

**TESIS**

**KLASIFIKASI CITRA DAUN SAWI BERBASIS CNN UNTUK  
MEMBEDAKAN TANAMAN SEHAT DAN TERKENA HAMA**



Disusun oleh:

**Nama : M. Hafidurrohman**  
**NIM : 23.55.1398**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 PJJ INFORMATIKA  
PROGRAM PASCASARJANA  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2020**

**TESIS**

**KLASIFIKASI CITRA DAUN SAWI BERBASIS CNN UNTUK  
MEMBEDAKAN TANAMAN SEHAT DAN TERKENA HAMA**

**CNN-BASED CLASSIFICATION OF MUSTARD LEAF IMAGES TO  
DISTINGUISH HEALTHY AND PEST INFESTED PLANTS**



Disusun oleh:

**Nama : M. Hafidurrohman**  
**NIM : 23.55.1398**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI PJJ S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2026**

## HALAMAN PERSETUJUAN

### KLASIFIKASI CITRA DAUN SAWI BERBASIS CNN UNTUK MEMBEDAKAN TANAMAN SEHAT DAN TERKENA HAMA

### CNN-BASED CLASSIFICATION OF MUSTARD LEAF IMAGES TO DISTINGUISH HEALTHY AND PEST INFESTED PLANTS

Dipersiapkan dan Disusun oleh :

**M. Hafidurrohman**

**23.55.1398**

Telah disetujui oleh Dosen Pembimbing Tesis  
pada hari Kamis, 08 Januari 2026

**Dosen Pembimbing,**

**Prof. Dr. Kusriani, M.Kom.**

**NIK. 190302106**

## HALAMAN PENGESAHAN

### KLASIFIKASI CITRA DAUN SAWI BERBASIS CNN UNTUK MEMBEDAKAN TANAMAN SEHAT DAN TERKENA HAMA

### CNN-BASED CLASSIFICATION OF MUSTARD LEAF IMAGES TO DISTINGUISH HEALTHY AND PEST INFESTED PLANTS

Dipersiapkan dan Disusun oleh  
**M. Hafidurrohman**  
**23.55.1398**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Kamis, 08 Januari 2026

#### Susunan Dewan Penguji

##### Nama Penguji

**Hanafi, S.Kom., M.Eng., Ph.D.**  
**NIK. 190302024**

**Dr. Andi Sunyoto, S.Kom., M.Kom.**  
**NIK. 190302052**

**Prof. Dr. Kusriani, M.Kom.**  
**NIK. 190302106**

##### Tanda Tangan



Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, tanggal ujian tesis

Dekan Fakultas Ilmu Komputer



**Prof. Dr. Kusriani, M.Kom.**  
**NIK. 190302106**

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : M. Hafidurrohman  
NIM : 23.55.1398  
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:

**Klasifikasi Citra Daun Sawi Berbasis CNN untuk Membedakan Tanaman Sehat dan Terkena Hama**

Dosen Pembimbing Utama : Prof. Dr. Kusri, M.Kom.

Dosen Pembimbing Pendamping : -

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 08 Januari 2026

Yang Menyatakan,



M. Hafidurrohman

## HALAMAN PERSEMBAHAN

Segala puji dan syukur kehadiran Allah SWT atas berkah, rahmat dan hidayah-Nya yang senantiasa dilimpahkan kepada kami, sehingga dapat merampungkan Tesis yang berjudul **“KLASIFIKASI CITRA DAUN SAWI BERBASIS CNN UNTUK MEMBEDAKAN TANAMAN SEHAT DAN TERKENA HAMA”**. Semoga dapat diterima sebagai salah satu amal kebaikan. Selaras dengan harapan kami.

Penelitian ini kami persembahkan kepada kedua orang tua saya. Dengan segala bentuk dukungan lahir dan batin, kami bisa menyelesaikan studi Program Magister (S2) Program Studi Pendidikan Jarak Jauh Magister Teknik Informatika (PJJ-MTI) di Universitas Amikom Yogyakarta.

## HALAMAN MOTTO

Kesalahan bukan akhir dari penelitian, melainkan bagian dari proses menemukan kebenaran.

Pengetahuan bertumbuh melalui pertanyaan, ketekunan, dan keberanian untuk mencoba.

Never give up, even if you fail many times !



## KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT atas berkah, rahmat dan hidayah-Nya yang senantiasa dilimpahkan kepada penulis, sehingga bisa merampungkan Tesis yang berjudul “KLASIFIKASI CITRA DAUN SAWI BERBASIS CNN UNTUK MEMBEDAKAN TANAMAN SEHAT DAN TERKENA HAMA” ini dengan baik, sebagai syarat untuk menyelesaikan Program Magister (S2) pada Program Studi Pendidikan Jarak Jauh Magister Teknik Informatika (PJJ-MTI) Universitas Amikom Yogyakarta.

Dalam penyusunan Tesis ini banyak hambatan serta rintangan yang penulis hadapi namun pada akhirnya dapat dilalui berkat adanya dukungan dan bantuan dari berbagai pihak baik secara moral maupun spiritual. Untuk itu pada kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas Amikom Yogyakarta.
2. Ibu Prof. Dr. Kusrini, M.Kom., selaku Direktur Program Pasca Sarjana Universitas Amikom Yogyakarta yang telah memberikan kesempatan dan izin untuk menempuh studi lanjut di Program Studi Pendidikan Jarak Jauh Magister Teknik Informatika (PJJ-MTI).
3. Ibu Prof. Dr. Kusrini, M.Kom., selaku pembimbing utama. Yang bersedia menyempatkan waktu di sela-sela kesibukan beliau, untuk memberikan dukungan, membimbing, mengoreksi dan mengarahkan penulis demi kesempurnaan penulisan penelitian ini.

4. Tim Penguji dari SPT, SHPT, hingga UT yang telah memberikan arahan dan wawasan lebih dalam proses penyempurnaan penulisan.
5. Seluruh Dosen Pengajar di Program Studi Pendidikan Jarak Jauh Magister Teknik Informatika (PJJ-MTI) Universitas Amikom Yogyakarta dari semester pertama hingga terakhir yang memberikan arahan, dukungan, semangat, dan sharing pengetahuan sehingga penulis mendapatkan wawasan baru yang lebih luas dalam menyelesaikan tugas disetiap studi.
6. Segenap Civitas Akademika (Pengelola dan Admisi) Program Studi Pendidikan Jarak Jauh Magister Teknik Informatika (PJJ-MTI) Universitas Amikom Yogyakarta yang telah memberikan pelayanan dan bantuan sangat baik dalam kebutuhan studi.
7. Kepada Ibu dan Bapak tercinta, terima kasih atas setiap doa dan dukungan yang tiada henti. Penulis meyakini bahwa setiap langkah yang dimudahkan dan setiap keberhasilan yang diraih, bukan semata karena usaha pribadi, melainkan karena doa-doa tulus dari orang tua yang senantiasa mengiringi hingga menembus langit.
8. Untuk istriku tercinta, terima kasih atas segala dukungan dan doa yang senantiasa menyertai. Penulis percaya bahwa di balik setiap pencapaian ini, ada ketulusan hati yang diam-diam menjadi kekuatan, dan ada doa yang perlahan namun pasti menguatkan langkah ini hingga sampai pada titik ini.
9. Teman-teman Mahasiswa Program Studi Pendidikan Jarak Jauh Magister Teknik Informatika (PJJ-MTI) Universitas Amikom Yogyakarta Angkatan 2022 & 2023 yang telah memberikan pengalaman, suasana dan keluarga baru.

Dengan senang hati penulis menerima kritik dan saran yang membangun dari pembaca. Karena penulis menyadari dengan penuh bahwa dalam penyusunan penelitian ini masih banyak kekurangan. Akhir kata, semoga penelitian ini dapat memberikan manfaat bagi pembacanya.

