

**IDENTIFIKASI PENYAKIT TANAMAN CABAI MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN CITRA DAUN**

SKRIPSI

ANDI SAPUTRA SITORUS

191401048



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023**

**IDENTIFIKASI PENYAKIT TANAMAN CABAI MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN CITRA DAUN**

SKRIPSI

ANDI SAPUTRA SITORUS

191401048



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023**

PENGESAHAN

Judul : IDENTIFIKASI PENYAKIT TANAMAN CABAI
MENGUNAKAN METODE CONVOLUTIONAL
NEURAL NETWORK (CNN) BERDASARKAN CITRA
DAUN

Kategori : SKRIPSI

Nama : ANDI SAPUTRA SITORUS

Nomor Induk Mahasiswa : 191401048

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Telah diuji dan dinyatakan lulus di Medan, 5 Januari 2024

Dosen Pembimbing II



Dewi Sartika Br Ginting, S.Kom., M.Kom
NIP.199005042019032023

Dosen Pembimbing I



Prof. Dr. Poltak Sihombing, M.Kom
NIP.196203171991031001

Diketahui/Disetujui Oleh

Ketua Program Studi Ilmu Komputer



Dr. Amalia, S.T, M.T
NIP. 197812212014042001

PERNYATAAN**IDENTIFIKASI PENYAKIT TANAMAN CABAI MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN CITRA DAUN****SKRIPSI**

Saya menyatakan bahwa skripsi ini adalah hasil dari penelitian pribadi saya, dengan pengecualian beberapa kutipan dan ringkasan yang masing-masing sudah saya beri sumbernya.

Medan, 5 Januari 2024



Andi Saputra Sitorus

191401048

PENGHORMATAN

Dengan rasa syukur yang mendalam, penulis mengungkapkan rasa terima kasih yang sangat besar kepada Allah SWT atas anugerah-Nya, petunjuk-Nya, dan kekuatan-Nya yang terus melimpah selama proses penulisan skripsi ini, sehingga mungkin bagi penulis untuk menyelesaikan penyusunan skripsi ini.

Pada kesempatan ini, penulis ingin mengungkapkan rasa terima kasih dan penghargaan yang besar kepada:

1. Bapak Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lidya, B.Sc, M.Sc selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
3. Ibu Dr. Amalia, S.T., M.T selaku Ketua Prodi S-1 Ilmu Komputer Universitas Sumatera Utara
4. Bapak Jos Timanta Tarigan S.Kom., M.Sc selaku Dosen Pembimbing Akademik.
5. Bapak Prof. Dr. Poltak Sihombing, M.Kom, sebagai Dosen Pembimbing I, yang telah memberikan bimbingan, arahan, dan inspirasi yang sangat berharga. Keberhasilan penyelesaian skripsi ini tidak terlepas dari dukungan, pemahaman, dan kesabaran Bapak sebagai pembimbing.
6. Ibu Dewi Sartika Br Ginting, S.Kom., M.Kom, sebagai Dosen Pembimbing II, yang telah memberikan bimbingan, arahan, dan inspirasi yang sangat berharga. Keberhasilan penyelesaian skripsi ini tidak terlepas dari dukungan, pemahaman, dan kesabaran Ibu sebagai pembimbing.
7. Seluruh tenaga pengajar dan pegawai di Program Studi S-1 Ilmu Komputer yang telah memberikan dukungan dan memfasilitasi dalam berbagai tahapan penyusunan skripsi ini..
8. Orang tua, yakni Ayahanda Thamrin Sitorus dan Ibunda Sunarti yang senantiasa memberikan doa, dukungan dan kasih sayang yang tiada henti, serta bantuan dalam berbagai bentuk selama perjalanan penelitian ini.

9. Rekan-rekan seperjuangan yang selalu memberikan dorongan semangat kepada penulis.
10. Dan kepada semua pihak, baik individu maupun kelompok, yang telah memberikan bantuan, walaupun tidak dapat diungkapkan satu per satu.

Sebagai penutup, penulis menyampaikan rasa terima kasih yang setinggi-tingginya kepada semua yang ikut berperan dalam perjalanan penulisan skripsi ini. Semoga hasil penelitian ini dapat memberikan manfaat yang berarti serta kontribusi positif bagi kemajuan ilmu pengetahuan dan kesejahteraan masyarakat.

Medan, 5 Januari 2024



Andi Saputra Sitorus
191401048

ABSTRAK

Penyakit pada tanaman cabai merupakan masalah penting dalam dunia pertanian yang dapat mempengaruhi hasil panen dan produktivitas. Tujuan dari penelitian ini adalah mengidentifikasi penyakit pada tanaman cabai dengan menerapkan metode CNN dan memanfaatkan model VGG16 sebagai alat bantu, dengan penekanan citra daun sebagai indikator penyakit. Penelitian ini mencakup pengumpulan data citra daun cabai yang terkena berbagai penyakit, pelatihan model CNN, dan evaluasi hasil klasifikasi. Hasil penelitian menunjukkan bahwa penggunaan CNN dengan ekstraksi fitur warna daun dapat mengklasifikasikan penyakit pada tanaman cabai dengan tingkat akurasi yang memadai. Tingkat akurasi yang berhasil diperoleh mencapai 92%, dengan tingkat *precision* sekitar 92%. Tingkat *recall* juga mencapai 92%, sementara nilai *F1-score* mencapai 92%. Dari hasil tersebut menunjukkan bahwa kemampuan model dalam mengidentifikasi jenis-jenis penyakit pada tanaman cabai memberikan hasil yang cukup baik. Hasil penelitian ini memiliki potensi untuk membantu para petani menghemat waktu dan biaya, dapat mengetahui penyakit yang menyerang tanaman cabai dan dapat membantu dalam pencegahan secara efisien dan akurat.

Kata kunci: Tanaman Cabai, *Machine Learning*, *Neural Network*, *Convolutional Neural Network*, VGG16, Pertanian.

ABSTRACT

Diseases in chili plants are a significant issue in the world of agriculture that can affect harvest yields and productivity. The aim of this research is to identify diseases in chili plants by applying CNN (Convolutional Neural Network) methods and utilizing the VGG16 model as a tool, with an emphasis on leaf images as disease indicators. This study encompasses the collection of leaf images from chili plants affected by various diseases, the training of the CNN model, and the evaluation of classification results. The research results indicate that the use of CNN with leaf color feature extraction can classify diseases in chili plants with an adequate level of accuracy. The achieved accuracy rate is 92%, with a precision rate of approximately 92%. The recall rate also reaches 92%, while the F1-score value reaches 92%. These results demonstrate that the model's ability to identify various diseases in chili plants yields satisfactory outcomes. The findings of this research have the potential to assist farmers in saving time and costs, enabling them to identify diseases affecting chili plants and aiding in efficient and accurate prevention.

Keywords: *Chili Plants, Machine Learning, Neural Network, Convolutional Neural Network, VGG16, Agriculture.*

DAFTAR ISI

PENGESAHAN.....	iii
PERNYATAAN	iv
PENGHORMATAN	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB 1.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	5
1.7.1. BAB 1 Pendahuluan	5
1.7.2. BAB 2 Tinjauan Pustaka	5
1.7.3. BAB 3 Analisis dan Perancangan.....	5
1.7.4. BAB 4 Implementasi dan Pengujian	5
1.7.5. BAB 5 Kesimpulan dan Saran.....	5
BAB 2.....	6
TINJAUAN PUSTAKA.....	6
2.1. Tanaman Cabai.....	6
2.2. Penyakit Tanaman Cabai.....	6
2.2.1. Daun Keriting	6
2.2.2. Kutu Kebul	6

2.2.3. Penyakit Kekuningan.....	7
2.2.4. Bercak Daun	7
2.3. Deep Learning.....	7
2.4. <i>Machine Learning</i>	7
2.5. <i>Convolutional Neural Network (CNN)</i>	8
2.5.1. <i>Convolution Layer</i>	9
2.5.2. <i>Pooling Layer</i>	11
2.5.3. <i>Fully-Connected Layer</i>	12
2.5.4. <i>Softmax</i>	13
2.5.5. <i>ReLu (Rectified Linear Unit)</i>	14
2.6. Arsitektur VGG16.....	15
2.7. <i>Neural Network</i>	16
BAB 3	17
ARSITEKTUR DAN PERANCANGAN SISTEM	17
3.1. Analisis Sistem.....	17
3.1.1. Analisis Masalah	17
3.1.2. Analisis Kebutuhan.....	18
3.1.3. Analisis Proses.....	19
3.1.4. Arsitektur Umum.....	22
3.2. Pemodelan Sistem	23
3.2.1. <i>Use Case Diagram</i>	23
3.2.2. <i>Activity Diagram</i>	24
3.2.3. <i>Sequence Diagram</i>	25
3.3. <i>Flowchart</i>	26
3.4. Perancangan Antarmuka Sistem	28
3.4.1. Halaman Awal	28
3.4.2. Halaman Pengujian.....	28
BAB 4	30
IMPLEMENTASI DAN PENGUJIAN	30
4.1. Implementasi Sistem	30
4.1.1. Halaman Utama.....	30
4.1.2. Halaman Pengujian.....	31

4.2. Perancangan Model	32
4.2.1. Perancangan Model CNN.....	32
4.2.2. Perancangan Model VGG16	34
4.3. Penentuan Parameter Model	37
4.3.1. Pengaruh Banyaknya Epoch.....	37
4.4. Pengujian Sistem.....	38
4.4.1. Hasil Proses <i>Training</i>	38
4.4.2. Hasil Proses <i>Testing</i>	40
BAB 5.....	58
KESIMPULAN DAN SARAN.....	58
5.1. Kesimpulan	58
5.2. Saran.....	58
DAFTAR PUSTAKA.....	59

DAFTAR GAMBAR

2.1. Gambar Arsitektur Algoritma CNN	9
2.2. Gambar <i>Convolution Layer</i>	9
2.3. Gambar <i>Stride</i>	10
2.4. Gambar <i>Padding</i>	11
2.5. Gambar <i>Pooling Layer</i>	12
2.6. Gambar <i>Fully Connected Layer</i>	13
2.7. Gambar <i>ReLu Function</i>	14
2.8. Gambar Arsitektur VGG16	16
2.9. Gambar Arsitektur <i>Neural Network</i>	16
3.1. Gambar <i>Fishbone Diagram</i>	17
3.2. Gambar Daun Cabai Sehat	19
3.3. Gambar Daun Keriting	20
3.4. Gambar Daun Kutu Kebul.....	20
3.5. Gambar Daun Penyakit Kekuningan	21
3.6. Gambar Berak Daun	21
3.7. Gambar Arsitektur Umum Sistem.....	22
3.8. Gambar <i>Use Case Diagram</i>	24
3.9. Gambar <i>Activity Diagram</i>	25
3.10. Gambar <i>Sequence Diagram</i>	26
3.11. Gambar <i>Flowchart</i> Sistem.....	27
3.12. Gambar Rancangan Halaman Awal Sistem	28
3.13. Gambar Rancangan Halaman Pengujian Sistem	29
4.1. Gambar Halaman Awal Sistem	30
4.2. Gambar Halaman Pengujian Sistem.....	31
4.3. Gambar <i>Summary Model CNN</i>	32
4.4. Gambar Implementasi Model CNN.....	33
4.5. Gambar <i>Summary Model VGG16</i>	34

4.6. Gambar Implementasi Model CNN	35
4.7. Gambar Hasil Proses <i>Training</i>	39
4.8. Gambar Hasil Akurasi Proses <i>Training</i>	40

DAFTAR TABEL

3.1. Tabel Pembagian Dataset	22
4.1. Tabel <i>Epoch</i>	38
4.2. Tabel Pengujian Daun Cabai Normal.....	40
4.3. Tabel Pengujian Daun Keriting	43
4.4. Tabel Pengujian Bercak Daun	46
4.5. Tabel Pengujian Penyakit Kutu Kebul	48
4.6. Tabel Pengujian Penyakit Kekuningan	51
4.7. Tabel <i>Confusion Matrix</i>	54
4.8. Tabel <i>Precision</i>	55
4.9. Tabel <i>Recall</i>	56

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Tanaman cabai memiliki nilai ekonomi yang penting. Selain berperan sebagai tanaman pertanian yang esensial dalam kehidupan sehari-hari, cabai juga memberikan nilai tambah karena selain digunakan sebagai bumbu masakan, tanaman ini mengandung kapsaikin, vitamin A, vitamin C, dan antioksidan.

Tanaman cabai adalah salah satu komoditas pertanian yang memiliki peran penting dalam menyediakan bahan pangan dan mendukung ekonomi masyarakat. Namun, serangan penyakit pada tanaman cabai dapat menyebabkan penurunan hasil panen yang signifikan. Tanaman ini rentan terkena penyakit dan serangan hama. Penyakit tersebut dapat berasal dari berbagai faktor, termasuk serangan hama, infeksi penyakit, serta kondisi lingkungan yang tidak mendukung.

Deteksi penyakit pada tanaman cabai memerlukan pendekatan yang efektif dan cepat untuk mencegah penyebaran infeksi dan meminimalkan kerugian pada hasil pertanian, namun jika dilakukan secara manual dapat memerlukan investasi waktu dan biaya yang besar. Oleh karena itu, penerapan teknologi dalam metode identifikasi penyakit pada tanaman cabai dapat membantu petani menghemat waktu dan biaya, sehingga dapat berperan dalam pencegahan penyakit pada tanaman cabai secara efisien dan akurat.

Dalam usaha untuk mengatasi permasalahan ini, penggunaan metode identifikasi penyakit pada tanaman cabai dengan *Convolutional Neural Network* (CNN) dan pemanfaatan model VGG16 sebagai alat bantu dapat diimplementasikan. CNN digunakan untuk mengekstrak dan mengklasifikasi citra secara otomatis. Metode ini memungkinkan petani untuk melakukan diagnosis penyakit pada tanaman cabai dengan cepat dan akurat berdasarkan gambar daun yang terinfeksi penyakit..

Metode CNN memiliki kapabilitas untuk belajar dari data dan mengekstraksi fitur-fitur penting dari gambar yang digunakan dalam proses pelatihan. Setelah model CNN diberikan

pelatihan, model tersebut dapat digunakan untuk secara otomatis dan akurat mendiagnosis penyakit pada tanaman cabai. Hal ini memberikan dukungan kepada para petani untuk mengambil tindakan dengan cepat dan tepat dalam menangani masalah penyakit pada tanaman cabai, sehingga dapat meningkatkan produksi dan kualitas tanaman cabai.

