

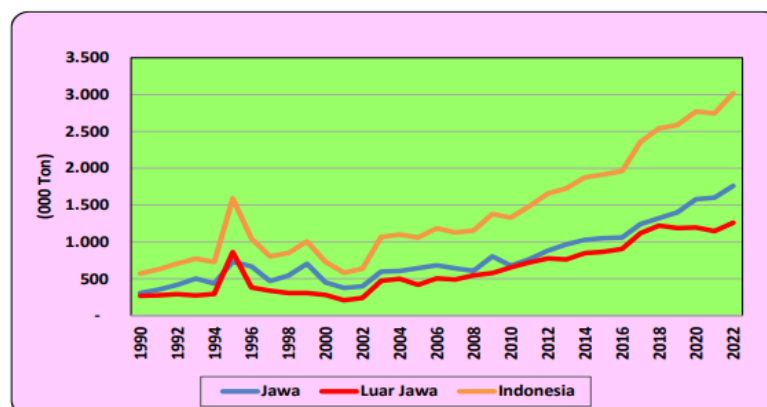
BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal sebagai negara agraris dengan mayoritas penduduknya bekerja di sektor pertanian. Keadaan ini didukung oleh keberadaan lahan kosong yang luas dan kondisi tanah yang subur, ideal untuk pertumbuhan berbagai tanaman. Salah satu tanaman hortikultura yang memiliki permintaan tinggi dari berbagai kalangan adalah cabai [1].

Cabai merupakan komoditas yang paling banyak dikonsumsi masyarakat yang berperan besar dalam pembentukan inflasi nasional, selain dari pada itu cabai juga memiliki dampak cukup besar terhadap kebutuhan masyarakat. Karena hal itu, cabai merupakan komoditas yang strategis di Indonesia. Dengan menjadi komoditas yang paling banyak dikonsumsi masyarakat, produksi cabai yang terus meningkat sangat diperlukan untuk memenuhi kebutuhan konsumen masyarakat di Indonesia [2].



Gambar 1. 1 Perkembangan Produksi Cabai Di Jawa, Luar Jawa Dan Indonesia, Tahun 1990-2022

Sumber : [3]

Produksi cabai di Indonesia terus mengalami peningkatan signifikan dari tahun ke tahun, sebagaimana terlihat pada tren grafik produksi nasional yang terus naik sejak tahun 1990 hingga 2022. Hal ini menunjukkan bahwa cabai merupakan komoditas pertanian yang semakin penting dan bernilai tinggi, baik di wilayah Jawa maupun luar Jawa [4].

Namun, di balik urgensi peningkatan produksi cabai, penyakit tanaman masih menjadi ancaman utama. *Food and Agriculture Organization (FAO)* mencatat bahwa setiap tahun penyakit tanaman merugikan perekonomian global lebih dari USD 220 miliar dan menyebabkan kehilangan hasil hingga 40 %. Kerugian ini memukul keras petani kecil yang bergantung pada budidaya hortikultura [4].

Berdasarkan hasil wawancara dengan seorang petani cabai yang berasal dari daerah Gunung Intan yaitu bapak Muktar Lusupi, terdapat beberapa jenis penyakit yang kerap menyerang daun tanaman cabai. Beberapa di antaranya adalah munculnya bercak pada daun, daun yang menjadi keriting, serta kondisi daun yang layu. Menurut penuturan beliau, salah satu cara untuk mengenali daun yang terserang keriting adalah dengan melihat bentuknya yang tampak melengkung atau tidak rata seperti daun sehat pada umumnya. Gejala-gejala ini sangat mengganggu proses pertumbuhan tanaman cabai secara keseluruhan. Tanaman yang terserang penyakit cenderung mengalami pertumbuhan yang tidak optimal, bahkan dalam beberapa kasus dapat menyebabkan gagal panen. Kondisi ini tentu sangat merugikan petani, baik dari segi kuantitas hasil panen maupun kualitasnya. Oleh karena itu, penanganan dini dan pencegahan terhadap penyakit daun pada tanaman cabai menjadi sangat penting agar produktivitas tetap terjaga.

Pendeteksian penyakit pada tanaman secara manual oleh petani sering kali memerlukan waktu dan keahlian yang cukup, serta dapat memberikan hasil yang tidak konsisten [5]. Oleh karena itu, pemanfaatan teknologi pengolahan citra digital dan kecerdasan buatan, khususnya *Convolutional Neural Network (CNN)*, menjadi solusi potensial dalam mendeteksi dan mengklasifikasikan penyakit daun secara otomatis.

CNN merupakan salah satu jenis *neural network* dimana metode ini dapat digunakan pada data *image*. *CNN* memiliki kemampuan untuk mengenali objek pada suatu *image*. *Convolutional Neural Network* merupakan salah satu model dari *Deep Learning* yang banyak digunakan untuk menganalisis citra/visual [6].

CNN telah terbukti efektif dalam berbagai tugas klasifikasi citra karena kemampuannya mengekstraksi fitur visual yang kompleks secara otomatis. Namun, tantangan utama dalam pengolahan citra daun cabai adalah adanya variasi pencahayaan, latar belakang, serta tingkat kematangan daun yang dapat mempengaruhi akurasi model *CNN*. Salah satu solusi untuk meningkatkan performa deteksi citra adalah dengan menggunakan teknik konversi warna, seperti *HSV (Hue, Saturation, Value)*, yang lebih robust terhadap perubahan pencahayaan.

Adapun beberapa penelitian yang dilakukan oleh peneliti sebelumnya, yaitu mengenai klasifikasi penyakit daun cabai telah dilakukan dengan berbagai pendekatan berbasis *Convolutional Neural Network (CNN)*. Studi oleh Bella Nurbuana dkk. menggunakan *CNN* standar dengan hasil akurasi mencapai 100% pada data yang sudah dikenali, namun menurun menjadi 45% ketika data baru ditambahkan, menandakan keterbatasan generalisasi model [7]. Penelitian lain oleh

Rahmat Arief Setyadi dkk. menerapkan transfer *learning* menggunakan arsitektur *ResNet101* dan menunjukkan hasil sangat tinggi dengan akurasi validasi sebesar 98.12% setelah augmentasi data, membuktikan efektivitas metode ini dalam klasifikasi penyakit daun cabai merah [1]. Sementara itu, studi oleh Muhammad Misbahul Munir dkk. menggunakan arsitektur *MobileNet* dan mendapatkan akurasi sebesar 97.22% dengan jumlah data terbatas, memperlihatkan bahwa *MobileNet* sebagai arsitektur ringan tetap mampu menghasilkan performa yang baik dalam klasifikasi citra daun cabai [8]. Ketiga penelitian ini menyoroti pentingnya pemilihan arsitektur *CNN* yang tepat, strategi augmentasi data, dan kebutuhan sistem deteksi yang adaptif terhadap variasi penyakit untuk mendukung deteksi dini dan peningkatan hasil pertanian. Meski menjanjikan, sebagian besar studi tersebut masih menggunakan ruang warna *RGB* sehingga sensitif terhadap variasi pencahayaan. Dan penelitian sebelumnya yang menggunakan *CNN* belum ada yang menggunakan model *CNN-HSV* dengan evaluasi augmentasi sistematis.

Dengan menggabungkan teknik konversi warna *HSV* dan *CNN*, diharapkan akurasi klasifikasi penyakit daun cabai dapat meningkat secara signifikan. Penelitian ini bertujuan untuk mengembangkan model klasifikasi penyakit daun cabai berbasis *CNN* yang didukung oleh *pre-processing* menggunakan ruang warna *HSV*, sehingga mampu memberikan hasil yang lebih andal dan efisien.

Berdasarkan latar belakang tersebut, maka penelitian ini diberikan judul **“Implementasi Metode *Convolutional Neural Network* Dan Teknik Konversi Warna *HSV* Untuk Klasifikasi Penyakit Daun Cabai”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah bagaimana merancang, membangun dan mengimplementasikan metode *Convolutional Neural Network (CNN)* dalam mengklasifikasikan penyakit pada daun cabai dengan memanfaatkan teknik konversi warna *HSV* untuk meningkatkan akurasi deteksi citra daun?.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang, membangun dan mengimplementasikan metode *Convolutional Neural Network (CNN)* dalam mengklasifikasikan penyakit pada daun cabai dengan memanfaatkan teknik konversi warna *HSV* guna meningkatkan akurasi deteksi citra daun.

1.4 Batasan Masalah

Agar pembahasan masalah menjadi lebih jelas dan terarah, maka diperlukan adanya batasan masalah. Adapun Batasan penelitian ini adalah:

1. Penyakit daun cabai diklasifikasikan kedalam 3 jenis yaitu daun sehat, daun keriting, dan daun kuning.
2. Jumlah dataset yang digunakan dalam penelitian ini adalah 1.200. data validasi 574 dan data latih 1160 gambar.
3. Citra daun cabai yang digunakan merupakan data citra 2D statis.
4. Pengujiannya menggunakan *precision*, *recall*, *confusion matrix* dan *F1-score* dibangun dengan menggunakan bahasa pemrograman *Python* dengan menggunakan *Google Colab* dengan *library Numpy*, *Pandas*, *Tensorflow* dan *Keras*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberi alternatif untuk mengetahui apakah daun cabai yang dimiliki berpenyakit atau tidak menggunakan algoritma *CNN*.
2. Menjadi acuan dalam pengembangan pengklasifikasian secara otomatis jenis penyakit pada daun cabai.

1.6 Sistematika Penulisan

Untuk memberikan gambaran umum dalam penulisan Tugas Akhir ini, maka sistematika penulisan secara garis besar dibagi menjadi lima bab, yaitu:

BAB 1 PENDAHULUAN

Bab ini berisi uraian mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, tahapan penelitian, serta sistematika penulisan. Komponen ini disusun untuk memberikan gambaran awal mengenai pentingnya penelitian klasifikasi penyakit daun cabai dengan metode *CNN* dan konversi warna *HSV*.

BAB 2 LANDASAN TEORI

Bab ini mengulas berbagai teori dan konsep yang relevan, seperti pengolahan citra digital, ruang warna *HSV*, serta arsitektur dan penerapan *Convolutional Neural Network (CNN)*. Selain itu, bab ini juga memuat kajian terhadap beberapa penelitian terdahulu yang berhubungan dengan klasifikasi penyakit tanaman dan pengolahan warna citra.