Yogyakarta, 08 Januari 2026

  
M. Hafidurrohman

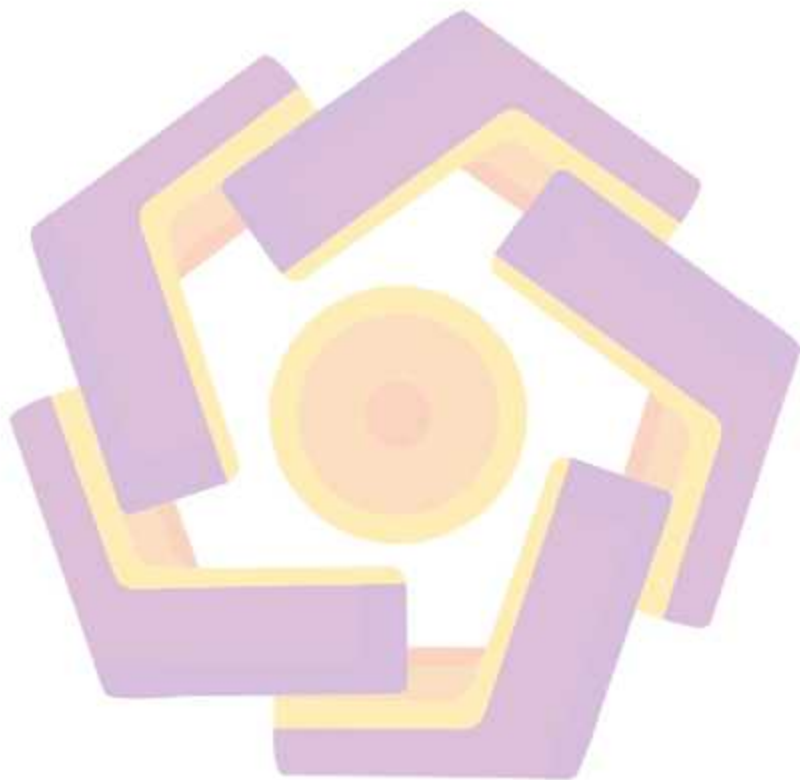


## DAFTAR ISI

|                                         |       |
|-----------------------------------------|-------|
| TESIS .....                             | ii    |
| HALAMAN PENGESAHAN.....                 | 3     |
| HALAMAN PERSETUJUAN.....                | 4     |
| HALAMAN PERNYATAAN KEASLIAN TESIS ..... | v     |
| HALAMAN PERSEMBAHAN .....               | vi    |
| HALAMAN MOTTO .....                     | vii   |
| KATA PENGANTAR.....                     | viii  |
| DAFTAR ISI.....                         | xi    |
| DAFTAR TABEL.....                       | xv    |
| DAFTAR GAMBAR.....                      | xvi   |
| INTISARI.....                           | xviii |
| <i>ABSTRACT</i> .....                   | xix   |
| Bab I.....                              | 1     |
| 1.1. Latar Belakang Masalah.....        | 1     |
| 1.2. Rumusan Masalah.....               | 5     |
| 1.3. Batasan Masalah.....               | 6     |
| 1.4. Tujuan Penelitian .....            | 6     |
| 1.5. Manfaat Penelitian .....           | 7     |

|                                                   |    |
|---------------------------------------------------|----|
| BAB II .....                                      | 8  |
| 2.1. Tinjauan Pustaka .....                       | 8  |
| 2.2. Keaslian Penelitian .....                    | 12 |
| 2.3. Landasan Teori .....                         | 16 |
| 2.3.1. Deep Learning .....                        | 16 |
| 2.3.2. Convolutional Neural Network .....         | 17 |
| 2.3.3. VGG19 .....                                | 19 |
| 2.3.4. Xception .....                             | 20 |
| 2.3.5. Dataset .....                              | 21 |
| 2.3.6. Data Preprocessing .....                   | 22 |
| 2.3.7. Data Augmentation .....                    | 23 |
| 2.3.8. Learning Rate .....                        | 25 |
| 2.3.9. Dropout .....                              | 25 |
| 2.3.10. Bayesian Optimization .....               | 26 |
| 2.3.11. Confussion Matrix .....                   | 27 |
| BAB III .....                                     | 28 |
| 3.1. Jenis, Sifat dan Pendekatan Penelitian ..... | 28 |
| 3.2. Metode Pengumpulan Data .....                | 29 |
| 3.3. Metode Analisis Data .....                   | 30 |
| 3.4. Alur Penelitian .....                        | 33 |

