layer

model_mobilenetV2_ori.add(Dense(3,activation='softmax'))

# print model summary

model_mobilenetV2_ori.summary()
```

#3. PERBANDINGAN AKURASI MODEL

```
al_acc_skenario1 = hist_mobilenetV2_ori_skenario1.history['val_accuracy']
val_acc_skenario2 = hist_mobilenetV2_ori_skenario2.history['val_accuracy']
val_acc_skenario3 = hist_mobilenetV2_ori_skenario3.history['val_accuracy']
```

```
val_acc_skenario4 = hist_mobilenetV2_ori_skenario4.history['val_accuracy']  
val_acc_skenario5 = hist_mobilenetV2_ori_skenario5.history['val_accuracy']  
val_acc_skenario6 = hist_mobilenetV2_ori_skenario6.history['val_accuracy']  
val_acc_skenario7 = hist_mobilenetV2_ori_skenario7.history['val_accuracy']  
val_acc_skenario8 = hist_mobilenetV2_ori_skenario8.history['val_accuracy']  
val_acc_skenario9 = hist_mobilenetV2_ori_skenario9.history['val_accuracy']  
val_acc_skenario10 = hist_mobilenetV2_ori_skenario10.history['val_accuracy']  
val_acc_skenario11 = hist_mobilenetV2_ori_skenario11.history['val_accuracy']  
val_acc_skenario12 = hist_mobilenetV2_ori_skenario12.history['val_accuracy']
```

```
plt.figure(figsize=(15, 5))  
plt.plot(val_acc_skenario1)  
plt.plot(val_acc_skenario2)  
plt.plot(val_acc_skenario3)  
plt.plot(val_acc_skenario4)  
plt.plot(val_acc_skenario5)  
plt.plot(val_acc_skenario6)  
plt.plot(val_acc_skenario7)  
plt.plot(val_acc_skenario8)  
plt.plot(val_acc_skenario9)  
plt.plot(val_acc_skenario10)  
plt.plot(val_acc_skenario11)  
plt.plot(val_acc_skenario12)
```

```
plt.title('Model Validation Accuracy')
```

```
plt.xlabel("Epoch")
```

```
plt.ylabel("Accuracy")

# 'Model Skenario 7','Model Skenario 8','Model Skenario 9','Model Skenario
10','Model Skenario 11','Model Skenario 12',

plt.legend(['Model Skenario 1','Model Skenario 2','Model Skenario 5','Model
Skenario 4','Model Skenario 5','Model Skenario 6',

'Model Skenario 7','Model Skenario 8','Model Skenario 9','Model Skenario
10','Model Skenario 11','Model Skenario 12'], loc='lower right')

plt.grid(True)

plt.show()
```

#4. EVALUASI MODEL (Confusion Matrix & Classification Report)

```
batch_size=32
target_size=(224,224)
test_path='/content/drive/MyDrive/My_Dataset/hama-daun-jambu/test'

test_generator = test_datagen.flow_from_directory test_path
target_size=target_size
batch_size=batch_size,
class_mode=None,
shuffle=False)

test_generator.reset()
```

Calling the saved model for making predictions

```
model_prediksi = load_model("/content/drive/MyDrive/My
Model/mobilenetV2_ori_skenario11.h5")
```

```
pred = model_prediksi.predict(test_generator, verbose=1)
predicted_class_indices = np.argmax(pred, axis=1)
labels = (test_generator.class_indices)
labels = dict((v, k) for k, v in labels.items())
predictions = [labels[k] for k in predicted_class_indices]
filenames = test_generator.filenames
results = pd.DataFrame({"Filename": filenames,
                        "Predictions": predictions})

# create a function for visualizing model performance
def PerformanceReports(conf_matrix, class_report, labels):
    ax = plt.figure(figsize=(15, 10))
    ax = plt.subplot(2, 2, 1)
    sns.heatmap(conf_matrix, annot=True, fmt='.4g', linewidths=2,
                linecolor='white', cbar=True, ax=ax)
    # labels, title and ticks
    ax.set_xlabel('Predicted labels', fontsize=12, color="darkblue", labelpad=24)
    ax.set_ylabel('True labels', fontsize=12, color="darkblue", labelpad=24)
    ax.set_title('Confusion Matrix Model Skenario 11', fontsize=12, pad=24)
    ax.xaxis.set_ticklabels(labels)
    ax.yaxis.set_ticklabels(labels)
    #plt.show()

    ax = plt.subplot(2, 2, 2)
    sns.heatmap(pd.DataFrame(class_report).iloc[:-1, :].T,
                annot=True, ax=ax)
```

