

**ANALISIS KINERJA *HYPERPARAMETER CONVOLUTIONAL
NEURAL NETWORK* PADA KASUS KLASIFIKASI PENYAKIT
DAUN CABAI**

SKRIPSI

Diajukan Oleh:

RAHMAT ARIEF SETYADI

208160026



**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2024**

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 31/1/25

SKRIPSI

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



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

MEDAN

2024

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 31/1/25

HALAMAN PENGESAHAN

Judul Skripsi : Analisis Kinerja Hyperparameter Convolutional Neural Network
Pada Kasus Klasifikasi Penyakit

Nama : Rahmat Arief Setyadi

NPM : 208160026

Fakultas : Teknik

Prodi : Teknik Informatika



Tanggal Lulus : 26 Agustus 2024

UNIVERSITAS MEDAN AREA

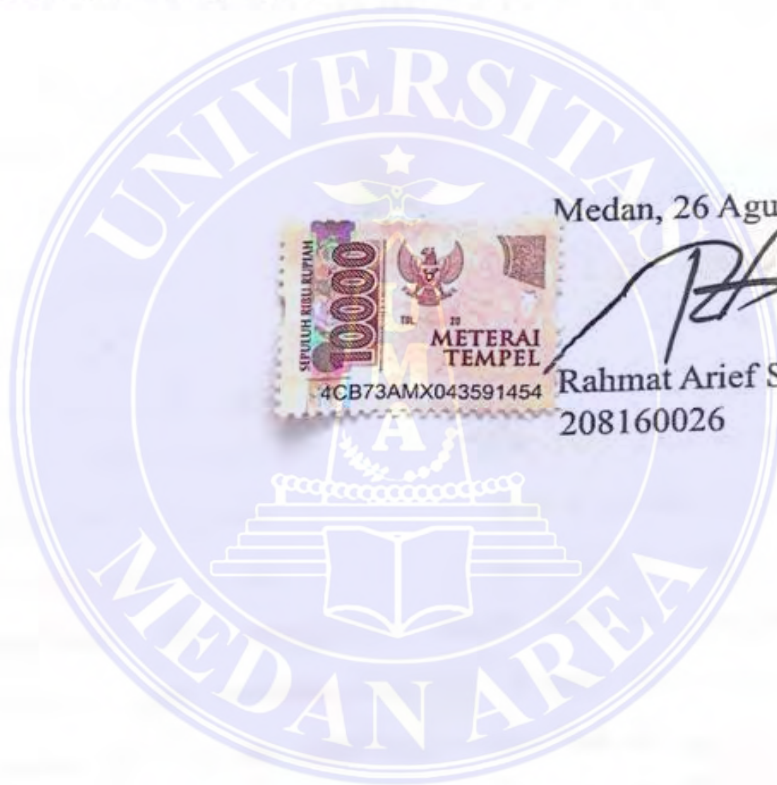
© Hak Cipta Di Lindungi Undang-Undang

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

HALAMAN PERNYATAAN

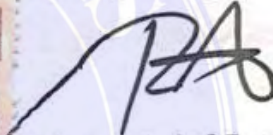
Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian – bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah,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

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



Medan, 26 Agustus 2024




Rahmat Arief Setyadi
208160026

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

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

Nama : Rahmat Arief Setyadi

NPM : 208160026

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 Kinerja Hyperparameter Convolutional Neural Network Pada Kasus Klasifikasi Penyakit 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 : 26 Agustus
2024

Yang
Menyatakan



Rahmat Arief Setyadi
208160026

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 31/1/25

Access From (repository.uma.ac.id)31/1/25

RIWAYAT HIDUP

Penulis dilahirkan di Medan, Pada tanggal 9 April 2001 dari ayah Ribut Budi Santoso dan ibu Desi Sofia Aulia. Penulis merupakan putra pertama dari 3 (tiga) bersaudara. Penulis menyelesaikan pendidikan sekolah dasar di SD Islam Kasih Bunda pada tahun 2013. Pada tahun yang sama penulis melanjutkan pendidikan sekolah menengah pertama di SMP Swasta Islam An-Nizam selama 3 (tiga) tahun dan selesai pada tahun 2016. Penulis melanjutkan tingkat pendidikan selanjutnya di SMAN 5 Medan dan lulus pada tahun 2019. Tahun 2020 penulis juga terdaftar sebagai mahasiswa Teknik Informatika Universitas Medan Area.



KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh Puji dan syukur saya panjatkan kehadirat Allah SWT atas segala Rahmat dan Karunia-Nya sehingga skripsi ini berhasil diselesaikan. Tema yang dipilih dalam penelitian ini ialah *Deep Learning* dengan judul "Analisis Kinerja *Hyperparameter Convolutional Neural Network* Pada Kasus Klasifikasi Penyakit Daun Cabai".

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. Eng. Supriatno, ST, MT. 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 Dr. Sayuti Raman. ST. M.Kom., selaku Dosen pembimbing yang telah membantu penulis sehingga penulis dapat menyelesaikan skripsi ini.
5. Orang tua penulis yaitu Bapak Ribut Budi Santoso dan Ibu Desi Sofia Aulia 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. Adik penulis yaitu Rifky Pradipta dan Shofi Aisyah Aqila yang telah memberikan semangat kepada penulis dalam menyelesaikan skripsi ini dengan sebaik baiknya.
7. Seluruh Dosen dan Staf Program Studi Teknik Informatika Universitas Medan Area.
8. Seluruh pihak yang tidak dapat disebutkan satu persatu yang membantu dalam penyelesaian skripsi ini.

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

Medan, 1 November 2024

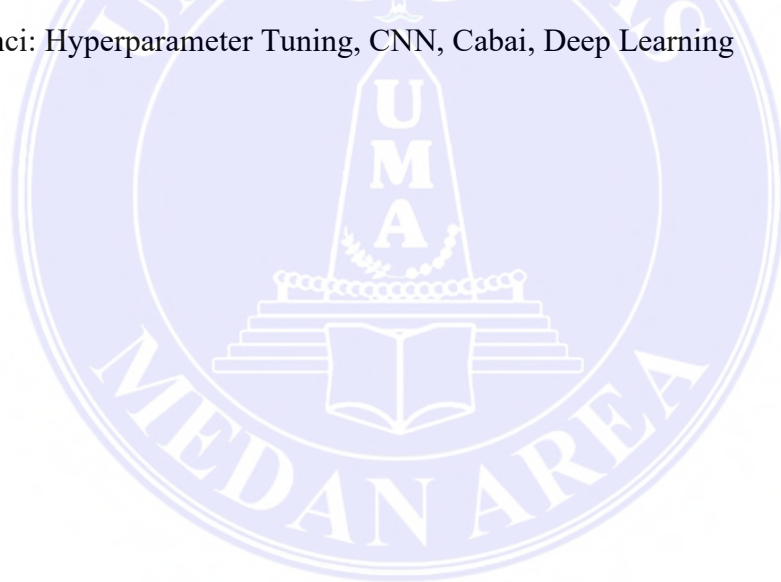
Rahmat Arief Setyadi

NPM 208160026

ABSTRAK

Penyakit yang menyerang daun pada tanaman cabai merupakan tantangan dalam pertanian modern yang dapat diatasi dengan teknologi pengolahan citra dan kecerdasan buatan. Penelitian ini mengusulkan metode hyperparameter tuning, citra daun cabai digunakan dalam dua kondisi: tanpa augmentasi dan dengan augmentasi. Pada arsitektur ResNet101 tanpa augmentasi dataset, akurasi pelatihan mencapai 98.3529% dan akurasi validasi 89.7196%. Dengan augmentasi, akurasi pelatihan sebesar 97.8261% dan akurasi validasi meningkat signifikan menjadi 97.1831%. Perbandingan kinerja arsitektur CNN lainnya menunjukkan ResNet50 dengan akurasi pelatihan 98.4724% dan validasi 97.4178%, sementara ResNet101 dengan hyperparameter tuning mencapai akurasi pelatihan 97.6498% dan validasi tertinggi 98.1221%. hyperparameter dengan fungsi aktivasi Tanh dengan learning rate schedule ReduceLR pada ResNet101 menghasilkan akurasi pelatihan 97.7673% dan validasi tertinggi 99.5305%. Hasil evaluasi menunjukkan bahwa hyperparameter tuning meningkatkan akurasi klasifikasi secara signifikan. Metode ini berkontribusi dalam pengembangan sistem deteksi dini penyakit pada tanaman cabai secara efektif dan efisien. Hyperparameter tuning juga menunjukkan potensi yang besar untuk diterapkan pada berbagai jenis tanaman lain dalam pertanian membantu petani dalam mengidentifikasi dan mengelola penyakit tanaman dengan lebih baik.

Kata kunci: Hyperparameter Tuning, CNN, Cabai, Deep Learning



ABSTRACT

Leaf diseases in chili plants pose a significant challenge in modern agriculture, which can be addressed using image processing and artificial intelligence technologies. This study proposes a hyperparameter tuning method, utilizing chili leaf images in two conditions: without augmentation and with augmentation. For the ResNet101 architecture without dataset augmentation, the training accuracy reached 98.3529%, and the validation accuracy was 89.7196%. With augmentation, the training accuracy was 97.8261%, and the validation accuracy significantly improved to 97.1831%. A comparison of other CNN architectures showed that ResNet50 achieved a training accuracy of 98.4724% and validation accuracy of 97.4178%, while ResNet101 with hyperparameter tuning attained a training accuracy of 97.6498% and the highest validation accuracy of 98.1221%. Hyperparameter tuning with the Tanh activation function and a ReduceLR learning rate schedule on ResNet101 resulted in a training accuracy of 97.7673% and the highest validation accuracy of 99.5305%. The evaluation results indicate that hyperparameter tuning significantly enhances classification accuracy. This method contributes to the development of an effective and efficient early disease detection system for chili plants. Furthermore, hyperparameter tuning demonstrates great potential for application to other types of crops in agriculture, aiding farmers in better identifying and managing plant diseases.

Keywords: Hyperparameter Tuning, CNN, Chili, Deep Learning.



DAFTAR ISI

RIWAYAT HIDUP	vi
KATA PENGANTAR	vii
ABSTRAK	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan 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 Cabai Merah Besar	8
2.2 Penyakit Tanaman	8
2.3 <i>Deep Learning</i>	11
2.4 <i>Convolutional Neural Network (CNN)</i>	11
2.4.1 <i>Convolution Layer</i>	12
2.4.2 Rectified Linear Unit.....	13
2.4.3 <i>Pooling Layer</i>	14
2.4.4 <i>Fully Connected Layer</i>	14
2.4.5 <i>Softmax Activation</i>	15
2.5 ResNet	15
2.6 Augmentasi Data	17
2.7 Fungsi Aktivasi.....	17
2.8 Optimasi	18
2.9 Eksploitasi CNN.....	18
2.9.1 <i>Hyperparameter Tuning</i>	18
2.9.2 Transfer Learning	19
2.10 Pengukuran	19
2.11 Penelitian Terdahulu.....	20
BAB III METODELOGI PENELITIAN.....	24
3.1 Kerangka Kerja Penelitian.....	24
3.2 Waktu dan Tempat	26

3.3	Alat dan Bahan	26
3.4	Analisis Proses Tahapan Hyperparameter Tuning ResNet101	27
3.5	Rancangan Form.....	36
BAB IV HASIL DAN PEMBAHASAN		38
4.1	Hasil Pelatihan dan Fine Tuning Existing CNN.....	38
4.1.1	Hasil Pelatihan Dataset Tidak Diaugmentasi dan Diaugmentasi	39
4.1.2	Hasil Pelatihan Existing CNN.....	40
4.2	Perbandingan Lapisan ResNet.....	41
4.3	Hyperparameter Tuning Optimizer	42
4.4	Hyperparameter Tuning Fungsi Aktivasi	45
4.5	Hyperparameter <i>Tuning Learning Rate Schedule</i>	48
4.6	Implementasi Form Klasifikasi	50
BAB V KESIMPULAN		52
5.1	Kesimpulan.....	52
5.2	Saran.....	52
DAFTAR PUSTAKA		53
LAMPIRAN.....		56

DAFTAR GAMBAR

Gambar 2.1 Penyakit Pada Daun Cabai	9
Gambar 2.2 Arsitektur CNN	
Gambar 2.3 lapisan Konvolusi.....	13
Gambar 2.4 Rectified Linear Unit.....	13
Gambar 2.5 Max-Pooling.....	14
Gambar 3.1 adalah arsitektur ResNet101 yang akan digunakan	27
Gambar 3.2 <i>Flowchart Hyperparameter Tuning</i> CNN.....	34
Gambar 3.3 Rancangan Form	37
Gambar 4.1 Grafik Perbandingan Akurasi latih dan Validasi Optimasi.....	43
Gambar 4.1 Grafik validasi Hasil Hasil Fine Tuning Optimizer	44
Gambar 4.2 Grafik Perbandingan Akurasi latih dan validasi Fungsi Aktivasi.....	47
Gambar 4.3 Grafik Perbandingan Akurasi Latih dan Validasi Learning Rate Schedule	49
Gambar 4.4 Tampilan Awal Form Klasifikasi.....	50
Gambar 4.5 Tampilan Halaman Menampilkan Hasil Klasifikasi	51

DAFTAR TABEL

TABEL 1.1 PENELITIAN TERDAHULU	20
TABEL 2.1 PEMBAGIAN DATASET CITRA	36
TABEL 4. 1 HASIL PELATIHAN DATASET TIDAK DIAUGMENTASI DAN DIAUGMENTASI	39
TABEL 4. 2 HASIL PELATIHAN EXISTING CNN.....	40
TABEL 4. 3 HASIL PERBANDINGAN RESNET.....	41
TABEL 4. 4 HASIL PERBANDINGAN BERBAGAI OPTIMASI	42
TABEL 4. 5 HASIL BERBAGAI FUNGSI AKTIVASI DI FINE TUNING	45
TABEL 4. 6 PERBANDINGAN AKURASI <i>LEARNING RATE SCHEDULE</i>	48



BAB I

PENDAHULUAN

1.1 Latar Belakang

Cabai merah adalah salah satu jenis sayuran yang memiliki peran penting di Indonesia, baik untuk kebutuhan konsumsi dalam negeri maupun sebagai produk ekspor, namun tanaman cabai sering kali diserang dari beberapa jenis penyakit salah satu diantaranya adalah penyakit yang disebabkan oleh faktor hama dan lingkungan yang dapat menginfeksi tanaman cabai yang merusak kesehatan daun cabai dan faktor lingkungan (Astuti dkk., 2021).