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| 3.4.1. Tahapan Pengumpulan Data .....                                        | 35 |
| 3.4.2. Preprocessing Data.....                                               | 35 |
| 3.4.3. Pembagian Dataset.....                                                | 37 |
| 3.4.4. <i>Data Augmentation</i> .....                                        | 38 |
| 3.4.5. Model Xception, VGG19 dan pencarian Hyperparameter ...                | 39 |
| 3.4.6. Implementasi.....                                                     | 40 |
| 3.4.7. Model <i>Evaluation (Confusion Matrix)</i> .....                      | 41 |
| BAB IV.....                                                                  | 42 |
| 4.1. Pengumpulan data.....                                                   | 42 |
| 4.2. Preprocessing data.....                                                 | 44 |
| 4.3. Pembagian dataset.....                                                  | 46 |
| 4.4. Data Augmentation .....                                                 | 47 |
| 4.5. Pemilihan Model (VGG19 & Xception) dan Pencarian<br>Hyperparameter..... | 48 |
| 4.6. Pelatihan Model .....                                                   | 51 |
| 4.7. Evaluasi Model.....                                                     | 67 |
| BAB V.....                                                                   | 81 |
| 5.1. Kesimpulan .....                                                        | 81 |
| 5.2. Saran.....                                                              | 82 |
| Daftar Pustaka .....                                                         | 85 |



## DAFTAR TABEL

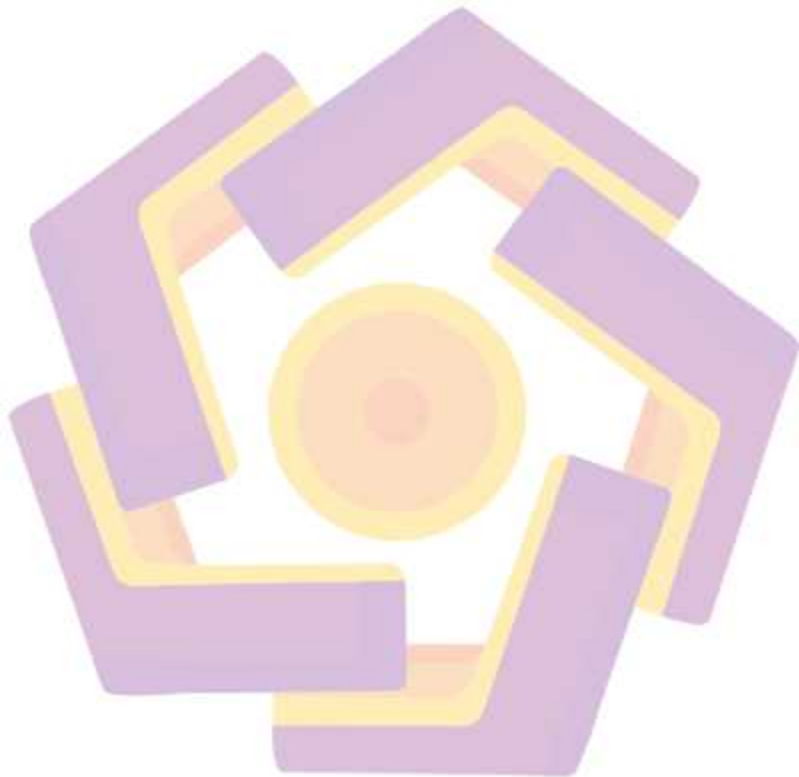
|                                                                                                                       |    |
|-----------------------------------------------------------------------------------------------------------------------|----|
| Tabel 2. 1 Matriks literatur review dan posisi penelitian.....                                                        | 12 |
| Tabel 4. 1 Total gambar dan kelas dataset.....                                                                        | 45 |
| Tabel 4. 2 Hasil Pencarian <i>hyperparameter learning rate, dropout rate</i> , dan jumlah <i>filter</i> Xception..... | 50 |
| Tabel 4.3 Hasil Pencarian <i>hyperparameter learning rate, dropout rate</i> , dan jumlah <i>filter</i> VGG19.....     | 51 |
| Tabel 4.4 Skenario Pelatihan yang akan dilakukan.....                                                                 | 53 |
| Tabel 4.5 Perbandingan dengan Penelitian Terdahulu.....                                                               | 68 |