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan, meliputi pendekatan penelitian, teknik pengumpulan data, metode *CNN* yang diusulkan, proses konversi warna *HSV*, tahapan implementasi, dan rancangan pengujian model klasifikasi. Penjabaran pada bab ini menjadi dasar dari pelaksanaan penelitian secara teknis dan sistematis.

BAB 4 ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas hasil analisis kebutuhan sistem serta perancangan sistem klasifikasi citra daun cabai, mulai dari alur kerja, pemrosesan data, arsitektur *CNN* yang digunakan, hingga desain antarmuka (jika sistem berbentuk aplikasi). Bab ini juga menyertakan penjelasan mengenai perbandingan hasil klasifikasi menggunakan warna *RGB* dan *HSV*.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi sistem klasifikasi penyakit daun cabai menggunakan *CNN* yang telah dirancang. Pengujian dilakukan untuk mengevaluasi performa model dalam dua skenario: tanpa dan dengan konversi warna *HSV*. Evaluasi menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Bab ini juga menyajikan interpretasi terhadap hasil yang diperoleh dan efektivitas metode yang diusulkan.

BAB 6 KESIMPULAN DAN SARAN

Bab ini merupakan penutup dari laporan tugas akhir yang berisi kesimpulan dari hasil penelitian yang telah dilakukan, serta saran-saran yang dapat dijadikan acuan dalam pengembangan lebih lanjut, baik dari segi metode, data, maupun penerapan sistem klasifikasi citra dalam bidang pertanian cerdas.

BAB 2

LANDASAN TEORI

2.1 Tanaman Cabai

Cabai adalah salah satu jenis buah dan sayuran yang populer di kalangan masyarakat Indonesia. Buah ini mengandung berbagai nutrisi, di antaranya kalori, protein, serta vitamin A, B1, dan C. Gejala penyakit pada daun cabai dapat ditandai dengan perubahan warna daun yang terjadi lebih awal dari seharusnya, warna daun yang tidak merata, serta munculnya bintik-bintik pada permukaan daun [9].

Cabai merupakan tanaman jenis sayuran yang banyak digunakan sebagai bahan pembuatan berbagai macam bumbu masakan oleh masyarakat Indonesia. Seiring dengan peningkatan pertumbuhan penduduk dan industri makanan yang menggunakan bahan baku cabai menyebabkan permintaan cabai di Indonesia terus meningkat. Produksi cabai di Indonesia justru menurun. Data BPS menunjukkan bahwa selama tahun 2020-2022 mengalami fluktuasi. Sebagai contoh: produksi tahun 2020 sebanyak 1,51 juta ton menurun menjadi 1,39 juta ton tahun 2021, kemudian meningkat lagi menjadi 1,55 juta ton tahun 2022 [10].

2.2 Penyakit Tanaman Cabai

Beberapa penyakit pada tanaman cabai dapat dikenali melalui pengamatan visual pada daun, seperti virus kuning dan penyakit keriting. Meskipun gejala dapat dilihat dari perubahan bentuk dan warna daun, daun cabai cenderung memiliki kemiripan dan kesamaan dalam hal bentuk dan warna, sehingga membedakan penyakit hanya berdasarkan visual terkadang sulit [11].

2.2.1 Penyakit Keriting (*Leafcurl*)

Penyakit daun keriting pada daun cabai disebabkan oleh organisme pengganggu tanaman (OPT) seperti kutu yang bersarang di daun. Daun keriting umumnya bagian tepi daun menggulung ke bagian dalam sehingga membentuk cekungan [11]

Penyakit daun keriting (*yellow leaf curl disease*) pada cabai adalah gangguan yang umumnya disebabkan oleh infeksi virus dari kelompok *Geminiviridae*, khususnya jenis *Begomovirus* seperti *PepYLCV* (*Pepper Yellow Leaf Curl Virus*) [12]. seperti pada gambar 2.1.



Gambar 2. 1 Penyakit Keriting (Leafcurl)

2.2.2 Penyakit Virus Kuning (*Yellowwvss*)

Helai daun mengalami *vein clearing* yang dimulai dari pucuk daun berkembang menjadi warna kuning, tulang daun menebal dan daun menggulung ke atas. Infeksi selanjutnya dari virus kuning adalah daun akan mengecil dan berwarna kuning terang, tanaman kerdil dan produksi buah berkurang [11].

Penyakit virus kuning pada cabai-disebut juga penyakit bulai atau penyakit kerdil kuning-umumnya disebabkan oleh virus dari famili *Geminiviridae*, khususnya genus *Begomovirus* seperti *Pepper yellow leaf curl virus* (*PepYLCV*) dan *Tomato yellow leaf curl virus* (*TYLCV*)[12].



Gambar 2. 2 Penyakit Virus Kuning (Yellowwviss)

2.2.3 Daun Sehat

Daun cabai sehat adalah daun yang tampak hijau segar, utuh, dan tidak menunjukkan tanda-tanda kerusakan atau penyakit. Ciri-cirinya meliputi warna hijau merata, tidak ada bercak, lubang, atau bentuk yang tidak normal. Daun yang sehat menandakan bahwa tanaman cabai tumbuh dengan baik dan siap menghasilkan buah secara optimal[11].



Gambar 2. 3 Daun Sehat

2.3 Klasifikasi

Klasifikasi merupakan salah satu tahap paling krusial dalam proses data mining. Proses ini bertujuan untuk mengelompokkan atau memberi label pada data atau objek baru berdasarkan karakteristik tertentu. Teknik klasifikasi dilakukan dengan menganalisis variabel-variabel yang diambil dari data yang tersedia. Tujuan utamanya adalah untuk memprediksi kelas dari data yang belum diketahui. Proses klasifikasi terdiri dari tiga tahap utama, yaitu pembangunan model, penerapan

model, dan evaluasi. Pembangunan model dilakukan dengan menggunakan data pelatihan yang sudah memiliki atribut dan kelas, yang kemudian digunakan untuk menentukan kelas dari data atau objek baru. Setelah itu, model yang telah dibuat dievaluasi untuk mengukur tingkat akurasi ketika diterapkan pada data baru [13].

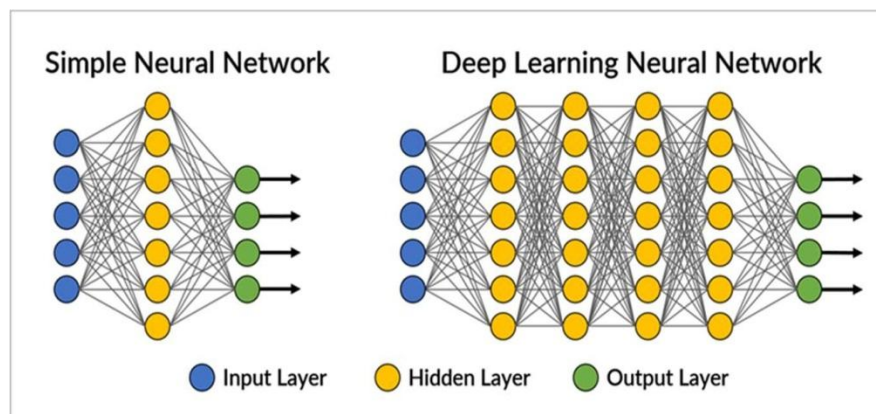
Klasifikasi adalah salah satu teknik yang paling sering digunakan dalam data mining. Proses klasifikasi mencakup dua tahap utama, yaitu pembelajaran (*learning*) dan pengklasifikasian. Pada tahap pembelajaran, data pelatihan dianalisis menggunakan algoritma klasifikasi. Sementara itu, pada tahap klasifikasi, data diuji dengan mengukur tingkat akurasi dari aturan klasifikasi yang telah dibentuk [14].

Maka dapat diambil Kesimpulan bahwasanya Klasifikasi merupakan proses penting dalam data mining yang bertujuan untuk memprediksi kelas dari data baru berdasarkan karakteristik tertentu. Proses ini mencakup pembangunan model dengan data pelatihan, penerapan model untuk mengklasifikasikan data baru, serta evaluasi akurasi model. Klasifikasi terdiri dari dua tahap utama, yaitu pembelajaran (*learning*) dan pengklasifikasian, yang dilakukan dengan menggunakan algoritma tertentu untuk menghasilkan aturan klasifikasi yang efektif.

2.4 Deep Learning

Deep learning merupakan cabang dari algoritma *machine learning* yang memiliki tingkat kompleksitas tinggi. Model dalam *deep learning* dilatih menggunakan sejumlah besar data berlabel dan memanfaatkan arsitektur jaringan saraf (*neural network*) yang mampu mempelajari fitur secara langsung dari data *input*, tanpa memerlukan proses ekstraksi fitur secara manual [15].

Algoritma *deep learning* bekerja dengan memproses data melalui sejumlah lapisan dalam jaringan saraf. Setiap lapisan dalam jaringan tersebut akan menyederhanakan data sebelum mengirimkannya ke lapisan berikutnya. Sementara algoritma *machine learning* tradisional biasanya efektif untuk data terstruktur yang tersusun dalam bentuk baris dan kolom, algoritma tersebut sering mengalami kesulitan ketika dihadapkan pada data tidak terstruktur, seperti gambar, yang membutuhkan proses prapemrosesan yang kompleks. Oleh karena itu, dikembangkanlah algoritma *deep learning* yang dirancang khusus untuk menganalisis dan memproses data visual secara mendalam melalui sejumlah lapisan yang saling terhubung. Lapisan awal berfungsi untuk mengenali detail atau fitur dasar dalam gambar, sedangkan lapisan-lapisan selanjutnya menggabungkan informasi tersebut untuk membentuk representasi yang lebih kompleks dan bermakna [16].



Gambar 2. 4 Struktur Simple *Deep Learning*

Sumber : [16]

Deep Learning dipandang sebagai salah satu terobosan teknologi paling signifikan setelah kemunculan internet, dan kerap digambarkan sebagai inovasi

yang berpotensi menggantikan peran manusia dalam berbagai bidang pekerjaan [17].