Dengan kemajuan teknologi informasi saat ini memungkinkan identifikasi penyakit pada daun cabai secara otomatis dapat dilakukan dengan menggunakan sistem komputer. Pemrosesan gambar memungkinkan komputer menganalisis dan mengidentifikasi penyakit yang dilihat dan direkam oleh kamera. Perkembangan ilmu pengetahuan terkini telah memfasilitasi penemuan cara mendeteksi penyakit pada daun cabai secara otomatis menggunakan komputer. Deteksi penyakit menggunakan komputer. penyakit berbasis komputer dapat menggunakan metode deep learning dikarenakan deteksi yang dihasilkan komputer cukup akurat (Dzaky, 2021). (Mustamu dkk., 2023).

Convolutional Neural Network (CNN) merupakan salah satu algoritma *deep learning* yang dirancang untuk pengolahan gambar, (Karno dkk., 2022). Menurut (Anggraeni dkk., 2022) *convolutional neural network* (CNN) memiliki kemampuan untuk menerima input berupa citra dan dapat mempelajari aspek atau objek yang

terdapat di dalamnya. Selain itu, CNN mampu melakukan ekstraksi fitur secara otomatis dari citra dan kemudian mengklasifikasikan citra berdasarkan fitur-fitur yang diperoleh. Secara umum, CNN terdiri dari empat komponen utama yaitu : *Convolution layer*, *ReLU layer (Activation Function)*, *Pooling layer*, dan *Fully connected layer*. CNN memiliki beberapa arsitektur di dalamnya satu diantaranya adalah Residual Network (ResNet), Ciri utama dari arsitektur ResNet adalah penanganan masalah gradien yang menghilang (Ridhovan & Suharso, 2022). Masalah ini muncul karena banyaknya jumlah lapisan (layer) yang mengakibatkan penurunan fungsi loss dalam pencarian bobot yang sesuai.

Penelitian ini mengimplementasikan algoritma CNN untuk mengukur tingkat akurasi prediksi yang dihasilkan. Untuk mengoptimalkan algoritma CNN, perlu dilakukan *hyperparameter tuning*, salah satunya adalah penyesuaian nilai Epoch, fungsi aktivasi, optimasi dan *batch size* untuk mendapatkan akurasi terbaik dari model yang diamati.

Hyperparameter tuning adalah proses menyesuaikan parameter atau bobot dengan dataset yang baru atau yang sudah ada. *Hyperparameter tuning* berfokus pada mencari konfigurasi terbaik dari parameter yang mengontrol proses pelatihan model untuk meningkatkan performa secara keseluruhan. *Hyperparameter tuning* biasanya dilakukan pada dataset baru ataupun yang sudah ada dan masalah yang sama atau masalah yang baru (Poojary dkk., 2021).

Hyperparameter Tuning memainkan peran penting dalam *machine learning* dan *deep learning*, karena parameter yang dihasilkan sangat mempengaruhi kinerja model CNN. Oleh karena itu, Hyperparameter Tuning diusulkan untuk

mengoptimalkan pemrosesan citra pada model CNN yang dikembangkan. Adapun objek yang akan diklasifikasi adalah kondisi penyakit pada daun cabai merah.

Dengan menggunakan metode CNN arsitektur ResNet101, dengan fungsi aktivasi tanh, optimasi SGD, batch size 32 dan dilatih sebanyak 50 epoch. klasifikasi penyakit pada daun cabai dapat dilakukan secara otomatis dan akurat. Selain itu untuk melatih lapisan dapat menyesuaikan parameter atau bobot maka perlu dilakukan tuning hyperparameter untuk mencapai hasil yang lebih baik.

Dalam penelitian yang berjudul “Optimasi Hyperparameter dalam Algoritma CNN” menyatakan klasifikasi penyakit daun penelitian ini mengoptimalkan model CNN (MobileNet-V2) untuk klasifikasi penyakit padi di Indonesia dengan hyperparameter terbaik (epoch 100, batch size 16, learning rate 0,1, optimizer Nadam). Hasil menunjukkan akurasi 86%, presisi 86%, recall 84%, dan f1-score 84%. Penelitian ini berfokus pada penyakit daun padi, dengan rekomendasi penelitian selanjutnya untuk daun cabai dan perbandingan dengan arsitektur CNN lain seperti DenseNet, ResNet, dan AlexNet (Rizal Amegia, 2024).

Dalam Penelitian (Chavez & Hernando, 2023) yang berjudul “Implementasi Sistem Pendeteksi Penyakit Pada Daun Singkong Dan Daun Cabai Berbasis Machine Learning” memiliki nilai akurasi training 87.50%, sedangkan akurasi testing 82.50%. dalam penelitian (Ridhovan & Suharso, 2022) yang berjudul “Penerapan Metode Residual Network (ResNet) Dalam Klasifikasi Penyakit Pada Daun Gandum” ResNet mampu mengklasifikasikan penyakit dengan tingkat akurasi lebih dari 95% menggunakan pemisahan data yang berbeda. Tingkat akurasi tertinggi terdapat pada skenario 2, yang menunjukkan nilai akurasi dan matriks

kebingungan yang lebih tinggi dibandingkan dengan skenario lainnya. Skenario 2 mencapai tingkat akurasi sebesar 98% dan matriks kebingungan sebesar 0,35.

Berdasarkan latar belakang dan penelitian terkait yang telah dijelaskan sebelumnya, inovasi dalam penelitian ini yang membedakannya dari penelitian terdahulu yaitu penggunaan hyperparameter Convolutional Neural Networks (CNN) dengan arsitektur ResNet101 untuk mengklasifikasikan penyakit pada daun cabai merah. penelitian ini diberi judul "Analisis Kinerja *Hyperparameter Convolutional Neural Network* dalam Klasifikasi Penyakit pada Daun Cabai Merah."

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas maka terdapat rumusan masalah yaitu:

1. Bagaimana menerapkan teknik hyperparameter tuning CNN dengan menggunakan arsitektur ResNet101 untuk mengklasifikasikan penyakit pada daun cabai?
2. Apakah model yang dihasilkan melalui proses *Hyperparameter tuning* mampu mengenali dan mengklasifikasikan penyakit pada daun cabai dengan akurasi yang lebih baik dari pada penelitian sebelumnya?

1.3 Batasan Masalah

Permasalahan yang diangkat dalam penelitian ini memiliki cakupan yang luas, sehingga diberikan sejumlah batasan untuk memastikan penelitian lebih terarah dan selaras dengan tujuan yang ingin dicapai, yaitu sebagai berikut:

1. Penelitian ini difokuskan pada klasifikasi penyakit pada daun tanaman cabai (*Capsicum annuum*).
2. Penelitian ini akan membatasi jenis penyakit yang diklasifikasikan pada daun cabai.
3. Penelitian ini akan menggunakan metode CNN yang di *tuning* menggunakan arsitektur ResNet101 yang telah dilatih sebelumnya sebagai pendekatan utama untuk mengklasifikasikan penyakit pada daun cabai.
4. Dataset citra penyakit pada daun cabai yang digunakan dalam penelitian ini bersumber dari Menddeley Dataset M P, Aishwarya; Reddy, Padmanabha (2024), “chilli dataset”, Mendeley Data, V2.
5. Evaluasi kinerja model dilakukan berdasarkan metrik klasifikasi standar yaitu akurasi latih dan validasi

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengatasi permasalahan yang diidentifikasi di rumusan masalah dengan mengusulkan pendekatan *hyperparameter tuning* maka tujuan yang ingin dicapai dalam penelitian ini, yaitu sebagai berikut:

1. Mempelajari dan menganalisis efektivitas pendekatan hyperparameter tuning dalam mengklasifikasikan penyakit pada daun cabai.
2. Menguji dan mengevaluasi kinerja model yang dihasilkan dalam mengenali dan mengklasifikasikan penyakit pada daun cabai.
3. Menghasilkan model CNN *hyperparameter tuning* ResNet101 yang dapat diimplementasikan.

1.5 Manfaat Penelitian

Adapun Manfaat penelitian ini yaitu :

1. Penelitian ini diharapkan memberikan berbagai manfaat baik secara akademis ilmu pengetahuan digital dan pengolahan citra, khususnya dalam konteks deteksi penyakit pada tanaman cabai menggunakan teknologi deep learning.
2. Penelitian ini diharapkan memberikan kontribusi dalam meningkatkan produktivitas tanaman cabai, dengan memungkinkan untuk mengidentifikasi penyakit dengan cepat, tindakan pencegahan dan pengendalian yang tepat dapat diambil untuk mengurangi kerugian hasil panen yang disebabkan oleh penyakit.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini di sampaikan kepada para pembaca untuk mempermudah pemahaman dari isi penelitian ini. Berikut ini sistematika penulisan dari penelitian :

BAB I PENDAHULUAN

Pada bagian pendahuluan dibahas latar belakang penelitian ini bagaimana cara mengapa penelitian ini dilakukan, kemudian dibahas rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini teori-teori yang relevan dengan topik penelitian dikumpulkan dan disajikan untuk memperluas basis informasi dan menjadi dasar argumentasi. Dasar

teori yang digunakan dalam penelitian ini meliputi *Deep Learning*, *Convolutional Neural Network*, Arsitektur ResNet101, serta beberapa teori pendukung lainnya yang terkait dengan pengembangan model CNN untuk klasifikasi penyakit pada daun cabai.

BAB III METODELOGI PENELITIAN

Bab ini menjelaskan tahapan-tahapan yang akan dilakukan dalam penelitian ini. Setiap rencana dari tahapan penelitian dijelaskan secara rinci.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menampilkan hasil dari penelitian yang telah dilakukan, termasuk hasil dari setiap tahapan penelitian.

BAB V KEIMPULAN DAN SARAN

Pada bab ini akan disajikan kesimpulan dari penelitian yang telah dilakukan. Kesimpulan ini merupakan rangkuman diperoleh dari hasil penelitian. Kesimpulan akan merujuk kembali pada tujuan penelitian dan menjawab rumusan masalah yang telah diidentifikasi seperti keefektivitasan model CNN khususnya arsitektur ResNet101, dalam klasifikasi penyakit pada daun cabai.

Sedangkan dibagian saran memberikan rekomendasi yang dapat berguna bagi penelitian selanjutnya. Saran ini didasarkan pada keterbatasan penelitian saat ini dan potensi perbaikan atau pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Cabai Merah Besar

Capsicum annum L., tanaman cabai merah besar adalah tanaman semak atau perdu yang termasuk dalam keluarga terong (Solanaceae). Cabai merah merupakan tanaman semusim atau berumur pendek dan berasal dari benua Amerika, Eropa, dan Asia termasuk Indonesia. Adapun beberapa jenis cabai yang dikenal masyarakat yakni cabai besar, cabai keriting, cabai rawit, dan paprika. (Astuti dkk., 2021).