## DAFTAR GAMBAR

|             |                                                                                      |    |
|-------------|--------------------------------------------------------------------------------------|----|
| Gambar 2. 1 | Arsitektur Deep Learning.....                                                        | 16 |
| Gambar 3. 1 | Alur penelitian.....                                                                 | 35 |
| Gambar 4. 1 | Dataset Tanaman Sawi Ada Hama dan Sawi Tanpa Hama....                                | 44 |
| Gambar 4.2  | Hasil grafik <i>Xception</i> dengan 20 <i>epoch</i> .....                            | 55 |
| Gambar 4.3  | Hasil grafik <i>Xception</i> dengan 50 <i>epoch</i> .....                            | 55 |
| Gambar 4.4  | Hasil grafik <i>Xception</i> dengan 100 <i>epoch</i> .....                           | 56 |
| Gambar 4.5  | Hasil grafik <i>VGG19</i> dengan 20 <i>epoch</i> .....                               | 58 |
| Gambar 4.6  | Hasil grafik <i>VGG19</i> dengan 50 <i>epoch</i> .....                               | 58 |
| Gambar 4.7  | Hasil grafik <i>VGG19</i> dengan 100 <i>epoch</i> .....                              | 59 |
| Gambar 4.8  | Hasil grafik <i>Xception</i> dengan 50 <i>epoch</i> .....                            | 60 |
| Gambar 4.9  | Hasil grafik <i>Xception</i> dengan 100 <i>epoch</i> .....                           | 61 |
| Gambar 4.10 | Hasil grafik <i>VGG19</i> dengan 50 <i>epoch</i> .....                               | 63 |
| Gambar 4.11 | Hasil grafik <i>VGG19</i> dengan 100 <i>epoch</i> .....                              | 64 |
| Gambar 4.12 | Hasil <i>confussion matrix</i> hyperparameter <i>Xception</i> 20 <i>epoch</i> .....  | 70 |
| Gambar 4.13 | Hasil <i>confussion matrix</i> hyperparameter <i>Xception</i> 50 <i>epoch</i> .....  | 70 |
| Gambar 4.14 | Hasil <i>confussion matrix</i> hyperparameter <i>Xception</i> 100 <i>epoch</i> ..... | 71 |
| Gambar 4.15 | Hasil <i>confussion matrix</i> hyperparameter <i>VGG19</i> 20 <i>epoch</i> .....     | 73 |
| Gambar 4.16 | Hasil <i>confussion matrix</i> hyperparameter <i>VGG19</i> 50 <i>epoch</i> .....     | 74 |
| Gambar 4.17 | Hasil <i>confussion matrix</i> hyperparameter <i>VGG19</i> 100 <i>epoch</i> .....    | 74 |
| Gambar 4.18 | Hasil <i>confussion matrix</i> <i>Xception</i> 50 <i>epoch</i> .....                 | 77 |
| Gambar 4.19 | Hasil <i>confussion matrix</i> <i>Xception</i> 100 <i>epoch</i> .....                | 77 |

Gambar 4.20 Hasil *confussion matrix* VGG19 50 epoch.....79

Gambar 4.21 Hasil *confussion matrix* VGG19 100 epoch.....80



## INTISARI

Penelitian dengan judul "Klasifikasi Citra Daun Sawi Berbasis CNN untuk Membedakan Tanaman Sehat dan Terkena Hama" ini bertujuan untuk mengembangkan sistem identifikasi yang akurat dan efisien guna meminimalisir kerusakan hasil panen. Penelitian ini membandingkan dua arsitektur *Convolutional Neural Network* (CNN), yaitu VGG19 dan Xception, dengan mengintegrasikan metode Bayesian Optimization untuk menentukan *hyperparameter* optimal secara sistematis pada aspek *learning rate*, *dropout rate*, dan jumlah filter. Hasil eksperimen menunjukkan performa yang sangat tinggi, di mana model Xception mencapai akurasi maksimal 100% dan VGG19 sebesar 99,00%, melampaui capaian penelitian terdahulu yang menggunakan parameter manual. Dari sisi efisiensi komputasi, Xception mencatatkan waktu inferensi lebih cepat yaitu 27,58 ms per citra dibandingkan VGG19 sebesar 33,37 ms. Keunggulan kecepatan Xception didorong oleh penggunaan *depthwise separable convolution* yang efektif mereduksi beban komputasi tanpa mengorbankan kualitas ekstraksi fitur, sedangkan VGG19 unggul dalam stabilitas akurasi pada jangka panjang. Meskipun sangat akurat, analisis kegagalan menunjukkan adanya potensi salah prediksi pada daun dengan lubang fisik non-hama akibat kemiripan tekstur visual. Kesimpulan penelitian menegaskan bahwa penggunaan Bayesian Optimization memberikan kontribusi besar dalam pencapaian konvergensi model yang presisi. Arsitektur Xception sangat direkomendasikan untuk implementasi pada perangkat dengan sumber daya terbatas (*edge devices*) karena keseimbangan optimal antara akurasi tinggi dan efisiensi waktu respons.