Salah satu penelitian yang menerapkan metode CNN dilakukan oleh Errissya Rasywir, Rudolf Sinaga, dan Yovi Pratama. Dalam penelitian ini, CNN digunakan untuk menganalisis dan mengimplementasikan diagnosis penyakit pada kelapa sawit. Dengan menguji 2490 citra kelapa sawit yang terkategori sebagai penyakit dalam 11 kategori, penelitian ini mencapai hasil akurasi tertinggi sebesar 0.89, terendah 0.87, dan rata-rata akurasi sebesar 0.87. Penelitian lainnya yang relevan dilakukan oleh Abdul Jalil Rozaqi, Andi Sunyoto, dan Rudyanto Arief. Pada penelitian tersebut, metode CNN digunakan untuk mendeteksi penyakit pada daun kentang, mencapai nilai akurasi sebesar 95%, sementara akurasi validasi mencapai 94%.

Dikarenakan hasil akurasi yang memuaskan dari penelitian-penelitian sebelumnya yang mengaplikasikan metode CNN, algoritma ini akan diadopsi untuk membangun sistem yang mampu menganalisis dan mengklasifikasikan jenis penyakit pada tanaman cabai, seperti Daun keriting, kutu kebul, penyakit kekuningan, dan bercak daun. Dalam situasi ini, representasi warna daun diartikan sebagai gambar, dan CNN memiliki kemampuan untuk memahami pola warna daun yang terkait dengan jenis penyakit tertentu. Dengan begitu, metode ini diharapkan dapat memberikan kontribusi dalam upaya mengetahui penyakit pada tanaman cabai dengan cara yang efisien dan akurat.

1.2. Rumusan Masalah

Deteksi penyakit pada tanaman cabai memerlukan pendekatan yang efektif dan cepat untuk mencegah penyebaran infeksi dan meminimalkan kerugian pada hasil pertanian, namun jika dilakukan secara manual dapat memerlukan investasi waktu dan biaya yang besar. Oleh karena itu, dibutuhkan suatu sistem yang dapat membantu petani menghemat waktu dan biaya yang diperlukan dalam mengidentifikasi penyakit pada tanaman cabai,

sehingga mampu memberikan dukungan yang efektif dan akurat kepada petani dalam upaya pencegahan penyakit pada tanaman cabai.

1.3. Batasan Masalah

Dalam melakukan penelitian, batasan masalah yang akan diteliti mencakup:

1. Melakukan identifikasi antara daun cabai sehat dan 4 jenis penyakit tanaman cabai, diantaranya yaitu penyakit daun keriting, kutu kebul, penyakit kekuningan dan bercak daun.
2. Menerapkan metode CNN dan memanfaatkan model VGG 16.
3. *Output* yang dihasilkan adalah hasil klasifikasi jenis penyakit daun.
4. Sistem aplikasi dibangun berbasis *web* dengan menggunakan bahasa pemrograman *python*.

1.4. Tujuan Penelitian

Tujuan penelitian ini berfokus pada identifikasi penyakit tanaman cabai dengan memanfaatkan metode CNN dan menitikberatkan pada citra daun sebagai pendekatan utama.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini di antara lain yaitu:

1. Dapat membantu petani cabai untuk mengidentifikasi penyakit tanaman cabai dengan lebih efisien dan akurat.
2. Dapat membantu petani cabai untuk meminimalkan kerugian ekonomi yang diakibatkan oleh penyakit tanaman cabai.
3. Dapat digunakan sebagai referensi untuk penelitian berikutnya di bidang *image processing*.

1.6. Metodologi Penelitian

Metodologi penelitian yang dilakukan dalam penelitian ini adalah :

1. Studi Literatur

Tahap ini dilakukan guna mengumpulkan data sebagai referensi penelitian ini dari berbagai jurnal, buku, artikel, dan sumber referensi lainnya yang relevan dengan pengolahan citra, penyakit pada tanaman cabai, dan algoritma *Convolutional Neural Network* (CNN).

2. Analisis dan Perancangan

Pada tahap ini, dilakukan perancangan arsitektur sistem yang sesuai dengan analisis permasalahan yang telah dilakukan. Dalam penelitian ini, akan melibatkan perancangan *flowchart* algoritma, perancangan aplikasi, dan antarmuka pengguna (UI) aplikasi.

3. Implementasi

Implementasi dilakukan di komputer yang menjalankan sistem operasi *Microsoft Windows* dan menggunakan bahasa pemrograman *Python*.

4. Pengujian

Pada fase ini, uji coba aplikasi dilakukan untuk memastikan bahwa aplikasi yang telah dikembangkan berfungsi sesuai dengan yang diinginkan, dan melakukan perbaikan jika masih ditemukan kesalahan dalam aplikasi.

5. Dokumentasi dan Penyusunan Laporan

Langkah akhir yang dijalankan melibatkan pembuatan dokumentasi dan penyusunan laporan berdasarkan temuan dari penelitian yang telah dilakukan.

1.7. Sistematika Penulisan

1.7.1. BAB 1 PENDAHULUAN

Bagian ini mencakup latar belakang penelitian, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

1.7.2. BAB 2 TINJAUAN PUSTAKA

Pada bagian ini dijelaskan teori, metode, dan komponen-komponen yang relevan mengenai tanaman cabai, citra digital, *machine learning*, algoritma CNN, serta pengenalan penyakit pada daun cabai.

1.7.3. BAB 3 ANALISIS DAN PERANCANGAN

Bagian ini mencakup perumusan masalah penelitian, analisis proses, dan perancangan sistem aplikasi. Tujuan dari tahap ini adalah menghasilkan desain penelitian yang sesuai dengan tujuan penelitian dan dapat menangani masalah yang telah dirumuskan sebelumnya.

1.7.4. IMPLEMENTASI DAN PENGUJIAN

Fokus utama dari bagian ini adalah penerapan algoritma CNN dalam aplikasi untuk mengklasifikasikan penyakit pada tanaman cabai, sekaligus membahas dan melakukan pengujian kepada sistem yang dirancang sesuai dengan permasalahan yang sudah dijelaskan sebelumnya.

1.7.5. KESIMPULAN DAN SARAN

Bagian ini mencakup ringkasan, hasil dan penilaian terhadap pencapaian tujuan penelitian, serta saran yang dapat mendukung penelitian di masa mendatang.

BAB 2

TINJAUAN PUSTAKA

2.1. Tanaman Cabai

Tanaman cabai merupakan jenis tanaman rempah dengan nilai ekonomis tinggi dan terkenal sebagai sumber bumbu pedas yang digunakan dalam masakan di seluruh dunia. Tanaman cabai termasuk dalam keluarga *Solanaceae* dan umumnya tumbuh sebagai tanaman tahunan. Dalam proses budidaya tanaman cabai, tanah yang subur dan kondisi kelembapan yang baik sangat dibutuhkan untuk mendapatkan hasil yang optimal. Selain itu, faktor-faktor seperti penyiraman teratur, pemupukan, dan pengendalian hama serta penyakit juga perlu diperhatikan dalam pemeliharaan tanaman cabai.

2.2. Penyakit Tanaman Cabai

Penyakit pada tanaman cabai memiliki potensi untuk mengurangi produksi dan kualitas buah cabai, yang tentunya dapat memberikan kerugian signifikan bagi para petani cabai. Beberapa penyakit tanaman cabai meliputi:

2.2.1. Daun Keriting

Penyakit ini diinduksi oleh virus yang tersebar melalui serangga pemakan daun seperti *whitefly*. Gejalanya mencakup perubahan bentuk daun yang menjadi keriting dan membengkak, serta munculnya warna gelap atau coklat pada daun. Virus ini juga dapat menyebar ke buah cabai yang dapat mengakibatkan penurunan hasil panen.

2.2.2. Kutu Kebul

Penyakit ini diinduksi oleh serangan kutu kebul (*Aphis* spp.) sering disebut sebagai penyakit kutu kebul pada cabai. Serangan kutu kebul dapat merusak tanaman cabai dengan mengganggu pertumbuhan, menyebabkan daun menguning, dan menimbulkan kerusakan lainnya. Selain itu, kutu kebul juga dapat berperan sebagai penyebar virus pada tanaman cabai.

2.2.3. Penyakit Kekuningan

Penyakit ini dapat disebabkan oleh beragam faktor, seperti infeksi bakteri, virus maupun kondisi lingkungan khusus. Gejala penyakit ini biasanya mencakup perubahan warna daun menjadi kuning atau coklat, penurunan pertumbuhan tanaman, dan penurunan produksi buah.

2.2.4. Bercak Daun

Penyakit ini dapat disebabkan oleh beberapa jenis jamur dan virus. Gejalanya termasuk munculnya bercak atau bintik pada daun, yang dapat memiliki warna putih, coklat, atau hitam. Penyebaran bercak daun dapat melibatkan seluruh tanaman dan berpotensi mengurangi produktivitas tanaman tersebut.

2.3. Deep Learning

Deep learning adalah subbidang dalam *machine learning* yang memanfaatkan arsitektur jaringan saraf tiruan yang sangat kompleks, yang sering disebut sebagai jaringan saraf dalam (*deep neural networks*). Tujuan dari *deep learning* adalah untuk melatih mesin agar mampu memahami dan memodelkan data yang kompleks dan memiliki hierarki.

Deep Learning memfasilitasi pengembangan model komputasi yang terdiri dari beberapa tahap pemrosesan, memungkinkannya untuk menggali informasi dari data dengan beragam tingkat abstraksi. Melalui penggunaan algoritma *Convolutional Neural Network* (CNN), *Deep Learning* mampu mengidentifikasi struktur yang kompleks dalam himpunan data besar. CNN memungkinkan mesin untuk mengatur parameter internal guna menghasilkan representasi pada tiap lapisan, yang didasarkan pada representasi lapisan sebelumnya. (Lecun, 2015).

2.4. Machine Learning

Bidang *machine learning* bertumpu pada pengembangan algoritma dan model komputer yang memungkinkan sistem belajar dari data serta meningkatkan kinerjanya secara otomatis dalam menangani masalah tertentu. Istilah "*machine learning*" mengacu pada kemampuan mesin atau komputer untuk memproses data dan belajar dari pengalaman

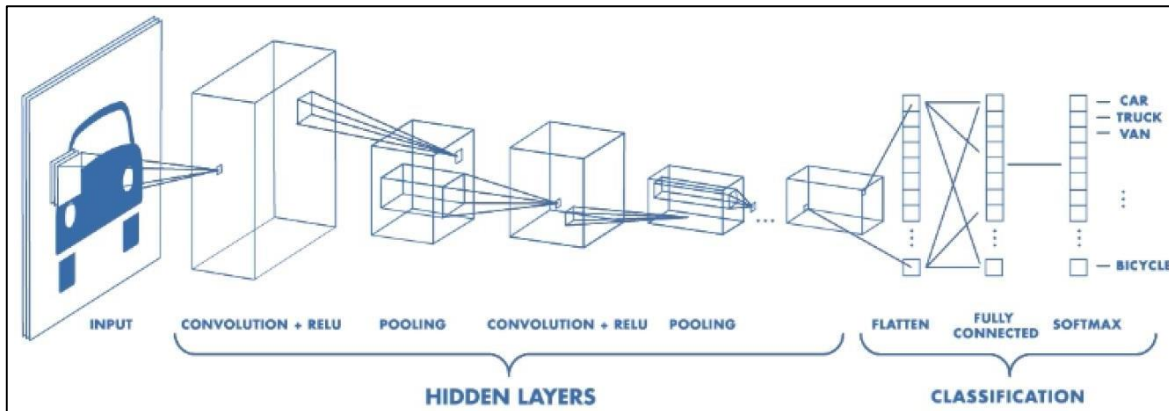
mereka untuk meningkatkan kinerja dalam tugas tertentu. Aplikasi *machine learning* meliputi pengenalan wajah, pengenalan suara, analisis sentimen, kendaraan otonom, peringkat produk, pengelompokan data dan masih banyak lagi. Keberlanjutan perkembangan *machine learning* membuatnya semakin penting dalam berbagai bidang.

2.5. Convolutional Neural Network (CNN)

Convolutional Neural Network dirancang secara spesifik untuk memproses data gambar dan citra. Cara kerja CNN melibatkan pengambilan gambar sebagai input, melakukan operasi konvolusi guna menarik fitur-fitur penting dari gambar, dan selanjutnya melakukan proses klasifikasi dengan menggunakan lapisan-lapisan khusus yang terdapat dalam struktur CNN.

CNN pertama kali diperkenalkan oleh Yann Le Cun pada tahun 1988. CNN merupakan salah satu metode yang mengawali kemunculan dan kesuksesan Deep Learning. Yang membedakan CNN dengan ANN adalah CNN memiliki arsitektur tambahan yang dioptimisasi untuk fitur yang ada pada citra input. Komponen-komponen utama yang ada di dalam CNN antara lain : *Input Layer*, *Convolution Layer*, *Activation Layer*, *Pooling Layer*, *Fully Connected Layer*. (Errissya Rasywir, Rudolf Sinaga & Yovi Pratama, 2020).

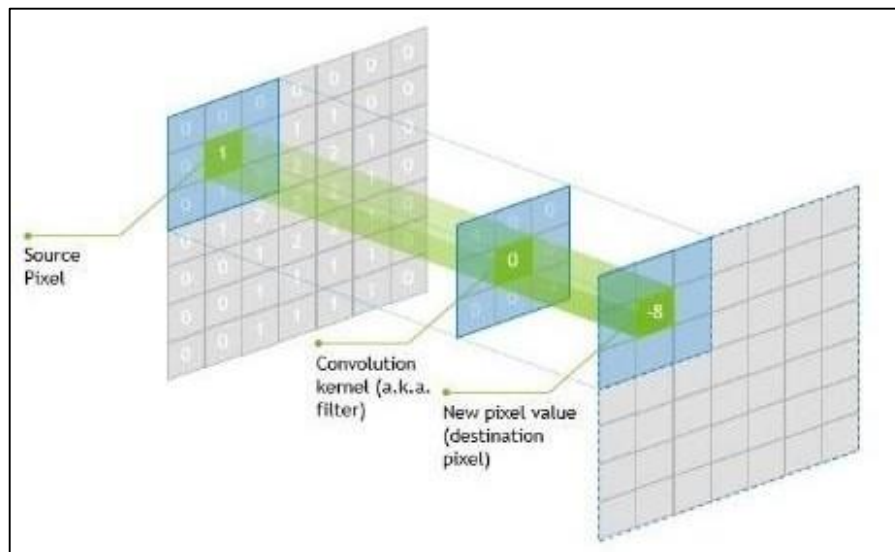
Algoritma CNN memiliki dua tahap pemrosesan utama, yaitu tahap *feature learning* dan tahap *classification*. Pada tahap *feature learning*, sistem secara otomatis mempelajari untuk mengekstrak fitur-fitur penting dari data mentah tanpa campur tangan manusia. Pada tahap ini, gambar yang dimasukkan dan diekstraksi untuk memahami nilai-nilai yang ada di dalamnya. Setelah tahap *feature learning*, selanjutnya akan dilakukan tahap *classification*, model *machine learning* dimanfaatkan untuk memproyeksikan kelas atau label dari data yang belum pernah diproses sebelumnya.



