

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sektor pertanian memainkan peran yang sangat penting dalam perekonomian global, khususnya dalam menyediakan kebutuhan pangan bagi populasi dunia yang terus berkembang. Dalam konteks ini, teknologi yang diterapkan dalam pertanian dapat membantu meningkatkan hasil pertanian dan mengurangi kerugian akibat faktor-faktor yang mempengaruhi kualitas tanaman, termasuk hama dan penyakit. Di Indonesia, sektor pertanian merupakan salah satu penyumbang utama bagi perekonomian negara. Pertanian menyumbang lebih dari 13% terhadap produk domestik bruto (PDB) Indonesia, dengan berbagai komoditas utama seperti padi, jagung, kedelai, serta sayuran dan buah-buahan yang diproduksi dalam jumlah besar setiap tahunnya [1]. Salah satu komoditas sayuran yang banyak dibudidayakan adalah sawi, yang memiliki potensi pasar domestik yang besar, serta permintaan ekspor yang terus meningkat.

Namun, dalam proses budidaya tanaman, petani sering menghadapi masalah serius berupa serangan hama yang dapat merusak tanaman dan mengurangi hasil panen secara signifikan. Hama pada tanaman, seperti kutu daun, ulat, atau tikus, dapat mengurangi kualitas dan kuantitas hasil pertanian jika tidak terdeteksi dan ditangani secara cepat dan efektif. Serangan hama pada tanaman sayuran seperti sawi dapat mengakibatkan kerusakan yang cukup parah, seperti lubang pada daun, pengguguran bunga, serta penurunan kualitas dan kuantitas hasil panen yang pada akhirnya berdampak pada keuntungan petani. Berdasarkan data dari Kementerian Pertanian, sekitar 30% dari hasil pertanian Indonesia setiap tahunnya hilang akibat serangan hama dan penyakit [2].

Selain hama, prediksi waktu panen yang tepat juga merupakan tantangan besar

yang dihadapi oleh petani. Waktu panen yang tidak tepat dapat mengurangi kualitas produk pertanian, mengakibatkan kerugian ekonomi, dan memperburuk sistem distribusi hasil pertanian. Beberapa faktor yang mempengaruhi kesiapan panen antara lain kondisi cuaca, kualitas tanah, dan kesehatan tanaman. Oleh karena itu, prediksi waktu panen yang akurat dapat membantu petani mengoptimalkan hasil pertanian, mengurangi kerugian akibat *overripe* atau *underripe*, dan meningkatkan efisiensi dalam distribusi hasil pertanian.

Tradisionalnya, deteksi hama dan prediksi kesiapan panen dilakukan secara manual oleh petani, yang tentunya memerlukan banyak waktu dan tenaga. Oleh karena itu, pemanfaatan teknologi berbasis kecerdasan buatan (AI) dan pengolahan citra menjadi salah satu solusi yang sangat potensial dalam memecahkan masalah ini. Teknologi pengolahan citra menggunakan kamera atau sensor untuk memantau kondisi tanaman, kemudian data citra tersebut dianalisis menggunakan algoritma *deep learning* untuk mendeteksi hama atau mengidentifikasi tanda-tanda kesiapan panen.

Kecerdasan buatan (AI) khususnya dalam bidang *deep learning*, telah menunjukkan hasil yang sangat baik dalam berbagai aplikasi pengolahan citra, seperti klasifikasi objek, deteksi penyakit, dan identifikasi spesies tanaman. Salah satu metode *deep learning* yang sering digunakan dalam aplikasi pertanian adalah *Convolutional Neural Networks* (CNN). CNN merupakan salah satu arsitektur jaringan saraf tiruan yang sangat efektif untuk tugas-tugas pengolahan citra karena kemampuannya untuk mengenali pola visual dengan tingkat akurasi yang sangat tinggi [3]. CNN dapat digunakan untuk menganalisis gambar daun tanaman, memeriksa keberadaan hama atau penyakit, serta mendeteksi perubahan pada tanaman yang menunjukkan bahwa tanaman telah mencapai tahap siap panen. Selain itu, penelitian oleh [4] mengaplikasikan *Convolutional Neural Networks* (CNN) untuk mendeteksi hama pada tanaman pertanian, yang menunjukkan efektivitas CNN dalam klasifikasi dan identifikasi masalah pada tanaman.

Namun, dalam konteks pertanian, selain CNN, ada juga model *deep learning*

lain yang dapat memberikan hasil yang lebih baik, yaitu VGG16 dan *Xception*. VGG16 adalah model *deep learning* yang terkenal dengan arsitekturnya yang dalam, dengan banyak lapisan konvolusi yang memungkinkan model ini untuk menangkap detail yang sangat halus dalam gambar [5]. Keunggulan utama dari model VGG16 adalah kemampuannya dalam ekstraksi fitur dari gambar, yang memungkinkan model ini mengenali karakteristik tanaman secara mendalam. Sementara itu, *Xception* adalah model *deep learning* yang merupakan pengembangan dari arsitektur *Inception* dengan menggunakan teknik *convolution* yang lebih efisien, yaitu *depthwise separable convolutions* [6]. *Xception* terkenal karena performanya yang sangat baik dalam aplikasi pengolahan citra, dengan penggunaan parameter yang lebih sedikit namun tetap memberikan hasil yang sangat akurat. Selain itu, model *Xception*, yang lebih efisien dalam penggunaan parameter, juga telah terbukti efektif dalam mendeteksi penyakit tanaman dalam berbagai kondisi yang lebih kompleks, seperti yang ditunjukkan oleh [7]. Penelitian lebih lanjut oleh [8] menyoroti penerapan AI dalam deteksi penyakit dan hama pada pertanian, menunjukkan bahwa teknologi ini memiliki potensi besar untuk meningkatkan hasil pertanian dengan meminimalkan kerugian akibat serangan hama dan penyakit.

Meskipun ketiga model tersebut (CNN, VGG16, dan *Xception*) telah diterapkan dalam berbagai penelitian untuk mendeteksi penyakit tanaman atau mengidentifikasi tanaman secara umum, perbandingan kinerja antara model-model tersebut dalam konteks deteksi hama dan prediksi kesiapan panen pada tanaman sawi masih terbatas. Beberapa studi terbaru menunjukkan bahwa model CNN dapat mendeteksi penyakit pada tanaman dengan tingkat akurasi yang cukup tinggi, namun ketika diterapkan pada tanaman tertentu seperti sawi, tantangan besar masih ada terkait dengan kondisi lingkungan yang dapat mempengaruhi kualitas gambar [9]. Demikian pula, penelitian lain menunjukkan bahwa *Xception* dapat memberikan akurasi yang lebih tinggi dalam mendeteksi penyakit tanaman dalam kondisi yang lebih kompleks dan beragam [10].

Selain itu, prediksi kesiapan panen dengan menggunakan model deep learning juga merupakan area yang masih berkembang. Beberapa penelitian telah mencoba memanfaatkan citra tanaman untuk memprediksi waktu panen berdasarkan faktor-faktor seperti ukuran daun, perubahan warna, atau tekstur daun. Namun, aplikasi model-model seperti CNN dengan VGG16, dan *Xception* untuk prediksi panen pada tanaman sawi masih memerlukan penelitian lebih lanjut. Penggunaan citra dengan resolusi tinggi, pengolahan citra *multiview*, dan penyesuaian model untuk mendeteksi ciri-ciri khas pada daun sawi dapat menjadi faktor penentu dalam meningkatkan akurasi prediksi kesiapan panen. Beberapa penelitian telah membandingkan kinerja antara berbagai model *deep learning* dalam aplikasi pertanian. [11] membandingkan CNN, VGG16, dan *Xception* untuk deteksi penyakit tanaman pada daun dan hasilnya menunjukkan bahwa *Xception* memberikan hasil yang lebih baik dalam klasifikasi dan deteksi. Sebuah studi oleh [12] juga menunjukkan bahwa *Xception* memiliki keunggulan dalam mendeteksi penyakit pada tanaman sayuran dengan akurasi lebih tinggi daripada CNN dan VGG16.

[13] dalam penelitiannya juga mengimplementasikan algoritma CNN untuk deteksi hama pada tanaman caisim, yang menunjukkan potensi besar dari teknologi ini dalam membantu deteksi otomatis hama pada tanaman sayuran, termasuk sawi. Penelitian ini menekankan pentingnya penerapan algoritma CNN dalam deteksi dini serangan hama, yang dapat meningkatkan hasil pertanian dan mengurangi kerugian yang disebabkan oleh hama.