Deep learning terbagi 3 kategori pendekatan, yaitu:

1. *Supervised learning*

Supervised learning merupakan salah satu cabang dari *machine learning* yang bekerja dengan menggunakan dataset yang telah diberi label. Tujuannya adalah menghasilkan *output* atau solusi yang sesuai berdasarkan data tersebut. Algoritma dalam *Supervised learning* mempelajari pola hubungan antara data dan label yang menyertainya. Beberapa metode yang umum digunakan dalam pendekatan ini antara lain *regresi linier*, *random forest*, dan *support vector machine (SVM)* [15].

2. *UnSupervised learning*

UnSupervised learning merupakan pendekatan deskriptif yang digunakan untuk mengelompokkan atau mengkategorikan data. Tidak seperti *Supervised learning*, metode ini tidak memerlukan dataset yang telah diberi label karena Tujuannya bukan untuk membuat prediksi, melainkan untuk mempelajari struktur dari data yang tersedia. Proses ini dilakukan dengan mengelompokkan data berdasarkan kemiripan atau kedekatan antar data, yang dikenal sebagai *clustering*. Metode ini menganalisis data yang belum berlabel untuk mengidentifikasi pola tersembunyi serta menentukan hubungan atau korelasi di antara data tersebut. Beberapa algoritma yang umum digunakan dalam *UnSupervised learning* antara lain *K-Means*, *Hierarchical Clustering*, *DBSCAN*, dan *Fuzzy C-Means* [15].

3. *Reinforcement learning*

Reinforcement learning merupakan pendekatan pembelajaran mesin yang menggunakan sistem penghargaan (*reward*) dan hukuman (*penalty*) untuk belajar. Algoritma ini berfokus pada bagaimana agen perangkat lunak (*software agent*) mengambil keputusan dalam suatu lingkungan. Tujuan utamanya adalah mempelajari strategi pengambilan keputusan terbaik secara bertahap agar dapat memaksimalkan keberhasilan atau performa dalam situasi nyata [15].

Berdasarkan penjelasan di atas, maka dapat disimpulkan bahwasanya *Deep Learning* merupakan cabang dari *Machine Learning* yang memiliki tingkat kompleksitas tinggi dan menggunakan arsitektur jaringan saraf (*neural network*) untuk mempelajari fitur langsung dari data berlabel tanpa ekstraksi manual. Teknologi ini dianggap sebagai salah satu inovasi besar setelah internet karena potensinya dalam menggantikan peran manusia di berbagai bidang. *Deep learning* terbagi menjadi tiga pendekatan utama, yaitu *Supervised learning* yang bekerja dengan data berlabel untuk mempelajari hubungan antara *input* dan *output*, *UnSupervised learning* yang menganalisis data tanpa label untuk menemukan pola atau struktur tersembunyi melalui teknik seperti *clustering*, dan *Reinforcement learning* yang menggunakan sistem *reward* dan *penalty* untuk melatih agen dalam mengambil keputusan optimal di lingkungan tertentu.

2.5 *Machine Learning*

Machine learning merupakan cabang dari kecerdasan buatan yang menitikberatkan pada penggunaan algoritma dan metode tertentu untuk melakukan

prediksi, mengenali pola, serta melakukan klasifikasi. Beberapa algoritma yang umum digunakan dalam *machine learning* antara lain *neural network*, *decision tree*, *k-nearest neighbor*, *naïve Bayes*, *random forest*, dan lainnya [18].

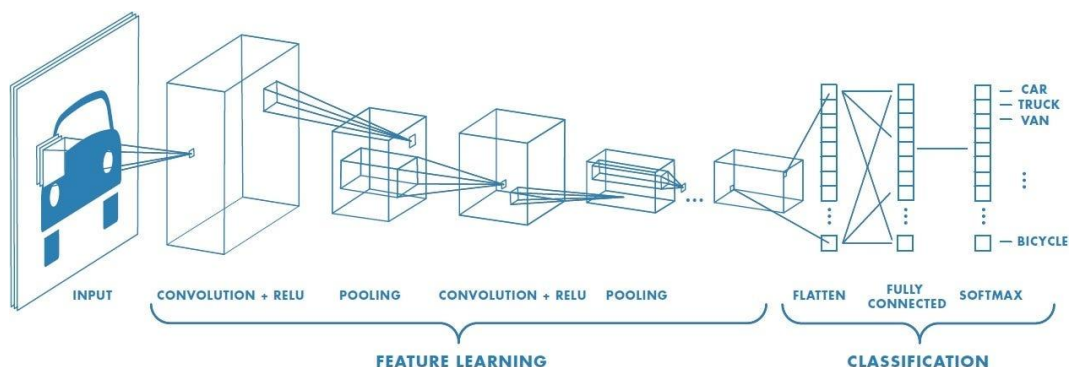
Machine learning memiliki cabang yang memanfaatkan algoritma jaringan saraf tiruan untuk belajar dan beradaptasi terhadap volume data yang besar. Algoritma jaringan saraf tiruan dalam *deep learning* terinspirasi dari cara kerja otak manusia. Dengan pendekatan ini, mesin dapat mengenali pola dalam data yang tidak terstruktur atau data yang memiliki fitur yang sulit ditentukan secara eksplisit, seperti gambar, teks, audio, dan video. Proses pembelajaran yang dimaksud berbeda dengan cara manusia belajar, karena mesin dianggap "belajar" saat ia dapat memperbarui parameternya, di mana parameter ini secara umum merepresentasikan pengetahuan yang dimiliki oleh mesin tersebut [15].

Berdasarkan penjelasan di atas, maka dapat disimpulkan bahwasanya *Machine Learning* merupakan cabang dari kecerdasan buatan (AI) yang berfokus pada penggunaan algoritma untuk melakukan prediksi, mengenali pola, dan mengklasifikasikan data. Algoritma yang umum digunakan antara lain *neural network*, *decision tree*, *k-nearest neighbor*, *naïve Bayes*, dan *random forest*. Salah satu cabang penting dari *machine learning* adalah *deep learning*, yang menggunakan jaringan saraf tiruan (*artificial neural networks*) yang terinspirasi dari cara kerja otak manusia. Pendekatan ini memungkinkan mesin untuk belajar dari data berukuran besar dan tidak terstruktur, seperti gambar, teks, audio, dan video, dengan cara memperbarui parameter internalnya sehingga dapat

mengadaptasi dan mengenali pola secara otomatis tanpa perlu ekstraksi fitur secara eksplisit.

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu algoritma dalam *deep learning* yang dikembangkan dari arsitektur *Multilayer Perceptron (MLP)*, dan dirancang khusus untuk memproses data berdimensi dua, seperti gambar dan suara. *CNN* biasanya digunakan untuk mengklasifikasikan data berlabel dengan pendekatan *Supervised learning*. *CNN* termasuk jenis *neural network* yang paling sering diterapkan pada data citra atau gambar. Karena memiliki banyak lapisan, jaringan *CNN* termasuk dalam kategori *Deep Neural Network* dan sering dimanfaatkan dalam analisis data visual [19].



Gambar 2. 5 Proses Convolutional Neural Network (CNN)

Sumber : [20]

Secara umum, arsitektur *Convolutional Neural Network (CNN)* terdiri dari beberapa lapisan (layers) yaitu *Input Layer*, *Convolutional Layer*, Fungsi Aktivasi *ReLu*, *Pooling Layer*, *Flatten Layer*, *Fully Connected Layer (ANN)*, Fungsi Aktivasi *Softmax*, dan *Dropout*. Setiap lapisan ini memiliki fungsi spesifik dalam proses

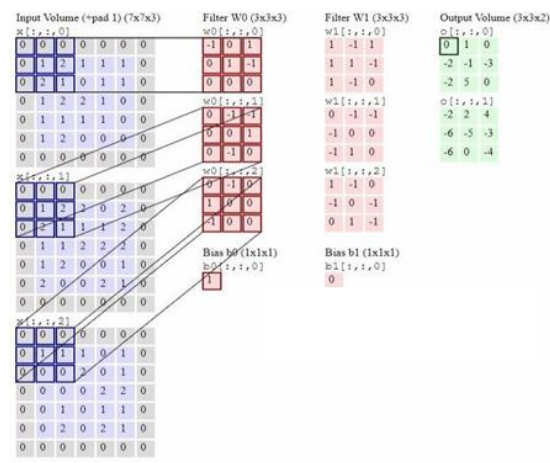
klasifikasi gambar. Berikut adalah penjelasan tentang bagaimana *CNN (feed forward)*

1. Input layer

Input layer pada *Convolutional Neural Network (CNN)* berfungsi untuk merepresentasikan gambar yang dimasukkan ke dalam model. Misalnya, jika gambar yang diinput memiliki resolusi 224x224 piksel dan menggunakan format *RGB (Red, Green, Blue)*, maka gambar tersebut akan diubah menjadi *array* multidimensi dengan dimensi 224x224x3, dimana angka 3 menunjukkan saluran warna *RGB*.

2. Convolution Layer

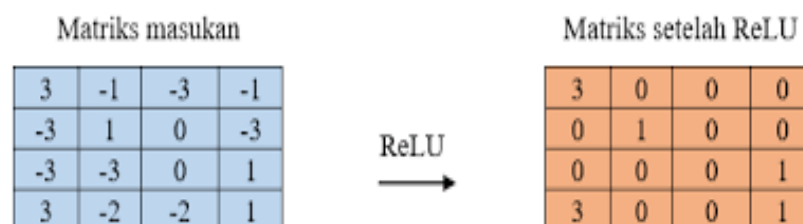
Convolution layer merupakan bagian inti dari *Convolutional Neural Network (CNN)* yang berperan penting dalam melakukan perhitungan lebar, tinggi, dan kedalaman gambar serta kernel. Lapisan ini berfungsi untuk memproses filter dan feature map pada gambar input [21].