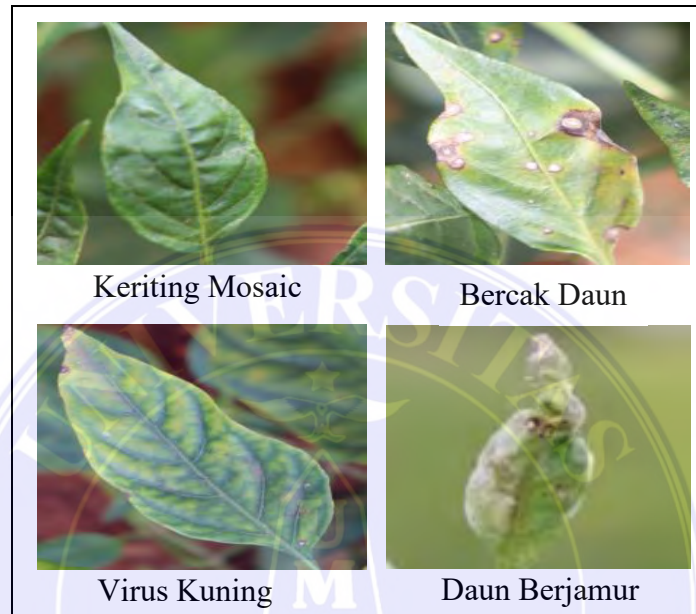
Penelitian ini dilakukan bertujuan untuk mengklasifikasikan penyakit pada daun cabai dengan menerapkan metode *Hyperparameter Convolutional Neural Network* (CNN) menggunakan arsitektur ResNet101 yang mampu mengolah informasi citra secara efektif. Dengan metode CNN dan arsitektur ResNet101, klasifikasi penyakit pada daun cabai dapat dilakukan secara otomatis dan akurat.

2.2 Penyakit Tanaman

Tanaman cabai memiliki kendala dalam membudidayakannya terutama dikerenakan terjerat penyakit. Penyakit pada tanaman memiliki hubungan yang kuat dengan patogen. Istilah patogen merujuk pada hal-hal yang menyebabkan tanaman mengalami kerusakan atau gangguan. Oleh karena itu, patogen tidak selalu berupa organisme hidup (*animate pathogen*), tetapi juga dapat berupa faktor tak hidup (*inanimate pathogen*), seperti kekurangan unsur hara, air, atau penyebab lainnya. (Wakhidah dkk., 2021).

Penyakit yang sering menyerang tanaman terutama tanaman cabai salah satunya dibagian daun, penyakit yang menyerang daun cabai yaitu meliputi keriting mosaic, bercak pada daun, jamur pada daun, dan daun cabai berwarna kekuningan.

Berikut ini gambar jenis penyakit pada daun cabai.



Gambar 2.1 Penyakit Pada Daun Cabai

Sumber : <https://data.mendeley.com/datasets/tf9dtfz9m6/2>

Berikut penjelasan penyakit yang menyerang daun cabai yaitu sebagai berikut :

1. Daun Keriting (*Keriting Mosaic*)

Penyakit ini disebabkan oleh *Cucumber Mosaic Virus* (CMV) dengan gejala berupa perubahan warna daun yang menjadi belang antara hijau tua dan hijau muda. Kadang-kadang, penyakit ini juga disertai perubahan bentuk daun, seperti menjadi cekung, keriting, atau memanjang, sebagaimana ditampilkan pada Gambar 2.1. Penyebaran penyakit ini terjadi melalui kutu daun, sehingga salah satu langkah pencegahan yang efektif adalah penggunaan insektisida untuk mengendalikan populasi kutu daun (Dzaky, 2021).

2. Bercak Daun

Penyakit bercak daun sering menyerang tanaman cabai dan disebabkan oleh jamur *Cercospora capsici*. Penyakit ini dikenal pula dengan sebutan Penyakit Mata Katak atau Bercak Daun Serkospora. Intensitas serangan cenderung meningkat akibat jarak tanam yang terlalu rapat dan sistem drainase yang buruk (Devianto dkk., 2023).

3. Virus Kuning

Virus kuning adalah penyakit yang disebabkan oleh virus gemini yang menyerang daun tanaman cabai. Gejala awal penyakit ini meliputi pemutihan pada tulang daun, diikuti perubahan warna daun menjadi kuning, penebalan tulang daun, dan daun yang melengkung ke atas. Ilustrasi gejala ini dapat dilihat pada Gambar 2.1. Penyakit ini menyebar melalui hama kutu kebul yang berasal dari tanaman yang telah terinfeksi. Peningkatan populasi kutu kebul berkontribusi terhadap perkembangan penyakit ini. Selain itu, penyakit ini juga dapat disebabkan oleh penggunaan bibit tanaman yang telah terinfeksi sebelumnya (Dzaky, 2021).

4. Daun Berjamur

Gejala penyakit ini ditandai dengan kemunculan bercak-bercak bulat kecil pada daun yang tampak basah. Bercak-bercak tersebut dapat meluas, dengan bagian tengahnya tampak pucat dan tepinya lebih gelap. Seiring waktu, bercak ini dapat berlubang. Jika intensitasnya tinggi, daun akan cepat menguning dan gugur. Dalam beberapa kasus, daun bahkan bisa gugur tanpa mengalami klorosis atau penguningan terlebih dahulu (Nur Inaya et al., 2022).

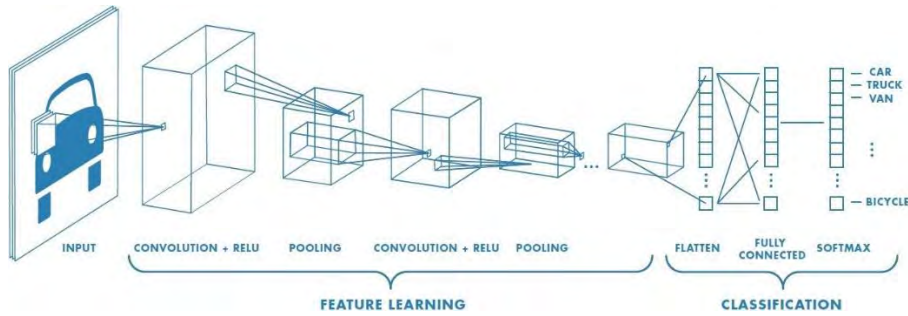
2.3 *Deep Learning*

Kecerdasan buatan (AI) saat ini berkembang sangat pesat, masalah-masalah yang rumit sulit dipecahkan oleh manusia kini dapat diatasi dengan bantuan kecerdasan buatan, oleh karena itu, dengan adanya kecerdasan buatan, permasalahan tersebut dapat diselesaikan dengan mudah. Kecerdasan buatan memiliki masalah kemampuan untuk memahami dengan cepat dalam pengetahuannya sehingga untuk menyelesaikan masalah tersebut digunakan konsep *deep learning* (Rosadi & Lutfi, 2021).

Deep learning merupakan salah satu cabang ilmu dari *machine learning* yang berbasis pada *Artificial Neural Networks* (ANN) atau bisa dikatakan merupakan evolusi dari ANN. Pada *deep learning* komputer belajar mengklasifikasikan objek, gambar, teks, atau data lainnya dengan mengidentifikasi pola yang kompleks maupun abstrak dalam bentuk data. Dengan menggunakan jaringan saraf yang dalam seperti *Convolutional Neural Network* (CNN) (Almansour dkk., 2022)

2.4 *Convolutional Neural Network* (CNN)

Convolutional Neural Network adalah salah satu jenis algoritma *deep learning* yang berfungsi dengan menerima input berupa gambar. CNN dapat digunakan untuk pelatihan data, baik dalam *supervised learning* maupun *unsupervised learning* (Geza Jeremia Bu'ulölö, 2021). Arsitektu CNN terdiri dari beberapa lapisan yaitu *convolution layers*, *pooling layers*, *fully connected layers*, dan *softmax fuction* (Dzaky, 2021).



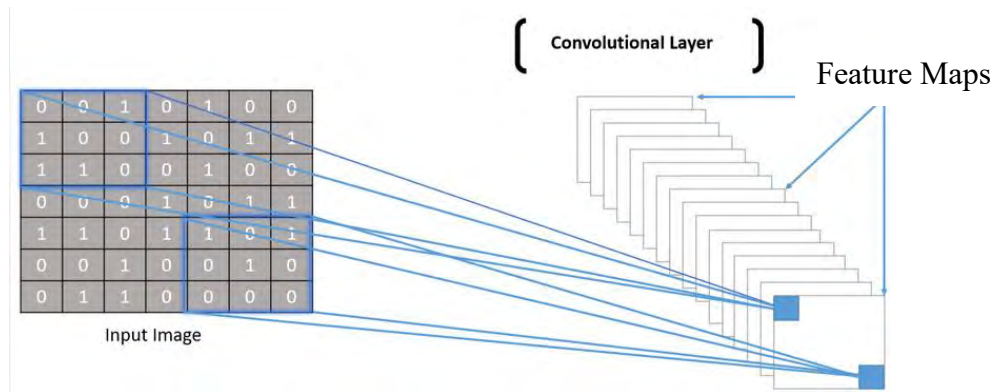
Gambar 2.2 Arsitektur CNN

Sumber : (Rahman dkk., 2021)

Arsitektur CNN terdiri dari dua bagian utama, yaitu *Feature Learning* dan *Classification*, seperti yang terlihat pada Gambar 2.2. *Feature Learning* terdiri dari lapisan konvolusi, fungsi aktivasi ReLU (*Rectified Linear Unit*), dan *pooling*. Lapisan-lapisan ini sering disusun secara bertumpuk sesuai kebutuhan arsitektur yang digunakan. Bagian klasifikasi terdiri dari lapisan *flatten*, *fully connected*, dan *softmax* (Alzubaidi dkk., 2021). Setiap citra input melewati beberapa lapisan konvolusi dan *pooling*, diikuti oleh beberapa lapisan *fully connected*. Selain itu, fungsi *softmax* diterapkan di akhir untuk melakukan klasifikasi multikelas (Nisa dkk., 2020). CNN sangat cocok untuk pemrosesan dengan input citra. Karena itu pada CNN biasa digunakan untuk mengklasifikasikan citra, deteksi objek, dan pengenalan suara (Lesmana dkk., 2022).

2.4.1 Convolution Layer

Lapisan konvolusional (*convolutional layer*) adalah tempat di mana output, yang juga dikenal sebagai Peta Aktivasi atau Peta Fitur, dihasilkan melalui operasi konvolusi antara input dan nilai filter (Tilasefana & Putra, 2023). Filter akan diterapkan ke seluruh gambar. Gambar 2.3 di bawah ini menggambarkan operasi dari lapisan konvolusional :

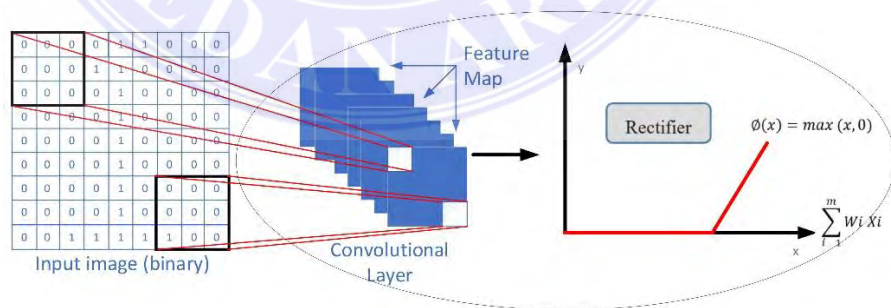


Gambar 2.3 lapisan Konvolusi

Sumber : (Tilasefana & Putra, 2023)

2.4.2 Rectified Linear Unit

Rectified Linear Unit (ReLU) digunakan untuk memperkenalkan fungsi non-linear ke dalam jaringan konvolusi. ReLU biasanya diterapkan pada lapisan konvolusi maupun lapisan pooling. Fungsi aktivasi ReLU membatasi keluaran dari lapisan pooling, mempercepat proses pelatihan dengan memetakan nilai negatif menjadi nol, dan mempertahankan nilai positif untuk diproses lebih lanjut di lapisan berikutnya (Nihayatul Husna dkk., 2022).