```
ax.set_title('Classification Report Model Skenario 11', fontsize=12, pad=24)

plt.show()

labels = ['Lalat Buah', 'Ulat Pagoda', 'Walang Sangit']
test_labels = [fn.split('/')[0] for fn in filenames]
cm = confusion_matrix(test_labels, predictions)

print(cm)
```



Lampiran 3.Sk Pembimbing Tugas Akhir

 **UNIVERSITAS MEDAN AREA**
FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 ☎ (061) 7366878, 7360168, 7364348, 7366781, Fax.(061) 7366998 Medan 20223
Kampus II : Jalan Seliabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 565/FT.6/01.10/VI/2023 25 Juli 2023
Lamp : -
Hal : **Perubahan Judul Tugas Akhir**

Yth, Pembimbing Tugas Akhir
Muhathir, ST, M. Kom
di
Tempat

Dengan hormat, Sehubungan dengan adanya perubahan judul tugas akhir maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa tersebut :

N a m a : Fitriansyah Putra
N P M : 198160053
Jurusan : Teknik Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :

Muhathir, ST, M. Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :


“Analisis Arsitektur Deep Learning MobileNet dalam Mengklasifikasi Hama Daun Jambu Madu”.

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.


Dr. Rahmad Syah, S. Kom, M. Kom

Lampiran 4. Surat pengantar Riset

 **UNIVERSITAS MEDAN AREA**
FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 ☎(061) 7366878, 7360168, 7364348, 7366781, Fax (061) 7366998 Medan 20223
Kampus II : Jalan Setia Budi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanama@uma.ac.id

Nomor : 256 /FT.6/01.10/III/2023 31 Maret 2023
Lamp : -
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

Yth. Pimpinan CV. Sae Akademi Digital
Komplek Puri Zahara II F 12, Jln. Rinte Raya
Di
Medan

Dengan hormat,
Kami mohon kesediaan Bapak/Ibu berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut di bawah ini :


NO	N A M A	N P M	PRODI
1	Fitriyan Syahputra	198160053	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir pada perusahaan/Instansi yang Bapak/Ibu Pimpin.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan ilmiah dan Skripsi yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul penelitian :


Analisis Arsitektur Deep Learning Eficiennet dalam Mengklarifikasi Hama Daun Jambu Madu

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.


Dr. Raluad Syah, S. Kom, M. Kom

Tembusan :
1. Ka. BAMAI
2. Mahasiswa
3. File

Lampiran 5. Surat Selesai Riset



Tempat PKL Medan

Alamat: Komplek Puri Zahara 2 Blok M No.6, Jalan Bunga Rinte Raya
Simpang Selayang, Kec. Medan Tuntungan, Kota Medan, Sumatera Utara 20135
www.tempatpklmedan.com | Instagram @tempatpklmedan | 0813-7017-7719

SURAT KETERANGAN SELESAI PRAKTEK KERJA LAPANGAN

Yang bertanda tangan di bawah ini :

Nama : Asrul Helmandi, S.Kom

Jabatan : Manajer

Menyatakan bahwa yang beridentitas di bawah ini :

Nama : Fitriyan Syahputra

NPM : 198160053


Jurusan : Teknik Informatika

Telah selesai melaksanakan kegiatan praktek kerja lapangan di Tempat PKL Medan dari tanggal 01 Agustus 2022 sampai dengan 31 Agustus 2022 sesuai dengan surat permohonan dari Universitas Potensi Utama

Selama melaksanakan kegiatan praktek kerja lapangan di perusahaan/intansi kami, peserta sangat antusias dan dapat menjalankan tugas-tugas yang kami berikan dengan baik dan bisa dipertanggung jawabkan .

Demikian surat keterangan ini kami buat, atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Medan, 27 Agustus 2022


Asrul Helmandi, S.Kom