Gambar 2.1 Arsitektur Algoritma CNN (Sumber : www.trivusi.web.id)

2.5.1. Convolution Layer

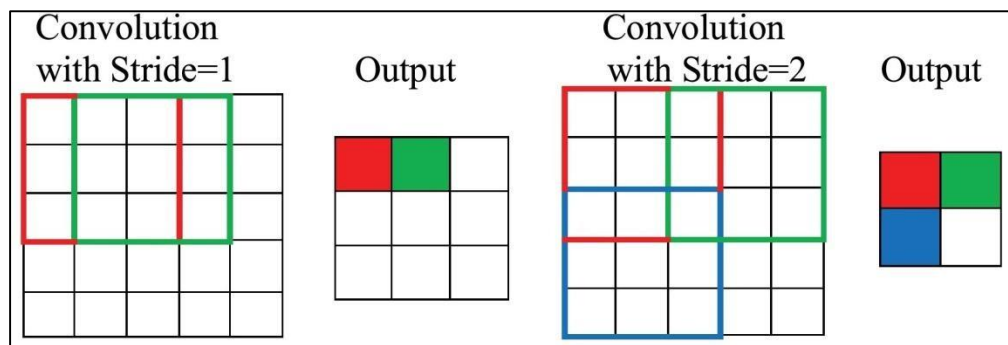
Convolution layer adalah elemen yang sangat penting dalam struktur jaringan saraf tiruan, terutama pada model jaringan saraf konvolusional (CNN). Lapisan ini secara khusus dirancang untuk mengolah data yang memiliki struktur spasial, seperti gambar atau citra. Data input yang memasuki lapisan ini akan melalui proses penyaringan menggunakan filter, yang kemudian akan menghasilkan peta fitur sebagai hasil keluaran (*output*).



Gambar 2.2 Convolution Layer (Sumber : www.cadalyst.com)

2.5.1.1 *Stride*

Stride merupakan parameter dalam operasi konvolusi pada lapisan konvolusi jaringan saraf tiruan. *Stride* berfungsi untuk mengatur sejauh mana filter berpindah melintasi input saat proses konvolusi berlangsung. Jarak pergeseran ini dapat diatur untuk mempengaruhi ukuran peta fitur hasil dan dapat memengaruhi cara jaringan menyusutkan atau mempertahankan informasi spasial.

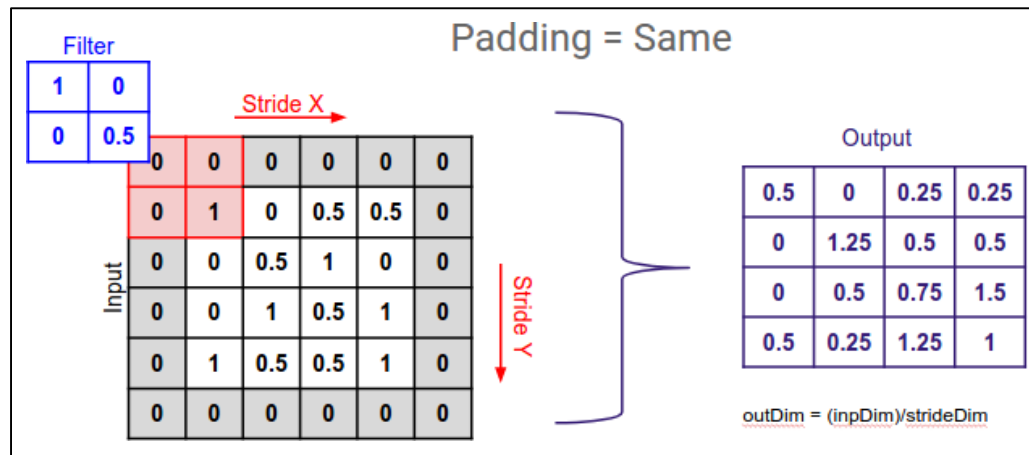


Gambar 2.3 *Stride* (Sumber : Computer.org)

Stride menentukan bagaimana filter melakukan konvolusi pada matriks input, yaitu seberapa jauh pergeseran piksel yang terjadi. Ketika nilai *stride* diatur menjadi 1, filter bergerak satu piksel setiap kali, dan ketika nilai *stride* diatur menjadi 2, filter bergerak dua piksel setiap kali. Penggunaan nilai *stride* yang lebih kecil menghasilkan keluaran yang lebih kecil, dan sebaliknya.

2.5.1.2. *Padding*

Padding adalah metode yang digunakan dalam pemrosesan data dengan menggunakan konvolusi pada jaringan saraf tiruan. Prinsip teknik ini melibatkan penambahan piksel atau nilai-nilai nol di sekitar batas input sebelum melaksanakan operasi konvolusi. Penggunaan *padding* memungkinkan kita untuk memelihara informasi di wilayah batasan input dan menghindari pengurangan ukuran peta fitur.



Gambar 2.4 *Padding* (Sumber : <http://medium.com>)

Penerapan *padding* dapat membantu mengurangi risiko *overfitting* karena piksel-piksel tersebut dijelajahi dengan cara yang seragam. Meskipun demikian, perlu dicatat bahwa penggunaan *padding* juga akan menghasilkan peta fitur yang lebih besar sebagai hasil keluaran.

$$\square \square \square \square \square \square = \frac{\square - \square + 2\square}{\square} + 1$$

Keterangan:

W = Panjang *input*

N = Panjang filter

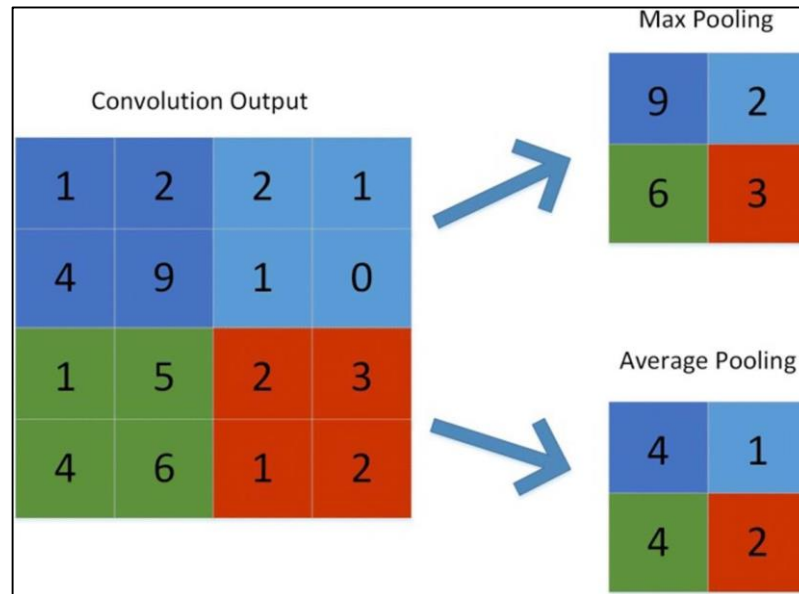
P = *Zero padding*

S = *Stride*

2.5.2. Pooling Layer

Pooling Layer merupakan komponen dari struktur jaringan saraf tiruan, khususnya dalam model arsitektur jaringan saraf konvolusional (CNN). Lapisan ini secara khusus dirancang untuk mengurangi dimensi spasial (ukuran) dari *feature map* yang dihasilkan oleh lapisan sebelumnya. Ini membantu mengurangi kompleksitas serta jumlah parameter, mempercepat proses pelatihan model, dan mencegah terjadinya *overfitting*.

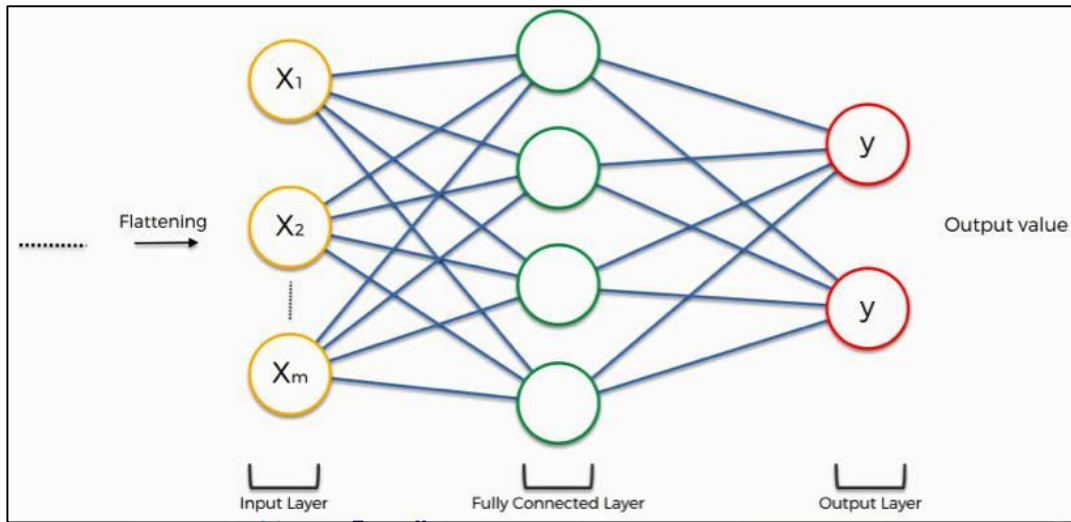
Terdapat dua varian umum dari pooling yang sering digunakan, yaitu *max pooling* (nilai maksimum) dan *average pooling* (nilai rata-rata).



Gambar 2.5 Pooling Layer (Sumber : www.analyticsvidhya.com)

2.5.3. Fully-Connected Layer

Fully-Connected Layer adalah komponen dalam arsitektur jaringan saraf tiruan di mana setiap neuron atau unit dalam lapisan tersebut memiliki koneksi ke semua neuron di lapisan sebelumnya dan setelahnya. Fungsi utama dari *fully-connected layer* adalah mengintegrasikan fitur-fitur yang diekstraksi dari lapisan-lapisan sebelumnya dan menghasilkan output yang sesuai dengan tujuan jaringan, seperti klasifikasi gambar, deteksi objek, atau tugas-tugas lainnya. Jumlah neuron dalam lapisan ini dapat berubah-ubah bergantung pada tingkat kompleksitas dari tugas yang diinginkan.



Gambar 2.6 *Fully-Connected Layer* (Sumber : <https://warstek.com>)

$$z_j = \sum_{i=0}^n w_{ij} x_i + b_j$$

Keterangan:

W = Weight

X = Input dari neuron sebelumnya

B = Bias

Terdapat beberapa jenis fungsi aktivasi yang sering diterapkan dalam *fully-connected layer*, seperti *sigmoid*, *softmax*, dan ReLu (*Rectified Linear Unit*).

2.5.4. *Softmax*

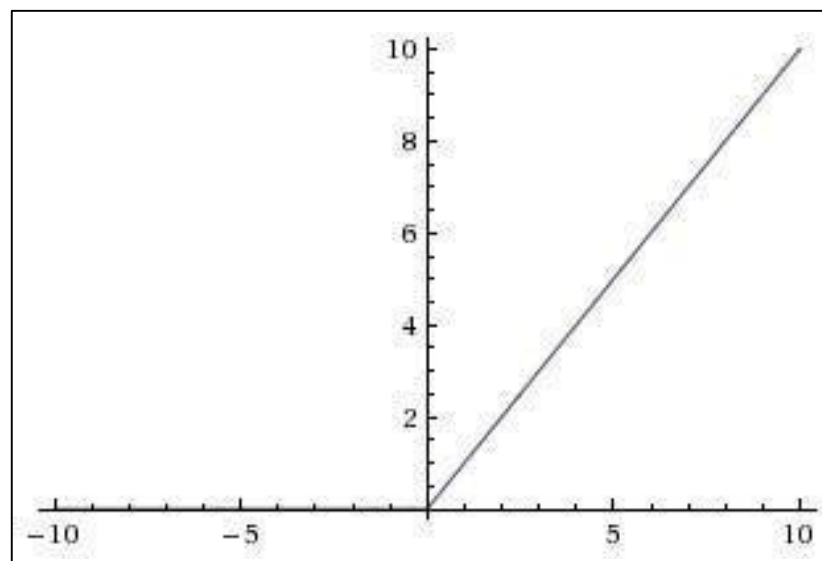
Softmax merupakan jenis fungsi aktivasi yang diterapkan dalam jaringan saraf tiruan untuk menghasilkan distribusi probabilitas yang telah dinormalisasi dari sejumlah nilai. Penggunaan umum dari fungsi *softmax* terjadi dalam situasi klasifikasi multikelas, di mana model harus menyajikan probabilitas untuk setiap kelas yang ada.

Fungsi *softmax* melakukan transformasi vektor nilai real menjadi vektor probabilitas, sehingga total probabilitas dari semua kelas menjadi 1. Proses ini

melibatkan penghitungan setiap elemen dalam vektor output softmax dengan menerapkan fungsi eksponensial (e^x) pada nilai aslinya, dan kemudian hasilnya dinormalisasi dengan membaginya oleh jumlah total nilai eksponensial.

2.5.5. ReLu (*Rectified Linear Unit*)

ReLU adalah jenis fungsi aktivasi yang sering diterapkan dalam jaringan saraf tiruan. Fungsi ini memberikan tanggapan linear positif terhadap input positif, sementara menghasilkan nilai nol untuk input yang bersifat negatif. Keuntungan utama dari ReLU adalah pengenalan non-linearitas ke dalam model, hal ini sangat krusial untuk memungkinkan jaringan saraf belajar merepresentasikan data dengan tingkat kompleksitas yang lebih tinggi.