Gambar 2. 6 Proses Convolution Layer [13]

3. Fungsi Aktivasi *ReLU*

Fungsi aktivasi *ReLU* (*Rectified Linear Unit*) digunakan setelah proses konvolusi dan sebelum *pooling* (di tahap *feature extractor*), serta diterapkan pada setiap node di hidden layer (di tahap *fully connected*). *ReLU* merupakan lapisan aktivasi dengan fungsi $f(x) = \max(0, x)$, yang mengubah semua nilai piksel kurang dari nol pada suatu gambar menjadi 0. Fungsi ini bertujuan untuk mengurangi linearitas yang muncul dari proses konvolusi, sehingga *Convolutional Neural Network* (*CNN*) lebih mudah mencapai nilai optimum [10].



Gambar 2. 7 Ilustrasi Proses Relu

4. *Pooling Layer*

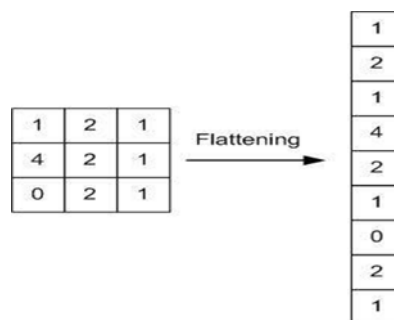
Pooling layer merupakan filter dengan ukuran tertentu yang bergerak melintasi seluruh area feature map. Fungsi utama *pooling layer* adalah mengurangi dimensi atau melakukan downsampling pada *feature map*, sehingga mempercepat proses komputasi karena jumlah bobot yang perlu diperbarui menjadi lebih sedikit [10]. Terdapat beberapa metode pooling yang umum digunakan, seperti *max pooling* dan *average pooling*. *Max pooling* berfungsi untuk memilih nilai maksimum dari area tertentu, sementara *average pooling* menghitung nilai rata-rata dari area tertentu [12].



Gambar 2. 8 Max Pooling dan Average Pooling [14]

5. Flatten Layer

Flatten layer adalah langkah yang dilakukan sebelum memasuki *Fully Connected Layer*. Pada tahap ini, feature map yang dihasilkan dari proses *feature extraction* masih berupa *array* multidimensi, sehingga perlu dilakukan proses *flattening* untuk mengubahnya menjadi vektor satu dimensi. Proses ini memastikan fitur dapat digunakan sebagai input untuk tahap klasifikasi atau *fully connected layer*. Dengan demikian, informasi dari feature map tetap dipertahankan, namun dalam format yang sesuai untuk diproses oleh lapisan *fully connected* [13].



Gambar 2. 9 Ilustrasi Proses Flatten Layer

Gambar 2.9 menunjukkan hasil akhir dari *pooling* yang diubah menjadi satu vektor. Proses flattening ini kemudian digunakan sebagai input untuk *Fully Connected Layer*. *Flatten* adalah proses yang berfungsi untuk mengubah *feature*

map berbentuk matriks multidimensi menjadi vektor, sehingga dapat dimasukkan ke dalam *Fully Connected Layer* sebagai input [13].

6. *Fully Connected Layer*

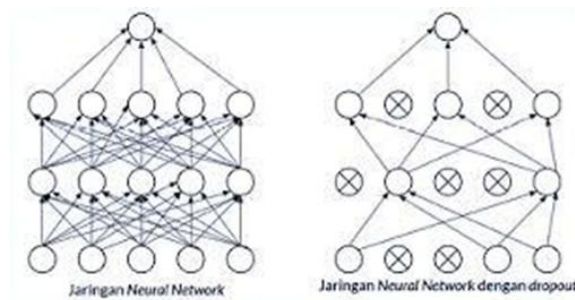
Fully Connected Layer terdiri dari *Input Layer*, *Hidden Layer*, dan *Output Layer*, yang memiliki struktur mirip dengan lapisan pada jaringan saraf tiruan (*ANN*). Oleh karena itu, beberapa literatur mengidentifikasi *fully connected layer* sebagai *ANN*. Pada *fully connected layer*, setiap *node* (*neuron*) pada layer sebelumnya terhubung penuh dengan setiap *node* di layer berikutnya. Setiap *hidden layer* memiliki fungsi aktivasi, yang umum digunakan yaitu *ReLU*, dan begitu pula di *output layer* juga memiliki fungsi aktivasi. Untuk klasifikasi dengan lebih dari satu kelas, fungsi *Softmax* sering digunakan. *Fully connected layer* bertujuan untuk mengolah data agar dapat diklasifikasikan, dan hasilnya berupa probabilitas untuk setiap kategori (jika menggunakan *Softmax*) [10].

7. Fungsi Aktivasi *Softmax*

Fungsi aktivasi *Softmax* digunakan dalam kasus klasifikasi *multi-class* (lebih dari dua kelas) dan ditempatkan pada output layer dalam tahap *fully connected* atau klasifikasi. *Softmax* berfungsi untuk menghitung probabilitas dari setiap kelas target (kategori) atas semua kelas target yang memungkinkan, sehingga dapat membantu menentukan kelas target untuk suatu citra yang diberikan. Keunggulan *Softmax* adalah outputnya dalam rentang probabilitas antara 0 hingga 1, dan jumlah keseluruhan probabilitasnya sama dengan satu [10].

8. Dropout Regularization

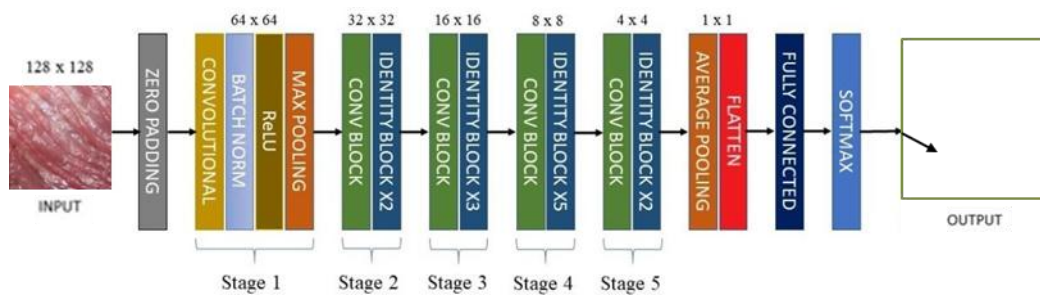
Dropout merupakan teknik regularisasi yang digunakan untuk mengurangi *overfitting* pada *neural network*, di mana beberapa neuron dipilih secara acak untuk dinonaktifkan selama proses pelatihan (dengan kata lain: menghilangkan fitur/mengurangi variansi). *Neuron* ini diabaikan secara acak, sehingga kontribusinya dihentikan sementara, dan tidak ada bobot baru yang diterapkan pada neuron tersebut selama *backpropagation*. Selain membantu mengurangi *overfitting*, penggunaan *Dropout* juga mempercepat proses pelatihan (training) [10].



Gambar 2. 10 Ilustrasi Penggunaan *Dropout* [10]

2.7 CNN-ResNet

ResNet (*Residual Network*) merupakan salah satu arsitektur deep learning yang sangat populer dan efektif dalam pengenalan citra. *ResNet* diperkenalkan untuk mengatasi masalah degradasi kinerja pada jaringan saraf dalam yang semakin dalam, dengan memperkenalkan konsep *residual learning*. Dalam arsitektur ini, *shortcut connection* memungkinkan jaringan untuk melewati beberapa lapisan sehingga menghindari hilangnya informasi penting selama proses propagasi [22].



Gambar 2. 11 Arsitektur *Resnet-50*

Sumber: [23]

ResNet telah terbukti unggul dalam klasifikasi penyakit tanaman. Misalnya, Ji et al. (2024) mengembangkan *ICS-ResNet* yang ringan dan efisien untuk klasifikasi penyakit daun jagung dan melaporkan kinerja tinggi. Chen et al. (2024) memodifikasi *ResNet34* dengan menambahkan *CBAM* (*Convolutional Block Attention Module*) dan *LeakyReLU* sehingga meningkatkan akurasi hingga 98.6 % pada klasifikasi daun tomat. Upadhyay dan Saxena (2024) mengevaluasi *ResNet-50* yang diperluas melalui transfer learning dan augmentasi lanjutan, menghasilkan akurasi lebih dari 95 %. Studi Jayashree dan Sumalatha (2024) juga menunjukkan efektivitas *ResNet-50* dalam mendeteksi penyakit daun tanaman setelah tahap pemrosesan citra [24] [25] [22].

2.8 Konversi Warna *HSV*

Hue, *Saturation*, *Value* (*HSV*) merupakan ruang warna yang bersifat perseptual karena merepresentasikan warna pada citra berdasarkan cara manusia memahami warna. Model warna *HSV* terdiri dari tiga komponen utama: *Hue*, yaitu panjang gelombang dominan yang menunjukkan jenis warna seperti merah, kuning, atau biru; *Saturation*, yang menggambarkan tingkat kejenuhan atau kemurnian warna; dan *Value*, yang menunjukkan tingkat kecerahan dari warna tersebut. Salah

satu keunggulan ruang warna *HSV* adalah kemampuannya dalam mendeskripsikan berbagai warna dasar, tidak terbatas hanya pada warna merah, hijau, dan biru. Warna seperti oranye atau hijau kebiruan tetap dapat dikenali sebagai warna dominan dalam model *HSV* tanpa harus bergantung pada komponen *RGB*. Keunggulan ini membuat ruang warna *HSV* lebih stabil dan lebih sesuai digunakan dalam proses segmentasi warna [9].

2.9 *HSV* Dalam Segmentasi Dan Klasifikasi Penyakit Daun

Penggunaan ruang warna *HSV* (*Hue-Saturation-Value*) telah terbukti memberikan keuntungan signifikan dalam segmentasi dan klasifikasi penyakit tanaman karena kemampuannya dalam memisahkan komponen warna dari intensitas cahaya. Pan et al. (2023) melakukan segmentasi pada pohon pinus yang terinfeksi menggunakan kombinasi model U-Net dan fitur HSV, dan menemukan bahwa citra delapan-band *RGBHSV* meningkatkan akurasi segmentasi dibandingkan hanya RGB. Pada skala kecil, dalam studi Elena dkk. (2023), HSV digunakan sebagai dasar threshold untuk mendeteksi lesi daun tomat; metode ini memberikan segmentasi otomatis lebih baik dibanding manual berbasis RGB dan meningkatkan akurasi klasifikasi penyakit hingga 97,4 %. Juga, dalam penelitian lanjutan HT-CNN untuk klasifikasi penyakit daun gandum (2023), konversi HSV diikuti oleh peningkatan kontras dan segmentasi menggunakan *K-means* menghasilkan identifikasi area penyakit yang lebih tajam dibanding tanpa HSV [26] [27].