Kata Kunci : *Convolutional Neural Network*, *Xception*, *VGG19*, Daun Sawi, *Bayesian Optimization*

## ABSTRACT

*This study, titled "CNN-Based Classification of Mustard Leaf Images to Distinguish Healthy and Pest-Infested Plants," aims to develop an accurate and efficient identification system to minimize crop damage. This study compares two Convolutional Neural Network (CNN) architectures, namely VGG19 and Xception, by integrating the Bayesian Optimization method to systematically determine the optimal hyperparameters in terms of learning rate, dropout rate, and number of filters. The experimental results show very high performance, with the Xception model achieving a maximum accuracy of 100% and VGG19 achieving 99.00%, surpassing the achievements of previous studies that used manual parameters. In terms of computational efficiency, Xception recorded a faster inference time of 27.58 ms per image compared to VGG19 at 33.37 ms. Xception's speed advantage is driven by the use of depthwise separable convolution, which effectively reduces the computational load without sacrificing feature extraction quality, while VGG19 excels in long-term accuracy stability. Despite its high accuracy, failure analysis shows the potential for mispredictions on leaves with non-pest physical holes due to visual texture similarities. The research conclusion confirms that the use of Bayesian Optimization contributes significantly to achieving model convergence with precision. The Xception architecture is highly recommended for implementation on devices with limited resources.*

*Keywords: Convolutional Neural Network, Xception, VGG19, Mustard Greens, Bayesian Optimization*

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang Masalah

Daun sawi (*Brassica juncea*), sering dikenal sebagai sawi hijau, merupakan salah satu jenis sayuran hijau yang sangat populer di berbagai negara, terutama di Asia. Daun sawi memiliki ciri-ciri daunnya yang lebar bergelombang dan memiliki batang berwarna putih dan tebal. Daun sawi mudah ditemukan di berbagai tempat, utamanya di daerah yang memiliki pengairan yang cukup dengan karakteristik kering (Samiha, 2023). Tanaman ini tidak hanya mudah tumbuh, tetapi juga memiliki banyak manfaat gizi dan kesehatan, salah satunya biji sawi yang dapat digunakan untuk membuat minyak sawi (Sood et al., 2023). Selain itu, daun sawi juga mengandung banyak vitamin dan bermanfaat jika dijadikan sayur karena kaya akan nutrisi.

Daun sawi merupakan salah satu sayuran yang memiliki nilai ekonomi penting dan banyak dikonsumsi oleh masyarakat. Namun, tanaman ini rentan terhadap berbagai penyakit yang dapat menyerang daunnya, sehingga sering kali menyebabkan kerusakan signifikan pada tanaman sebelum masa panen (Utami & Erwin Dwika Putra, 2023). Penyakit yang menyerang daun sawi dapat berdampak besar pada kualitas dan kuantitas hasil panen, yang pada gilirannya mempengaruhi ketersediaan produk di pasaran serta pendapatan petani. Deteksi dini terhadap penyakit pada daun sawi menjadi langkah penting dalam upaya menjaga kesehatan tanaman dan memastikan hasil panen yang berkualitas (Abd Algani et al., 2023).

Penyakit pada tanaman sawi umumnya menyerang daun dan batang, dengan gejala utama berupa bercak daun dan perubahan warna yang menjadi indikator infeksi bakteri atau jamur. Kondisi ini dapat menyebabkan kerusakan parah pada tanaman, mengurangi hasil panen, dan menurunkan kualitas biji. Seiring dengan munculnya tantangan tersebut, penelitian terbaru mulai memanfaatkan teknologi pintar seperti pembelajaran mesin untuk mengidentifikasi dan mengklasifikasikan

penyakit pada daun sawi secara lebih efektif. Salah satu pendekatan yang banyak digunakan adalah *Convolutional Neural Network (CNN)*.