Selain itu, prediksi kesiapan panen dengan menggunakan model deep learning juga merupakan area yang masih berkembang. Beberapa penelitian telah mencoba memanfaatkan citra tanaman untuk memprediksi waktu panen berdasarkan faktor-faktor seperti ukuran daun, perubahan warna, atau tekstur daun. Namun, aplikasi model-model seperti CNN, VGG16, dan *Xception* untuk prediksi panen pada tanaman sawi masih memerlukan penelitian lebih lanjut. Penggunaan citra dengan resolusi tinggi, pengolahan citra *multiview*, dan penyesuaian model untuk mendeteksi ciri-ciri

khas pada daun sawi dapat menjadi faktor penentu dalam meningkatkan akurasi prediksi kesiapan panen [14].

Berdasarkan uraian di atas, penelitian ini bertujuan untuk membandingkan kinerja model CNN dengan VGG16, dan Xception dalam dua tugas utama: deteksi hama pada daun sawi dan prediksi kesiapan panen. Penelitian ini diharapkan dapat memberikan kontribusi dalam penerapan teknologi deep learning di bidang pertanian, khususnya dalam meningkatkan produktivitas dan efisiensi di sektor pertanian sayuran, serta memberikan solusi yang lebih cepat dan akurat dalam mengidentifikasi masalah hama dan menentukan waktu panen yang optimal.

Penerapan teknologi deep learning dalam pertanian dapat memberikan berbagai manfaat, seperti mengurangi ketergantungan pada pengamatan manual, meningkatkan efisiensi dan akurasi deteksi hama dan penyakit, serta memberikan prediksi yang lebih akurat untuk waktu panen. Hasil dari penelitian ini diharapkan dapat memberikan landasan bagi pengembangan sistem pemantauan otomatis yang lebih efektif, yang dapat digunakan oleh petani untuk mengelola tanaman mereka dengan lebih efisien, mengurangi kerugian akibat hama, dan meningkatkan hasil pertanian secara keseluruhan.

1.2.Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, terdapat beberapa permasalahan yang akan dibahas dalam penelitian ini, yaitu:

1. Bagaimana penerapan algoritma CNN Arsitektur Xception dan CNN Arsitektur VGG-16 dalam klasifikasi hama dan siap panen pada citra daun caisim?
2. Bagaimana hasil analisis perbandingan tingkat akurasi CNN Arsitektur Xception dan CNN Arsitektur VGG-16 dalam klasifikasi hama dan siap panen pada citra daun caisim?

3. Model manakah yang memiliki performa terbaik untuk deteksi hama dan prediksi kesiapan panen berdasarkan parameter akurasi, presisi, dan recall?

1.3. Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan sebagai berikut:

1. Menganalisis penerapan algoritma CNN Arsitektur Xception dan CNN Arsitektur VGG-16 dalam klasifikasi hama dan prediksi kesiapan panen pada citra daun caisim.
2. Mengevaluasi dan membandingkan tingkat akurasi, presisi, dan recall dari CNN Arsitektur Xception dan CNN Arsitektur VGG-16 dalam mendeteksi hama dan memprediksi kesiapan panen pada citra daun caisim.
3. Mengidentifikasi model deep learning yang memiliki performa terbaik berdasarkan parameter akurasi, presisi, dan recall untuk deteksi hama dan prediksi kesiapan panen pada daun caisim.

1.4. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

1. Manfaat Teoritis:

- a. Memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang kecerdasan buatan (AI) dan pengolahan citra, khususnya dalam penerapan model deep learning untuk aplikasi pertanian.
- b. Memperluas penelitian mengenai penerapan model CNN dengan arsitektur VGG16, dan Xception untuk deteksi hama dan prediksi kesiapan panen di sektor pertanian.

2. Manfaat Praktis:

- a. Memberikan solusi berbasis teknologi yang dapat membantu petani dalam mengidentifikasi hama dan memprediksi waktu panen secara lebih akurat dan efisien.
- b. Mengurangi ketergantungan pada metode manual yang memerlukan waktu dan tenaga kerja yang besar.
- c. Membantu meningkatkan produktivitas pertanian dengan mengurangi kerugian akibat serangan hama dan ketidaktepatan waktu panen.
- d. Memberikan dasar bagi pengembangan sistem otomatis berbasis kecerdasan buatan yang dapat diimplementasikan di lapangan untuk mendukung pengelolaan tanaman yang lebih baik.

3. Manfaat Ekonomi:

- a. Meningkatkan hasil panen dan kualitas produk pertanian, sehingga berdampak pada peningkatan pendapatan petani.
- b. Mengurangi kerugian finansial akibat serangan hama dan kesalahan dalam penentuan waktu panen.

BAB II

TINJAUN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian mengenai penggunaan algoritma Convolutional Neural Network (CNN) untuk klasifikasi hama dan prediksi kesiapan panen telah banyak dilakukan. Penelitian-penelitian ini memberikan dasar yang kuat dalam memahami pendekatan deep learning untuk analisis citra daun tanaman.

Penelitian pertama dilakukan oleh [11], yang berfokus pada penerapan algoritma CNN untuk deteksi penyakit pada tanaman padi. Model yang dikembangkan berhasil mencapai tingkat akurasi sebesar 85%. Proses ekstraksi fitur dilakukan dengan menerapkan lapisan konvolusi dan pooling untuk menangkap pola pada citra daun. Studi ini menekankan pentingnya preprocessing data seperti augmentasi citra untuk meningkatkan performa model. Penelitian lain oleh [12] membandingkan kinerja beberapa arsitektur CNN, termasuk VGG16 dan Xception, untuk deteksi penyakit pada daun tanaman. Hasil eksperimen menunjukkan bahwa model Xception mencapai akurasi tertinggi sebesar 92%, sementara VGG16 juga memberikan hasil yang kompetitif dengan akurasi 90%. Evaluasi dilakukan menggunakan metrik akurasi, presisi, dan recall untuk menilai performa model. Penelitian ini menyoroti efektivitas arsitektur yang lebih dalam dalam menangkap pola kompleks dari citra tanaman. Penelitian yang lebih spesifik pada tanaman caisim dilakukan oleh [13], yang mengimplementasikan algoritma CNN untuk mendeteksi hama pada daun caisim. Model ini mampu mendeteksi hama dengan akurasi 88%. Penelitian ini juga mencatat bahwa penggunaan teknik augmentasi dan normalisasi citra mampu meningkatkan ketahanan model terhadap variasi citra yang berbeda. Hasil penelitian ini menegaskan pentingnya penggunaan CNN untuk deteksi dini hama. Selanjutnya, penelitian oleh [14] mengeksplorasi prediksi kesiapan panen berdasarkan tekstur dan ukuran daun menggunakan model deep learning. Model yang digunakan berhasil meningkatkan

efisiensi prediksi dengan akurasi 87%. Penelitian ini menekankan pentingnya parameter tekstur dan warna dalam menentukan tingkat kematangan daun caisim.

[15] mengembangkan sistem deteksi penyakit pada tanaman padi menggunakan Convolutional Neural Network (CNN). Model ini menerapkan beberapa lapisan konvolusi untuk mengekstrak fitur penting dari gambar daun padi. Hasil penelitian menunjukkan bahwa CNN mampu mencapai akurasi sebesar 89%, menjadikannya metode yang efektif untuk deteksi penyakit tanaman. Dengan arsitektur yang sederhana namun kuat, penelitian ini membuka jalan bagi pengembangan lebih lanjut dalam aplikasi serupa. [16] menerapkan arsitektur VGG16 untuk mendeteksi hama pada tanaman tomat. Model ini diimplementasikan dengan pendekatan transfer learning yang meningkatkan efisiensi dan akurasi klasifikasi. Hasil pengujian menunjukkan akurasi sebesar 91%, membuktikan keunggulan VGG16 dalam mengklasifikasikan hama dengan presisi tinggi. Keberhasilan ini menyoroti potensi transfer learning untuk mempercepat pengembangan sistem berbasis AI. [17] menggunakan model Xception dalam mengklasifikasikan penyakit pada daun cabai. Dengan arsitekturnya yang memanfaatkan depthwise separable convolutions, Xception mampu mengekstrak fitur yang lebih kompleks. Hasil pengujian menunjukkan akurasi 94%, yang lebih unggul dibandingkan CNN konvensional. Model ini menunjukkan bahwa arsitektur yang lebih dalam dan efisien mampu menangani data dengan tekstur yang lebih rumit.