2.10 *Augmented Datasets*

Augmentasi data merupakan proses menambah variasi pada dataset pelatihan dengan teknik manipulasi citra seperti rotasi, *flipping*, *zoom*, brightness shift, dan penambahan noise. Teknik ini sangat penting dalam klasifikasi citra berbasis CNN karena membantu mengurangi *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan pada data pelatihan dan gagal mengenali pola baru pada data uji. Dengan membuat model terekspos pada variasi yang lebih luas, augmentasi dapat meningkatkan generalisasi dan ketahanan terhadap noise dunia nyata [28].

Augmentasi data telah menjadi strategi penting untuk memperbaiki performa model CNN dalam klasifikasi penyakit tanaman, terutama ketika dataset alami terbatas atau tidak seimbang. Min et al. (2023) mengembangkan metode augmentasi berbasis *CycleGAN*-attention yang menerjemahkan citra sehat ke citra penyakit dan menghasilkan gambar sintetis yang lebih realistis; dengan menggunakan dataset *Plant Village* (apel, kentang, anggur), mereka berhasil meningkatkan akurasi klasifikasi menunjukkan efektifitas augmentasi berbasis GAN dibanding augmentasi konvensional [28].

Sementara itu, Suryawanshi et al. (2024) memperkenalkan teknik augmentasi berbasis latent space menggunakan transformasi Hadamard untuk mencampur fitur citra; ketika diuji pada arsitektur CNN (VGG16/19 dan ResNet-50), pendekatan ini meningkatkan generalisasi dan akurasi dibanding augmentasi tradisional [29]. Studi-studi tersebut menunjukkan bahwa augmentasi canggih seperti GAN atau latent-space mixing bukan hanya memperkaya data, tetapi juga membantu mengurangi *overfitting* dan meningkatkan ketahanan terhadap variasi dunia nyata.

2.11 *Flowchart*

Flowchart adalah representasi logis dari suatu prosedur penyelesaian masalah, yang digambarkan melalui simbol-simbol khusus. Diagram ini menggambarkan langkah-langkah sistematis dalam menyelesaikan suatu permasalahan, sekaligus menunjukkan alur logika program secara visual [30].

Flowchart merupakan representasi grafis dari langkah-langkah dan urutan prosedur dalam suatu program. Flowchart sistem menggambarkan rangkaian proses dalam suatu sistem, termasuk menunjukkan perangkat input, output, serta jenis media penyimpanan yang digunakan selama pemrosesan data [31].

2.12 *Unified Model Language (UML)*

UML adalah sekelompok alat yang digunakan untuk perangkat lunak abstrak atau sistem berbasis objek. Alat lain untuk memfasilitasi pembuatan aplikasi berkelanjutan adalah *Unified Model Language*. Dalam aplikasi atau sistem yang tidak terdokumen dapat menghambat pengembangan karena memerlukan pengembangan untuk menavigasi dan memeriksa kode komputer. Uml juga dapat digunakan sebagai alat bagi pengembangan untuk berbagi informasi satu sama lain mengenai sistem atau aplikasi yang akan datang. Siapa pun dapat memahami sistem dengan uml, tidak hanya pengembangan dapat profesional bisnis [32].

2.12.1 *Use Case Diagram*

Use case diagram merupakan model untuk perilaku (*behaviour*) dari sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa yang ada dalam sistem informasi dan siapa yang memiliki kewenangan untuk menggunakan fungsi tersebut [33].

2.12.2 Class Diagram

Class Diagram Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan [33].

2.12.3 Activity Diagram

Activity Diagram yang menggambarkan atau memodelkan berbagai aktivitas aliran dalam sistem yang sedang dikembangkan, serta bagaimana setiap aliran berawal dari keputusan yang mungkin terjadi, dan bagaimana setiap aktivitas berakhir [33].

2.12.4 Sequence Diagram

Sequence diagram digunakan untuk memodelkan urutan dari aktivitas-aktivitas dalam sebuah proses yang melibatkan beberapa objek dalam sistem. *Sequence diagram* menunjukkan bagaimana pesan atau perintah diproses oleh objek-objek dalam sistem [34].

2.13 Python

Python merupakan bahasa pemrograman berorientasi objek yang dapat dijalankan secara interaktif. Bahasa ini menyediakan struktur data tingkat tinggi dan bersifat interpretatif, dengan berbagai fungsi bawaan. *Python* dirancang dengan penekanan pada kejelasan sintaks dan kemudahan pemahaman, sehingga dianggap sebagai bahasa yang menggabungkan kekuatan fungsional dengan kesederhanaan

penulisan kode. Tujuan utama dari perancangannya adalah untuk membantu programmer dalam membangun program secara efisien, mempercepat proses pengembangan, serta menjaga kompatibilitas dengan berbagai sistem. *Python* dapat digunakan baik untuk membuat aplikasi mandiri maupun untuk pemrograman skrip [35].

Python merupakan bahasa pemrograman yang sangat fleksibel dan memiliki berbagai macam kegunaan. Dengan dukungan alat dan pustaka (*library*) yang sesuai, siapa pun dapat berinovasi secara maksimal. Untuk mulai belajar *Python*, dibutuhkan keberanian, komitmen, waktu, dan mungkin sedikit tambahan semangat. Oleh karena itu, penting untuk menentukan tujuan belajar sejak awal dan memahami berbagai potensi penggunaan *Python* [36].

Python merupakan salah satu perangkat lunak yang saat ini tengah populer. Bahasa ini memungkinkan penggunanya untuk melakukan analisis data, menyelesaikan perhitungan statistik yang rumit dan memakan waktu, membuat visualisasi data, menerapkan algoritma *machine learning*, serta memanipulasi data dan menyelesaikan berbagai persoalan matematika lainnya. Keunggulan *Python* terletak pada kemampuannya dalam menghasilkan *output* yang lebih akurat dan efisien dibandingkan dengan metode manual. Dalam konteks pembelajaran matematika, khususnya pada materi Permukaan Ruang, *Python* dapat dimanfaatkan untuk memvisualisasikan grafik fungsi dua variabel. Penggunaan *Python* dalam konteks ini sangat membantu dalam memperdalam pemahaman serta mendukung proses pengajaran konsep-konsep matematika yang kompleks [37].

Berdasarkan penjelasan di atas, maka dapat disimpulkan bahwasanya *Python* merupakan bahasa pemrograman berorientasi objek yang fleksibel, interpretatif, dan mudah dipahami karena sintaksnya yang sederhana. *Python* dirancang untuk membantu pengembang dalam membangun program secara efisien dan kompatibel dengan berbagai sistem, serta dapat digunakan baik untuk aplikasi mandiri maupun skrip. Kepopuleran *Python* didukung oleh kemampuannya dalam menyelesaikan berbagai tugas, mulai dari analisis data, perhitungan statistik, visualisasi data, hingga penerapan algoritma *machine learning*. Dengan dukungan pustaka yang beragam, *Python* memungkinkan penggunaannya untuk berinovasi secara maksimal. Dalam bidang pendidikan, khususnya pada pembelajaran matematika seperti materi Permukaan Ruang, *Python* juga bermanfaat untuk memvisualisasikan grafik fungsi dua variabel guna memperdalam pemahaman konsep yang kompleks.

2.14 Streamlit

Streamlit merupakan opensource framework yang memungkinkan pengembangan aplikasi web berbasis python, khususnya dalam bidang sains data dan *machine learning*. Fitur – fiturnya membantu mempermudah pengembangan model *machine learning* dengan visualisasi melalui antarmuka interaktif untuk berkomunikasi dengan user. *Streamlit* juga mendukung pembuatan presentasi dan kolaborasi, serta memungkinkan ekspor dokumen dalam format aplikasi data interaktif. Proses eksekusi *script python* dilakukan dari baris atas hingga bawah setiap kali user berinteraksi dengan aplikasi [38].

Streamlit merupakan platform yang sangat berguna untuk mengembangkan antarmuka pengguna yang interaktif. Dengan menggunakan Streamlit, pengguna

dapat dengan mudah memasukkan spesifikasi laptop yang mereka cari, seperti merk, warna, tipe prosesor, RAM, penyimpanan, ukuran layar, dan parameter lainnya melalui widget input yang disediakan. Input dari pengguna kemudian diproses secara otomatis dan digunakan sebagai masukan untuk model prediksi yang telah disiapkan sebelumnya [39].

2.15 *Google Collab*

Google Collab, atau *Google Colaboratory*, merupakan *Platform* berbasis cloud dari *Google* yang memungkinkan pengguna menulis dan menjalankan kode *Python* langsung melalui browser tanpa perlu melakukan instalasi atau konfigurasi tambahan. *Platform* ini memanfaatkan infrastruktur cloud *Google* dan menyediakan lingkungan pengembangan yang tangguh, termasuk akses gratis ke *GPU* dan *TPU* (*Tensor Processing Unit*). Salah satu keunggulan utama *Google Colab* adalah fitur kolaboratifnya, yang memungkinkan pengguna untuk berbagi dan mengerjakan notebook secara real-time bersama orang lain. Karena kemudahan akses dan penggunaannya, *Google Colab* menjadi pilihan populer di kalangan pengembang, peneliti, dan pelajar untuk mengembangkan proyek-proyek *Python* tanpa perlu khawatir tentang perangkat keras atau pengaturan system [40].

Google Colab berjalan di atas Jupyter Notebook dan mendukung berbagai bahasa pemrograman populer seperti *Python*, *R*, dan *Julia*. *Platform* ini menyediakan sumber daya komputasi yang kuat, termasuk akses ke mesin virtual berbasis Linux dengan *GPU* canggih yang dapat digunakan hingga 12 jam. Fitur-fitur ini menjadikan *Colab* sebagai pilihan yang praktis untuk berbagai keperluan,

terutama dalam menjalankan tugas-tugas terkait pembelajaran mesin dan pembelajaran mendalam [41].