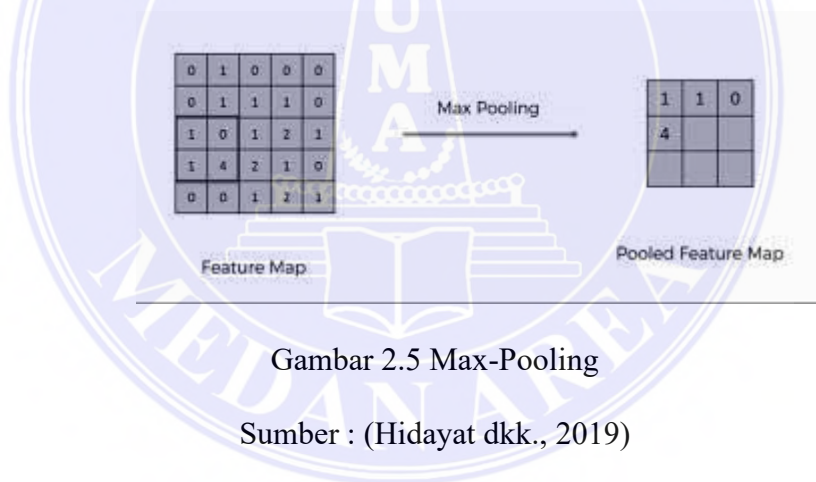


Gambar 2.4 Rectified Linear Unit

Sumber : (Caesarendra dkk., 2021)

2.4.3 Pooling Layer

Lapisan *pooling* atau *pooling layer* berfungsi untuk mengurangi ukuran dari kumpulan ciri hasil konvolusi atau mengurangi ukuran dari gambar masukan. untuk meringankan beban komputasi selama pemrosesan data (Alzubaidi et al., 2021). Proses pooling biasanya mengambil nilai terbesar (*max pooling*) atau nilai rata-rata (*average pooling*) pixel dari feature map yang dihasilkan *convolution layer* (Irfansyah et al., 2021). Pada tahap max pooling, proses pooling membagi hasil keluaran dari jaringan konvolusional menjadi sejumlah grid kecil. Kemudian, nilai maksimum dari setiap grid akan dimasukkan ke dalam matriks citra yang telah direduksi oleh lapisan pooling. Contoh operasi lapisan max pooling dapat dilihat pada gambar berikut ini:



Gambar 2.5 Max-Pooling

Sumber : (Hidayat dkk., 2019)

2.4.4 Fully Connected Layer

Fully connected layer adalah lapisan dalam CNN yang dipergunakan kedalam tahapan klasifikasi objek gambar. *Fully connected layer* juga merupakan tahapan akhir dalam CNN hingga bisa mendapatkan hasil klasifikasi terhadap kelas objek (Tilasefana & Putra, 2023). Masukan dapat diterima *fully connected layer* mengharuskan berbentuk vektor atau satu dimensi, sehingga perlu dilakukan

flatten, lapisan ini umumnya diletakan setelah lapisan pooling dalam CNN. Lapisan ini adalah proses merubah matrik menjadi bentuk vector tunggal atau perataan bentuk feature map sebelum masuk ke tahap klasifikasi (Karno dkk., 2022). *Fully-connected layer* terdiri atas 13 weights (bobot) dan bias, neuron yang terdapat nantinya bisa saling terhubung dengan menggunakan nilai bobot dan bias. Neuron yang terhubung sama seperti susunan puzzle, setiap ciri akan saling berhubungan hingga bisa didapatkannya hasil klasifikasi dari kelas objek.

2.4.5 Softmax Activation

Softmax activation adalah jenis fungsi aktivasi yang biasa digunakan pada layer output dari *fully connected layer* untuk menghitung distribusi probabilitas dari vektor bilangan real. Fungsi ini secara matematis mengonversi angka dalam bentuk vektor menjadi vektor probabilitas. Softmax umumnya digunakan dalam proses klasifikasi dan penentuan kelas suatu objek. Setiap neuron atau fitur objek yang terhubung setelah bobotnya dijumlahkan akan menghasilkan nilai probabilitas yang menentukan kelas objek tersebut. Softmax mengubah output berupa total bobot menjadi nilai probabilitas yang merepresentasikan kemungkinan suatu objek termasuk dalam kelas tertentu, Fungsi ini sangat sesuai digunakan untuk klasifikasi dengan banyak kelas (multiclass classification). (Dzaky, 2021).

2.5 ResNet

ResNet, atau Residual Network, adalah salah satu bentuk arsitektur *Convolutional Neural Network* (CNN) yang dibedakan oleh penggunaan *skip connection* dan *batch normalization*, serta tidak menyertakan *fully connected layers*

di bagian akhir (Lukas Hansel & Bunyamin, 2021). Fitur utama dari ResNet adalah kemampuannya dalam mengatasi masalah vanishing gradients. Masalah ini sering terjadi pada jaringan yang sangat dalam karena kemampuan fungsi kerugian untuk mengoptimalkan bobot yang sesuai menjadi berkurang. Seiring bertambahnya jumlah lapisan dalam pembelajaran mendalam (*Deep Learning*), gradien dapat menjadi semakin kecil hingga akhirnya menghilang akibat perkalian berulang. Selain itu, pelatihan jaringan saraf yang dalam dengan banyak parameter memerlukan tingkat pembelajaran (*learning rate*) yang cukup tinggi. Jenis lapisan dalam desain ResNet mencakup 18, 34, 50, 101, dan 152 lapisan (Lasniari dkk., 2022). Tujuan dari arsitektur ResNet adalah melakukan pemetaan identitas pada data citra dengan menggunakan prinsip *skip connection* atau melompati beberapa lapisan untuk menghindari hilangnya gradien. Dapat dilihat seperti pada Gambar 2.4 dibawah ini.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2 _x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3 _x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4 _x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5 _x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

Gambar 2.7 Lapisan Layer ResNet

(Togatorop & Fauzi, 2022).

2.6 Augmentasi Data

Data Augmentation adalah proses memperkaya data pelatihan guna menghindari overfitting. Proses ini melibatkan beberapa langkah, yaitu horizontal flip, shear range, dan zoom range. Masing-masing nilai shear range dan zoom range adalah 0,2. Langkah horizontal flip bertujuan untuk menambah jumlah data pelatihan dengan memutar gambar atau citra secara horizontal sebesar 90 derajat (Solihin dkk., 2022). Langkah shear range menggunakan metode shear transformation untuk menambah variasi citra dengan merotasi citra pada sudut tertentu. Sedangkan langkah zoom range memperbesar citra dengan skala tertentu dari citra asli (Naufal & Kusuma, 2021).

2.7 Fungsi Aktivasi

Fungsi aktivasi adalah fungsi non-linear yang memungkinkan Jaringan Syaraf Tiruan (JST) mentransformasi data input ke dalam dimensi yang lebih tinggi. Transformasi ini memungkinkan pemotongan *hyperplane* sederhana untuk melakukan klasifikasi. Dalam Convolutional Neural Network (CNN), fungsi aktivasi yang digunakan adalah fungsi aktivasi Tanh.

Cara kerja fungsi aktivasi Tanh dalam jaringan saraf adalah input diterima oleh jaringan melalui lapisan input. Di neuron pada lapisan selanjutnya, jumlah bobot dari setiap input dihitung dan biasanya ditambahkan dengan bias. Jumlah tersebut kemudian diproses menggunakan fungsi aktivasi (Firmansyah & Hayadi, 2022).

2.8 Optimasi

Optimasi berfungsi untuk meminimalkan fungsi kerugian (*loss function*) dan mencari serangkaian nilai input yang menghasilkan *output* berupa nilai minimum atau maksimum dari suatu fungsi. Optimasi mempelajari cara mengatur parameter model agar prediksinya lebih akurat dan sesuai dengan data pelatihan. Optimasi merupakan hyperparameter yang diuji untuk mengurangi nilai kerugian (Ken Ratri, 2023). Pada penelitian ini, tiga percobaan optimasi yaitu Adam, AdamW dan SGD.

2.9 Eksploitasi CNN

Eksploitasi *Convolutional Neural Networks* (CNN), yang dilakukan dalam penelitian ini adalah untuk mengoptimalkan model CNN dengan arsitektur ResNet101. Model CNN ini digunakan untuk klasifikasi penyakit pada daun cabai dengan memanfaatkan transfer learning (transfer pengetahuan), pruning (pemangkasan) dan fine tuning (penyetelan) untuk meningkatkan keakuratan model CNN dengan arsitektur ResNet101. Eksploitasi CNN terdiri dari beberapa model yaitu :

2.9.1 Hyperparameter Tuning

Hyperparameter tuning adalah proses mencari nilai parameter optimal yang dimulai dengan menentukan daftar parameter dan rentang pencarian untuk setiap parameter. Proses ini membantu model menemukan nilai parameter yang tepat untuk setiap dataset, sehingga kinerja model dapat maksimal (Emma, 2022).

Dilakukan parameter tuning yang bertujuan untuk mengukur tingkat akurasi dari model yang diterapkan. Proses ini juga melibatkan pelatihan data latih dan data

validasi hingga mencapai tingkat akurasi yang optimal. Salah satu parameter yang digunakan yaitu fungsi aktivasi, optimasi dan epoch, yang diatur untuk meningkatkan nilai akurasi dan validasi dilatih guna menghasilkan model dengan akurasi yang optimal (Indra, 2022).

2.9.2 Transfer Learning

Transfer learning bertujuan untuk meningkatkan performa pada tugas target dengan memanfaatkan pengetahuan yang telah diperoleh dari tugas sumber dan menerapkannya pada tugas target. Pada saat menggunakan pembelajaran transfer, yang paling penting adalah target task dan source task domain harus serupa (Falakhi dkk., 2022).

Transfer learning dapat menjadi solusi efektif untuk mengatasi keterbatasan data, karena model yang telah dilatih dapat diadaptasi untuk menyelesaikan tugas yang berbeda dengan menggunakan dataset yang relatif sedikit (Wiratama dkk., 2023).

2.10 Pengukuran

Performa CNN dapat diukur menggunakan beberapa cara diantaranya dengan mengukur tingkat akurasi klasifikasi latih dan validasi. Akurasi merupakan sebagai tolak ukur kebenaran hasil dari penelitian, yang menunjukkan sejauh mana hasil penelitian tersebut akurat (IBRAHIM dkk., 2022). Berikut adalah persamaan matematis akurasi Perhitungan tersebut dapat dilihat pada persamaan berikut.

$$\text{Akurasi} = \frac{\text{Jumlah Citra Benar atau Dikenali}}{\text{Jumlah Total Citra Keseluruhan}} \times 100 \quad (2.1)$$

2.11 Penelitian Terdahulu

Penelitian ini didasari oleh hasil-hasil penelitian sebelumnya yang memperluas dan memperdalam teori yang digunakan dalam penelitian ini. Penelitian sebelumnya sering menjadi bahan kutipan yang penting dalam penelitian yang lebih baru, dan hasil penelitian sebelumnya juga merupakan sumber inspirasi yang sangat membantu selama proses penelitian. Adapun penelitian terdahulu dalam penelitian ini dapat di lihat pada tabel :

Tabel 1.1 Penelitian Terdahulu

No.	Nama Penelitian dan tahun	Judul	Metode	Hasil Penelitian
1	(Sasongko dkk., 2023)	Analisis Efek Augmentasi Dataset Dan Fine Tune Pada Algoritma Pre Trained Convolutional Neural Network (Cnn)	transfer learning CNN	Penelitian menunjukkan bahwa teknik augmentasi random erase dan zoom range secara signifikan meningkatkan akurasi model transfer learning CNN (mobilenet, efficientnet, dan nasnetmobile) pada dataset kecil, dengan peningkatan akurasi sekitar 0,03% hingga 0,1%.
2	(Leo dkk., 2022)	Fine Tuning CNN Pre-trained CNN Berdasarkan Pencitraan Termal untuk Deteksi Dini Obesitas	DesNet	Dalam studi ini, kami menyempurnakan model CNN yang telah dilatih sebelumnya dalam mengklasifikasikan kumpulan data gambar termal obesitas. Model DenseNet201 mengungguli model lain dalam hal akurasi klasifikasi (akurasi validasi 83,33 %) dan kinerja pembelajaran

				<p>(kesenjangan terdekat 13,33 % antara akurasi pelatihan dan validasi). Sedangkan model MobileNet mengungguli model lain dalam hal kecepatan komputasi (waktu berlalu pelatihan 12 detik/ epoch). Model DenseNet201 yang diusulkan cocok untuk diterapkan sebagai sistem skrining dini obesitas bagi petugas kesehatan atau dokter karena kinerja model dalam kinerja klasifikasi. Sementara itu, model MobileNet yang diusulkan cocok untuk deteksi dini/diagnosis obesitas pada aplikasi seluler karena kinerja model dalam hal kecepatan komputasi.</p>
3	(Dzaky, 2021)	Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network	AlexNet	<p>Convolutional Neural Network (CNN) telah terbukti mampu memberikan akurasi yang sangat tinggi dalam klasifikasi, terutama dalam menemukan fitur-fitur yang ada pada citra secara otomatis. Dalam penelitian ini, penggunaan arsitektur AlexNet juga menghasilkan akurasi yang sangat baik. Untuk meningkatkan akurasi CNN, teknik data augmentation dapat digunakan untuk meningkatkan jumlah data, serta mencari learning rate yang optimal agar proses pembelajaran menjadi lebih cepat dan efektif.</p>