Gambar 2.7 ReLu Function (Sumber : <https://medium.com>)

Rumus umum dari ReLu untuk suatu input x adalah:

$$\text{ReLu}(x) = \max(0, x)$$

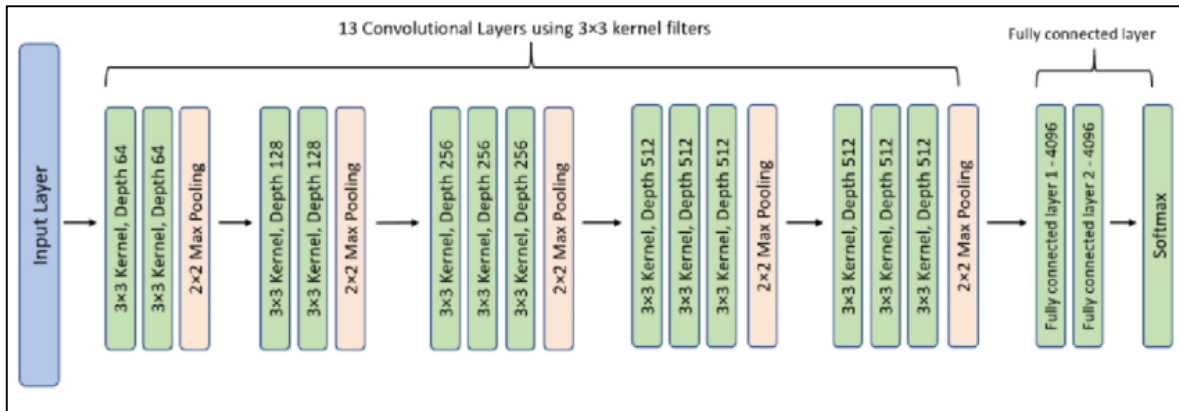
Artinya, jika nilai x positif, ReLu mengembalikan nilai x , sedangkan jika x negatif atau nol, ReLu mengembalikan nilai nol.

2.6. Arsitektur VGG16

Arsitektur VGG16 terkenal karena kedalamannya yang luar biasa dan kegunaannya dalam tugas-tugas pengenalan gambar. Di bawah ini adalah beberapa poin kunci terkait dengan VGG16:

1. Kedalaman:
VGG16 memiliki total 16 lapisan, yang terdiri dari 13 lapisan konvolusional dan 3 lapisan sepenuhnya terhubung (*fully connected*).
2. Kernel Konvolusi:
Kernel konvolusi pada semua lapisan konvolusional memiliki ukuran 3x3.
3. Pooling:
Setelah setiap blok konvolusional, terdapat lapisan max pooling dengan kernel 2x2 dan stride sebanyak 2 untuk mengurangi dimensi data.
4. Fully Connected Layers:
Tiga lapisan sepenuhnya terhubung dengan aktivasi ReLu mengikuti blok konvolusional.
5. Aktivasi ReLu:
Fungsi aktivasi ReLu digunakan di seluruh jaringan kecuali pada lapisan *output*.
6. Output Layer:
Lapisan output terdiri dari 1000 neuron yang menggunakan aktivasi softmax, sesuai dengan jumlah kelas dalam dataset, dan lapisan ini awalnya digunakan untuk melatih model.

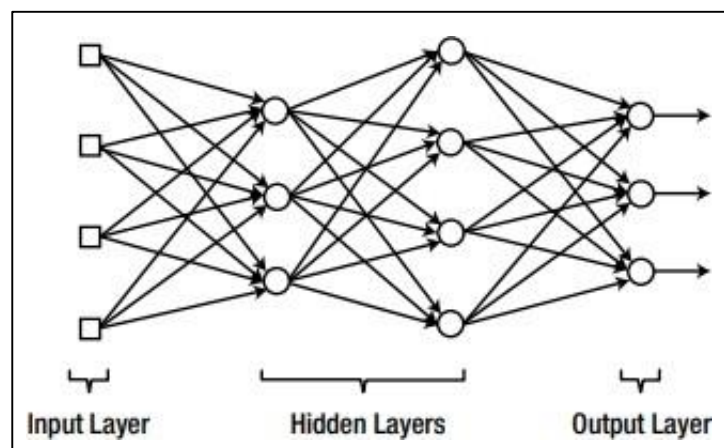
VGG16 sering dijadikan fondasi atau acuan dalam pengembangan arsitektur jaringan saraf konvolusional lainnya. Meskipun arsitektur ini memiliki jumlah parameter yang besar dan memerlukan sumber daya komputasi yang signifikan, prestasinya dalam tugas pengenalan gambar telah menjadikannya salah satu model yang diminati di kalangan peneliti dan praktisi industri.



Gambar 2.8 Arsitektur VGG16 (Sumber : www.researchgate.net)

2.7. Neural Network

Neural Network bertujuan untuk memproses informasi dan melakukan pembelajaran mesin untuk menemukan pola dalam data. *Neural network* terdiri dari unit pemrosesan informasi yang disebut neuron yang diatur dalam lapisan-lapisan. Pada awalnya, struktur model jaringan saraf cukup sederhana, hanya melibatkan input layer dan output layer yang disebut sebagai jaringan saraf satu lapisan. Seiring waktu, perkembangan ini berkembang menjadi jaringan saraf multi-layer dengan penambahan lapisan tersembunyi yang ditempatkan di antara lapisan *input* dan *output*.



Gambar 2.9 Arsitektur *Neural Network* (Kim, P. 2017)

BAB 3

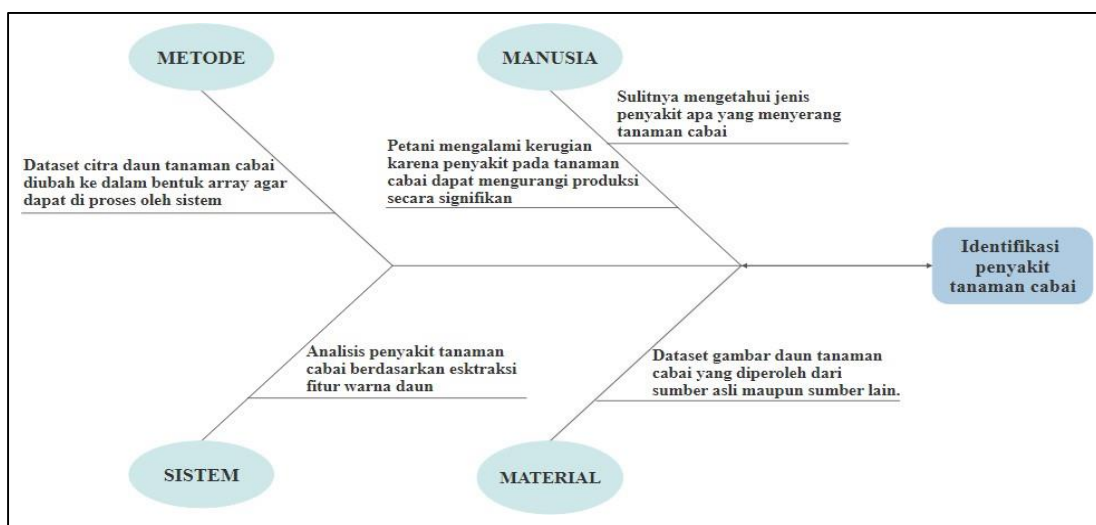
ARSITEKTUR DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem adalah langkah pertama yang dilakukan guna memahami dan mengevaluasi sistem yang sudah ada atau yang akan dibangun. Proses ini membentuk dasar untuk pengembangan, perbaikan, atau peningkatan sistem dengan tujuan memastikan bahwa hasilnya akan efektif dan efisien dalam memenuhi kebutuhan serta tujuan yang telah ditetapkan. Analisis sistem melibatkan serangkaian langkah penting yang mencakup pemahaman mendalam, perencanaan, dan pengambilan keputusan untuk menentukan bagaimana sistem tersebut akan beroperasi.

3.1.1. Analisis Masalah

Analisis masalah merupakan proses identifikasi, pemahaman, serta pemecahan masalah. Proses ini menjadi langkah kritis dalam menyelesaikan masalah dan pengambilan keputusan dalam berbagai konteks. Dalam konteks ini, digunakan Diagram Ishikawa (Fishbone Diagram) sebagai alat untuk menguraikan akar permasalahan ke dalam empat bagian yang diilustrasikan sebagai berikut:



Gambar 3.1 *Fishbone Diagram*

3.1.2. Analisis Kebutuhan

Langkah ini digunakan untuk mengidentifikasi kebutuhan yang diperlukan dan diinginkan dalam perancangan dan pembangunan suatu sistem. Fokus utama pada tahap ini adalah untuk mengenali, mendokumentasikan, dan menguraikan persyaratan yang harus dipenuhi oleh sistem yang sedang dikembangkan. Secara umum, analisis kebutuhan dibagi menjadi dua, yaitu kebutuhan fungsional yang berkaitan dengan fungsi-fungsi spesifik sistem, dan kebutuhan non-fungsional yang mencakup aspek-aspek seperti performa, keamanan, dan keandalan sistem.

1. Kebutuhan Fungsional

Aspek utama yang perlu dipenuhi dan diterapkan oleh sistem yang sedang dalam proses pengembangan.. Dalam penelitian ini, beberapa kebutuhan fungsional yang harus diperhatikan, seperti berikut:

- a. Sistem harus dapat menerima unggahan *user* yang berupa gambar atau foto.
- b. Input atau tanggapan *user* yang berkaitan dengan data daun tanaman cabai yang akan diperiksa harus dapat diterima oleh sistem.
- c. Sistem harus dapat menunjukkan apakah tanaman cabai terkena penyakit dan mengidentifikasi jenis penyakit berdasarkan foto daun yang diunggah oleh *user*.

2. Kebutuhan non-fungsional

Kebutuhan non-fungsional merujuk pada aspek-aspek yang secara langsung tidak terkait dengan fungsi-fungsi spesifik sistem, melainkan berperan sebagai pendukung tambahan untuk memastikan kinerja yang optimal. Dalam penelitian ini, yang harus diperhatikan sebagai berikut:

- a. Sistem yang akan dirancang menggunakan basis *website* dan memiliki tampilan yang sederhana agar mudah digunakan.
- b. Sistem dibangun dengan menggunakan metode CNN serta model VGG16 dan hasil identifikasinya didasarkan pada citra daun cabai dengan fitur unggahan gambar ke sistem.

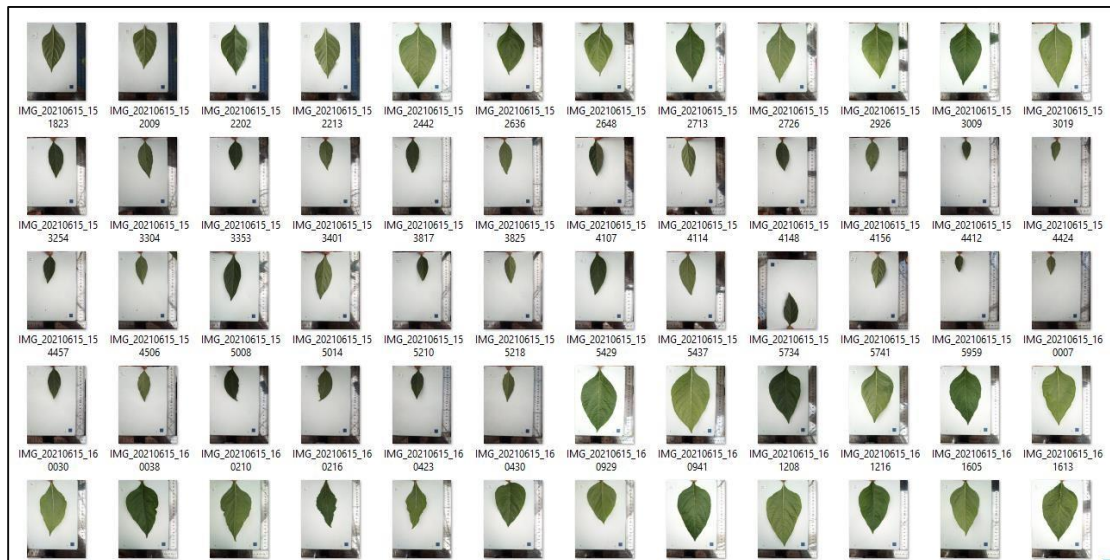
- c. Sistem yang akan dibangun tidak membutuhkan koneksi internet saat digunakan.

3.1.3. Analisis Proses

Analisis proses melibatkan evaluasi dan pemahaman yang mendalam terhadap serangkaian langkah yang dilalui dalam penelitian.

1. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data berupa gambar atau foto daun tanaman cabai, seperti daun cabai sehat, daun keriting, penyakit kutu kebul, penyakit kekuningan dan bercak daun. Berikut ini adalah gambar atau foto daun cabai sesuai dengan klasifikasi yang telah ditetapkan:



Gambar 3.2 Daun Cabai Sehat

Gambar 3.2 di atas merupakan gambar dari daun cabai yang dalam kondisi sehat. Citra daun tersebut akan digunakan sebagai bahan pelatihan dalam proses pembelajaran model *Convolutional Neural Network* (CNN).



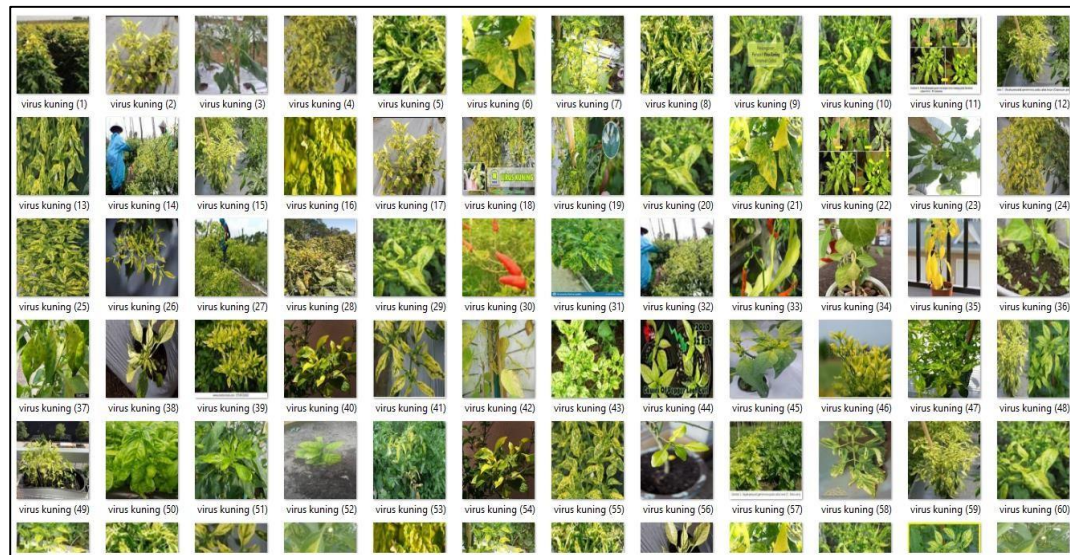
Gambar 3.3 Daun Keriting

Gambar 3.3 di atas adalah gambar yang menunjukkan daun cabai yang mengalami penyakit daun keriting. Citra daun tersebut akan dijadikan sebagai bahan pelatihan dalam proses pembelajaran model Convolutional Neural Network (CNN).