Google Colab merupakan *Platform* komputasi awan yang menyerupai Jupyter Notebook dan dikembangkan oleh *Google Research*. *Platform* ini memungkinkan pengguna menulis serta menjalankan kode *Python* langsung melalui browser, tanpa perlu instalasi tambahan. Dengan demikian, pengguna dapat memanfaatkan sumber daya server untuk melakukan pemrosesan dan analisis data secara efisien dan cepat [42].

Berdasarkan penjelasan di atas, maka dapat disimpulkan bahwasanya *Google Colab* atau *Google Colaboratory* adalah *Platform* komputasi awan berbasis browser yang dikembangkan oleh *Google* dan memungkinkan pengguna menulis serta menjalankan kode *Python* tanpa perlu instalasi atau konfigurasi tambahan. *Platform* ini berjalan di atas Jupyter Notebook dan menyediakan akses gratis ke sumber daya komputasi yang kuat, termasuk *GPU* dan *TPU*, yang sangat bermanfaat untuk tugas-tugas seperti pembelajaran mesin dan pembelajaran mendalam. Selain itu, *Google Colab* mendukung bahasa pemrograman populer seperti *Python*, *R*, dan *Julia*, serta memiliki fitur kolaboratif yang memungkinkan pengguna berbagi dan bekerja secara real-time. Karena kemudahan akses, fleksibilitas, serta efisiensi dalam pemrosesan data, *Google Colab* menjadi pilihan populer di kalangan pengembang, peneliti, dan pelajar.

2.16 Penelitian Terkait

Berikut ini merupakan penelitian terkait dengan Skripsi tersebut :

Tabel 2. 1 Penelitian Terkait

No	Penulis	Judul	Metode	Kelebihan	Kelemahan	Hasil
1.	Bella Nurbuana Tri Cahya Ningrum, Erlina Nasrinatun Ni'mah, Miranda Putri Arifin, dan Made Ayu Dusea Widya Dara (2024)	Klasifikasi dan Pengenalan Pola Penyakit Cabai dengan Metode <i>CNN</i> (<i>Convolutional Neural Network</i>)	Penelitian ini menggunakan metode <i>Convolutional Neural Network</i> (<i>CNN</i>) sebagai pendekatan utama dalam proses klasifikasi dan pengenalan pola penyakit pada buah cabai. Pengembangan sistem dilakukan dengan pendekatan metode <i>Waterfall</i> , yang terdiri dari lima tahapan: requirement, design, implementation, verification	Model menggunakan <i>CNN</i> dan ruang warna HSV untuk klasifikasi penyakit cabai dengan pendekatan praktis dan implementasi awal berbasis citra asli dari kamera smartphone; pendekatan HSV memberikan fleksibilitas terhadap variasi warna.	Akurasi menurun drastis saat diuji dengan data penyakit baru (dari 100% menjadi 45%), belum mencakup proses augmentasi sistematis dan belum diuji generalisasinya terhadap variasi lingkungan dan jenis penyakit lain secara menyeluruh.	Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu mengklasifikasi penyakit cabai dengan sangat baik pada data yang telah dikenali sebelumnya. Saat diuji menggunakan 162 citra yang sudah dikenali, model <i>CNN</i> mencapai akurasi sebesar 100%. Namun, ketika ditambahkan 74 data baru, termasuk jenis

			<p>n, dan maintenance. Pengolahan citra diawali dengan segmentasi warna <i>RGB</i> yang kemudian dikonversi ke ruang warna <i>HSV</i> (Hue, Saturation, Value). Selanjutnya dilakukan thresholding untuk menghasilkan citra biner, dan kemudian dilakukan ekstraksi ciri morfologi berdasarkan parameter <i>eccentricity</i> dan <i>metric</i>. Dataset yang digunakan terdiri dari 162 citra cabai jenis Asmoro, baik mentah maupun</p>		<p>penyakit yang belum dikenali oleh sistem seperti penyakit lalat, akurasi sistem menurun secara signifikan menjadi 45%. Hal ini menunjukkan bahwa meskipun model <i>CNN</i> efektif dalam mengidentifikasi jenis penyakit yang sudah dikenali, performanya masih terbatas ketika menghadapi variasi penyakit baru. Oleh karena itu, pengembangan sistem lebih lanjut dengan dataset yang lebih luas dan representat</p>
--	--	--	--	--	---

			<p>kering, yang dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian. Proses pengolahan dan pelatihan model dilakukan menggunakan perangkat lunak Jupyter dan Visual Studio Code, serta pustaka seperti TensorFlow Keras, OpenCV (cv2), dan NumPy.</p>			<p>if sangat diperlukan agar sistem dapat mengenali lebih banyak jenis penyakit dan tetap memiliki akurasi yang tinggi dalam berbagai kondisi.</p>
2.	<p>Alisyia Kornelia Ulandari, Ghina Kamilah Ramdhani, Wahyuningsih, M. Naufal Arwan</p>	<p>Klasifikasi Jeruk Segar dan Busuk Melalui GLCM dan HSV dengan Menggunakan Metode ANN</p>	<p><i>Artificial Neural Network (ANN)</i></p>	<p>Menggabungkan fitur tekstur (GLCM) dan warna (HSV) dalam klasifikasi buah jeruk dengan dataset besar (3.000+</p>	<p>Belum menggunakan CNN; fokus pada klasifikasi biner (segar/busuk) saja; belum menguji HSV dan GLCM</p>	<p>Dari 12 skenario pengujian yang dilakukan, hasil terbaik diperoleh pada skenario ke-9, yaitu dengan arsitektur Dense 32, rasio data train-test</p>

	syuri, dan Fitri Biman toro (2024)			<p>citra). Evaluasi 12 skenario arsitektur menunju kkan akurasi tinggi (maks. 86,88%). Cocok untuk klasifikas i berbasis warna dan tekstur di komodita s buah.</p>	<p>secara independ den; generalis asi ke penyakit tanaman atau skenario multi- kelas belum dibahas.</p>	<p>70:30, dan dropout 20%. Pada konfigurasi ini, model berhasil mencapai akurasi sebesar 86,88% pada data pelatihan dan 86,13% pada data pengujian. Hasil ini menunjukk an bahwa arsitektur model dan parameter dropout yang tepat sangat memengar uhi performa. Pengguna an dropout sebesar 50% pada beberapa skenario (seperti skenario ke-5, 10, dan 12) justru menurunka n akurasi model secara signifikan. Dengan demikian,</p>
--	---	--	--	--	---	--

						dapat disimpulkan bahwa model ANN dengan kombinasi fitur GLCM dan HSV terbukti efektif dalam mengklasifikasi buah jeruk segar dan busuk, meskipun masih ada ruang untuk peningkatan lebih lanjut dengan mencoba metode lain seperti CNN di masa mendatang .
3.	Fifi Febrianti Usman , Purnawansyah, Herdianti Darwis, dan Erick	Klasifikasi Penyakit Tanaman Bawang Merah Menggunakan <i>Convolutional Neural Network</i> dan K-Nearest Neighbor	Penelitian ini bertujuan untuk mengklasifikasi dua jenis penyakit pada tanaman bawang merah, yaitu	Penelitian ini membuktikan bahwa penggunaan teknik konversi warna HSV yang dikombinasikan	Penelitian hanya menggunakan dua kelas penyakit dan jumlah dataset yang terbatas (320	Hasil penelitian menunjukkan bahwa klasifikasi dengan metode CNN, baik yang menggunakan fitur HSV maupun

	Irawadi Alwi (2023)		<p>bercak ungu dan moler, dengan menggunakan dua pendekatan algoritma yaitu <i>Convolutional Neural Network (CNN)</i> dan <i>K-Nearest Neighbor (KNN)</i>. Proses penelitian diawali dengan pengumpulan dataset sebanyak 320 citra daun bawang merah dari perkebunan di Desa Batu Noni, Kabupaten Enrekang. Citra tersebut kemudian melalui tahap preprocessing berupa cropping dan resize menjadi ukuran 600x600 piksel. Selanjutny</p>	<p>dengan CNN dapat meningkatkan akurasi klasifikasi citra penyakit daun secara signifikan, mencapai akurasi 100%, sehingga menunjukkan efektivitas metode dalam mengidentifikasi gejala visual pada daun tanaman.</p>	<p>gambar), yang membantu generalisasi model dan potensi performanya pada kasus nyata yang lebih kompleks dan bervariasi seperti pada daun cabai.</p>	<p><i>CNN</i> murni, mampu memberikan hasil maksimal dengan akurasi, presisi, recall, dan f1-score sebesar 100% untuk kedua jenis penyakit, yaitu bercak ungu dan moler. Sementara itu, klasifikasi menggunakan <i>HSV + KNN</i> juga menunjukkan performa yang baik, khususnya saat nilai parameter $k = 1$, yang juga menghasilkan nilai akurasi hingga 97%. Namun, nilai performa menurun pada k yang lebih</p>
--	---------------------	--	--	--	---	--