[18] meneliti prediksi kesiapan panen tanaman kentang menggunakan CNN dengan transfer learning. Model ini dioptimalkan untuk mengklasifikasikan citra daun kentang yang menunjukkan tanda-tanda kematangan. Akurasi yang diperoleh sebesar 88%, membuktikan efektivitas CNN dalam analisis citra. Penelitian ini menyoroti bagaimana transfer learning dapat mempercepat pelatihan model dengan dataset yang lebih kecil. [19] membandingkan performa CNN dan VGG16 untuk deteksi penyakit tanaman jagung. VGG16 menunjukkan akurasi 93%, sedangkan CNN hanya mencapai 87%. Perbandingan ini menyoroti keunggulan arsitektur VGG16 dalam menangani

data citra yang lebih kompleks dan menegaskan pentingnya memilih arsitektur yang tepat untuk tugas klasifikasi spesifik. [20] mengimplementasikan model Xception untuk mendeteksi penyakit daun tembakau. Model ini dioptimalkan dengan augmentasi data dan fine-tuning parameter. Hasil penelitian mencatat akurasi 95%, membuktikan keunggulan Xception dalam mengolah data citra dengan variasi tekstur yang tinggi. Hasil ini memperkuat posisi Xception sebagai arsitektur yang efisien dalam pengolahan citra tanaman. [21] menggunakan CNN untuk mengidentifikasi tingkat kematangan buah mangga berdasarkan warna dan tekstur. Penelitian ini memanfaatkan metode ekstraksi fitur otomatis yang diintegrasikan dalam CNN. Hasilnya, model ini berhasil mencapai akurasi 90% dalam klasifikasi kematangan, memberikan dasar yang kuat untuk aplikasi serupa pada berbagai jenis buah dan tanaman.

[22] berfokus pada deteksi dini hama ulat pada tanaman sawi menggunakan CNN. Model ini dirancang untuk mengenali berbagai jenis hama berdasarkan pola dan warna pada gambar daun. Hasil pengujian menunjukkan akurasi sebesar 89%, membuktikan potensi CNN untuk aplikasi monitoring tanaman secara real-time. Keberhasilan ini menyoroti pentingnya integrasi teknologi dalam pemantauan pertanian. [23] membandingkan model CNN, VGG16, dan Xception dalam mendeteksi penyakit pada daun bayam. Model Xception mencapai akurasi tertinggi sebesar 96%, diikuti oleh VGG16 (92%) dan CNN (89%). Hasil ini menegaskan bahwa Xception unggul dalam menangani data dengan kompleksitas tinggi, menjadikannya pilihan utama untuk klasifikasi citra yang menuntut akurasi tinggi. [24] menggunakan metode transfer learning dengan arsitektur VGG16 untuk deteksi hama pada tanaman hortikultura. Penelitian ini menilai performa berdasarkan akurasi, presisi, dan *recall*, yang masing-masing mencapai 93%, 91%, dan 92%. Hasil ini

menunjukkan keandalan model VGG16 dalam deteksi hama dengan performa yang konsisten dan presisi tinggi.

Berdasarkan penelitian-penelitian di atas, dapat disimpulkan bahwa CNN, VGG16, dan Xception merupakan algoritma yang efektif untuk analisis citra tanaman. Namun, perbandingan langsung antara ketiga model ini khususnya untuk klasifikasi hama dan prediksi kesiapan panen pada daun caisim masih belum banyak dieksplorasi. Oleh karena itu, penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan membandingkan performa model berdasarkan metrik evaluasi seperti akurasi, presisi, dan *recall*.

2.2. Landasan Teori

2.2.1 Daun Caisim dalam Konteks Pertanian

Daun caisim (*Brassica juncea*) adalah bagian dari tanaman caisim yang memiliki peran penting dalam proses fotosintesis, yakni menghasilkan energi yang diperlukan untuk pertumbuhan tanaman. Sebagai salah satu tanaman hortikultura yang memiliki nilai ekonomi tinggi, caisim banyak dibudidayakan di Indonesia, terutama di wilayah dengan iklim tropis. Tanaman ini terkenal dengan kemampuannya tumbuh cepat dan mudah dipelihara, menjadikannya pilihan populer di pasar sayuran segar [25].

Namun, seperti tanaman lainnya, caisim rentan terhadap serangan hama dan penyakit yang dapat mempengaruhi kualitas dan hasil panen. Beberapa hama umum yang menyerang tanaman caisim adalah kutu daun, ulat, serta penyakit jamur dan bakteri yang menyebabkan bercak daun atau pembusukan. Serangan hama dan penyakit ini dapat mengakibatkan penurunan kualitas daun, mengurangi nilai jual, dan bahkan merusak seluruh tanaman jika tidak ditangani dengan tepat [26]. Oleh karena itu, deteksi dini terhadap keberadaan hama dan penyakit sangat penting untuk mengurangi kerugian dan memastikan hasil panen yang optimal.

Untuk itu, pengolahan citra digital dan penerapan metode machine learning menawarkan solusi yang efisien untuk mendeteksi gejala serangan hama pada daun

caisim. Dengan menggunakan teknologi ini, petani dapat dengan cepat mengidentifikasi kondisi tanaman tanpa perlu pemeriksaan manual yang memakan waktu. Sistem berbasis pengolahan citra digital mampu menganalisis gambar daun caisim dan mendeteksi perubahan morfologi yang disebabkan oleh hama atau penyakit. Selain itu, algoritma machine learning dapat dilatih untuk mengenali pola-pola yang khas dari gejala-gejala tersebut, memudahkan proses pemantauan yang lebih cepat dan akurat [27].

2.2.2 Citra Digital

Citra digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang continuous menjadi gambar diskrit melalui proses sampling. Gambar yang dimaksud dapat berupa gambar diam atau foto dan gambar bergerak atau video. Pada komputer, citra digital ini merupakan Kumpulan pixel pada suatu matriks 2 dimensi. piksel-piksel tersebut memiliki angka yang berbeda-beda sesuai warna. Pada pengolahannya ada suatu ilmu yang mempelajari tentang Teknik dalam mengolah suatu citra. Ilmu tersebut bernama pengolahan citra digital.

2.2.3 Machine Learning

Machine learning adalah metode pembelajaran pada mesin agar dapat mengklasifikasi / mengenali data dengan mudah. Machine learning dibagi menjadi 2 yaitu supervised dan unsupervised. Unsupervised machine learning adalah algoritma yang digunakan untuk mengklasifikasi data yang tidak diketahui kelasnya, sedangkan supervised machine learning adalah algoritma yang digunakan untuk mengklasifikasi dan memberi label pada data [28].

2.2.4 Deep Learning

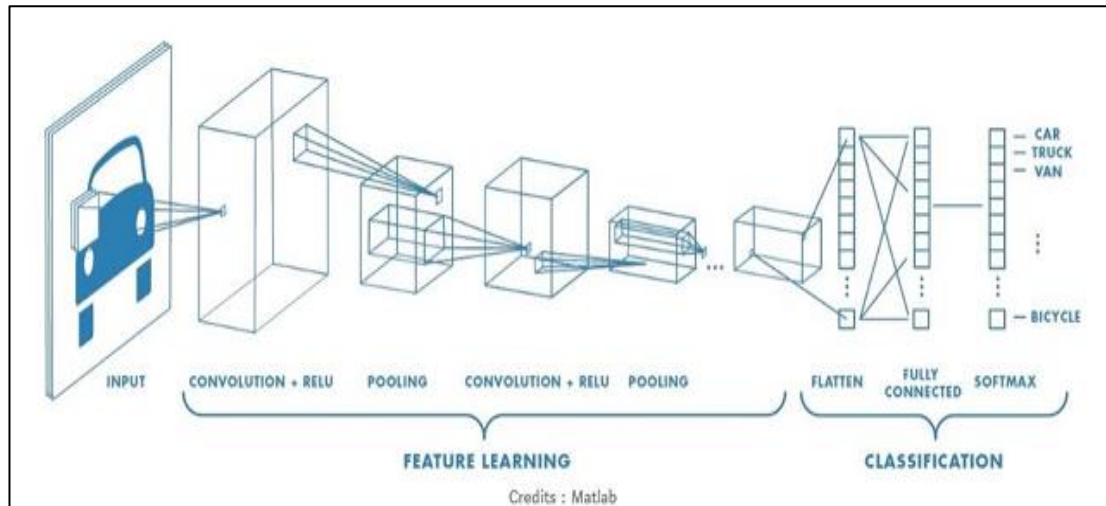
Deep learning adalah sub-bidang dari machine learning yang menggunakan jaringan saraf tiruan dengan banyak lapisan (layers) untuk menganalisis data dalam bentuk yang lebih kompleks, seperti gambar, suara, dan teks. Deep learning memiliki kemampuan untuk mengekstraksi fitur secara otomatis dari data mentah tanpa memerlukan teknik pemrograman manual yang rumit. Dalam konteks pertanian, deep

learning digunakan untuk berbagai aplikasi, termasuk deteksi penyakit pada tanaman, klasifikasi hama, dan prediksi hasil panen [29].