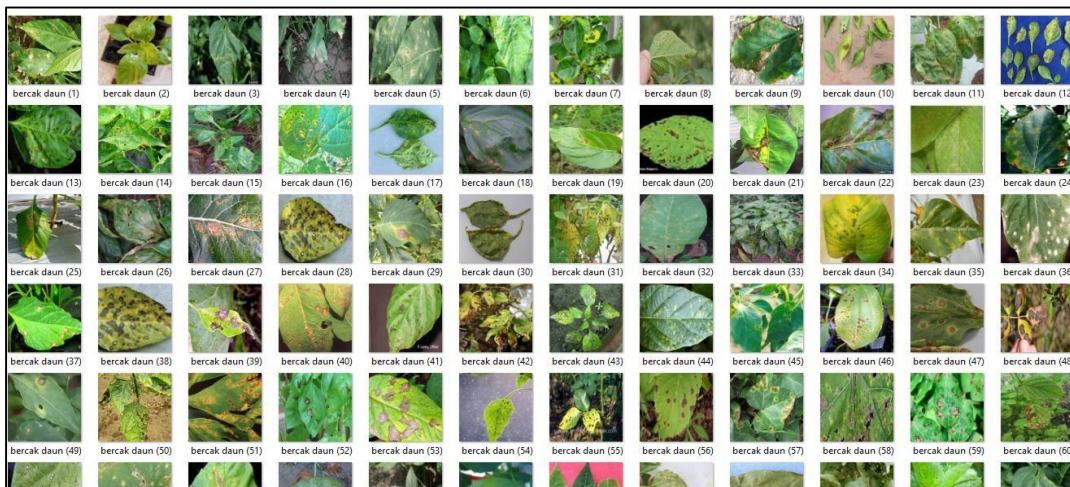
Gambar 3.4 Kutu Kebul

Gambar 3.4 di atas adalah ilustrasi dari daun cabai yang terinfeksi penyakit kutu kebul. Citra daun tersebut akan dijadikan sebagai bahan pelatihan dalam proses pembelajaran model Convolutional Neural Network (CNN).



Gambar 3.5 Penyakit Kekuningan

Gambar 3.5 di atas adalah ilustrasi dari daun cabai yang terinfeksi penyakit kekuningan. Citra daun tersebut akan dijadikan sebagai bahan pelatihan dalam proses pembelajaran model Convolutional Neural Network (CNN).



Gambar 3.6 Bercak Daun

Gambar 3.6 di atas merupakan daun cabai yang terinfeksi penyakit bercak daun. Citra daun tersebut akan dijadikan sebagai bahan pelatihan dalam proses pembelajaran model Convolutional Neural Network (CNN).

2. Menentukan Dataset

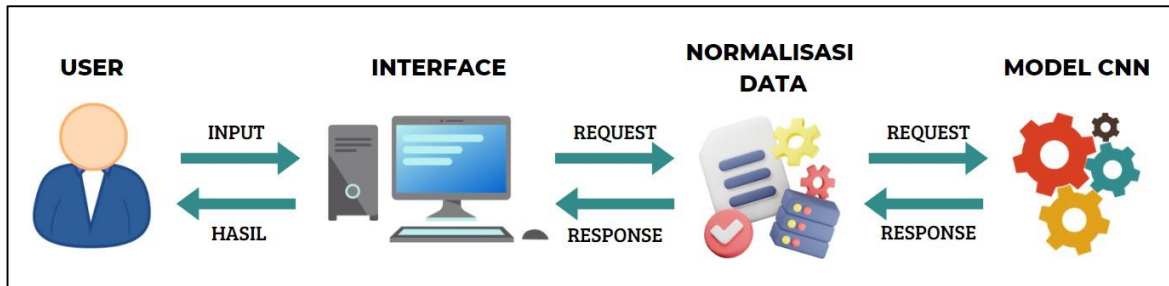
Berdasarkan data yang dikumpulkan, terdapat 200 citra untuk setiap klasifikasi, sehingga total keseluruhan mencapai 1000 citra. Kemudian, setiap kelompok citra dari berbagai klasifikasi dipisahkan menjadi dua bagian, yaitu data pelatihan (*training*) dan data pengujian (*testing*). Detail pembagian dataset ini dapat ditemukan dalam tabel berikut:

Tabel 3.1 Tabel Pembagian Dataset

Klasifikasi Daun	Total Dataset	Dataset Latih	Dataset Uji
Daun Sehat	200	190	10
Daun Keriting	200	190	10
Kutu Kebul	200	190	10
Penyakit Kekuningan	200	190	10
Bercak Daun	200	190	10

3.1.4. Arsitektur Umum

Arsitektur umum merujuk pada gambaran mengenai struktur dan jalannya suatu sistem, mencakup konsep-konsep yang digunakan untuk merancang serta mengevaluasi sistem tertentu.



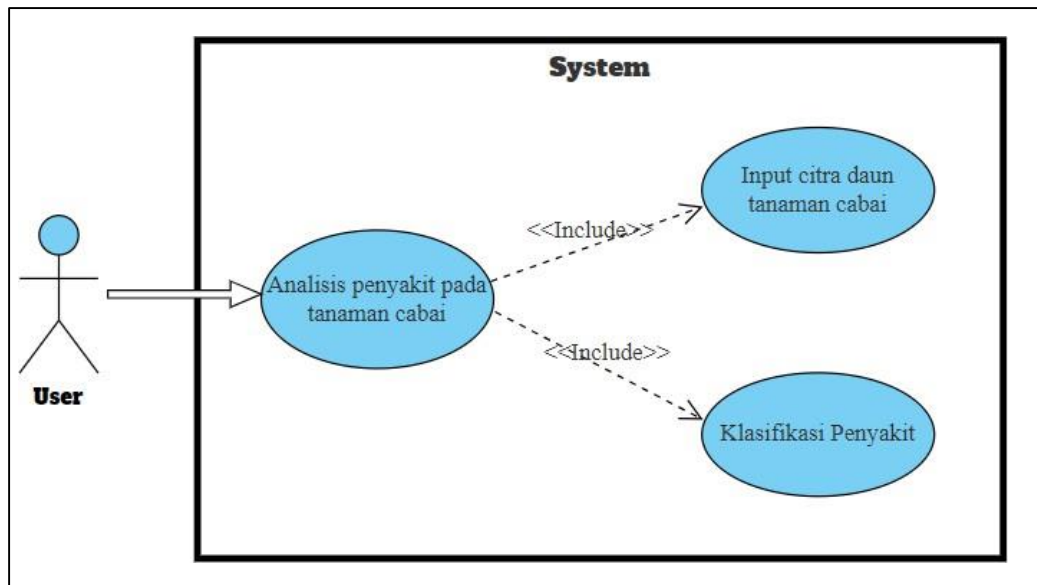
Gambar 3.7 Ilustrasi Arsitektur Umum Sistem

- Pada penelitian ini proses awal yaitu pengguna mengunggah gambar daun tanaman cabai.
- Modul pengelola *file* dalam sistem menerima gambar yang diunggah.
- Kemudian sistem akan melakukan normalisasi data pada gambar daun tanaman cabai yang diunggah.
- Setelah data gambar mengalami proses normalisasi, langkah selanjutnya data tersebut akan diteruskan ke model CNN dan VGG16.
- Proses *training* akan dilakukan oleh model CNN dan VGG16 yang akan menghasilkan *file* Belajar40_N.h5. Kemudian *file* tersebut akan digunakan dalam analisis dan klasifikasi penyakit pada tanaman cabai.
- Modul klasifikasi akan menghasilkan penyakit apa yang menyerang tanaman cabai berdasarkan analisis fitur dari gambar.
- Data hasil analisis dan klasifikasi tersebut akan disimpan, dimana hasilnya akan ditampilkan pada aplikasi klasifikasi penyakit pada tanaman cabai kepada pengguna.

3.2. Pemodelan Sistem

3.2.1. Use Case Diagram

Diagram ini digunakan untuk mendiskripsikan fungsionalitas sistem dari perspektif pengguna .

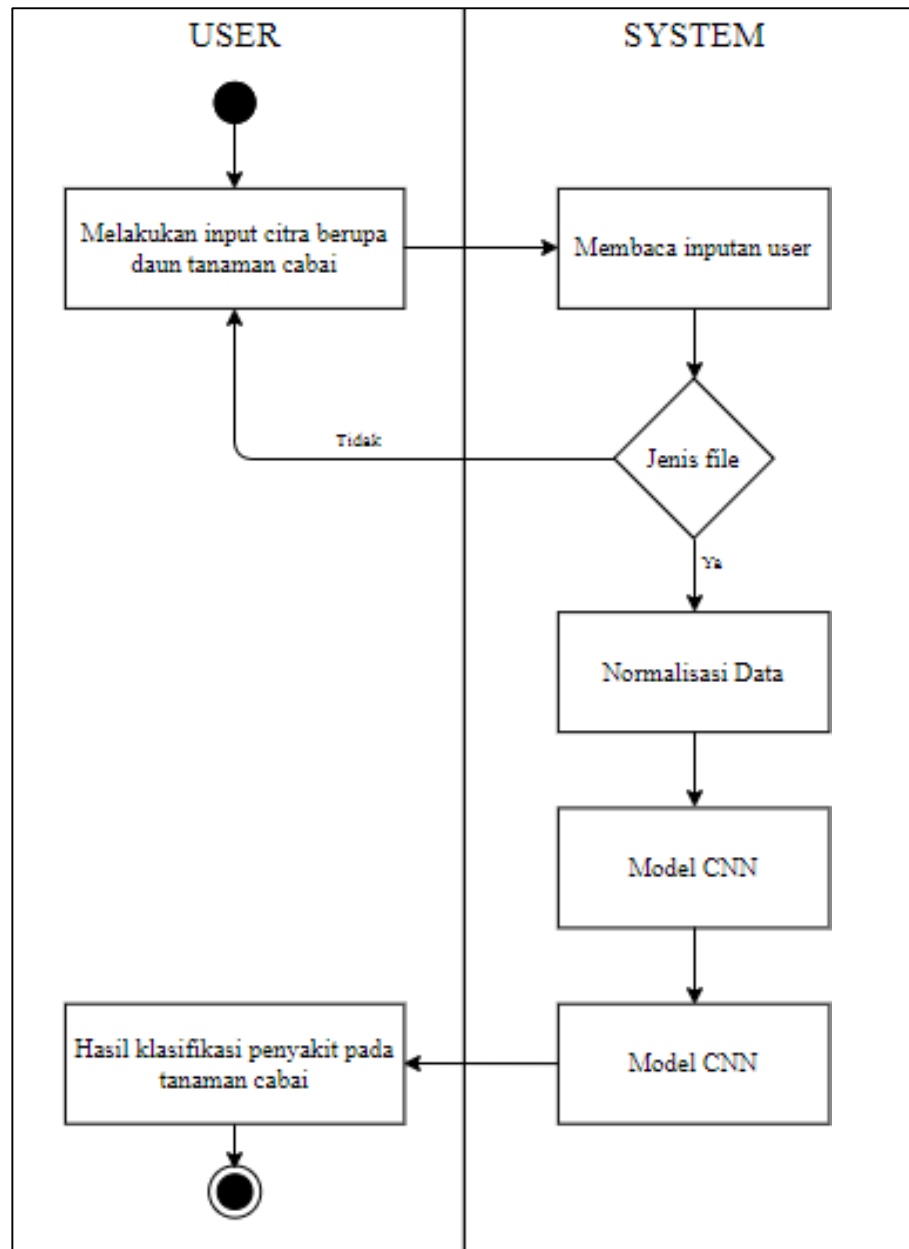


Gambar 3.8 *Use Case Diagram*

Dalam kerangka penelitian ini, operasional sistem terjadi saat pengguna mengunggah gambar daun tanaman cabai. Setelahnya, sistem akan melakukan klasifikasi penyakit pada tanaman cabai berdasarkan citra yang diunggah oleh pengguna.

3.2.2. Activity Diagram

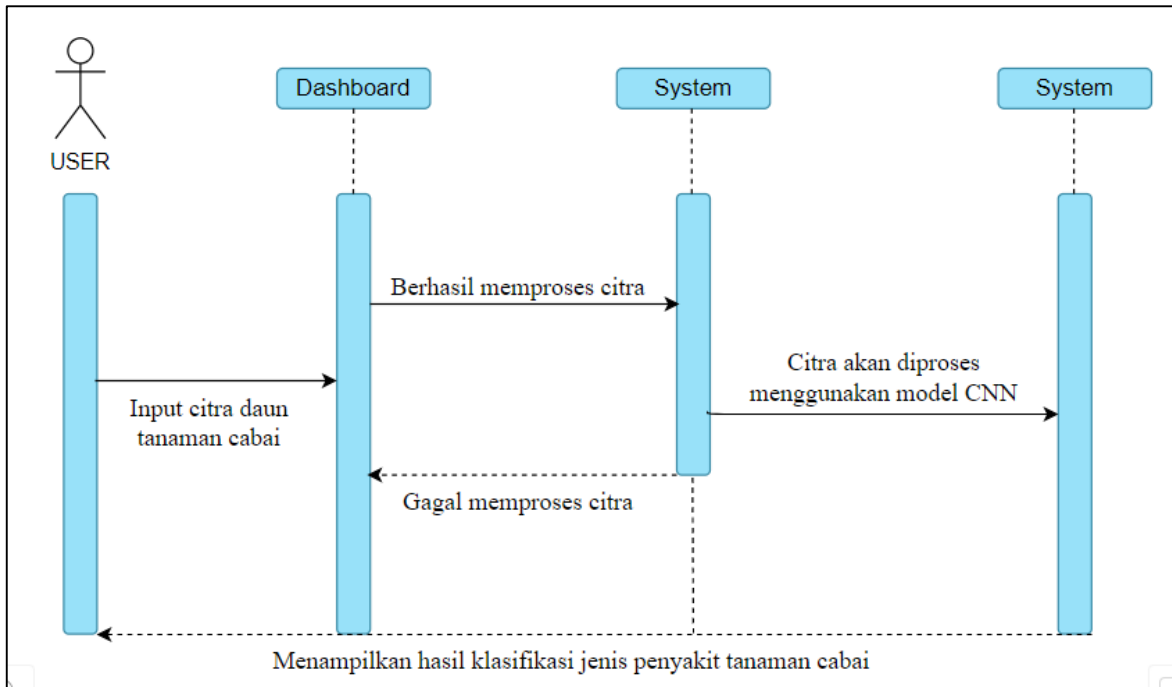
Diagram ini bertujuan guna membantu dalam merepresentasikan proses sistem atau alur kerja dari awal hingga akhir. Ilustrasi pada **Gambar 3.9** di bawah ini akan menggambarkan aktivitas dan alur kerja sistem dalam konteks penelitian ini.