			<p>a dilakukan ekstraksi fitur warna menggunakan model Hue Saturation Value (<i>HSV</i>). Data kemudian diklasifikasikan menggunakan dua metode: pertama, dengan menggabungkan ekstraksi fitur <i>HSV</i> dan algoritma KNN menggunakan jarak Euclidean; kedua, dengan metode <i>CNN</i> baik dengan maupun tanpa ekstraksi fitur <i>HSV</i>. <i>CNN</i> yang digunakan terdiri dari beberapa lapisan utama, termasuk convolutio</p>		<p>besar seperti 3 atau 5. Berdasarkan hasil ini, metode <i>CNN</i> terbukti lebih unggul dalam mengklasifikasikan penyakit daun tanaman bawang merah dibandingkan KNN. Oleh karena itu, <i>CNN</i> dinyatakan sebagai pendekatan yang efektif, akurat, dan layak diterapkan dalam sistem deteksi penyakit tanaman secara otomatis.</p>
--	--	--	--	--	---

			n layer, pooling layer, dan fully connected layer, dengan kemampuan ekstraksi fitur otomatis. Evaluasi performa model dilakukan menggunakan confusion matrix dan metrik evaluasi seperti akurasi, presisi, recall, dan f1-score.			
4.	Puji Catur Siswip raptin, Abdul Haris, Winda Novita Sari (2023)	Klasifikasi Citra Penyakit Daun Cabai Menggunakan Algoritma <i>Learning Vector Quantization</i>	Metode <i>Learning Vector Quantization (LVQ)</i>	tingkat akurasi yang cukup tinggi (80%), presisi 80%, recall 82%, dan f1-score 81%, serta proses identifikasi yang relatif cepat dan	jumlah data uji yang terbatas dan sensitivitas terhadap kualitas dan pencahayaan gambar	Hasil penelitian menunjukkan bahwa model ini cukup efektif dalam membantu identifikasi dini tiga jenis penyakit daun cabai, yaitu Fitoftora, Antraknosa, dan

				sederhana		Cercospora, meskipun terdapat beberapa kesalahan klasifikasi antar jenis penyakit.
5.	Fani Nurona Cahya, Nila Hardi, Dwiza Riana, Sri Hadianti (2021)	Klasifikasi Penyakit Mata Menggunakan <i>Convolutional Neural Network</i> (CNN)	<i>Convolutional Neural Network</i> (CNN)	akurasi tinggi sebesar 98,37%, serta menunjukkan bahwa CNN mampu mengenali pola kompleks pada citra medis secara efisien	terdapat kesalahan klasifikasi, terutama pada gambar katarak yang dikenali sebagai mata normal	penelitian ini membuktikan bahwa CNN efektif untuk klasifikasi citra medis dan berpotensi diterapkan pada domain lain seperti pertanian.
6.	Budi Nugroho, Eva Yulia Puspaningrum (2021)	Kinerja Metode CNN Untuk Klasifikasi Pneumonia Dengan Variasi Ukuran Citra Input	CNN dan CNN-ELM	Penelitian ini menggabungkan CNN dengan ELM yang meningkatkan akurasi. Selain itu, diuji berbagai ukuran citra untuk mengetahui pengaruh resolusi	Tidak dibandingkan dengan arsitektur CNN lain secara langsung dan tidak menggunakan data eksternal untuk uji generalisasi.	CNN-ELM menunjukkan performa terbaik, terutama pada ukuran citra 200x200 piksel dengan akurasi 93,59% dan F1 Score 0,9474, mengungguli metode CNN biasa di semua

				terhadap performa.		skenario.
7.	Muhammad Atsilhan, Raditia Patmasari, S.T., M.T., RYunendah NurFu'adah, S.T., M.T (2021)	Sistem klasifikasi Penyakit kulit menggunakan convolutional Neural Network (Cnn)	<i>Convolutional Neural Network (CNN)</i>	Penelitian ini berhasil mencapai performa tinggi dengan akurasi 96,53%, menggunakan arsitektur CNN yang dioptimasi melalui uji jumlah hidden layer, jenis optimizer, dan nilai learning rate. Sistem ini efektif untuk klasifikasi multi-kelas penyakit kulit.	Penelitian terbatas pada ukuran citra kecil (64x64) yang bisa mengurangi informasi visual detail. Selain itu, tidak ada uji banding dengan metode lain atau validasi eksternal menggunakan dataset berbeda.	Model CNN dengan 5 hidden layer dan parameter terbaik berhasil mencapai akurasi 96,53%, precision 95%, recall 95%, F1-score 95%, dan loss 0,2486, tanpa indikasi overfitting pada data uji.
8.	Herdianto, Darmeli Nasution (2023)	Implementasi Metode CNN Untuk Klasifikasi Objek	Convolutional Neural Network (CNN)	CNN dapat mengekstraksi ciri dari gambar secara otomatis dan mendalam, serta	Dataset yang digunakan relatif kecil, hanya 120 citra secara total, sehingga berpotensi	Model CNN berhasil mencapai akurasi 98% dalam klasifikasi objek setelah pelatihan dengan 15

				mampu mengklasifikasikan objek dengan tingkat akurasi tinggi. Sistem menunjukkan kemampuan mendeteksi beragam objek seperti manusia, sepeda motor, kambing, dan lainnya dengan hasil yang tepat.	si menurun generalisasi model. Selain itu, pengujian belum dibandingkan dengan metode lain untuk validasi performa secara komprehensif.	epoch. Pengujian menunjukkan sistem mampu mengklasifikasi sebagian besar citra dengan benar, meskipun ada kesalahan pada beberapa citra seperti monitor yang terdeteksi sebagai manusia.
9.	Diki Hananta Firdaus, Bahtiar Imran, Lalu Darmawan Bakti, Emi Suryadi (2022)	Klasifikasi Penyakit Katarak Pada Mata Menggunakan Metode <i>Convolutional Neural Network (CNN)</i> Berbasis Web	Convolutional Neural Network (CNN)	Model CNN yang dibangun memiliki akurasi sangat tinggi pada tahap pelatihan, mencapai 99,74% pada epoch ke-25. Sistem juga	Pengujian hanya dilakukan dengan 60 gambar, yang relatif sedikit untuk mengukur generalisasi sistem. Selain itu, model	Hasil pelatihan terbaik mencapai akurasi 99,74%, sedangkan hasil pengujian pada 60 data uji menunjukkan akurasi 93,33%, dengan 56 gambar diklasifikasi benar

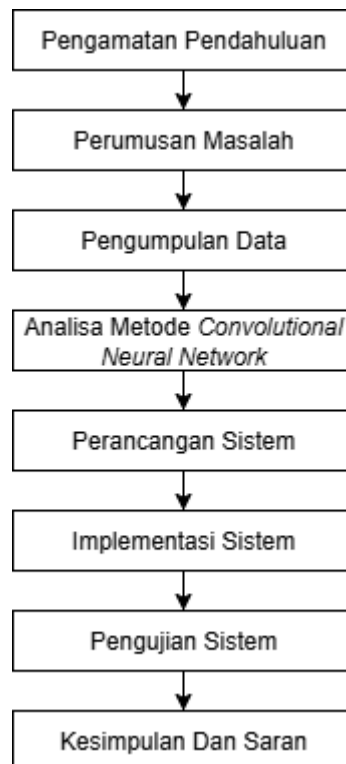
				berhasil diimplementasikan secara praktis dalam aplikasi web, sehingga memudahkan pengguna awam dalam mendeteksi katarak.	hanya menanganinya dua kelas (biner), tanpa eksplorasi multi-klasifikasi jenis penyakit mata lainnya.	dan gambar salah. Model CNN dinilai sangat efektif untuk identifikasi awal katarak.	4
10.	Grama ndha Wega Intyant o (2021)	Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: <i>CNN</i> (<i>Convolution Neural Network</i>)	<i>CNN</i> (<i>Convoluti onal Neural Network</i>)	Penelitian ini membandingkan performa dua arsitektur CNN yang berbeda, sehingga memberikan wawasan empiris mengenai efektivitas VGG16 dibandingkan arsitektur sederhana buatan sendiri. Proses klasifikasi dilakukan dengan	Arsitektur pertama menunjukkan gejala overfitting, ditunjukkan dari perbedaan signifikan antara performa pelatihan dan validasi. Selain itu, evaluasi hanya dilihat dari akurasi tanpa tambaha	Arsitektur buatan sendiri menghasilkan akurasi 62%, sedangkan arsitektur VGG16 mencapai akurasi 80%, dengan loss lebih rendah. VGG16 terbukti lebih stabil dan efisien dalam mengenali gambar bunga.	

				tahapan lengkap (konvolusi, ReLU, pooling, flatten, dense, <i>Softmax</i>).	n seperti precision, recall, atau F1-score.	
--	--	--	--	--	---	--

BAB 3

METODOLOGI PENELITIAN

Tahapan penelitian yang akan dilakukan dan penyelesaian masalah terhadap Klasifikasi Penyakit Daun Cabai Menggunakan *CNN* Dan Teknik Konversi Warna *HSV* Untuk Meningkatkan Akurasi Deteksi Citra Daun. Adapun tahapan metodologi yang dilakukan selama penelitian dapat dilihat pada Gambar 3.1, yang mana merupakan proses yang dimulai dari studi literatur hingga diperoleh kesimpulan.



Gambar 3. 1 Tahapan Penelitian

Tahapan dalam penelitian ini dijabarkan secara sistematis untuk menjamin alur kerja yang logis dan terstruktur, dimulai dari pengamatan pendahuluan hingga Kesimpulan dan saran:

3.1 Pengamatan Pendahuluan

Pengamatan pendahuluan dilakukan sebagai langkah awal untuk memahami permasalahan yang terjadi pada budidaya tanaman cabai, khususnya terkait dengan penyakit yang menyerang daun tanaman. Dalam tahap ini, peneliti melakukan wawancara informal dengan petani cabai di daerah Gunung Intan yaitu Bapak Muktar Lusupi untuk mengidentifikasi jenis-jenis penyakit daun yang umum terjadi serta dampaknya terhadap pertumbuhan tanaman dan hasil panen.

Hasil dari pengamatan menunjukkan bahwa terdapat beberapa gejala umum yang sering dijumpai pada daun cabai yang terserang penyakit, seperti daun menguning (*yellowwiss*), daun menggulung atau keriting (*leafcurl*), dan daun menjadi layu. Permasalahan ini sangat berdampak pada produktivitas tanaman karena penyakit daun dapat menghambat proses fotosintesis dan pertumbuhan tanaman secara keseluruhan.

Dalam proses pengamatan, ditemukan bahwa sebagian besar petani masih mengandalkan metode visual dan pengalaman pribadi untuk mendeteksi penyakit tanaman. Namun, metode manual ini memiliki beberapa kelemahan, antara lain ketergantungan pada keahlian individu, subjektivitas dalam pengamatan, serta waktu deteksi yang cukup lama. Kondisi ini menunjukkan perlunya sebuah sistem deteksi penyakit daun cabai yang otomatis, cepat, dan akurat.