2.2.5 Convolutional Neural Network

Convolutional neural network dikembangkan pertama kali oleh seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Kunihiko Fukushima, dengan nama NeoCognitron. Konsep dari CNN tersebut dimatangkan oleh seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA, Yann LeChun dengan model CNN bernama LeNet yang digunakan oleh LeChun untuk melakukan pengenalan pada angka dan tulisan tangan [30]. CNN ini merupakan hasil pengembangan dari multilayer perception (MLP) yang digunakan untuk mengelola data dua dimensi. Inputan dari CNN berupa data 2 dimensi yang kemudian dipropagasikan pada jaringan dengan parameter pada bobot dan operasi linear berbeda. Operasi linear yang digunakan adalah operasi konvulsi dan bobot yang digunakan berbentuk 4 dimensi yang merupakan kumpulan dari kernel konvulsi [30]. CNN merupakan salah satu metode pada deep learning. Seperti neural network pada umumnya, sistem ini juga akan dilatih dengan backpropagation. Metode CNN memiliki banyak jenis layer, yaitu convolution layer, sumsampling/pooling layer, dan Sully connected layer. CNN juga memiliki beberapa fungsi aktivasi yang digunakan,

contohnya yaitu Relu dan softmax. Berikut ini adalah salah satu contoh arsitektur CNN:



Gambar 2.1 Arsitektur CNN

Pada gambar 2.1 dapat diketahui bahwa CNN memiliki 2 tahap. Tahap pertama yaitu ekstraksi fitur dan tahap kedua yaitu klasifikasi fitur citra. Berikut ini adalah penjelasan dari tahap-tahapan tersebut:

2.2.5.1 Ekstraksi Fitur

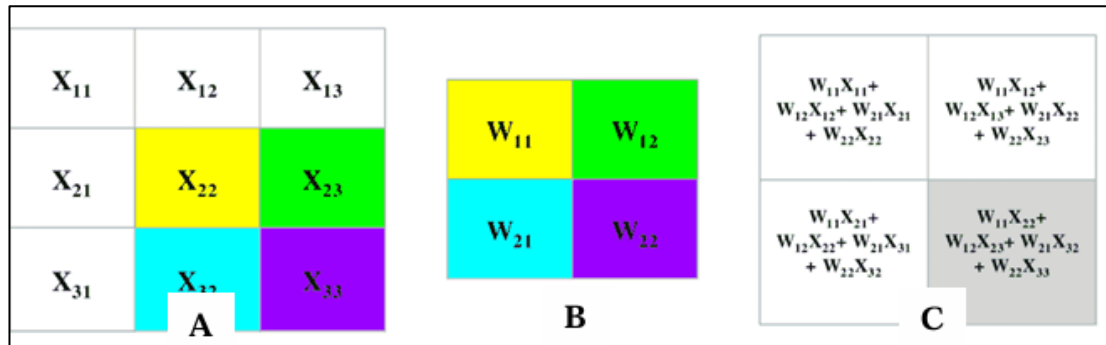
Pada tahap ekstraksi fitur citra ini, terdapat 3 layer utama yaitu:

convolution layer dan subsampling / pooling layer. Fungsi aktivasi yang digunakan pada tahap ini biasanya yaitu Relu. Jumlah layer untuk menghasilkan hasil akurasi yang baik adalah dengan melakukan banyak percobaan. Urutan layer-nya pun tidak selalu sama. Berikut ini adalah penjelasan dari masing-masing layer dan fungsi aktivasi:

1. Convolution Layer

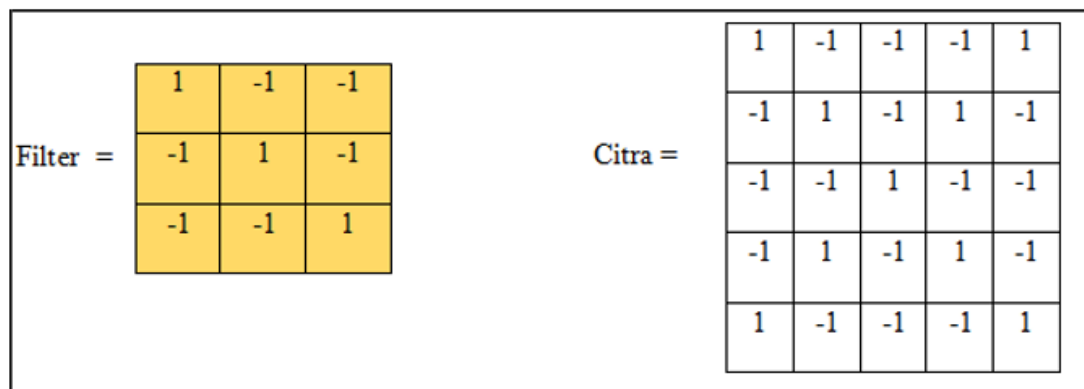
Convolution layer merupakan layer yang digunakan untuk mengekstraksi fitur dari citra yang diinputkan. Citra terdiri dari banyak piksel dengan ukuran dimensi tertentu. Convolution layer merupakan hasil dari perkalian antara kernel dengan citra yang telah diinputkan. Pada perkalian ini, kernel bergerak dari kiri atas kernel hingga kanan

bawah. Tujuan dari perkalian ini yaitu untuk menghasilkan dimensi citra yang lebih kecil.



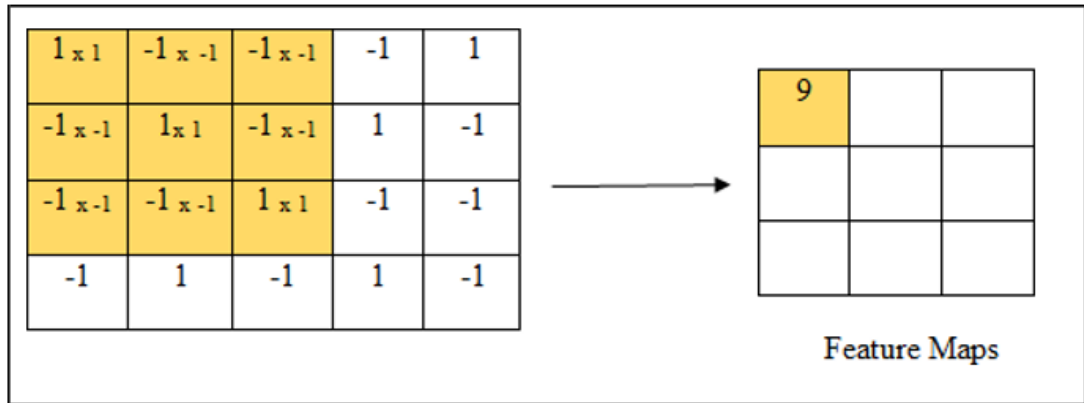
Gambar 2.2 Convolution Layer

Gambar 2.2 menunjukkan proses pada convolution layer. Bagian A merupakan inputan citra 2 dimensi dengan ukuran 3 x 3. Kemudian bagian B merupakan kernel 2 dimensi dengan ukuran 2 x 2. C merupakan hasil perkalian antara citra dengan kernel. Dapat dilihat bahwa kernel bergerak dari kiri atas ke kanan bawah dengan stride / pergeseran kernel bernilai 1. Contoh implementasi proses convolution layer yaitu pada gambar 2.3 dan 2. 4 misalkan citra yang digunakan berukuran 5 x 5 dan filter berukuran 3 x 3 :



Gambar 2.3 Filter dan Citra

Maka, proses konvolusinya yaitu :



Gambar 2.4 Proses Konvolusi

Untuk mendapatkan nilai pada feature map, citra yang diinputkan dikalikan dengan filter yang telah ditentukan. Filter bergerak dari kiri atas ke kanan bawah hingga semua bagian dari feature maps terisi. Pada contoh diatas, nilai 9 didapat dari $(1 \times 1) + (-1 \times -1) + (-1 \times -1) + (-1 \times -1) + (1 \times 1) + (-1 \times -1) + (-1 \times -1) + (-1 \times -1) + (1 \times 1) = 9$. Rumus perhitungan yang digunakan pada proses konvolusi (2.1) yaitu :

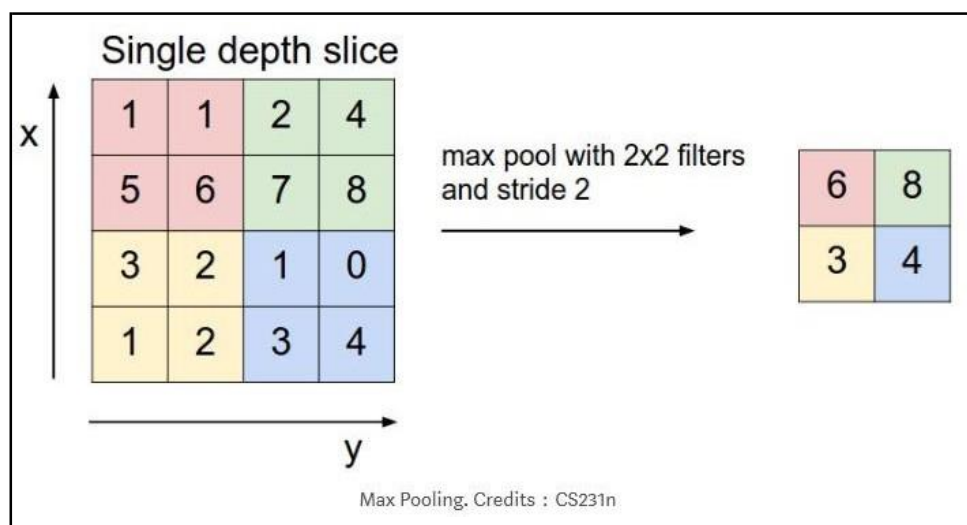
$$Q_j = f \left(\sum_{i=1}^N l_{i,j} * k_{i,j} + B_j \right) \quad (2.1)$$

Q merupakan elemen dari hasil matriks tunggal pada proses konvolusi. $l_{i,j}$ merupakan matrik dari citra yang diinputkan sedangkan $k_{i,j}$ merupakan matriks dari kernel, B_j merupakan nilai bias. Hasil matriks ini merupakan hasil dari fungsi aktifasi f . Fungsi aktifasi yang digunakan rectified linear unit (ReLU) dengan persamaan (2.2).