CNN merupakan metode *deep learning* yang terbukti memberikan kontribusi signifikan dalam bidang *computer vision*, khususnya klasifikasi dan deteksi objek dari citra. CNN mampu mengekstraksi fitur-fitur visual yang kompleks melalui lapisan konvolusi sehingga dapat membedakan daun sehat dan daun yang terinfeksi hama. Model ini memiliki keunggulan karena dapat mengenali pola-pola penyakit berdasarkan karakteristik visual seperti warna, tekstur, dan bentuk gejala pada daun sawi (Syech Ahmad & Sugiarto, 2023).

Penelitian yang dilakukan oleh (Sakshi et al., 2024) menggunakan empat model CNN, yaitu AlexNet, ResNet-50, VGG, dan Sequential CNN, untuk klasifikasi biner (sehat vs. tidak sehat) dan multiclass (lima penyakit spesifik: BBMD, WRMD, SRMD, PMMD, dan DMMD). Hasilnya menunjukkan akurasi rata-rata 97,50% untuk klasifikasi biner dan 93,84% untuk klasifikasi multiclass. Temuan ini membuktikan bahwa CNN dapat menjadi alat yang efektif bagi petani dalam mendeteksi penyakit tanaman secara cepat dan efisien. Deteksi dini berbasis CNN tidak hanya membantu mencegah penyebaran penyakit, tetapi juga meningkatkan kualitas hasil panen melalui penanganan yang lebih tepat sasaran. Dengan dataset citra daun sawi yang melibatkan kondisi daun yang sehat maupun yang terinfeksi penyakit, CNN dapat dilatih untuk mengklasifikasikan daun secara otomatis berdasarkan fitur visualnya (Al-Lami et al., 2023). Pendekatan ini memanfaatkan teknologi pembelajaran mesin untuk memberikan solusi praktis bagi sektor pertanian, terutama dalam klasifikasi penyakit daun sawi dengan akurasi yang lebih baik dibandingkan metode tradisional.

Selain CNN, ada metode lain juga yang sering digunakan dalam kombinasi untuk meningkatkan performa dan akurasi deteksi. Penggabungan arsitektur CNN dengan metode lain, seperti Support Vector Machine (SVM), juga telah dilakukan untuk membandingkan efektivitasnya. SVM adalah metode pembelajaran mesin yang populer digunakan dalam klasifikasi dua kelas, termasuk dalam identifikasi penyakit tanaman. Dalam penelitian terbaru, perbandingan antara metode CNN dan SVM telah dilakukan untuk mendeteksi penyakit tanaman dan hama. Hasil

eksperimen menunjukkan bahwa metode CNN dengan arsitektur ResNet mencapai akurasi 95,83%, sedangkan metode SVM menunjukkan akurasi sebesar 87,65% (Tian et al., 2020). Hasil ini menegaskan bahwa CNN lebih unggul dibandingkan SVM dalam hal akurasi deteksi penyakit dan hama pada tanaman, memberikan indikasi bahwa teknologi berbasis *deep learning* seperti CNN sangat menjanjikan untuk diterapkan dalam sistem pertanian pintar.

Dengan demikian, penggunaan arsitektur CNN dan berbagai metode pendukung dalam deteksi daun sawi sehat dan daun sawi terkena hama diharapkan dapat menghasilkan solusi yang efektif dan efisien dalam menjaga kesehatan tanaman, meningkatkan produktivitas, dan memastikan kualitas produk yang sampai ke konsumen (Lu et al., 2021). Saat ini, banyak penelitian tentang deteksi penyakit pada daun sawi telah dikerjakan. Berbagai tantangan dihadapi, antara lain bagaimana menentukan kombinasi antara preprocessing yang optimal, menentukan algoritma klasifikasi terbaik, bagaimana meningkatkan akurasi dari algoritma yang digunakan, dan bahkan bagaimana data dibersihkan.

Penelitian yang dilakukan oleh (Da Rocha et al., 2021) mengeksplorasi tiga arsitektur CNN, yaitu AlexNet, ResNet-50, dan SqueezeNet. Studi ini juga mengimplementasikan optimasi *hyperparameter* menggunakan *Bayesian optimization* untuk menentukan parameter pelatihan yang optimal, serta menerapkan teknik augmentasi data guna mengatasi masalah distribusi data yang tidak seimbang. Hasil eksperimen dari penelitian ini menunjukkan bahwa ResNet-50 dan SqueezeNet memberikan performa terbaik dengan F1-score rata-rata mencapai 96%.