Berdasarkan hasil pengamatan ini, peneliti kemudian menggagas solusi berbasis teknologi kecerdasan buatan, yaitu menggunakan metode *Convolutional Neural Network (CNN)* yang dikombinasikan dengan teknik konversi warna *HSV*. Teknologi ini diharapkan mampu memberikan akurasi tinggi dalam

mengidentifikasi dan mengklasifikasikan penyakit daun cabai secara otomatis, sekaligus menjadi alat bantu yang efektif bagi para petani dalam melakukan deteksi dini terhadap serangan penyakit pada tanaman mereka.

3.2 Perumusan Masalah

Perumusan masalah dalam penelitian ini didasarkan pada hasil pengamatan awal dan studi literatur yang telah dilakukan. Penyakit pada daun cabai menjadi salah satu faktor utama yang memengaruhi penurunan produktivitas tanaman cabai di Indonesia. Proses identifikasi penyakit secara manual memiliki keterbatasan dalam hal kecepatan, konsistensi, dan akurasi, sehingga dibutuhkan pendekatan berbasis teknologi untuk membantu petani dalam mendeteksi penyakit daun secara otomatis.

Dari kondisi tersebut, dirumuskan permasalahan utama sebagai berikut: Bagaimana merancang dan mengimplementasikan model klasifikasi penyakit daun cabai dengan metode *Convolutional Neural Network (CNN)* yang diintegrasikan dengan teknik konversi warna *HSV*, guna meningkatkan akurasi deteksi citra daun cabai?

Permasalahan ini menjadi dasar bagi seluruh tahapan penelitian, mulai dari pengumpulan data, pemrosesan citra, pembangunan model *CNN*, hingga evaluasi performa klasifikasi dengan dan tanpa penggunaan ruang warna *HSV*.

3.3 Pengumpulan Data (*Data Collection*)

Dataset yang digunakan dalam penelitian ini diperoleh dari *Kaggle* yang terdapat pada link berikut ini:

<https://www.kaggle.com/datasets/ratnasarii/penyakitdauncabai/data>

yaitu kumpulan citra daun cabai yang mengandung berbagai jenis penyakit serta citra daun cabai yang sehat. Dataset tersebut sudah memiliki label klasifikasi yaitu daun sehat, *leafcurl* dan *yellowwiss*. Data diunduh dalam format gambar (.jpg/.png) dan kemudian digunakan untuk pelatihan dan pengujian model *CNN*. Penggunaan dataset dari *Kaggle* dipilih karena menyediakan data dengan kualitas baik, jumlah memadai, dan telah digunakan dalam berbagai penelitian sebelumnya, sehingga memperkuat validitas dan reliabilitas penelitian ini.

3.4 Analisa Metode *Convolutional Neural Network*

Metode *Convolutional Neural Network (CNN)* merupakan pendekatan deep learning yang paling banyak digunakan untuk klasifikasi citra karena kemampuannya dalam mengekstraksi dan mengenali pola visual secara otomatis. *CNN* bekerja dengan menggabungkan beberapa lapisan (*layers*) seperti *convolutional layer*, *activation function*, *pooling layer*, *flatten*, dan *fully connected layer*, yang secara bertahap memproses citra dari bentuk mentah menjadi representasi fitur yang dapat digunakan untuk klasifikasi.

Pada penelitian ini, *CNN* digunakan untuk mengklasifikasikan tiga jenis citra daun cabai, yaitu:

1. Daun Sehat (*Healthy*)
2. Daun Terkena *Leafcurl* (Keriting)
3. Daun Terkena *Yellowwiss* (Virus Kuning)

Arsitektur *CNN* yang akan dianalisis dan dibangun memiliki alur sebagai berikut:

1. *Input Layer*

Menerima input citra daun dengan ukuran tetap, yaitu 224x224 piksel dengan 3 kanal warna (*RGB* atau setelah dikonversi ke *HSV*).

2. *Convolutional Layer*

Lapisan ini mengekstraksi fitur lokal dari citra menggunakan filter (*kernel*). Filter akan mendeteksi pola visual seperti tepi, tekstur, atau bentuk gejala penyakit pada daun.

3. *Activation Function (ReLU)*

Fungsi aktivasi *ReLU* digunakan setelah proses konvolusi untuk menambah non-linearitas dan mempercepat proses konvergensi.

4. *Pooling Layer (MaxPooling)*

Pooling bertujuan untuk mereduksi dimensi fitur dan mempertahankan informasi penting. *MaxPooling* digunakan untuk mengambil nilai maksimum dari area fitur.

5. *Flatten Layer*

Mengubah matriks hasil *pooling* menjadi vektor satu dimensi yang bisa diproses oleh lapisan *fully connected*.

6. *Fully Connected Layer*

Lapisan ini berfungsi untuk menggabungkan fitur dan melakukan klasifikasi akhir berdasarkan fitur yang telah diekstraksi.

7. *Output Layer*

Menggunakan fungsi aktivasi *Softmax* untuk mengklasifikasikan citra daun ke dalam tiga kelas berdasarkan probabilitas tertinggi.

8. *Dropout*

Dropout diterapkan pada dense layer untuk mengurangi *overfitting* dengan menonaktifkan neuron secara acak saat pelatihan.

Model dikembangkan dan dilatih menggunakan platform *Google Colab* dengan bantuan *library* seperti *TensorFlow*, *Keras*, dan *OpenCV*. Evaluasi performa dilakukan dengan menghitung metrik akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix*.

Analisis metode ini bertujuan untuk mengetahui konfigurasi *CNN* terbaik yang menghasilkan akurasi tinggi dan mampu membedakan penyakit daun cabai dengan baik, khususnya setelah diterapkannya teknik konversi warna *HSV*.

3.5 Perancangan Sistem

Perancangan sistem pada penelitian ini bertujuan untuk membangun alur klasifikasi citra daun cabai yang efisien dan akurat menggunakan metode *Convolutional Neural Network (CNN)* yang diintegrasikan dengan teknik konversi warna *HSV*. Sistem ini dirancang untuk memproses citra daun dari tahap awal (input citra) hingga menghasilkan output berupa kelas penyakit yang terdeteksi.

Perancangan sistem ini dibagi menjadi beberapa tahapan sebagai berikut:

1. Input Data

Sistem menerima masukan berupa citra daun cabai dalam format *JPG* atau *PNG*. Gambar dikonversi ke ukuran tetap (224x224 piksel) agar seragam saat diproses oleh model *CNN*.

2. Pra-Pemrosesan (*Preprocessing*)

Resize gambar ke resolusi 224x224 piksel.

Normalisasi piksel ke skala 0–1.

Konversi warna dari *RGB* ke *HSV*, untuk memisahkan informasi warna dari pencahayaan dan meningkatkan ketahanan terhadap variasi intensitas cahaya.

Augmentasi data dilakukan dengan metode rotasi, *flipping*, *zoom*, dan brightness shift, agar model dapat mengenali pola dari berbagai variasi citra.

3. Ekstraksi Fitur Otomatis dengan *CNN*

Citra hasil pra-pemrosesan dimasukkan ke dalam model *CNN*.

CNN akan mengekstrak fitur penting seperti tekstur, pola kerusakan daun, bentuk gulungan, dan distribusi warna yang relevan dengan penyakit daun cabai.

4. Klasifikasi

Lapisan *fully connected* dan *output layer* akan memproses fitur hasil ekstraksi dan menentukan kelas citra daun, yaitu:

1. *Healthy* (Sehat)
2. *Leafcurl* (Keriting)
3. *Yellowwviss* (Virus Kuning)

5. Output Sistem

Sistem menghasilkan output berupa label kelas penyakit daun beserta probabilitas kepercayaannya.

Output juga ditampilkan dalam bentuk visual (grafik akurasi, *confusion matrix*, dan hasil prediksi).

3.6 Implementasi Sistem

Implementasi sistem dilakukan dengan merealisasikan seluruh tahapan yang telah dirancang dalam bentuk program berbasis *Python* menggunakan *platform Google Colab*. Dataset citra daun cabai yang telah diklasifikasikan ke dalam tiga kategori, yaitu daun sehat, *leafcurl*, dan *yellowwiss*, diunggah dan diproses melalui tahap pra-pemrosesan berupa *resize*, *normalisasi*, konversi warna dari *RGB* ke *HSV*, serta *augmentasi data* untuk meningkatkan variasi dan daya generalisasi model. Model *Convolutional Neural Network (CNN)* kemudian dibangun dengan beberapa lapisan seperti *convolutional*, *pooling*, *flatten*, dan *fully connected*, serta menggunakan fungsi aktivasi *ReLU* dan *Softmax* untuk proses klasifikasi akhir. Pelatihan dilakukan selama beberapa *epoch* dengan data *training* dan *validasi*, sedangkan pengujian dilakukan terhadap data uji menggunakan metrik evaluasi seperti *akurasi*, *precision*, *recall*, *F1-score*, dan *confusion matrix*. Seluruh proses implementasi ini bertujuan untuk membangun sistem klasifikasi penyakit daun cabai yang efisien, akurat, dan dapat digunakan untuk mendeteksi penyakit secara otomatis berdasarkan citra daun yang dimasukkan ke dalam sistem.

3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi kinerja model *CNN* dalam mengklasifikasikan citra daun cabai ke dalam tiga kelas, yaitu sehat, *leafcurl*, dan *yellowwiss*, baik dengan maupun tanpa penggunaan konversi warna *HSV*. Data pengujian yang telah dipisahkan sejak awal digunakan untuk mengukur kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya. Evaluasi dilakukan menggunakan metrik *akurasi*, *precision*, *recall*,

dan *F1-score*, serta ditunjang oleh confusion matrix untuk melihat distribusi prediksi terhadap masing-masing kelas. Hasil pengujian diinterpretasikan dalam bentuk grafik akurasi dan loss, serta visualisasi prediksi terhadap beberapa contoh citra, sehingga dapat disimpulkan sejauh mana sistem mampu mendeteksi penyakit daun cabai secara otomatis dan andal.

3.8 Kesimpulan Dan Saran

Tahapan terakhir adalah menarik kesimpulan dari hasil penelitian yang didapatkan dari Implementasi Metode *Convolutional Neural Network* Dan Teknik Konversi Warna *HSV* Untuk Klasifikasi Penyakit Daun Cabai. Pada tahapan ini juga berisikan saran peneliti bagi pembaca untuk melakukan pengembangan terhadap penelitian ini kedepannya.