2. Subsampling Layer

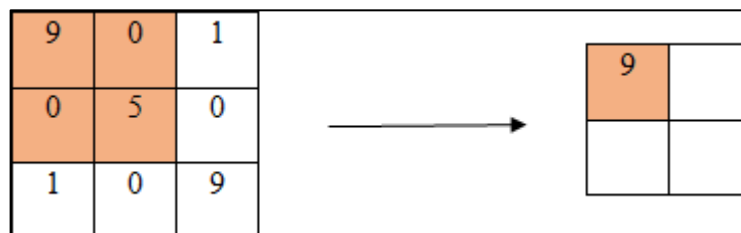
Subsampling layer merupakan layer yang digunakan untuk mereduksi ukuran dari dimensi citra. Subsampling layer ini merupakan kernel 2 dimensi dengan ukuran $m \times n$

n, biasanya kernel yang digunakan berukuran 2 x 2 sehingga ukuran dimensi citra akan berkurang setengah[31]. Citra akan dibagi menjadi beberapa bagian sesuai dengan ukuran layer yang telah ditentukan. Metode yang digunakan pada subsampling layer adalah pooling max, yaitu dengan memilih nilai terbesar pada matriks citra. Gambar 2.5 menunjukkan proses pada subsampling layer. Pada bagian kanan merupakan inputan dengan ukuran 4 x 4, kemudian dilakukan proses pooling max dengan kernel berukuran 2 x 2. Pada gambar 3.2 tersebut, stride / pergeseran kernel bernilai 2.



Gambar 2.5 Subsampling Layer

Gambar 2.6 adalah contoh implementasi *pooling layer* ukuran 2 x 2 dan stride 1 :



Gambar 2.6 Implementasi Subsampling Layer

3. ReLu

ReLU atau rectified linear unit merupakan salah satu dari fungsi aktivasi. Fungsi dari ReLU yaitu untuk menghilangkan nilai negatif pada citra. Cara kerja fungsi aktivasi ReLU ini yaitu dengan mengganti nilai negatif pada citra atau feature maps dengan nilai 0. Fungsi aktivasi ReLU ini memiliki formula (2.2) :

$$f(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (2.2)$$

2.2.6 Softmax

Softmax merupakan suatu fungsi aktivasi dengan range nilai antara 0 – 1. Fungsi aktivasi ini akan memastikan semua output bernilai 0 – 1 dan jika semua nilai output dijumlahkan maka hasilnya adalah 1[32]. Softmax ini digunakan untuk menentukan output yang sesuai. Rumus yang digunakan pada fungsi aktivasi softmax yaitu pada rumus (2.3) :

$$p(x) = \frac{e^x}{\sum_{k=1}^K e^x} \quad (2.3)$$

2.2.7 Dropout

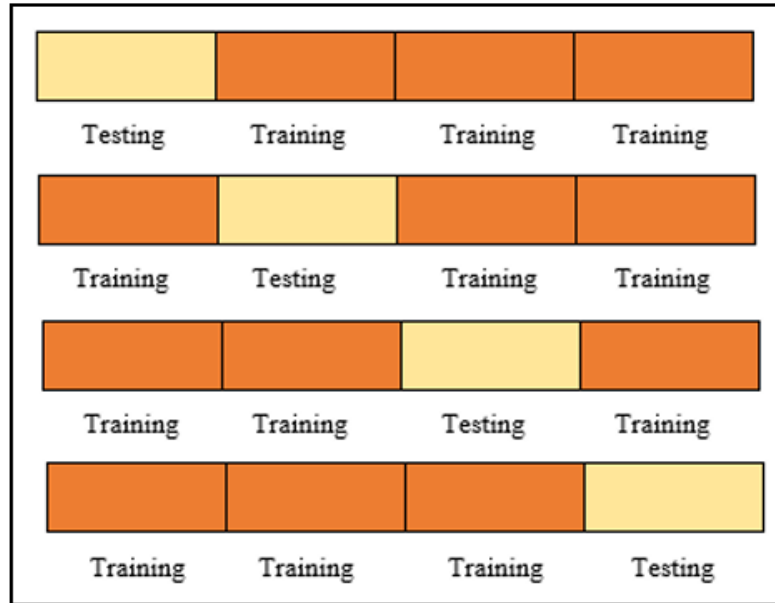
Dropout adalah proses yang digunakan untuk mencegah overfitting pada proses neural network dan juga menyediakan cara untuk menggabungkan antar neuron secara efisien. Cara kerja dari dropout ini yaitu dengan menghapus satu unit neuron dan koneksinya secara acak .

2.2.8 K-Fold Cross Validation

Cross validation merupakan metode yang digunakan untuk mengevaluasi kinerja dari suatu model yang dibuat. Biasanya, data akan dibagi menjadi k bagian. Masing – masing bagian k tersebut akan digunakan sebagai data training dan data

testing secara bergantian. Pada setiap iterasi akan menghasilkan nilai akurasi yang nantinya akan dijumlah dan dirata – rata untuk mendapatkan akurasi model [33] .

Gambar 2.7 merupakan ilustrasi dari K-Fold Cross Validation dengan nilai k = 4.



Gambar 2.7 K-Fold Crossing Validation

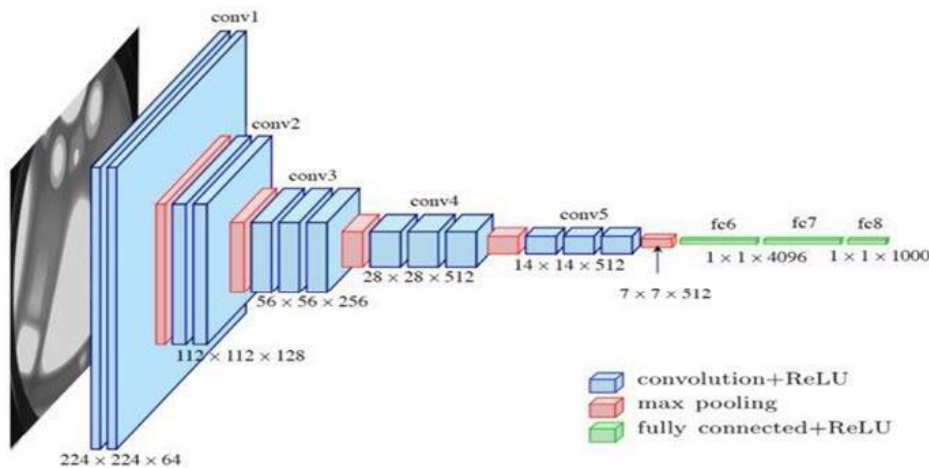
2.2.9 Cross Entropy Loss

Cross entropy loss merupakan fungsi yang digunakan untuk menghitung kinerja dari suatu model yaitu dengan menghitung error yang dihasilkan dari model tersebut. Rumus yang digunakan pada cross entropy loss yaitu ada pada rumus (2.4). Biasanya, cross entropy loss ini digunakan setelah fungsi softmax.

$$H(p, q) = - \sum_{i=1}^N p_i \log q_i \quad (2.4)$$

P merupakan target dari klasifikasi dan q merupakan output dari klasifikasi. Dalam hal ini output dari klasifikasi adalah output dari softmax.

2.2.10 VGG16



Gambar 2. 8 Arsitektur VGG-16

Pada Gambar 2.8 merupakan Arsitektur Visual Geometry Group-16 Weight Layer (VGG16). Arsitektur ini merupakan peningkatan dari arsitektur AlexNet dan dirancang oleh Karen Simonyan dan Andrew Zisserman dari universitas Oxford [5]. Arsitektur VGG16 memiliki total 16 weight layer yang terdiri dari Convolutional layer dan Fully connected layer. Keunggulan Arsitektur ini adalah menggantikan filter berukuran kernel besar dengan filter convolutional 3×3 dan pooling 2×2 dari awal hingga akhir. Kelemahan dari VGG16 adalah lebih berat untuk mengevaluasi dan menggunakan lebih banyak memori dan parameter mencapai 138 juta [35].