				Dengan pengaturan learning rate yang tepat, model CNN akan lebih cepat memahami citra dan memberikan akurasi yang sangat baik.
4	(Noprisson, 2022)	Fine-Tuning Model Transfer Learning VGG16 Untuk Klasifikasi Citra Penyakit Tanaman Padi	VGG16	Penelitian ini menganalisis kinerja VGG16 untuk klasifikasi penyakit padi dengan fine-tuning yang meningkatkan akurasi model. Tanpa fine-tuning, akurasi pelatihan, validasi, dan pengujian berturut-turut adalah 50.88%, 52.50%, dan 54.19%. Dengan fine-tuning, akurasi meningkat menjadi 63.50% (pelatihan), 62.08% (validasi), dan 62.21% (pengujian).
5	(Ridhovan & Suharso, 2022)	Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum	ResNet	Berdasarkan hasil penelitian, dapat disimpulkan bahwa metode Residual Network (ResNet) mampu melakukan klasifikasi penyakit dengan tingkat akurasi sangat tinggi, yaitu lebih dari 95%, menggunakan berbagai data. Skenario 2 menghasilkan hasil terbaik dengan akurasi 98% dan confusion matrix sebesar 0,35. Untuk penelitian selanjutnya, disarankan menggunakan lebih banyak data untuk meningkatkan akurasi dan confusion matrix. Selain itu, disarankan untuk mencoba variasi parameter lapisan masukan untuk

				mengevaluasi tingkat akurasi.
--	--	--	--	-------------------------------

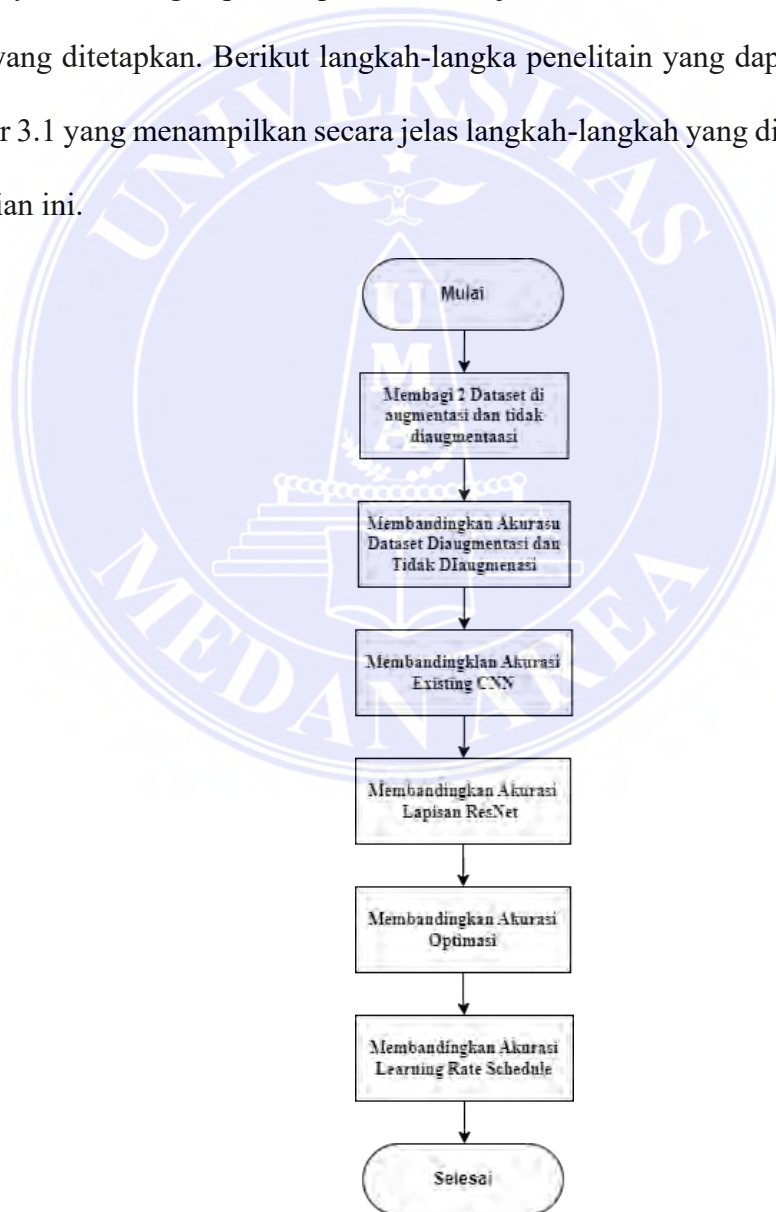


BAB III

METODELOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Dalam mendukung kelancaran dan keberhasilan penelitian diperlukan suatu tahapan rancangan penelitian atau kerangka kerja penelitian yang terstruktur. Tujuannya adalah agar proses penelitian berjalan terstruktur dan sesuai dengan target yang ditetapkan. Berikut langkah-langka penelitain yang dapat dilihat pada Gambar 3.1 yang menampilkan secara jelas langkah-langkah yang dilakukan dalam penelitian ini.



Gambar 3.1 Flowchart Alur Penelitian

Berdasarkan kerangka kerja penelitian pada Gambar 3.1 maka dapat dipaparkan pembahasan pada masing masing tahapan penelitian diuraikan sebagai berikut:

1. Membagi 2 Dataset diaugmentasi dan tidak diaugmentasi

Langkah pertama adalah membagi dataset menjadi dua bagian yaitu dataset yang telah diaugmentasi dan dataset yang tidak diaugmentasi, untuk keperluan perbandingan.

2. Membandingkan Akurasi Dataset Diaugmentasi dan Tidak Diaugmentasi

Setelah pembagian dataset, dilakukan perbandingan akurasi antara dataset yang diaugmentasi dan yang tidak diaugmentasi untuk melihat pengaruh augmentasi pada akurasi model.

3. Membandingkan Akurasi *Existing* CNN

Selanjutnya dilakukan perbandingan akurasi pada model CNN yang sudah ada untuk menentukan baseline atau acuan kinerja awal.

4. Membandingkan Akurasi Lapisan ResNet

Setelah itu, dilakukan perbandingan akurasi pada berbagai lapisan ResNet untuk melihat pengaruh jumlah lapisan terhadap akurasi model.

5. Membandingkan Akurasi Optimasi

Langkah berikutnya adalah membandingkan akurasi dari berbagai metode optimasi untuk menemukan metode yang paling efektif.

6. Membandingkan Akurasi *Learning Rate Schedule*

Selanjutnya, dilakukan perbandingan akurasi dengan berbagai strategi *learning rate schedule* untuk menentukan pengaturan *learning rate schedule* yang paling optimal.

3.2 Waktu dan Tempat

Penelitian ini dilaksanakan pada bulan Mei tahun 2024 di Universitas Medan Area, yang berlokasi di Jl. H. Agus Salim Siregar, Kenangan Baru, Kecamatan Medan Tembung, Kabupaten Deli Serdang, Sumatera Utara.

3.3 Alat dan Bahan

Dalam penelitian ini, berbagai alat dan bahan digunakan untuk melaksanakan eksperimen dan analisis data yang diperlukan. Bagian ini menyajikan daftar lengkap alat dan bahan yang digunakan dalam proses penelitian. Berikut ini alat dan bahan yang digunakan:

1. Komputer

Penelitian ini menggunakan laptop dengan spesifikasi prosesor AMD Ryzen 7 untuk penggunaan RAM dengan kapasitas 16 GB.

2. Perangkat lunak

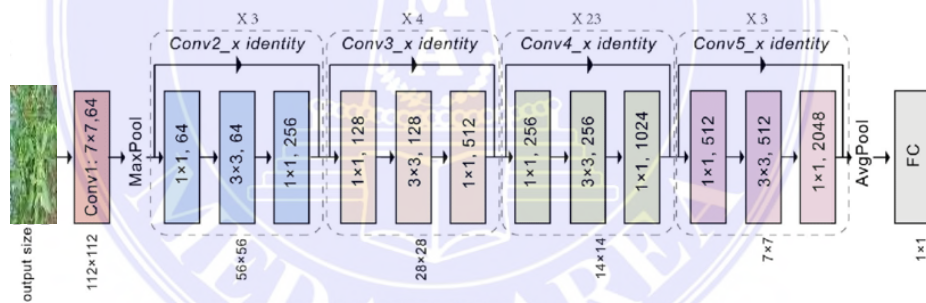
Dalam penelitian ini, peneliti menggunakan perangkat lunak yang menggunakan bahasa pemrograman python yaitu pytorch untuk aplikasi yang digunakan yaitu anaconda jupyter notebook untuk mengembangkan, melatih, dan mengevaluasi model *Convolutional Neural Network* (CNN) untuk klasifikasi penyakit pada daun cabai.

3. Bahan

Penelitian ini menggunakan dataset yang berisi gambar daun cabai dengan label kelas penyakit yang akan diklasifikasikan. Dataset ini terdiri dari 5 kelas yang terdiri dari cabai sehat, daun cabai keriting, bercak daun cabai, daun berjamur dan daun cabai berwarna kekuningan, dataset diambil dari Mendeley dataset .

3.4 Analisis Proses Tahapan Hyperparameter Tuning ResNet101

Penelitian Penelitian ini bertujuan untuk meningkatkan tingkat akurasi dalam mengklasifikasikan citra penyakit pada daun cabai dengan memanfaatkan model ResNet101 yang telah dilatih sebelumnya (pretrained). Beberapa hyperparameter akan diuji pada arsitektur ResNet101 meliputi skenario yang menggunakan dataset yang telah mengalami augmentasi dan yang tidak di augmentasi, serta skenario pelatihan atau penyetelan (tuning) pada lapisan ResNet, fungsi aktivasi, optimisasi, dan penjadwalan learning rate. Setiap skenario arsitektur model akan dilatih selama 30 epoch, yang menghasilkan beberapa model. Selanjutnya, masing-masing model akan diuji dengan data pengujian untuk mengevaluasi hasilnya, memilih model terbaik, dan akhirnya menerapkannya pada form GUI di Jupyter Notebook.



Gambar 3.1 adalah arsitektur ResNet101 yang akan digunakan

Sumber : <https://www.ikomia.ai/blog/mastering-resnet-deep-learning-image-recognition>

Hyperparameter Tuning adalah proses atau pengaturan parameter yang dilakukan untuk mendapatkan hasil model yang optimal. Berikut ini tahapan *Hyperparameter Tuning* ResNet101 :

1. pada bagian input layer, citra yang digunakan memiliki ukuran 227x227x3.

Angka 3 menunjukkan jumlah kanal citra (RGB) pada gambar berwarna (*true*

color). Sebelum citra dimasukkan ke dalam model, terlebih dahulu dilakukan proses augmentasi sesuai dengan yang telah dijelaskan pada bagian augmentasi dataset sebelumnya. Hasil augmentasi ini kemudian digunakan sebagai input untuk model pelatihan selanjutnya. Adapun proses konvolusi pada channel Red. Proses konvolusi pada layer pertama dengan stride 1 dan zero padding menggunakan persamaan (2.1) yang dapat di simulasikan sebagai berikut:

$$S(i,j) = (K * I)(i,j) = \sum_{i=1}^m \sum_{j=1}^n (I(i-m,j-n) * K)(m,n) \quad (2.1)$$

Keterangan :

S(i,j): Nilai hasil konvolusi di posisi (i,j)

I(i-m,j-n) : Elemen gambar yang sedang dikonvolusi.

K(m,n) : Elemen kernel pada posisi (m,n)

\sum : Penjumlahan nilai perkalian elemen-elemen dalam area kernel.

1. Posisikan kernel di sudut kiri atas piksel citra, kemudian hitung menggunakan rumus pada persamaan (2.1).

150	160	170	180	190	1	-1	1	60		
140	135	130	125	120	-1	2	-1			
110	115	105	100	95	-1	1	-1			
85	90	80	70	60						
50	55	65	75	85						

$$S(i,j) = (1 * 150) + (-1 * 160) + (1 * 170) + (-1 * 140) + (2 * 135) + (-1 * 130) \\ + (-1 * 110) + (1 * 115) + (-1 * 105) \\ = 60$$

2. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1).

150	160	170	180	190	1	-1	1	60	70	
140	135	130	125	120	-1	2	-1			
110	115	105	100	95	-1	1	-1			
85	90	80	70	60						
50	55	65	75	85						

$$S(i,j) = (1 * 160) + (-1 * 170) + (1 * 180) + (-1 * 135) + (2 * 130) + (-1 * 125) \\ + (-1 * 115) + (1 * 105) + (-1 * 100) \\ = 70$$

3. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1).