Gambar 3.9 Activity Diagram

3.2.3. Sequence Diagram

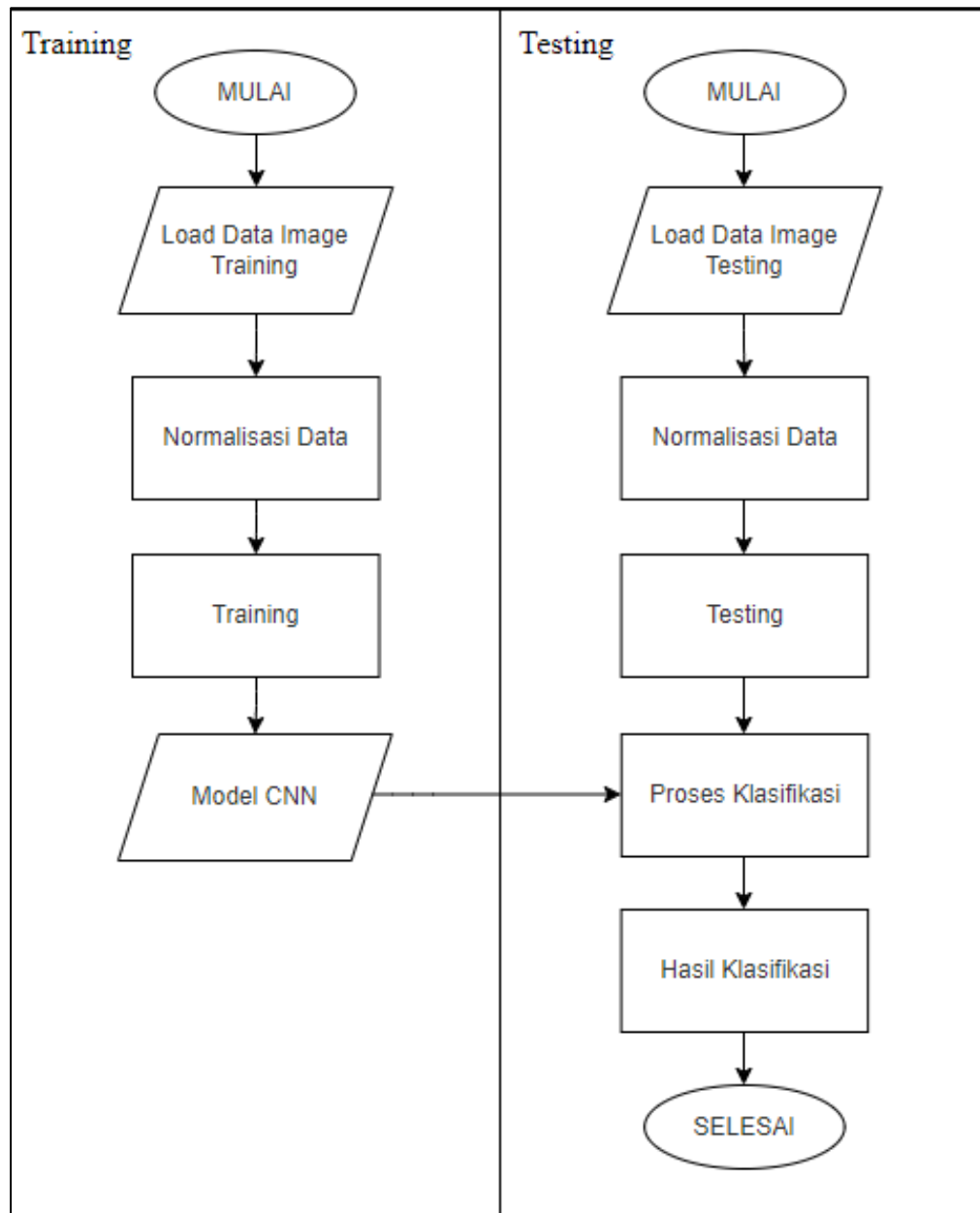
Diagram ini bertujuan untuk memvisualisasikan interaksi antara objek-objek dalam suatu sistem atau proses. Dibawah ini adalah gambar *sequence* diagram pada penelitian ini.



Gambar 3.10 *Sequence Diagram*

3.3. *Flowchart*

Flowchart adalah gambaran grafis dari alur suatu proses atau sistem. Flowchart membantu dalam memahami alur kerja suatu proses atau sistem secara visual dan mudah dipahami. Alur kerja atau proses dalam kerangka penelitian ini dijelaskan melalui gambar di bawah ini:



Gambar 3.11 *Flowchart* Sistem

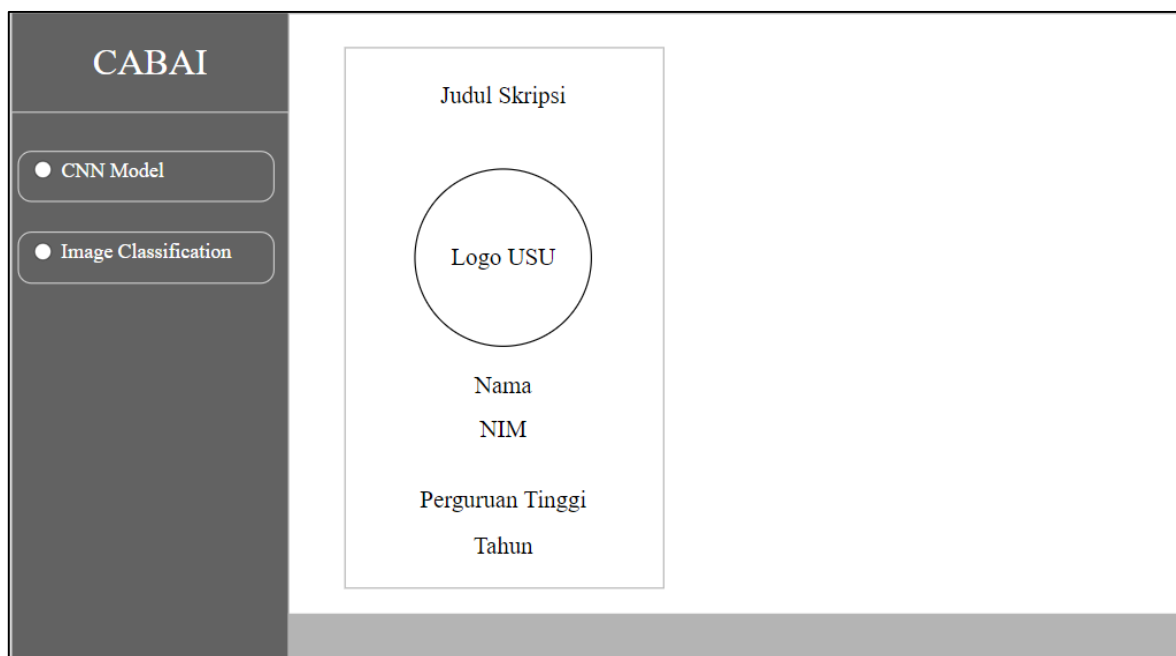
Seperti yang terlihat pada **Gambar 3.11**, langkah pertama yang dilakukan yaitu pemuatan dataset citra daun tanaman cabai untuk tahap pelatihan, diikuti oleh normalisasi data yang bertujuan untuk mempersiapkan data dan menyesuaikan ukuran gambar sebelum memasuki tahap pelatihan model. Setelah itu, data yang telah melalui langkah tersebut akan

digunakan dalam pelatihan model CNN, dengan menggunakan model VGG16. Setelah tahap pelatihan selesai, model CNN (Belajar40_N.h5) akan dihasilkan. Model tersebut kemudian digunakan dalam tahap pengujian di mana citra yang dimasukkan akan mengalami proses klasifikasi menggunakan model CNN. Setelah tahap ini selesai, hasil klasifikasi penyakit pada tanaman cabai akan diperoleh.

3.4. Perancangan Antarmuka Sistem

3.4.1. Halaman Awal

Halaman ini merupakan tampilan awal yang muncul ketika pengguna pertama kali mengakses aplikasi atau sistem.



The image shows a web interface design for a system named 'CABAI'. On the left is a dark grey sidebar with the title 'CABAI' at the top. Below the title are two buttons: 'CNN Model' and 'Image Classification', each with a small white circle icon. The main content area is white and contains a form with the following elements: a text input field labeled 'Judul Skripsi', a circular logo placeholder labeled 'Logo USU', and four text input fields labeled 'Nama', 'NIM', 'Perguruan Tinggi', and 'Tahun'.

Gambar 3.12 Rancangan Halaman Awal

3.4.2. Halaman Pengujian

Halaman pengujian merupakan tempat di mana metode CNN diterapkan untuk mengklasifikasi penyakit pada tanaman cabai. Proses pengujian ini bertujuan untuk

menentukan apakah daun tanaman cabai tersebut dalam keadaan sehat atau mengalami infeksi penyakit.

CABAI

CNN Model

Image Classification

Input Image

File input

Choose file

No file chosen

Submit

Image Classification Result

Uploaded Image

Prediction Result image 1:
Predicted Class Image 1: {{ predicted_class_vgg }}
Confidence Image 1: {{ confidence_vgg }}

Gambar 3.13 Rancangan Halaman Pengujian

BAB 4

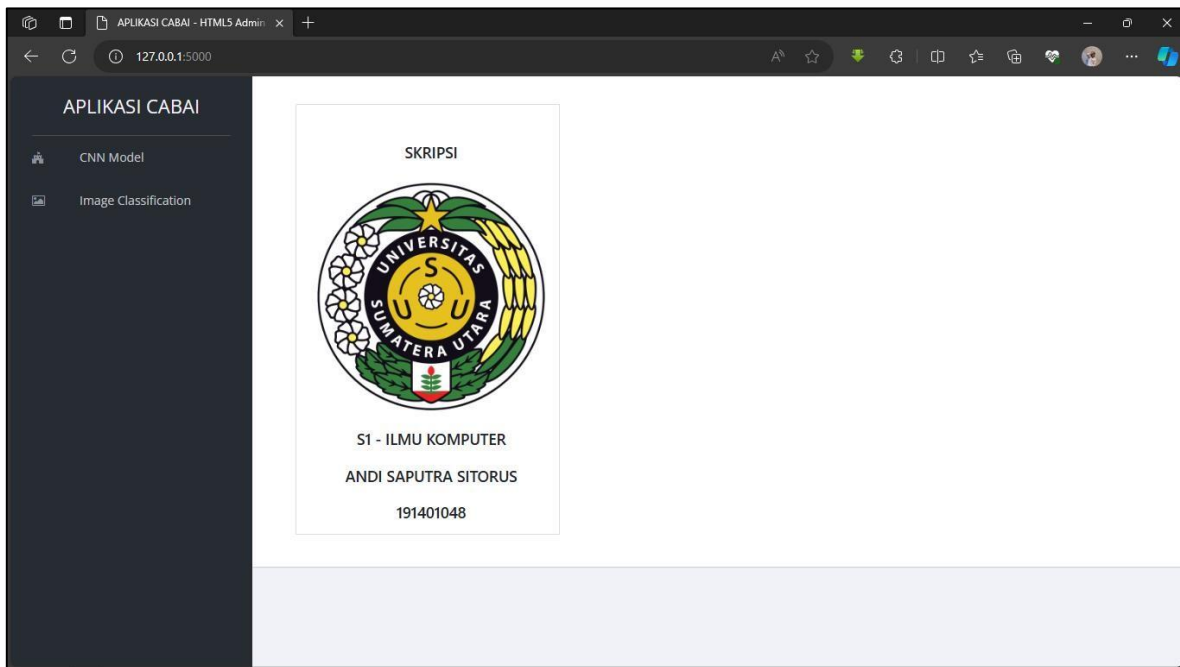
IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

Penelitian ini memanfaatkan bahasa pemrograman *Python* untuk membangun sistem aplikasi berbasis web. Aplikasi ini terdiri dari halaman utama dan halaman pengujian.

4.1.1. Halaman Utama

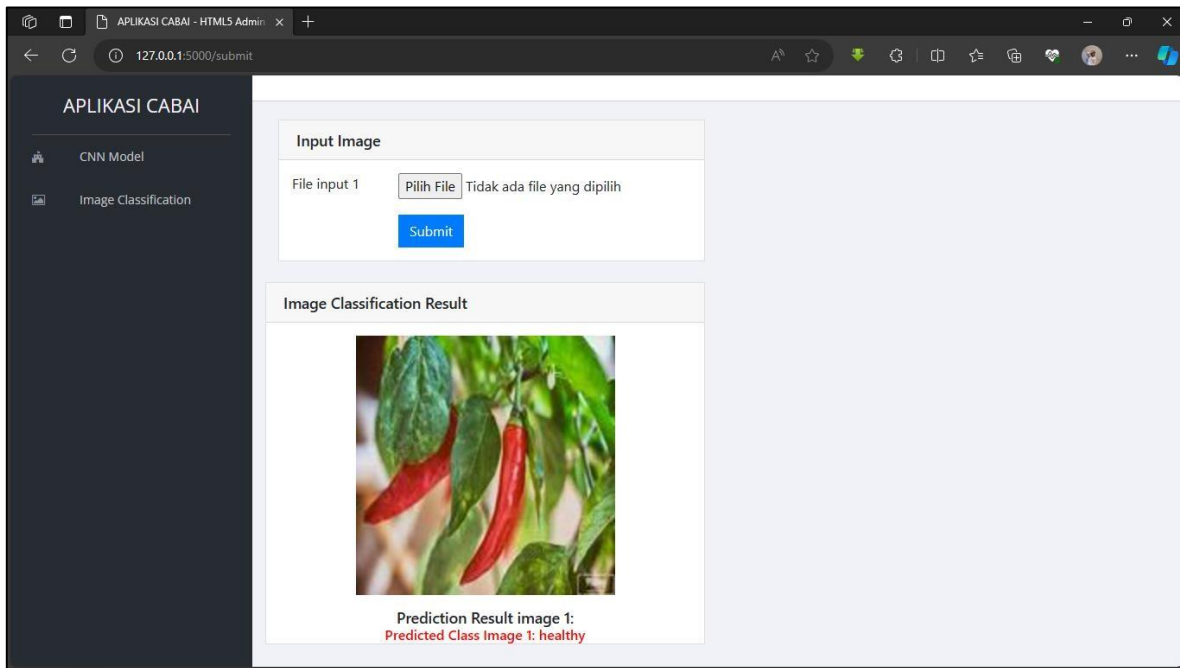
Gambar di bawah ini menggambarkan antarmuka halaman utama pada aplikasi. Ini adalah tampilan awal yang muncul ketika pengguna pertama kali mengakses aplikasi atau sistem.



Gambar 4.1 Halaman Utama Sistem

4.1.2. Halaman Pengujian

Gambar dibawah ini merupakan tampilan halaman pengujian pada aplikasi.