2.2.11 Xception

Xception (*Extreme Inception*) adalah arsitektur deep learning yang diperkenalkan oleh François Chollet pada tahun 2017 sebagai pengembangan lebih

lanjut dari Inception-v3. Xception mengadopsi prinsip dasar dari Inception-v3, namun menggunakan *depthwise separable convolution* sebagai lapisan konvolusinya, yang bertujuan untuk mengurangi jumlah parameter dan meningkatkan efisiensi komputasi [6].

Arsitektur Xception terdiri dari 36 lapisan, yang terbagi dalam 14 modul *depthwise separable convolution* yang disusun secara berurutan, diikuti oleh dua *fully connected layer* pada bagian akhir jaringan. Modul *depthwise separable convolution* menggantikan konvolusi standar dengan dua tahap proses: pertama, melakukan konvolusi pada setiap kanal input secara terpisah (*depthwise convolution*), kemudian menggabungkan hasilnya dengan konvolusi 1×1 untuk menghasilkan output (*pointwise convolution*). Proses ini menghasilkan pengolahan yang lebih efisien dibandingkan dengan konvolusi standar, yang memproses seluruh kanal secara bersamaan. Dengan menggunakan pendekatan ini, Xception mampu mengurangi jumlah parameter yang diperlukan tanpa mengorbankan akurasi [6].

Kelebihan utama dari Xception adalah kemampuannya untuk mencapai kinerja tinggi dengan lebih sedikit parameter, terutama dalam tugas-tugas yang melibatkan pengolahan citra. Pada pengujian menggunakan dataset ImageNet, Xception menunjukkan hasil yang lebih baik dibandingkan dengan model Inception-v3, dengan akurasi yang lebih tinggi dan waktu pelatihan yang lebih cepat. Selain itu, Xception juga efektif dalam menangani citra dengan tekstur yang lebih kompleks, membuatnya sangat cocok untuk berbagai aplikasi dalam pengolahan citra, termasuk deteksi objek dan klasifikasi citra dalam domain pertanian.

Namun, meskipun Xception memiliki keunggulan dalam efisiensi komputasi dan akurasi, arsitektur ini membutuhkan lebih banyak waktu dan daya komputasi

dibandingkan dengan beberapa model lain yang lebih sederhana, terutama pada perangkat dengan sumber daya terbatas.

Pada Xception, *depthwise separable convolution* dapat dijelaskan dengan dua langkah berikut:

1. **Depthwise Convolution:**

$$Y_d = X * K_d \quad (2.5)$$

Di mana X adalah input, K_d adalah filter *depthwise*, dan Y_d adalah output dari proses *depthwise convolution*. Proses ini dilakukan untuk setiap kanal secara independen.

2. **Pointwise Convolution:**

$$Y_p = X_d * K_p \quad (2.6)$$

Di mana X_d adalah output dari *depthwise convolution*, K_p adalah filter 1x1 (pointwise), dan Y_p adalah output akhir dari *pointwise convolution*.

Dengan cara ini, Xception mengurangi jumlah parameter dan komputasi, sehingga lebih efisien dalam pengolahan data citra besar.

2.2.12 Confusion Matrix

Pada tahap evaluasi dilakukan pengujian dengan menggunakan confusion matrix yang terdiri dari 4 bentuk kombinasi yang berbeda dari nilai aktual dan nilai prediksinya dalam menentukan matrix untuk mengukur klasifikasi.

Tabel 2. 1 Confussion Matrix

		Kelas Aktual	
		Negatif	Positif
Kelas Prediksi	Negatif	True Negatif	False Negatif
	Positif	False Positif	True Positif

Berdasarkan tabel di atas, terdapat empat elemen pada confusion matrix yaitu True Positive, True Negative, False Positive, dan False Negative.

- a. True Positive (TP) mewakili jumlah data yang diprediksi positif dan data aktualnya pun positif.
- b. True Negative (TN) mewakili jumlah data yang diprediksi negatif dan data aktualnya pun negatif.
- c. False Positive (FP) mewakili jumlah data yang diprediksi positif namun data aktualnya negatif.
- d. False Negative (FN) mewakili jumlah data yang diprediksi negatif namun data aktualnya positif.

Empat elemen tersebut selanjutnya dapat diproses untuk memperoleh tingkat performansi sebuah algoritma, yaitu *accuracy*, *precision*, *recall*, dan *f1- score*. Accuracy merupakan nilai hasil perbandingan dari jumlah data yang diprediksi benar dengan total seluruh data. Accuracy digunakan untuk mengukur tingkat keakuratan model algoritma dalam ketepatan hasil klasifikasi [34]. Secara matematis, *accuracy* dapat dijabarkan sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.7)$$

Precision merupakan nilai hasil perbandingan dari jumlah data yang diprediksi benar positif dengan total seluruh data yang diprediksi positif. Precision digunakan untuk mengukur seberapa sering prediksi itu benar. Ketika model algoritma memprediksi positif. Secara matematis, *precision* dapat dijabarkan sebagai berikut.

$$Precision = \frac{TP}{TP + FP} \quad (2.8)$$

Recall merupakan nilai hasil perbandingan dari jumlah data yang diprediksi benar positif dengan total seluruh data yang diprediksi benar. *Recall* digunakan untuk

mengukur seberapa sering prediksi itu positif ketika model algoritma memprediksi benar. Secara matematis, *recall* dapat dijabarkan sebagai berikut.

$$Recall = \frac{TP}{TP + FN} \quad (2.9)$$

F1 score digunakan untuk mengukur nilai rata-rata harmonik dari nilai precision dan recall. Secara matematis, F1 score dapat dijabarkan sebagai berikut.

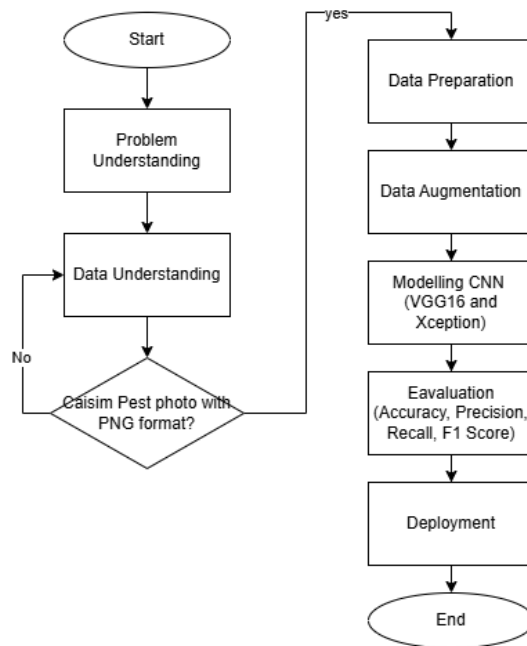
$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.10)$$

BAB III

METODOLOGI PENELITIAN

3.1 Alur Penelitian

Dalam menjalankan penelitian ini, dilakukan beberapa tahapan. Adapun alur tahapan dalam penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3. 1 Diagram Tahapan Penelitian

3.2 Identifikasi Masalah

Pada tahap ini, penting untuk memahami urgensi masalah, substansi dari pembagian kelas gambar, dan kebutuhan-kebutuhan yang ada. Tentukan tujuan dari pembagian kelas tersebut dan buatlah perencanaan strategis. Pemahaman terhadap masalah dalam penelitian ini merujuk pada substansi pembuatan klasifikasi, di mana hasil klasifikasi yang dihasilkan dapat membantu petani dalam penerapan pemupukan

yang lebih efektif di masa depan serta memberikan peringatan dini jika tanaman diserang oleh hama.

3.3 Pemahaman Data

Tahap pemahaman data melibatkan pengumpulan dataset dan eksplorasi data untuk mengidentifikasi data mana yang akan digunakan dalam analisis. Tahap ini juga bertujuan untuk mengidentifikasi masalah terkait kualitas data. Untuk penelitian ini, data yang digunakan diekstraksi dari sumber primer. Dataset yang terdiri dari gambar dalam format .png telah berhasil dikumpulkan, dengan total 1180 gambar. Model data gambar ini mencakup gambar daun caisim, yang bertujuan untuk digunakan dalam analisis lebih lanjut dan tugas klasifikasi terkait deteksi hama dan prediksi kesiapan panen. Dataset ini menjadi dasar untuk membangun dan melatih model machine learning dalam penelitian ini. Proses awal pengumpulan data diawali dengan menyiapkan direktori dataset. Setelah itu, setiap dataset akan memiliki direktorinya masing-masing sesuai dengan label mereka. Direktori masing masing gerakan akan otomatis terbuat ketika kita menambahkan label. Label yang akan digunakan dalam penelitian ini adalah *pest not ready*, *pest ready*, *no pest ready*, *no pest not ready*. Setiap gambar diatur dengan ukuran gambar 64 x 64 pixel dan akan tersimpan dalam format

file .PNG. Pada proses ini menghasilkan dataset dengan total berjumlah 1180 data. Dataset ini didapat dari kebun fakultas pertanian UPN “Veteran” Jawa Timur.