150	160	170	180	190	1	-1	1	60	70	80
140	135	130	125	120	-1	2	-1			
110	115	105	100	95	-1	1	-1			
85	90	80	70	60						
50	55	65	75	85						

$$S(i,j) = (1 * 170) + (-1 * 180) + (1 * 190) + (-1 * 130) + (2 * 125) + (-1 * 120) \\ + (-1 * 105) + (1 * 100) + (-1 * 95) \\ = 80$$

4. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1).

150	160	170	180	190
140	135	130	125	120
110	115	105	100	95
85	90	80	70	60
50	55	65	75	85

1	-1	1
-1	2	-1
-1	1	-1

60	70	80
65		

$$\begin{aligned}
 S(i,j) &= (1 * 140) + (-1 * 135) + (1 * 130) + (-1 * 110) + (2 * 115) + (-1 * 105) \\
 &+ (-1 * 90) + (1 * 85) + (-1 * 80) \\
 &= 65
 \end{aligned}$$

5. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1).

150	160	170	180	190
140	135	130	125	120
110	115	105	100	95
85	90	80	70	60
50	55	65	75	85

1	-1	1
-1	2	-1
-1	1	-1

60	70	80
65	35	

$$\begin{aligned}
 S(i,j) &= (1 * 135) + (-1 * 130) + (1 * 125) + (-1 * 115) + (2 * 105) + (-1 * \\
 &100) + (-1 * 90) + (1 * 80) + (-1 * 70) \\
 &= 25
 \end{aligned}$$

6. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1)

150	160	170	180	190	1	-1	1	60	70	80
140	135	130	125	120	-1	2	-1	65	35	45
110	115	105	100	95	-1	1	-1			
85	90	80	70	60						
50	55	65	75	85						

$$\begin{aligned}
 S(i,j) &= (1 * 130) + (-1 * 125) + (1 * 120) + (-1 * 105) + (2 * 100) + (-1 * 95) \\
 &+ (-1 * 80) + (1 * 70) + (-1 * 60) \\
 &= 45
 \end{aligned}$$

7. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1)

150	160	170	180	190	1	-1	1	60	70	80
140	135	130	125	120	-1	2	-1	65	35	45
110	115	105	100	95	-1	1	-1	55		
85	90	80	70	60						
50	55	65	75	85						

$$\begin{aligned}
 S(i,j) &= (1 * 110) + (-1 * 115) + (1 * 105) + (-1 * 85) + (2 * 90) + (-1 * 80) + \\
 &(-1 * 50) + (1 * 55) + (-1 * 65) \\
 &= 55
 \end{aligned}$$

8. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1).

150	160	170	180	190
140	135	130	125	120
110	115	105	100	95
85	90	80	70	60
50	55	65	75	85

1	-1	1
-1	2	-1
-1	1	-1

60	70	80
65	35	45
55	50	

$$S(i,j) = (1 * 115) + (-1 * 105) + (1 * 100) + (-1 * 90) + (2 * 80) + (-1 * 70) + (-1 * 55) + (1 * 65) + (-1 * 75) = 50$$

9. Pindahkan kernel satu piksel ke kanan, kemudian hitung menggunakan rumus pada persamaan (2.1)

150	160	170	180	190
140	135	130	125	120
110	115	105	100	95
85	90	80	70	60
50	55	65	75	85

1	-1	1
-1	2	-1
-1	1	-1

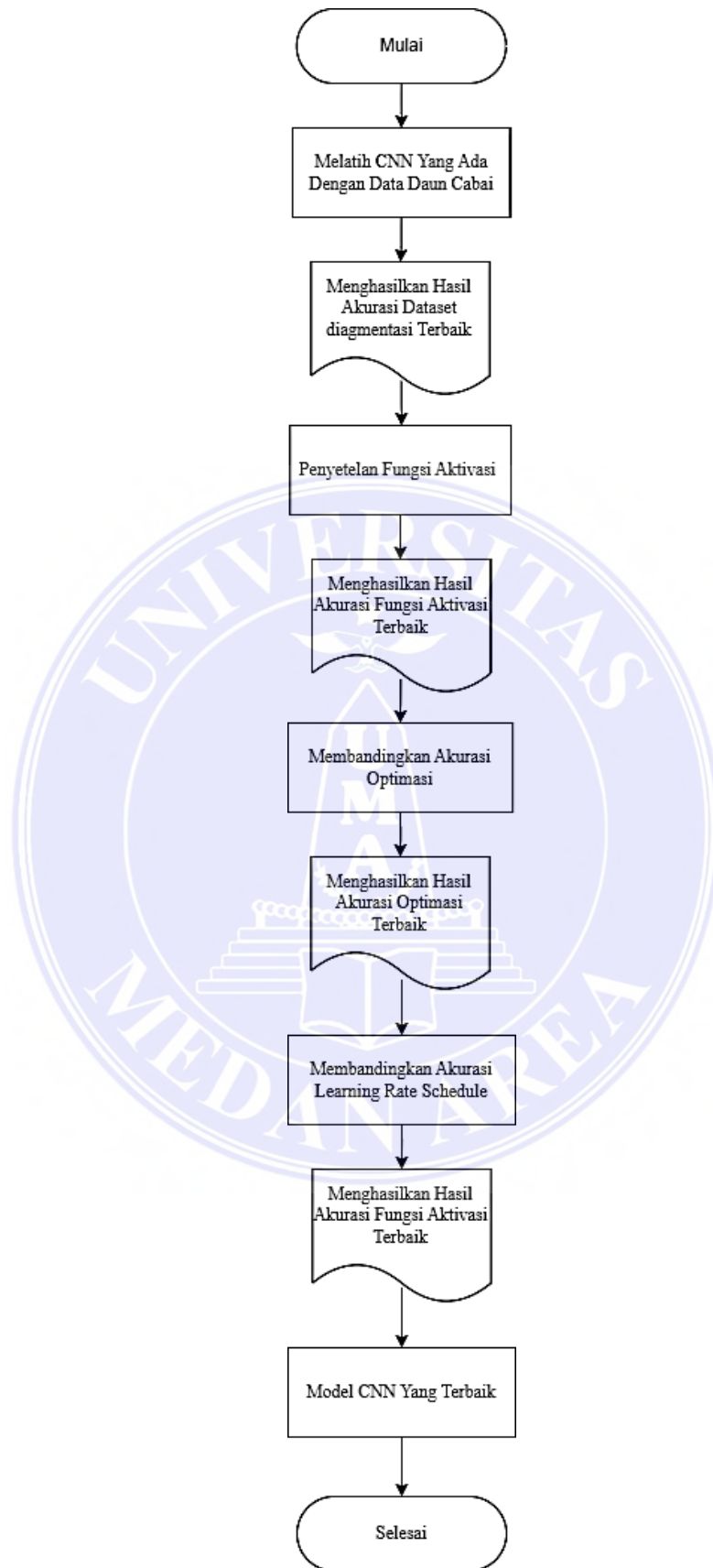
60	70	80
65	35	45
55	50	25

$$S(i,j) = (1 * 105) + (-1 * 100) + (1 * 95) + (-1 * 80) + (2 * 70) + (-1 * 60) + (-1 * 65) + (1 * 75) + (-1 * 85) = 25$$

2. Untuk mengekstraksi fitur-fitur dari dataset gambar, digunakan model pretrained dengan arsitektur ResNet101. Beberapa parameter dari arsitektur ResNet101 yang telah disesuaikan dalam penelitian ini meliputi fungsi aktivasi, optimasi, dan *learning rate schedule*.

3. Untuk mengembangkan model CNN terbaik dalam klasifikasi daun cabai, dilakukan berbagai langkah penting, Setiap tahapan-tahapan dirancang agar model mampu mengenali dan mengklasifikasikan daun cabai dengan akurasi tinggi, sehingga dapat menghasilkan performa yang optimal pada data uji. berikut ini *Flowchart Hyperparameter Tuning CNN* :





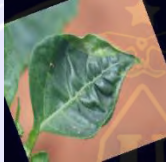




Gambar 3.2 *Flowchart Hyperparameter Tuning CNN*

Pada gambar 3.2 Menjelaskan diagram alir (*flowchart*) dari proses *fine tuning* ResNet101. Proses dimulai dengan melatih model CNN menggunakan data daun cabai untuk mendapatkan akurasi terbaik pada dataset yang telah diaugmentasi, dilanjutkan dengan penyetelan fungsi aktivasi untuk mencapai akurasi fungsi aktivasi terbaik, kemudian membandingkan metode optimasi untuk memperoleh hasil akurasi optimasi terbaik, serta membandingkan berbagai strategi *learning rate schedule* untuk mendapatkan akurasi optimal berdasarkan fungsi aktivasi yang paling sesuai, dan akhirnya menghasilkan model CNN terbaik, yang menandakan proses selesai.

Dataset yang dipakai dalam penelitian ini berupa kumpulan citra penyakit pada daun cabai merah yang bersumber dari mendeley data (M P, Aishwarya, Reddy, Padmanabha, 2024) Dataset ini mencakup lima kelas penyakit dan terbagi menjadi dua jenis, yaitu dataset yang telah diaugmentasi (*augmented*) dan yang tidak diaugmentasi. Sebelum dilakukan augmentasi, dataset terdiri atas 531 citra, sedangkan setelah proses augmentasi, jumlah citra meningkat menjadi 2.128, untuk data training sebanyak 1.702 citra daun cabai dengan rincian bercak daun 486, daun sehat 221, daun kriting 342, daun kekuningan 326, daun berjamur 327 sedangkan untuk data validasinya berjumlah 426 citra dengan rincian bercak daun 122, daun sehat 55, daun kriting 86, daun kekuningan 82, daun berjamur 81 citra yang telah diolah pada tahap praproses dan augmentasi dataset. Dataset ini terbagi menjadi data pelatihan dengan persentase 705 dan data validasi 20%.

Tabel 2.1 Pembagian Dataset Citra

Kelas	Gambar	Training	Validation	Total
Daun Berjamur		486	122	608
Daun Sehat		221	55	276
Daun Keriting		342	86	428
Bercak Daun		326	82	408
Daun Kekuningan		327	81	408
Total		1702	426	2128

Berdasarkan Tabel 3.1 dataset dibagi menjadi lima kelas yaitu daun cabai sehat, daun keriting, daun kekuningan, daun berjamur dan daun berbintik.

3.5 Rancangan Form

Pada sistem ini memiliki tampilan form yang akan di buat menggunakan pytorch yang menggunakan bahasa pemrograman *pyhton*. Berikut ini adalah Rancangan form klasifikasi penyakit daun cabai:

The image shows a web form titled "Form Penyakit Daun Cabai". It features three distinct sections: a file upload button labeled "Unggah File" with a small box containing the number "1"; a large rectangular area labeled "Gambar Hasil Klasifikasi" with a small box containing the number "2"; and a label "Prediksi Penyakit :" followed by a small box containing the number "3". The form is overlaid on a watermark of Universitas Medan Area.

Gambar 3.3 Rancangan Form

Adapun rancangan form aplikasi pendeteksi penyakit daun cabai terdapat beberapa bagian yaitu form gambar, pencarian, hasil klasifikasi dan hasil deskripsi penyakit daun cabai. Berikut penjelasan singkat dari bagian bagian desain rancangan form klasifikasi penyakit daun cabai :

1. Tombol ini digunakan untuk mengunggah file gambar daun cabai yang akan diklasifikasikan.
2. Bagian ini akan menampilkan gambar hasil klasifikasi setelah proses pengunggahan dan klasifikasi selesai.
3. Bagian ini akan menampilkan hasil prediksi penyakit berdasarkan gambar yang diunggah dan telah diklasifikasikan oleh sistem.

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai *hyperparameter tuning convolutional network* untuk klasifikasi penyakit pada daun cabai kesimpulannya bahwa pada arsitektur ResNet101, dengan penerapan augmentasi dataset, penggunaan fungsi aktivasi Tanh, optimizer SGD, dan *learning rate schedule* ReduceLR dengan patience 3, memberikan peningkatan akurasi yang sangat baik dalam klasifikasi penyakit pada daun cabai, dengan akurasi latih mencapai 97.7673% dan akurasi validasi mencapai 99.5305%.

5.2 Saran

saran untuk penelitian selanjutnya terus mengembangkan teknik augmentasi data guna meningkatkan variasi dataset yang terbukti dapat meningkatkan akurasi model secara signifikan. Selain itu, eksplorasi arsitektur CNN lainnya seperti EfficientNet atau MobileNet perlu dilakukan untuk membandingkan performa dan efisiensinya. Mengimplementasikan model dalam aplikasi untuk mengevaluasi efektivitas model dengan menggunakan aplikasi berbasis web maupun perangkat mobile. Dengan mengikuti saran-saran ini, diharapkan penelitian di bidang klasifikasi penyakit tanaman menggunakan *deep learning* dapat terus berkembang dan memberikan kontribusi dalam dunia akademik.