Gambar 4.2 Halaman Pengujian Sistem

Gambar 4.2 diatas menampilkan halaman selama proses pengujian untuk mengklasifikasi penyakit pada tanaman cabai, dimulai dari menginput citra hingga mendapatkan hasil klasifikasi. Proses ini melibatkan beberapa tahapan dalam penggunaan sistem ini:

1. Langkah awalnya, pengguna akan mengunggah gambar daun tanaman cabai pada *File Input 1*.
2. Setelah itu, pengguna akan menekan tombol *submit* untuk memproses gambar yang telah diunggah.
3. Hasil dari proses tersebut akan ditampilkan pada *Image Classification Result*. Disini akan diperlihatkan apakah daun tanaman cabai dalam kondisi sehat atau terinfeksi penyakit.

4.2. Perancangan Model

4.2.1. Perancangan Model CNN

Model: "sequential_2"		
Layer (type)	Output Shape	Param #
=====		
conv2d_16 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_8 (MaxPooling 2D)	(None, 63, 63, 32)	0
conv2d_17 (Conv2D)	(None, 61, 61, 64)	18496
max_pooling2d_9 (MaxPooling 2D)	(None, 30, 30, 64)	0
conv2d_18 (Conv2D)	(None, 28, 28, 128)	73856
max_pooling2d_10 (MaxPoolin g2D)	(None, 14, 14, 128)	0
flatten_2 (Flatten)	(None, 25088)	0
dense_5 (Dense)	(None, 128)	3211392
dense_6 (Dense)	(None, 5)	645
=====		
Total params: 3,305,285		
Trainable params: 3,305,285		
Non-trainable params: 0		

Gambar 4.3 *Summary Model CNN*

```

# Membuat objek model Sequential
model = models.Sequential()

# Layer Convolutional pertama
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)))
model.add(layers.MaxPooling2D((2, 2)))

# Layer Convolutional kedua
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

# Layer Convolutional ketiga
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

# Layer Flatten untuk mengubah output menjadi vektor 1D
model.add(layers.Flatten())

# Layer Dense (fully connected)
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(5, activation='softmax')) # Misalnya, 10 kelas output

# Compile model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Tampilkan ringkasan model
model.summary()

```

Gambar 4.4 Implementasi Model CNN

Pada penelitian ini, desain model CNN yang diperlihatkan dalam **Gambar 4.3** dan **Gambar 4.4** dirancang menggunakan 3 lapisan konvolusi, 3 lapisan *max pooling* (lapisan nilai maksimum), 1 lapisan perataan (*flatten layer*), dan 2 lapisan berdimensi penuh (*dense layer*).

Lapisan konvolusi pertama berperan sebagai lapisan masukan, memiliki dimensi 128x128 piksel dan menggunakan 3 *channel* warna (RGB). Tiap lapisan konvolusi diikuti oleh lapisan *max pooling* yang menggunakan *matrix* berukuran 2x2 dan *stride* sebesar 2, bertujuan untuk mengurangi dimensi dari *feature map* sebelumnya.

Flatten layer dimanfaatkan untuk mengonversi hasil *output* dari lapisan konvolusi menjadi vektor satu dimensi, hal ini disebabkan oleh kemampuan *dense layer* yang hanya dapat menerima *input* dalam bentuk vektor satu dimensi.

Jumlah neuron dalam *dense layer* adalah 128 dan 5, ditentukan berdasarkan hasil eksperimen dan penyesuaian. Di samping itu, 5 neuron di lapisan terakhir berfungsi sebagai *output* untuk klasifikasi, dan penggunaan fungsi *softmax* disesuaikan dengan jumlah kategori kelas pada dataset yang terdiri dari 5 kelas.

Pada proses pelatihan model, digunakan fungsi *loss categorical_crossentropy* dikarenakan dataset terdiri dari 5 kelas yang berbeda. *Optimizer* yang diterapkan adalah Adam yang terbukti efektif dalam menyesuaikan tingkat pembelajaran selama proses pelatihan. Evaluasi performa model akan dilakukan berdasarkan akurasi untuk menilai sejauh mana kemampuan model dalam mengklasifikasikan data dengan tepat dan benar.

4.2.2. Perancangan Model VGG16

Model: "sequential_3"		
Layer (type)	Output Shape	Param #
conv2d_19 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_20 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_11 (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_21 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_22 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_12 (MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_23 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_24 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_25 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_13 (MaxPooling2D)	(None, 28, 28, 256)	0
...		
Total params: 15,518,757		
Trainable params: 15,518,757		
Non-trainable params: 0		

Gambar 4.5 Summary Model VGG16


```

def build_vgg16(input_shape=(128, 128, 3), num_classes=5):
    model = Sequential()

    # Block 1
    model.add(Conv2D(64, (3, 3), activation='relu', padding='same', input_shape=input_shape))
    model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 2
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 3
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 4
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 5
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Fully connected layers
    model.add(Flatten())
    model.add(Dense(32, activation='relu'))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))

    return model

# Test the model
input_shape = (224, 224, 3)
num_classes = 5
vgg16_model = build_vgg16(input_shape, num_classes)
vgg16_model.summary()

```

Gambar 4.6 Implementasi Model VGG16

Perancangan model VGG16 dapat dilihat pada **Gambar 4.5** dan **Gambar 4.6**.

1. Blok 1
 - a. *Convolutional Layer* (Conv2D): Lapisan awal merupakan Convolutional Layer yang terdiri dari 64 filter. Setiap filter menggunakan kernel berukuran 3x3,

fungsi aktivasi ReLu, dan diterapkan *padding* "same" adalah untuk mempertahankan dimensi gambar input. Kemudian diikuti oleh Conv2D layer kedua yang memiliki karakteristik serupa.

- b. *Max Pooling Layer* (MaxPooling2D): Setelah dua lapisan Conv2D, berikutnya menggunakan MaxPooling2D dengan ukuran (2x2) dan *strides* (2x2) untuk mengurangi dimensi gambar.
 - c. Total jumlah *layer* pada blok 1:4.
2. Blok 2
 - a. *Convolutional Layer* (Conv2D): Dalam Blok kedua, terdapat dua lapisan Conv2D yang menggunakan 128 filter, aktivasi ReLu dan *padding* "same". Menggunakan lapisan MaxPooling2D dengan ukuran (2x2) dan pergeseran (*strides*) (2x2).
 - b. Total jumlah *layer* pada blok 2:3.
3. Blok 3
 - a. *Convolutional Layer* (Conv2D): Dalam blok ketiga ini terdapat tiga lapisan Conv2D yang terdiri dari 256 filter, aktivasi ReLu dan *padding* "same".
 - b. Menggunakan lapisan MaxPooling2D dengan ukuran (2x2) dan pergeseran (*strides*) (2x2).
 - c. Total jumlah *layer* pada blok 3:4.
4. Blok 4
 - a. *Convolutional Layer* (Conv2D): Dalam blok ketiga ini terdapat tiga lapisan Conv2D yang terdiri dari 512 filter, aktivasi ReLu dan *padding* "same".
 - b. Menggunakan lapisan MaxPooling2D dengan ukuran (2x2) dan pergeseran (*strides*) (2x2).
 - c. Total jumlah *layer* pada blok 4:4.
5. Blok 5
 - a. *Convolutional Layer* (Conv2D): Dalam blok ketiga ini terdapat tiga lapisan Conv2D yang terdiri dari 512 filter, aktivasi ReLu dan *padding* "same".

- b. Menggunakan lapisan MaxPooling2D dengan ukuran (2x2) dan pergeseran (strides) (2x2).
 - c. Total jumlah *layer* pada blok 5:4.
- 6. *Fully Connected Layers*
 - a. Setelah menyelesaikan lima blok konvolusi, tahapan berikutnya pada *Fully Connected Layers* terdiri dari tiga lapisan, dengan masing-masing berisi 32 neuron dan menggunakan aktivasi ReLu.
 - b. Total jumlah *Fully Connected Layer* adalah 3.
- 7. *Output Layer*
 - a. Lapisan terakhir merupakan *Dense Layer* dengan jumlah neuron sesuai dengan jumlah kelas yang diinginkan untuk melakukan prediksi (num_classes). Penggunaan fungsi aktivasi softmax bertujuan menghasilkan probabilitas untuk setiap kelas prediksi, dan penerapannya disesuaikan dengan jumlah kategori kelas pada dataset yang terdiri dari 5 kelas.
 - b. Total jumlah *output layer* adalah 1.

4.3. Penentuan Parameter Model

Dalam usaha mencari model terbaik, penting untuk mengenali nilai-nilai parameter yang paling optimal dalam konteks model CNN. Parameter tersebut meliputi berbagai faktor seperti jumlah epoch, ukuran gambar input, jumlah data latihan, dan learning rate. Tujuan utamanya adalah menetapkan parameter-model yang berbeda dan menemukan model terbaik dengan mempertimbangkan nilai-nilai parameter yang relevan.

4.3.1. Pengaruh Banyaknya Epoch

Epoch adalah siklus lengkap di mana seluruh dataset telah dijalani sekali oleh algoritma pembelajaran mesin untuk tujuan pelatihan. Proses ini melibatkan serangkaian iterasi di mana dataset dibagi menjadi *batch* yang lebih kecil. Epoch bertujuan untuk mengatur ulang parameter model, dan ada situasi di mana model melewati beberapa epoch untuk meningkatkan kinerja atau mendapatkan hasil yang

lebih baik. Berikut adalah perbandingan hasil epoch yang dihasilkan selama proses pelatihan model.

Tabel 4.1 Tabel Epoch

Epoch	Validation Accuracy	Validation Loss	Time Epoch
10	80%	10%	3s 284ms/step
20	85%	7%	3s 253ms/step
30	95%	6%	2s 236ms/step
40	98%	2%	2s 230ms/step

Dari **Tabel 4.1**, dapat dilihat bahwa tercapai akurasi yang signifikan, yakni mencapai 93%. Informasi dari tabel tersebut mengindikasikan bahwa semakin mendekati nilai 40 epoch yang digunakan, akurasi pada pengujian cenderung meningkat, sementara nilai loss mengalami penurunan.

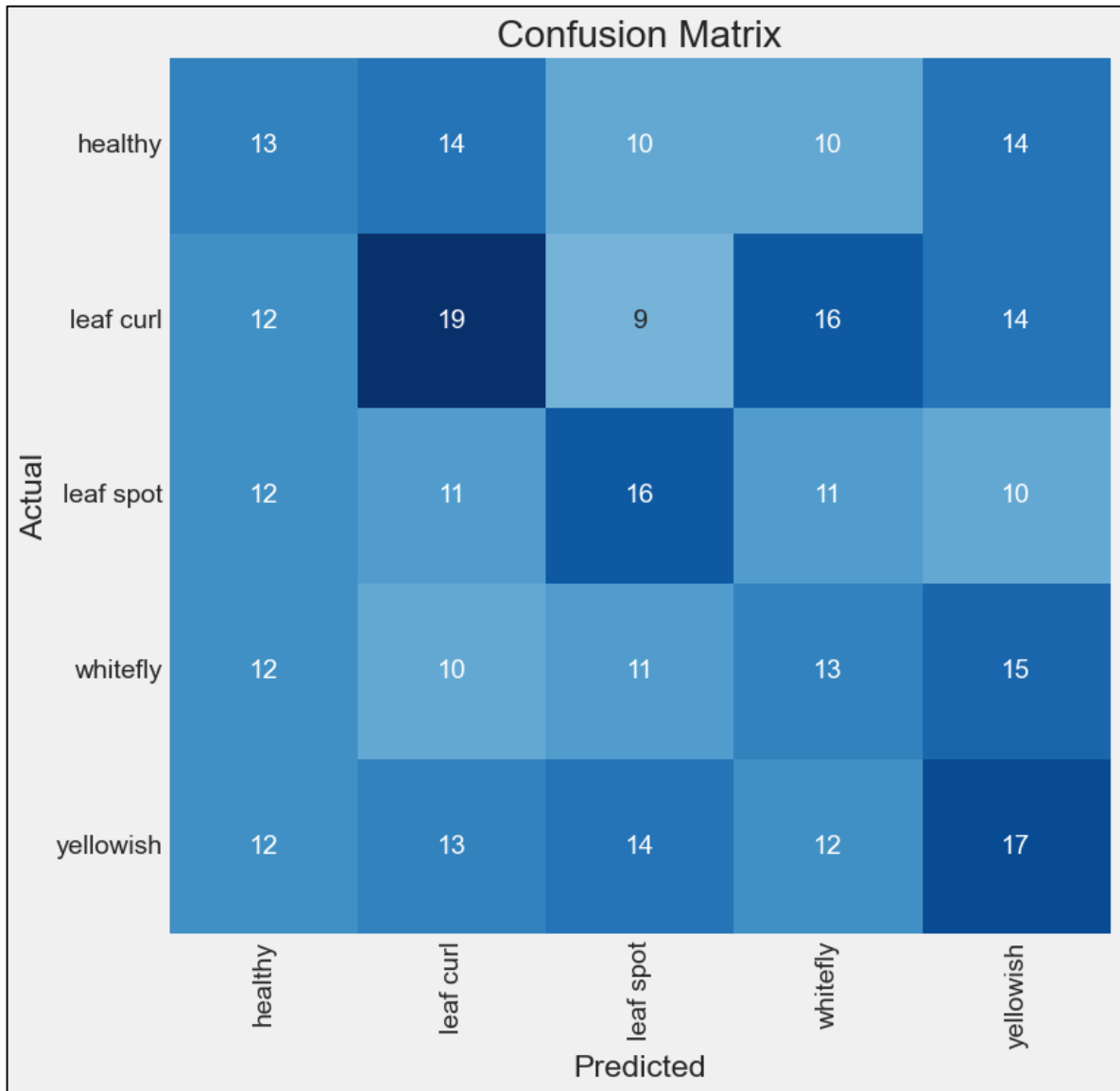
4.4. Pengujian Sistem

Hasil pengujian sistem menunjukkan evaluasi kinerja dan performa sistem. Tujuan pengujian sistem adalah untuk mengetahui seberapa baik sistem dapat memenuhi persyaratan, menemukan masalah atau kegagalan, dan memastikan bahwa sistem dapat berfungsi seperti yang direncanakan atau diharapkan.

4.4.1. Hasil Proses *Training*

Proses *training* dalam konteks pengembangan model *machine learning*, seperti *Convolutional Neural Network* (CNN) adalah tahapan di mana model itu sendiri belajar dari data yang disediakan untuk menghasilkan parameter atau bobot yang optimal. Proses ini melibatkan iterasi berulang, di mana model memperbarui parameter-nya berdasarkan kesalahan prediksi terhadap data pelatihan. Hasil dari proses *training* sangat penting untuk memastikan bahwa model *machine learning* dapat

digunakan secara efektif dalam memprediksi atau mengklasifikasikan data baru sesuai dengan tujuannya.