Gambar 3. 2 Pengumpulan Data

Adapun sample dataset yang digunakan terdapat pada table 3.1 dengan empat label yaitu : *pest no ready* (sawi dengan hama belum siap panen), *pest ready* (sawi dengan

hama dan siap panen), *no pest no ready*(sawi tidak ada hama dan belum siap panen), *no pest ready* (sawi tanpa hama dan siap panen).

Tabel 3. 1 Sampel dataset

Dataset Input			
 <p><i>No Pest Ready</i></p>		 <p><i>Pest Ready</i></p>	
 <p><i>Pest No Ready</i></p>		 <p><i>Not Pest No Ready</i></p>	

3.4 Preprocessing Data

3.4.1 Persiapan Data

Tahap persiapan data adalah tahap praproses data. Tahap ini mencakup pengubahan ukuran gambar, normalisasi gambar, serta pembagian data menjadi data pelatihan, data validasi, dan data pengujian. Pada tahap praproses ini, terdapat beberapa langkah yang harus dilakukan untuk menyajikan data agar siap diproses pada

tahap selanjutnya, yaitu tahap augmentasi data. Langkah-langkah tersebut adalah sebagai berikut:

1. Data yang Telah Diproses: Data dibagi menjadi 3 jenis, yaitu *Training* (Pelatihan), *Validation* (Validasi), dan *Test* (Pengujian). Data pelatihan akan diberikan komposisi lebih besar sehingga model dapat mempelajari lebih banyak pola. Fungsi dari data pelatihan adalah untuk melatih model, sementara data validasi digunakan selama proses pelatihan untuk menghasilkan fungsi kerugian yang dapat digunakan untuk mengevaluasi apakah model mengalami overfitting atau underfitting. Terakhir, data pengujian digunakan saat menguji model sebagai simulasi penggunaan model di dunia nyata. Data pengujian tidak boleh pernah dilihat oleh model yang telah dilatih sebelumnya.
2. Pengubahan Ukuran Gambar: Gambar diubah ukurannya menjadi 128x128. Ini dilakukan karena data gambar dari dataset Hama memiliki berbagai orientasi, sudut pandang, dan ukuran yang terlalu besar atau kecil. Oleh karena itu, pengubahan ukuran gambar diperlukan agar semua data memiliki ukuran yang seragam, yang akan mempercepat proses pelatihan model.

3.4.2 Augmentasi Data

Semakin banyak data dan variasi, semakin baik model dapat belajar. Oleh karena itu, semakin banyak data pelatihan yang digunakan, semakin tinggi akurasi model yang dihasilkan. Namun sebaliknya, semakin sedikit data yang tersedia, semakin banyak masalah yang dapat muncul, salah satunya adalah *overfitting*. Augmentasi data merupakan cara untuk mengatasi hal ini dengan memodifikasi data, seperti rotasi, *zoom*, *flip horizontal*, pergeseran lebar, pergeseran tinggi, dan sebagainya. Augmentasi data membantu memperluas variasi data pelatihan

sehingga model dapat belajar dengan lebih baik dan mengurangi kemungkinan *overfitting*.

3.5 Klasifikasi menggunakan CNN

Proses Pemodelan Sistem Deteksi Hama pada Daun Caisim menggunakan CNN

Proses pemodelan sistem deteksi hama yang ada pada daun caisim dilakukan dengan memanfaatkan citra daun caisim. Secara umum, CNN terdiri dari beberapa lapisan, termasuk lapisan konvolusi, lapisan pooling, lapisan dropout, lapisan flatten, dan lapisan fully connected.

1. *Convolutional Layer* (Lapisan Konvolusi): Lapisan-lapisan ini akan menghitung output dari neuron yang terhubung ke area lokal dalam input, dengan mempertimbangkan bobot titik-titik antara neuron dan area kecil yang terhubung dengan volume input yang sesuai. Di setiap lapisan, beberapa *hyperparameter* seperti ukuran filter, stride, dan padding dapat diubah. *Stride* adalah parameter yang menentukan seberapa jauh filter bergerak. Jika stride lebih kecil, maka informasi yang didapatkan dari input kemungkinan akan lebih banyak, namun memerlukan perhitungan yang lebih banyak dibandingkan dengan *stride* yang lebih besar. *Padding* adalah lapisan tambahan yang dapat ditambahkan pada batas gambar dengan menambah ukuran piksel di sekitar data input, sehingga area reseptif tidak terlalu kecil dan informasi tidak banyak yang hilang. Nilai padding ini biasanya adalah nol, yang dikenal dengan istilah *zero padding*. Proses konvolusi ini menghasilkan output yang akan digunakan sebagai input untuk lapisan konvolusi berikutnya. Fungsi aktivasi ReLU (*Rectified Linear Unit*) digunakan untuk memperkenalkan non-linearitas dan meningkatkan representasi serta model. Fungsi aktivasi ReLU akan menghilangkan masalah vanishing gradient

dengan menerapkan fungsi $f(x) = \max(0, x)$ yang akan mengaktifkan elemen ketika berada di atas threshold 0.

2. Pooling Layer (Lapisan Pooling):
Lapisan pooling bertujuan untuk mengurangi dimensi dari *feature map*, yang dapat mempercepat perhitungan karena lebih sedikit parameter yang perlu diperbarui dan juga dapat mengatasi overfitting. Nilai yang diambil dalam pooling adalah *max pooling*, yaitu nilai tertinggi dalam area filter.
3. Dropout Layer (Lapisan Dropout):
Dropout digunakan untuk mengurangi kompleksitas model yang telah dibangun dengan membangun jaringan syaraf yang tidak terpakai. Ukuran model tetap sama, tidak ada yang berubah, namun sebagian neuron secara acak dihilangkan selama pelatihan, sehingga model lebih generalisasi dan tidak terlalu terikat pada data pelatihan.
4. Flatten Layer (Lapisan Flatten):
Lapisan ini digunakan untuk merubah *feature map* menjadi vektor, sehingga dapat digunakan sebagai input untuk lapisan fully connected. Flatten mengubah matriks hasil konvolusi menjadi bentuk vektor satu dimensi.
5. Fully Connected Layer (Lapisan Fully Connected):
Lapisan ini sepenuhnya terhubung. *Feature map* yang dihasilkan masih berupa matriks filter, sehingga harus di-flatten agar bisa digunakan sebagai input untuk lapisan fully connected. Lapisan ini memiliki setiap neuron yang akan menghubungkan semua angka dan volume data, memungkinkan model untuk belajar dari informasi yang lebih kompleks dan memprediksi hasil yang lebih akurat.

3.6 Evaluasi

Setelah dilakukan implementasi model, tahap selanjutnya adalah melakukan pengujian terhadap citra atau gambar daun caisim, baik yang terinfeksi hama maupun

yang tidak. Tujuan dari pengujian ini adalah untuk mengukur tingkat keberhasilan model dalam melakukan klasifikasi, serta untuk mengetahui seberapa akurat model dalam memprediksi kondisi tanaman. Pengujian dilakukan dengan menggunakan dua metode, yaitu *Convolutional Neural Network* (CNN) VGG16 dan Xception, yang kemudian dievaluasi dengan menampilkan hasil pengujian dalam bentuk tabel.

Dalam skenario uji coba pada penelitian ini, enam skenario pengujian dilakukan pada model CNN dengan parameter-parameter sebagai berikut: ukuran *batch size* sebesar 32, *learning rate* 0.001, *optimizer* yang digunakan adalah Adam, dan fungsi aktivasi yang diterapkan adalah ReLU dan Softmax. Perbedaan dalam setiap skenario adalah penggunaan 2 epoch yang berbeda, yaitu 5 dan 10. Hasil dari uji coba ini akan dipresentasikan dalam tabel di bawah, yang akan memperlihatkan perbandingan antara skenario uji coba dan akurasi model yang didapatkan pada setiap epoch.

Tabel 3. 2 Skenario Uji Coba

Skenario Uji	Model	Epoch	Batch Size	Learning Rate	Optimizer	Fungsi Aktivasi
1	VGG16	5	32	0,001	Adam	ReLU dan SoftMax
2		10	32	0,001		
3	Xception	5	32	0,001		
4		10	32	0,001		

3.7 Deployment

Dalam penelitian ini, tahap implementasi dilakukan pada website menggunakan *framework Flask*. Tahap ini bertujuan untuk membangun aplikasi berbasis web yang dapat mengakomodasi model deteksi hama yang telah dilatih dan

dioptimalkan sebelumnya, serta memberikan antarmuka pengguna yang mudah digunakan.