DAFTAR PUSTAKA

- Almansour, A., Alawad, M., Aljouie, A., Almatar, H., Qureshi, W., Alabdulkader, B., Alkanhal, N., Abdul, W., Almufarrej, M., Gangadharan, S., Aldebasi, T., Alsomaie, B., & Almazroa, A. (2022). Peripapillary atrophy classification using CNN deep learning for glaucoma screening. *PLoS ONE*, *17*(10 October), 1–25. <https://doi.org/10.1371/journal.pone.0275446>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Dalam *Journal of Big Data* (Vol. 8, Nomor 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>
- Anggraeni, D. S., Widayana, A., Rahayu, P. D., & Rozikin, C. (2022). Metode Algoritma Convolutional Neural Network pada Klasifikasi Penyakit Tanaman Cabai. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, *7*(1), 73. <https://doi.org/10.30998/string.v7i1.13304>
- Astuti, F. S., Wanto, H. S., & Koesriwulandari, K. (2021). ELASTISITAS PERMINTAAN CABAI MERAH (*Capsicum annum* L.) DI KOTA SURABAYA. *Jurnal Ilmiah Sosio Agribis*, *21*(1), 76–93. <https://doi.org/10.30742/jisa21120211343>
- Caesarendra, W., Triwiyanto, T., Pandiyan, V., Glowacz, A., Permana, S. D. H., & Tjahjowidodo, T. (2021). A cnn prediction method for belt grinding tool wear in a polishing process utilizing 3-axes force and vibration data. *Electronics (Switzerland)*, *10*(12), 1–30. <https://doi.org/10.3390/electronics10121429>
- Chavez, K., & Hernando, L. (2023). Implementasi Sistem Pendeteksi Penyakit Pada Daun Singkong Dan Daun Cabai Berbasis Machine Learning. *Jurnal Quancom*, *1*(2), 1–5.
- Devianto, Y., Dwiasnati, S., Sukowo, B., Fauzi, A., & Baihaqi, K. A. (2023). Penerapan Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) untuk Mendiagnosa Penyakit Bercak Daun Cabai. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *3*(2), 136–142. <https://doi.org/10.57152/malcom.v3i2.850>
- Dzaky, A. T. R. (2021). Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network. *e-Proceeding of Engineering*, *8*(2), 3040–3055.
- Falakhi, B., Achmal, E. F., Rizaldi, M., Athallah, R. R. R., & Yudistira, N. (2022). Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Bunga Memanfaatkan Transfer Learning. *Jurnal Ilmu Komputer dan Agri-Informatika*, *9*(1), 70–78. <https://doi.org/10.29244/jika.9.1.70-78>
- Geza Jeremia Bu'ulölö, A. J. F. D. K. (2021). *jm_informatika*, +34208-73917-1-ED+375-382. *16*(4), 375–382.
- Hidayat, A., Darusalam, U., & Irmawati, I. (2019). Detection of Disease on Corn Plants Using Convolutional Neural Network Methods. *Jurnal Ilmu Komputer dan Informasi*, *12*(1), 51. <https://doi.org/10.21609/jiki.v12i1.695>
- IBRAHIM, N., LESTARY, G. A., HANAFLI, F. S., SALEH, K., PRATIWI, N. K. C., HAQ, M. S., & MASTUR, A. I. (2022). Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network.

- ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(1), 162. <https://doi.org/10.26760/elkomika.v10i1.162>
- Karno, A., Hastomo, W., Wardhana, I., Sutarno, & Arif, D. (2022). 29 Jenis Penyakit Tanaman Menggunakan Deep Learning EfficientNetB3 Identifikasi. *Insearch: Information System Research Journal*, 2(2), 35–45.
- Lasniari, S., Jasril, J., Sanjaya, S., Yanto, F., & Affandes, M. (2022). Klasifikasi Citra Daging Babi dan Daging Sapi Menggunakan Deep Learning Arsitektur ResNet-50 dengan Augmentasi Citra. *Jurnal Sistem Komputer dan Informatika (JSON)*, 3(4), 450. <https://doi.org/10.30865/json.v3i4.4167>
- Leo, H., Arnia, F., & Munadi, K. (2022). Fine Tuning CNN Pre-trained Model Based on Thermal Imaging for Obesity Early Detection. *Jurnal Rekayasa ElektriKa*, 18(1), 53–60. <https://doi.org/10.17529/jre.v18i1.25100>
- Lesmana, A. M., Fadhillah, R. P., & Rozikin, C. (2022). Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN). *Jurnal Sains dan Informatika*, 8(1), 21–30. <https://doi.org/10.34128/jsi.v8i1.377>
- Lukas Hansel, G., & Bunyamin, H. (2021). Penggunaan Augmentasi Data pada Klasifikasi Jenis Kanker Payudara dengan Model Resnet-34. *Jurnal Strategi*, 3(1), 187–193.
- Mustamu, N. E., Dalimunthe, B. A., Hartati, S., & Saragih, Y. (2023). Aplikasi Pemberian Abu Sekam Padi dan Pupuk Organik Cair (POC) Terhadap Produksi Tanaman Cabai Merah (*Capsicum annum L .*). *Jurnal Mahasiswa Agroteknologi (Jmatek)*, 4, 56–63.
- Naufal, M. F., & Kusuma, S. F. (2021). Pendeteksi Citra Masker Wajah Menggunakan CNN dan Transfer Learning. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(6), 1293. <https://doi.org/10.25126/jtiik.2021865201>
- Nihayatul Husna, I., Ulum, M., Kurniawan Saputro, A., Tri Laksono, D., & Neipa Purnamasari, D. (2022). Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN). *Seminar Nasional Fortei Regional*, 7(Vol. 5 No. 1 (2022): SinarFe7-5 2022), 3–8.
- Nisa', C., Puspaningrum, E. Y., & Maulana, H. (2020). Seminar Nasional Informatika Bela Negara (SANTIKA) Penerapan Metode Convolutional Neural Network untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data. *Seminar Nasional Informatika Bela Negara (SANTIKA)*, 1, 1–7.
- Noprisson, H. (2022). Fine-Tuning Model Transfer Learning VGG16 Untuk Klasifikasi Citra Penyakit Tanaman Padi. *JSAI (Journal Scientific and Applied Informatics)*, 5(3), 244–249. <https://doi.org/10.36085/jsai.v5i3.3609>
- Poojary, R., Raina, R., & Mondal, A. K. (2021). Effect of data-augmentation on fine-tuned cnn model performance. *IAES International Journal of Artificial Intelligence*, 10(1), 84–92. <https://doi.org/10.11591/ijai.v10.i1.pp84-92>
- Rahman, S., Ramli, M., Arnia, F., Muharar, R., & Sembiring, A. (2021). Performance analysis of mAlexnet by training option and activation function tuning on parking images. *IOP Conference Series: Materials Science and Engineering*, 1087(1), 012084. <https://doi.org/10.1088/1757-899x/1087/1/012084>
- Ridhovan, A., & Suharso, A. (2022). Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum. *JUPI (Jurnal Ilmiah*

- Penelitian dan Pembelajaran Informatika*), 7(1), 58–65.
<https://doi.org/10.29100/jipi.v7i1.2410>
- Rosadi, M. I., & Lutfi, M. (2021). Identifikasi Jenis Penyakit Daun Jagung Menggunakan Deep Learning Pre-Trained Model. *Jurnal Explore IT!*, 13(2), 36–42.
- Sasongko, T. B., Haryoko, H., & Amrullah, A. (2023). Analisis Efek Augmentasi Dataset dan Fine Tune pada Algoritma Pre-Trained Convolutional Neural Network (CNN). *Jurnal Teknologi Informasi dan Ilmu Komputer*, 10(4), 763–768. <https://doi.org/10.25126/jtiik.20241046583>
- Solihin, A., Mulyana, D. I., & Yel, M. B. (2022). Klasifikasi Jenis Alat Musik Tradisional Papua menggunakan Metode Transfer Learning dan Data Augmentasi. *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, 5(2), 36–44. <https://doi.org/10.47970/siskom-kb.v5i2.279>
- Tilasefana, R. A., & Putra, R. E. (2023). Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca. *Journal of Informatics and Computer Science (JINACS)*, 05(1), 48–57.
- Togatorop, P. R., & Fauzi, A. (2022). Klasifikasi Penggunaan Masker Wajah Menggunakan Squeezenet. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 9(1), 397–406. <https://doi.org/10.35957/jatisi.v9i1.642>
- Wakhidah, N., Kasrina, K., & Bustamam, H. (2021). KEANEKARAGAMAN JAMUR PATOGEN PADA TANAMAN CABAI MERAH (*Capsicum annum* L.) DI DATARAN RENDAH. *Konservasi Hayati*, 17(2), 63–68. <https://doi.org/10.33369/hayati.v17i2.17920>
- Wiratama, A. S., Rifqi, M., & Maesaroh, S. (2023). Efektivitas Transfer Learning Dalam Pendeteksian Penyakit Pneumonia Melalui Citra X-Ray Paru Manusia. *Jurnal Ilmiah Sains dan Teknologi*, 7(1), 43–52. <https://doi.org/10.47080/saintek.v7i1.2551>

LAMPIRAN

Lampiran 1 Kode Program Klasifikasi Menggunakan Anaconda jupyter Notebook

```

import torch

import torch.nn as nn

import torch.optim as optim

import torchvision

from torchvision import transforms, datasets

from torch.utils.data import DataLoader, random_split

def pilih_model_cnn_pretrained(nama_model, num_classes,
weights='IMAGENET1K_V1'):

    if nama_model not in ['resnet101', 'vgg16', 'alexnet',
'shufflenet']:

        raise ValueError("Nama model tidak valid. Pilih dari
'resnet101', 'vgg16', 'alexnet', atau 'shufflenet'.")

    if nama_model == "resnet101":

        model =
torchvision.models.resnet101(weights=torchvision.models.ResNet101_
Weights.IMAGENET1K_V1)

        num_features = model.fc.in_features

        model.fc = nn.Sequential(

            nn.Linear(num_features, num_classes),

            nn.Tanh()

        )

    elif nama_model == "vgg16":

        model =
torchvision.models.vgg16(weights=torchvision.models.VGG16_Weights.
IMAGENET1K_V1)

        num_features = model.classifier[6].in_features

        model.classifier[6] = nn.Sequential(

            nn.Linear(num_features, num_classes),

            nn.Tanh()

        )

    elif nama_model == "alexnet":

```

```

        model =
torchvision.models.alexnet(weights=torchvision.models.AlexNet_Weights.IMAGENET1K_V1)

        num_features = model.classifier[6].in_features

        model.classifier[6] = nn.Sequential(
            nn.Linear(num_features, num_classes),
            nn.Tanh()
        )

    elif nama_model == "shufflenet":

        model =
torchvision.models.shufflenet_v2_x1_0(weights=torchvision.models.ShuffleNet_V2_X1_0_Weights.IMAGENET1K_V1)

        num_features = model.fc.in_features

        model.fc = nn.Sequential(
            nn.Linear(num_features, num_classes),
            nn.Tanh()
        )

    return model

def latih_model(model, train_loader, val_loader, criterion,
optimizer, scheduler, num_epochs, patience):

    device = torch.device('cuda' if torch.cuda.is_available() else
'cpu')

    model.to(device)

    print(f"\nTesting with patience = {patience}\n")

    print("| epoch | Train Error | Train Accuracy |
Val Error | Val Accuracy |")

    for epoch in range(num_epochs):

        model.train()

        running_loss = 0.0

        correct_train = 0

        total_train = 0