Gambar 4.7 Hasil Proses *Training*

Gambar 4.7 diatas menggambarkan *confusion matrix* dari hasil proses pelatihan dengan menggunakan metode *Convolutional Neural Network* (CNN), dengan tingkat akurasi mencapai 99.06% sebagaimana terlihat pada **Gambar 4.8** dibawah ini.

```

10/10 [=====] - 1s 49ms/step - loss: 0.0219 - accuracy: 0.9906
Validation Loss: 0.02193283848464489
Validation Loss: 2.19%
Validation Accuracy: 99.06%

```



Gambar 4.8 Hasil Akurasi Proses *Training*





Dari **Gambar 4.8** diatas yang diberikan, terlihat bahwa tingkat akurasi dari proses pelatihan mencapai 99,06%, dan terdapat nilai *loss* sebesar 0,02%. Data yang menunjukkan tingkat akurasi ini dilakukan dengan parameter iterasi pelatihan sebanyak 40 epoch.





4.4.2. Hasil Proses *Testing*

Hasil evaluasi proses *testing* dipresentasikan dalam bentuk tabel berikut:

Tabel 4.2 Tabel Pengujian Daun Cabai Sehat




No	Citra Daun Normal	Aktual	Prediksi
1		Sehat	Sehat
2		Sehat	Sehat





3		Sehat	Sehat
4		Sehat	Sehat
5		Sehat	Sehat
6		Sehat	Sehat




7		Sehat	Sehat
8		Sehat	Sehat
9		Sehat	Sehat
10		Sehat	Sehat

Tabel 4.2 di atas mencerminkan hasil pengujian terhadap kondisi daun cabai yang sehat. Dari sepuluh citra daun cabai yang diuji, sistem berhasil memprediksi bahwa semua daun tersebut merupakan daun cabai yang sehat.

Tabel 4.3 Tabel Pengujian Daun Keriting


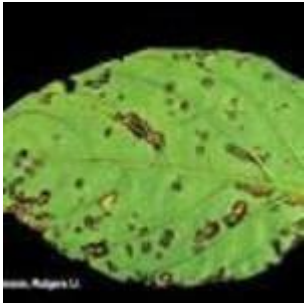


No	Citra Daun Keriting	Aktual	Prediksi
1		Daun Keriting	Daun Keriting
2		Daun Keriting	Daun Keriting
3		Daun Keriting	Daun Keriting





4		Daun Keriting	Daun Keriting
5		Daun Keriting	Daun Keriting
6		Daun Keriting	Daun Keriting
7		Daun Keriting	Daun Keriting



8		Daun Keriting	Daun Keriting
9		Daun Keriting	Daun Keriting
10		Daun Keriting	Penyakit Kekuningan

Tabel 4.3 di atas menunjukkan hasil pengujian terhadap penyakit daun cabai keriting. Dari 10 citra daun cabai yang diuji, sistem berhasil memprediksi bahwa 9 daun terkena penyakit cabai keriting, sementara satu daun mengalami penyakit kekuningan.

Tabel 4.4 Tabel Pengujian Penyakit Bercak Daun

No	Citra Bercak Daun	Aktual	Prediksi
1		Penyakit Bercak Daun	Penyakit Bercak Daun
2		Penyakit Bercak Daun	Penyakit Bercak Daun
3		Penyakit Bercak Daun	Penyakit Bercak Daun
4		Penyakit Bercak Daun	Penyakit Bercak Daun





5		Penyakit Bercak Daun	Penyakit Bercak Daun
6		Penyakit Bercak Daun	Penyakit Bercak Daun
7		Penyakit Bercak Daun	Penyakit Bercak Daun
8		Penyakit Bercak Daun	Penyakit Bercak Daun





9		Penyakit Bercak Daun	Penyakit Bercak Daun
10		Penyakit Bercak Daun	Penyakit Kekuningan

Tabel 4.4 di atas menunjukkan hasil pengujian terhadap penyakit bercak daun. Dari 10 citra daun cabai yang diuji, sistem berhasil memprediksi bahwa 9 daun terkena penyakit bercak daun, sementara 1 daun mengalami penyakit kekuningan.

Tabel 4.5 Tabel Pengujian Penyakit Kutu Kebul

No	Citra Penyakit Kutu Kebul	Aktual	Prediksi
1		Penyakit Kutu Kebul	Penyakit Kutu Kebul



2		Penyakit Kutu Kebul	Penyakit Kutu Kebul
3		Penyakit Kutu Kebul	Penyakit Kutu Kebul
4		Penyakit Kutu Kebul	Penyakit Kutu Kebul
5		Penyakit Kutu Kebul	Penyakit Kutu Kebul





6		Penyakit Kutu Kebul	Penyakit Kutu Kebul
7		Penyakit Kutu Kebul	Penyakit Kutu Kebul
8		Penyakit Kutu Kebul	Penyakit Kutu Kebul
9		Penyakit Kutu Kebul	Penyakit Bercak Daun





10		Penyakit Kutu Kebul	Penyakit Bercak Daun
----	---	---------------------	----------------------

Tabel 4.5 di atas mencerminkan hasil pengujian terhadap penyakit kutu kebul pada daun cabai. Dari 10 citra daun cabai yang diuji, sistem berhasil memprediksi bahwa 8 daun terinfeksi penyakit kutu kebul, sedangkan 2 daun mengalami penyakit bercak daun.

Tabel 4.6 Tabel Pengujian Penyakit Kekuningan

No	Citra Penyakit Kekuningan	Aktual	Prediksi
1		Penyakit Kekuningan	Penyakit Kekuningan
2		Penyakit Kekuningan	Penyakit Kekuningan

3		<p>Penyakit Kekuningan</p>	<p>Penyakit Kekuningan</p>
4		<p>Penyakit Kekuningan</p>	<p>Penyakit Kekuningan</p>
5		<p>Penyakit Kekuningan</p>	<p>Penyakit Kekuningan</p>
6		<p>Penyakit Kekuningan</p>	<p>Penyakit Kekuningan</p>

7		Penyakit Kekuningan	Penyakit Kekuningan
8		Penyakit Kekuningan	Penyakit Kekuningan
9		Penyakit Kekuningan	Penyakit Kekuningan
10		Penyakit Kekuningan	Penyakit Kekuningan

Tabel 4.6 di atas mencerminkan hasil pengujian terhadap penyakit kekuningan pada daun cabai. Dari 10 citra daun cabai yang diuji, sistem berhasil memprediksi bahwa semua daun tersebut merupakan daun cabai yang terkena penyakit kekuningan.

Dari hasil pengujian diatas, diperoleh sebuah tabel yang menunjukkan nilai *confusion matrix* sebagai berikut:

Tabel 4.7 Tabel *Confusion Matrix*

Prediksi		Penilaian <i>Confussion Matrix</i>				
		N	LC	LS	W	Y
Hasil Klasifikasi Sistem	N	10	0	0	0	0
	LC	0	9	0	0	0
	LS	0	0	9	2	0
	W	0	0	0	8	0
	Y	0	1	1	0	10

Keterangan:

- H = Normal (Daun Sehat)
- LC = *Leaf Curl* (Daun Keriting)
- LS = *Leaf Spot* (Bercak Daun)
- W = *Whitefly* (Kutu Kebul)
- Y = *Yellowish* (Penyakit Kekuningan)

Dalam *confusion matrix*, terdapat beberapa parameter evaluasi yang diuraikan sebagai berikut:

a. *Accuracy*

Perhitungan *accuracy* dilakukan dengan menambahkan semua klasifikasi yang benar dan kemudian membaginya dengan seluruh data hasil klasifikasi.

$$\frac{10 + 9 + 9 + 8 + 10}{50} \times 100\% = 92\%$$

Berdasarkan perhitungan diatas, nilai *accuracy* yang diperoleh adalah sebesar 92%.

b. *Precision*

Precision merupakan salah satu parameter evaluasi yang digunakan untuk menilai sejauh mana model klasifikasi mampu mengenali dengan benar kasus positif yang sebenarnya.

Tabel 4.8 Tabel *Precision*

	H	LC	LS	W	Y
<i>True Positive (TP)</i>	10	9	9	8	10
<i>False Positive (FP)</i>	0	1	1	2	0
<i>Precision</i> (TP)/(TP+FP)	1	0,9	0,9	0,8	1

Berdasarkan perhitungan yang telah dilakukan, didapatkan nilai *precision* sebesar 92% dengan detail sebagai berikut:

$$\frac{10}{10 + 0 + 1 + 1 + 2 + 0} \times 100\%$$

$$\frac{10}{14} \times 100\% = 92\%$$

c. *Recall*

Recall merupakan salah satu parameter evaluasi yang mengukur sejauh mana kemampuan model klasifikasi dalam mengenali dengan benar kasus positif yang sebenarnya.

Tabel 4.9 Tabel *Recall*

	H	LC	LS	W	Y
<i>True Positive (TP)</i>	10	7	8	8	10
<i>False Positive (FP)</i>	0	3	2	2	0
<i>Precision</i> (TP)/(TP+FP)	1	0,9	0,9	0,6	1

Berdasarkan perhitungan yang telah dilakukan, didapatkan nilai *recall* sebesar 92% dengan detail sebagai berikut:

$$\frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{10}{10 + 0} \times 100\%$$

$$\frac{10}{10} = \frac{4,6}{5} \times 100\% = 92\%$$

d. *F1-Score*

F1-score adalah parameter evaluasi yang mengintegrasikan *precision* dan *recall* menjadi satu nilai tunggal. Parameter ini digunakan guna memberikan gambaran keseluruhan tentang performa model klasifikasi. F1- score dihitung dengan rumus berikut:

$$F1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \times 100\%$$

$$F1 = \frac{2 \times (0,92 \times 0,92)}{0,92 + 0,92} \times 100\% = 92\%$$

Berdasarkan perhitungan yang telah dilakukan, didapatkan nilai *F-1 score* sebesar 92%.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Hasil dari penelitian yang dijalankan untuk mengidentifikasi penyakit pada tanaman cabai berdasarkan citra daun dengan menggunakan algoritma CNN menyimpulkan beberapa hal sebagai berikut:

1. Penerapan CNN sebagai metode untuk mengidentifikasi penyakit pada tanaman cabai telah terbukti berhasil. Model CNN mampu mengolah data citra dengan efisien dan memberikan hasil yang akurat dalam mengidentifikasi penyakit.
2. Menggunakan citra daun sebagai metode dalam mengidentifikasi penyakit pada tanaman cabai memiliki potensi yang signifikan. Citra daun adalah faktor penting dalam mengklasifikasikan jenis penyakit pada tanaman cabai.
3. Tingkat *accuracy* yang berhasil diperoleh mencapai 92%, dengan tingkat *precision* mencapai 92%. Tingkat *recall* juga mencapai 92%, sementara nilai *F1-score* mencapai 92%. Ini menunjukkan bahwa penggunaan CNN dalam mengidentifikasi penyakit pada tanaman cabai dapat memberikan manfaat kepada petani untuk mengelola tanaman dan mencegah penyakit.

5.2. Saran

Di bawah ini adalah beberapa saran yang dapat mendukung saat membangun sistem di penelitian selanjutnya:

1. Perluas dataset dengan mengumpulkan gambar dari berbagai kondisi pertumbuhan tanaman dan penyakit dapat meningkatkan keragaman data. Dengan dataset yang lebih kaya, model CNN memiliki potensi untuk lebih baik mengatasi variasi di lapangan.
2. Luaskan cakupan penelitian dengan menguji model pada jenis tanaman lain atau kombinasi tanaman. Hal ini dapat membuka peluang untuk penggunaan model yang sama pada variasi tanaman.

DAFTAR PUSTAKA

- Abdul Jalil Rozaqi, Andi Sunyoto, Rudyanto Arief. "Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode *Convolutional Neural Network*". *Citec Journal*, Vol.8, No.1, Januari 2021.
- Agustin, S., & Wijaya, R. (2019). Beef Image Classification using K-Nearest Neighbor Algorithm for Identification Quality and Freshness. *Journal of Physics: Conference Series*, 1179(1)
- Darmansah, Ni Wayan Wardani. "Analisa Penyebab Kerusakan Tanaman Cabai Menggunakan Metode *K-Means*". *Jurnal Teknik Informatika dan Sistem Informasi* Vol. 7, No.2, Agustus 2022, Hal. 126-134.
- Didit Iswantoro, Dewi Handayani UN. "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode *Convolutional Neural Network (CNN)*". *Jurnal Ilmiah Universitas Batanghari Jambi*, 22(2), Juli 2022, 900-905
- Errissya Rasywir, Rudolf Sinaga, Yovi Pratama". Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode *Convolutional Neural Network (CNN)*". *Jurnal Informatika dan Komputer* Vol.22, No.2 September 2020.
- Fahmi Zikra, Koredianto Usman, Raditiana Patmasari. "Deteksi Penyakit Cabai Berdasarkan Citra Daun Menggunakan Metode *Gray Level Co-Occurrence Matrix* Dan *Support Vector Machine*". Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat 2021 Institut Informatika dan Bisnis Darmajaya, 19 Agustus 2021.

- Febian Fitra Maulana, Naim Rochmawati. "Klasifikasi Citra Buah Menggunakan *Convolutional Neural Network*". *Journal of Informatics and Computer Science JINACS* : Volume 01 Nomor 02, 2019.
- Kim, P. (2017). MATLAB deep learning : with machine learning, neural networks and artificial intelligence. New York, NY: Apress.
- Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). *Implementation Of Training Convolutional Neural Networks*.
- LeCun, Y., Bengio, Y & Hinton, G. (2015). *Deep learning Nature*, 521(7553), 436-444.
- Meilin, Araz. 2014. "Hama dan Penyakit Pada Tanaman Cabai Serta Pengendaliannya" Jambi: Balai Pengkajian Teknologi Pertanian Jambi.
- Rhyosvaldo Aurellio Tilasefana, Ricky Eka Putra. "Penerapan Metode *Deep Learning* Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca". *Journal of Informatics and Computer Science JINACS*: Volume 05 Nomor 01, 2023.
- Susi Yuliany, Aradea, Andi Nur Rachman. "Implementasi *Deep Learning* pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode *Convolutional Neural Network* (CNN)". *Jurnal Buana Informatika*, Vol 13, No 1, April 2022:54-65.
- Teguh Setiawan, Donny Avianto. "Implementasi *Convolutional Neural Network* Untuk Pengenalan Warna Kendaraan". *Jurnal Program Studi Informatika Fakultas Teknologi Informasi dan Elektro Universitas Teknologi Yogyakarta* 2020.