Flask dipilih sebagai *framework* karena kemudahan penggunaannya dalam membangun aplikasi web berbasis Python dan fleksibilitasnya dalam menangani berbagai fitur yang diperlukan untuk sistem deteksi hama ini. Dengan Flask, aplikasi web dapat dengan mudah terintegrasi dengan model machine learning yang telah dibuat sebelumnya, memungkinkan pengguna untuk mengunggah gambar daun caisim, melakukan klasifikasi, dan mendapatkan hasil deteksi hama secara real-time.

Antarmuka pengguna yang ditampilkan pada aplikasi web ini dirancang untuk memberikan pengalaman yang sederhana dan intuitif. Pengguna dapat mengakses halaman utama yang menyediakan tombol untuk mengunggah gambar, serta melihat hasil deteksi yang berupa label klasifikasi gambar, apakah gambar tersebut mengandung hama atau tidak dan sudah siap panen atau belum.

Proses deployment yang dilakukan mencakup konfigurasi server Flask, pengaturan routing untuk setiap halaman aplikasi, serta integrasi dengan model machine learning untuk klasifikasi gambar. Flask memungkinkan pengembangan aplikasi yang cepat dan efisien, serta memberikan kontrol penuh terhadap alur kerja backend dan tampilan frontend yang dibutuhkan dalam sistem ini.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik. (2020). *Statistik Pertanian Indonesia 2020*. Jakarta: BPS.
- [2] Kementerian Pertanian RI. (2022). *Laporan Statistik Pertanian 2022*. Jakarta: Kementerian Pertanian Republik Indonesia.
- [3] LeCun, Y., et al. (2015). Deep Learning. *Nature*, 521(7553), 436-444.
- [4] Yulianto, H., Susanto, H., & Fadhilah, A. (2021). *Deep Learning for Agricultural Pest Detection Using Convolutional Neural Networks*. *International Journal of Computer Vision and Machine Learning*, 6(3), 189-200.
- [5] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- [6] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1251-1258.
- [7] Wang, H., Yang, Q., & Xu, L. (2023). *Xception-Based Model for Crop Disease Detection in Agricultural Applications*. *Computational Intelligence and Neuroscience*, 2023, 1-13.
- [8] Ning, X., Li, Z., & Wang, L. (2023). *Application of Artificial Intelligence in Pest and Disease Detection in Agriculture*. *Journal of AI and Machine Learning in Agriculture*, 5(1), 45-59.
- [9] Hadi, R., et al. (2020). *Performance of Convolutional Neural Networks in Detecting Plant Diseases*. *International Journal of Agricultural Technology*, 16(5), 915-930.
- [10] Sadeghi, H., et al. (2020). A Study on the Effectiveness of Xception in Plant Disease Detection. *International Journal of Machine Learning*, 27(4), 299-310.
- [11] Li, Y., et al. (2021). A Comparison of CNN, VGG16, and Xception Models for Plant Disease Detection. *Journal of Agricultural Informatics*, 12(2), 22-35.
- [12] Zhang, X., et al. (2023). Performance of Deep Learning Models for Plant Disease Classification in Vegetable Crops. *Agricultural Systems*, 194, 103346.
- [13] Nazalia, C. L., Palupiningsih, P., Prayitno, B., & Purwanto, Y. S. (2023, February). Implementation of Convolutional Neural Network Algorithm to Pest Detection in Caisim. In *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)* (pp. 609-614). IEEE.
- [14] Bello, O. S., Adedoyin, A. A., & Ayobami, O. (2020). *Artificial Intelligence in Crop Disease Detection and Harvest Prediction*. *Journal of Agricultural Research and Technology*, 32(4), 214-228.
- [15] Prasetyo, A., Purnomo, M., & Siregar, M. (2020). Pengembangan sistem deteksi penyakit pada tanaman padi menggunakan Convolutional Neural Network (CNN). *Jurnal Teknologi Pertanian*, 15(2), 112-120.

- [16] Sari, D., & Putra, A. (2021). Penerapan arsitektur VGG16 untuk deteksi hama pada tanaman tomat menggunakan transfer learning. *Jurnal Ilmu Komputer dan Sistem Informasi*, 7(3), 234-241.
- [17] Nugroho, R., Sutrisno, W., & Putri, S. (2022). Klasifikasi penyakit pada daun cabai dengan model Xception. *Jurnal Pengolahan Citra dan Sistem Informasi*, 5(1), 45-53.
- [18] Rahman, F., Satria, D., & Abdullah, H. (2020). Prediksi kesiapan panen tanaman kentang menggunakan CNN dan transfer learning. *Jurnal Agronomi dan Teknologi Pertanian*, 20(2), 78-85.
- [19] Dewi, R., & Aminah, Y. (2021). Perbandingan performa CNN dan VGG16 untuk deteksi penyakit tanaman jagung. *Jurnal Pertanian Digital*, 6(4), 88-95.
- [20] Kusuma, W., & Sari, S. (2023). Implementasi model Xception untuk deteksi penyakit daun tembakau dengan augmentasi data. *Jurnal Pengolahan Citra Tanaman*, 11(1), 101-108.
- [21] Ahmad, M., Nasution, Z., & Anwar, F. (2019). Identifikasi tingkat kematangan buah mangga menggunakan CNN dengan ekstraksi fitur otomatis. *Jurnal Teknologi Pertanian Tropis*, 14(2), 200-210.
- [22] Hidayat, A., & Sulaiman, R. (2022). Deteksi dini hama ulat pada tanaman sawi menggunakan CNN. *Jurnal Agroteknologi dan Citra Digital*, 8(2), 120-127.
- [23] Santoso, Y., & Firdaus, A. (2023). Perbandingan model CNN, VGG16, dan Xception dalam mendeteksi penyakit pada daun bayam. *Jurnal Teknologi Tanaman dan Kecerdasan Buatan*, 12(3), 300-310.
- [24] Utami, P., & Budi, W. (2021). Deteksi hama pada tanaman hortikultura menggunakan transfer learning dengan VGG16. *Jurnal Teknologi Informasi dan Pertanian*, 9(1), 50-58. [24] Utami, P., & Budi, W. (2021). *Deteksi hama pada tanaman hortikultura menggunakan transfer learning dengan VGG16*.
- [25] Hadiyanto, T., & Sunarto, D. (2020). *Tumbuhan Caisim dalam Konteks Pertanian Tropis*. *Jurnal Pertanian Tropika*, 28(3), 143-150.
- [26] Rini, M., & Eka, Y. (2019). Deteksi Dini Penyakit pada Daun Caisim Menggunakan Teknologi Pengolahan Citra. *Jurnal Teknologi Pertanian*, 12(1), 45-53.
- [27] Iskandar, F., & Nuraini, A. (2021). *Penerapan Machine Learning untuk Deteksi Hama pada Tanaman Caisim*. *Jurnal Ilmiah Pertanian*, 15(2), 102-110.
- [28] Turgut, S., Dağtekin, M. and Ensari, T. (2018) 'Microarray breast cancer data classification using machine learning methods', 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), pp. 1–3. doi: 10.1109/EBBT.2018.8391468.
- [29] Khan, S., Khan, M., & Akram, T. (2020). *Application of Deep Learning in Agriculture for Plant Disease Detection and Classification*. *International Journal of Computer Science in Agriculture*, 18(4), 256-263.

- [30] Suartika, I. W., Wijaya, A. Y. and Soelaiman, R. (2016) 'Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101', Jurnal Teknik ITS, 5(1), pp. 65–69. doi: 10.12962/j23373539.v5i1.15696.
- [31] Dewa, chandra kusuma, Fadhilah, amanda lailatul and Afiahayati, A. (2018) 'Convolutional Neural Networks for Handwritten Javanese Character Recognition', IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 12(1), p. 83. doi: 10.22146/ijccs.31144.
- [32] Graves, A. and Schmidhuber, J. (2005) 'Framewise phoneme classification with bidirectional LSTM and other neural network architectures.', Neural networks : the official journal of the International Neural Network Society, 18(5–6), pp. 602–610. doi: 10.1016/j.neunet.2005.06.042.
- [33] Yadav, S. and Shukla, S. (2016) 'Analysis of k-Fold Cross-Validation over Hold Out Validation on Colossal Datasets for Quality Classification', in 2016 IEEE 6th International Conference on Advanced Computing (IACC). IEEE, pp. 78–83. doi: 10.1109/IACC.2016.25.
- [34] Hendriyanto, Muhammad Diki. 2022. ANALISIS SENTIMEN ULASAN APLIKASI MOLA PADA GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE. Journal of Information Technology and Computer Science (INTECOMS). Volume 5 Nomor 1, Juni 2022.
- [35] Gunawan, I., Irawan, S., & Setianingsih, D. (2021). *Analisis Kinerja Arsitektur VGG16 dalam Pengolahan Citra Pertanian*. Jurnal Teknologi dan Informatika, 10(3), 211-220.