```



```
for inputs, labels in train_loader:
    inputs, labels = inputs.to(device), labels.to(device)

    optimizer.zero_grad()
    outputs = model(inputs)
    loss = criterion(outputs, labels)
    loss.backward()
    optimizer.step()

    running_loss += loss.item() * inputs.size(0)

    _, predicted_train = torch.max(outputs.data, 1)
    total_train += labels.size(0)
    correct_train += (predicted_train ==
labels).sum().item()

epoch_loss = running_loss / len(train_loader.dataset)
epoch_accuracy_train = 100 * correct_train / total_train

# Validasi
model.eval()
correct_val = 0
total_val = 0
val_loss = 0.0

with torch.no_grad():
    for inputs_val, labels_val in val_loader:
        inputs_val, labels_val = inputs_val.to(device),
labels_val.to(device)

        outputs_val = model(inputs_val)
        loss_val = criterion(outputs_val, labels_val)
        val_loss += loss_val.item() * inputs_val.size(0)
```

```

        _, predicted_val = torch.max(outputs_val.data, 1)
        total_val += labels_val.size(0)
        correct_val += (predicted_val ==
labels_val).sum().item()

    epoch_loss_val = val_loss / len(val_loader.dataset)
    epoch_accuracy_val = 100 * correct_val / total_val

    scheduler.step(epoch_loss_val) # Step the scheduler with
validation loss

    print(f"| {epoch+1:4d} | {epoch_loss:.4f} |
{epoch_accuracy_train:.4f} | {epoch_loss_val:.4f} |
{epoch_accuracy_val:.4f} |")

if __name__ == "__main__":
    nama_model = "resnet101" # Pilih model pre-trained yang
diinginkan ("resnet101", "vgg16", "alexnet", atau "shufflenet")
    num_classes = 5 # Jumlah kelas untuk tugas Anda
    pretrained = True # Menggunakan pre-trained weights
    num_epochs = 30

    train_path = r'D:\SKRIPSI\Datafix\Training'
    val_path = r'D:\SKRIPSI\Datafix\Validasi'

    train_transform = transforms.Compose([
        transforms.RandomResizedCrop(227),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
std=[0.229, 0.224, 0.225]),
    ])

    val_transform = transforms.Compose([
        transforms.Resize((227, 227)),
        transforms.ToTensor(),

```

```

        transforms.Normalize(mean=[0.485,      0.456,      0.406],
                             std=[0.229, 0.224, 0.225]),
    ])

    train_dataset = datasets.ImageFolder(root=train_path,
                                         transform=train_transform)

    val_dataset = datasets.ImageFolder(root=val_path,
                                       transform=val_transform)

    criterion = nn.CrossEntropyLoss()

    patience_values = [3, 5, 7, 10, 15]

    for patience in patience_values:
        # Perlu membuat ulang model dan optimizer untuk setiap nilai
        # patience
        model_custom = pilih_model_cnn_pretrained(nama_model,
                                                  num_classes, pretrained)

        optimizer = optim.SGD(model_custom.parameters(), lr=0.001,
                               momentum=0.9)

        scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer,
                                                         mode='min',
                                                         factor=0.1,
                                                         patience=patience,
                                                         verbose=True)

        train_loader = DataLoader(train_dataset, batch_size=32,
                                  shuffle=True)

        val_loader = DataLoader(val_dataset, batch_size=32,
                                shuffle=False)

        latih_model(model_custom, train_loader, val_loader,
                    criterion, optimizer, scheduler, num_epochs, patience)

    # Simpan model yang telah di-fine-tune

    torch.save(model_custom.state_dict(), 'fine_tuned_RR_ResNet101_Tanh
    _ReduceLR_patience_pretrained_model.pth')

```

Lampiran 2 Kode Program Gui Menggunakan Jupyter Notebook

```

import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk
import torch
import torchvision.transforms as transforms
import matplotlib.pyplot as plt
import torchvision.models as models
import torch.nn as nn

# Define transform untuk gambar baru
image_transform = transforms.Compose([
    transforms.Resize((227, 227)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
0.224, 0.225]),
])

def predict_image(image_path, model, image_transform):
    image = Image.open(image_path)
    image = image_transform(image).unsqueeze(0)
    outputs = model(image)
    _, predicted = torch.max(outputs, 1)
    confidence = torch.nn.functional.softmax(outputs,
dim=1)[0][predicted[0]].item()
    return predicted.item(), confidence

def display_image_classification(image_path):
    predicted_class, confidence = predict_image(image_path, model,
image_transform)
    class_names = ["Bercak Daun", "Daun
Sehat", "Kriting", "Kekuningan", "Daun Berjamur"]

    # Tampilkan gambar pada GUI
    image = Image.open(image_path)
    image = image.resize((300, 300)) # Sesuaikan ukuran gambar
    img = ImageTk.PhotoImage(image)

    panel = tk.Label(root, image=img)
    panel.image = img
    panel.pack()

    # Tampilkan hasil klasifikasi pada GUI
    result_label = tk.Label(root, text=f'Prediksi:
{class_names[predicted_class]} (Confidence: {confidence:.4f})')
    result_label.pack()

def browse_image():
    file_path = filedialog.askopenfilename()
    if file_path:
        display_image_classification(file_path)

model = models.resnet101(pretrained=True)
#model.fc = torch.nn.Linear(model.fc.in_features, 5) # Mengganti
layer fully connected untuk 2 kelas

```

```
num_features = model.fc.in_features
model.fc = nn.Sequential(
    nn.Linear(num_features, 5),
    nn.Tanh()
)

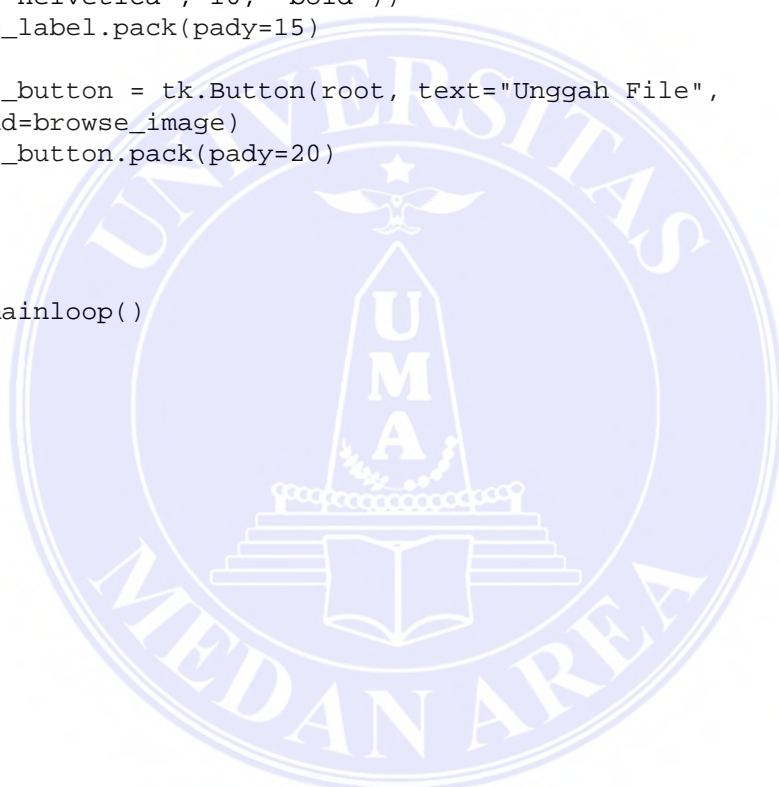
model.load_state_dict(torch.load('fine_tuned_RR_ResNet101_Tanh_Red
uceLR_patience_pretrained_model.pth'))
model.eval()

root = tk.Tk()
root.title("Form Penyakit Daun Cabai")
root.geometry("600x600") # Menyesuaikan ukuran jendela

header_label = tk.Label(root, text="Form Penyakit Daun Cabai",
font=("Helvetica", 16, "bold"))
header_label.pack(pady=15)

browse_button = tk.Button(root, text="Unggah File",
command=browse_image)
browse_button.pack(pady=20)

root.mainloop()
```



SKRIPSI RAHMAT ARIEF SETYADI 208160026 ANALISIS KINERJA HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK PADA KASUS KLASIFIKASI PENYAKIT DAUN CABAI .pdf

ORIGINALITY REPORT

23%
SIMILARITY INDEX

20%
INTERNET SOURCES

11%
PUBLICATIONS

3%
STUDENT PAPERS

PRIMARY SOURCES

1 jurnal.dharmawangsa.ac.id **5%**
Internet Source

2 repositori.uma.ac.id **4%**
Internet Source

3 digilib.unila.ac.id **2%**
Internet Source

4 Rahmat Arief Setyadi, Sayuti Rahman, Dionikxon Manurung, Mardiatul Hasanah, Asmah Indrawati. "IMPLEMENTASI TRANSFER LEARNING UNTUK KLASIFIKASI PENYAKIT PADA DAUN CABAI MENGGUNAKAN CNN", Djtechno: Jurnal Teknologi Informasi, 2024 **1%**
Publication

5 openlibrarypublications.telkomuniversity.ac.id **1%**
Internet Source

6 thesai.org **1%**
Internet Source

machinelearning.mipa.ugm.ac.id

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 31/1/25

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

Access From (repository.uma.ac.id)31/1/25



UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 (061) 7366878, 7360168, 7364348, 7366781, Fax.(061) 7366990 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.iastrik.uma.ac.id E-mail: univ_medanama@uma.ac.id

Nomor : 72/FT.6/01.10/I/2024
Lamp : -
Hal : Pembimbing Tugas Akhir

24 Januari 2024

Yth. Pembimbing Tugas Akhir
Dr. Sayuti Rahman, ST, M.Kom
di
Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

Nama : Rahmat Arief Setyadi
N P M : 208160026
Jurusan : Teknik Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :

Dr. Sayuti Rahman, ST, M.Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

"Fine Tuning CNN untuk Klasifikasi Penyakit pada Daun Cabai".

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.

Dekan,

Dr. Sayuti Rahman, ST, MT
FAKULTAS TEKNIK



UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kalam Nomor 1 Medan Estate ☎ (061) 7360168, 7366878, 7364348 📠 (061) 7368012 Medan 20223
Kampus II : Jalan Selabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 8225802 📠 (061) 8226331 Medan 20122
Website: www.uma.ac.id E-Mail: univ_medanarea@uma.ac.id

Nomor : 717/UMA/B/01.7/V/2024

25 Mei 2024.

Lamp. : -

Hal : Izin Penelitian Dan Pengambilan Data Tugas Akhir

Kepada Yth.

Dekan Fakultas Teknik

Universitas Medan Area

di - Medan

Dengan hormat, sesuai dengan surat dari Dekan Fakultas Teknik Universitas Medan Area nomor : 247/FT.6/01.10/V/2024 tertanggal 20 Mei 2024, perihal izin penelitian dan pengambilan data Tugas Akhir di Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area oleh mahasiswa sebagai berikut :

Nama : Rahmat Arief Setyadi
NPM : 208160026
Program Studi : Teknik Informatika

Dengan Judul Penelitian "*Fine Tuning Convolutional Neural Network Untuk Klasifikasi Penyakit Pada Daun Cabai*".

Pada Prinsipnya disetujui yang bersangkutan untuk melaksanakan pengambilan data di Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area guna penyusunan Skripsi untuk kepentingan Ilmiah dan Akademik.

Demikian surat ini diterbitkan untuk dipergunakan sebagaimana mestinya.

Wakil Rektor Bidang Mutu SDM dan



Dr. Debi Sabputra, S.Sos, MA

Tembusan :

1. Mahasiswa Ybs
2. File







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 Sialitabudi 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 : 247 /FT.6/01.10/V/2024 20 Mei 2024
Lamp : -
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

Yth. Wakil Rektor Bidang Mutu Sumber Daya Manusia dan Perekonomian
Jln. Kolam No.1
Di
Medan

Dengan hormat, kami mohon kesediaan bapak kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	N A M A	N P M	PRODI
1	Rahmat Arief Setyadi	208160026	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di **Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area**.

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 I Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

Fine Tuning Convolutional Neural Network untuk Klasifikasi Penyakit pada Daun Cabai.

Mohon kiranya tanggal Surat Izin Pengambilan Data Tugas Akhir agar disesuaikan dengan tanggal Terbitnya Surat ini.

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Dekan

Dekan, ST, MT

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



UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, 7366878, 7364348 ☎ (061) 7368012 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 8225602 ☎ (061) 8226331 Medan 20122
Website: www.uma.ac.id E-Mail: univ_medanarea@uma.ac.id

SURAT KETERANGAN

Nomor : 869/UMA/B/01.7/VI/2024

Rektor Universitas Medan Area dengan ini menerangkan bahwa :

Nama : Rahmat Arief Setyadi
No.Pokok Mahasiswa : 208160026
Program Studi : Teknik Informatika

Benar telah selesai Pengambilan Data di Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area dengan Judul Skripsi " *Fine Tuning Convolutional Neural Network Untuk Klasifikasi Penyakit Pada Daun Cabai*".

Dan kami harapkan Data tersebut kiranya dapat membantu yang bersangkutan dalam penyusunan Skripsi dan dapat bermanfaat bagi mahasiswa khususnya Fakultas Teknik Universitas Medan Area.

Demikian surat ini diterbitkan untuk dapat digunakan seperlunya

Medan, 21 Juni 2024.
a.n Rektor
Wakil Rektor Bidang Mutu SDM &
Ekonomian,



Medan Dr. Dedih Sahputra, S.Sos, MA

CC :
- Arsip