Kemudian penelitian yang dilakukan oleh (Nazalia et al., 2023) menerapkan algoritma CNN yang dibangun dari awal dengan 3 kali *trial and error* tanpa menggunakan arsitektur populer seperti ResNet-50, VGG, atau Xception untuk mendeteksi hama pada tanaman sawi. Hasil penelitian menunjukkan akurasi terbaik mencapai 92%, dan peneliti juga menyarankan penggunaan *early stopping* untuk mengurangi risiko *overfitting*. Penelitian ini membuktikan bahwa penerapan CNN dalam deteksi hama dapat membantu petani mengoptimalkan hasil tanaman

sawi, meminimalisasi kerugian akibat serangan hama, serta mendukung perkembangan sektor pertanian secara keseluruhan.

Beberapa penelitian di atas telah menunjukkan bahwa CNN mampu memberikan akurasi tinggi dalam klasifikasi penyakit tanaman sawi, namun tantangan lain seperti *overfitting*, *underfitting*, dan optimasi arsitektur model masih menjadi hambatan (Srinidhi et al., 2021). Pemilihan arsitektur seperti *ResNet-50* dan *Xception* telah berhasil, namun eksplorasi *hyperparameter* seperti jumlah filter, ukuran kernel, dan tingkat *dropout* masih terbatas. Selain itu, optimasi metode pelatihan seperti Adam atau SGD dan teknik data augmentasi menjadi kebutuhan untuk meningkatkan performa model. Oleh karena itu, pengembangan metode yang lebih optimal melalui eksplorasi parameter, augmentasi data, dan penggunaan dataset yang lebih variatif sangat penting untuk mendukung pertanian berkelanjutan dan meningkatkan efisiensi deteksi penyakit tanaman (Trihardianingsih et al., 2023).

Dalam konteks ini, tujuan dari penelitian ini adalah meningkatkan kinerja klasifikasi daun sawi sehat dan terkena hama dengan mengoptimalkan parameter seperti *learning rate*, *dropout rate*, dan jumlah filter pada model CNN yang dipilih, yaitu *Xception* dan *VGG19*. Penelitian ini menggunakan dataset publik dari Kaggle yang sebelumnya telah digunakan oleh peneliti lain, namun sebagai langkah inovatif, penelitian ini mengajukan penerapan Bayesian Optimization untuk mengeksplorasi ruang *hyperparameter* secara sistematis. Pendekatan ini ditargetkan mampu memberikan konfigurasi parameter yang lebih presisi dibandingkan metode *trial and error* atau manual yang umum digunakan pada penelitian terdahulu. Dengan optimalisasi sistematis ini, penelitian ini berupaya melampaui batasan akurasi *State-of-the-Art* sebelumnya yang masih berada di rentang 92-94%, guna mencapai performa klasifikasi yang lebih maksimal dan reliabel.

Selain peningkatan akurasi, kontribusi konkret yang ditawarkan penelitian ini terletak pada evaluasi mendalam mengenai *trade-off* antara kompleksitas arsitektur dan efisiensi komputasi. Dengan membandingkan *VGG19* dan *Xception*, penelitian ini dimaksudkan untuk menyediakan referensi teknis mengenai

kecepatan inferensi (ms/citra) yang dibutuhkan dalam implementasi perangkat *mobile* atau *edge devices*. Lebih lanjut, penelitian ini juga dirancang untuk menganalisis kausalitas kegagalan klasifikasi, seperti pada kasus daun berlubang yang sering kali menjadi pengecoh (*noise*) dalam deteksi hama. Melalui pendekatan ini, diharapkan penelitian mampu memberikan kontribusi nyata bagi ilmu pengetahuan dan sektor pertanian melalui penyediaan solusi deteksi yang tidak hanya akurat, tetapi juga efisien dan memiliki landasan analisis kegagalan yang kuat untuk membantu petani mengambil tindakan secara tepat sasaran.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan sebelumnya, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

- a. Bagaimana analisis kinerja (akurasi) dan efisiensi komputasi (waktu inferensi dan jumlah parameter) antara arsitektur *deep* (VGG19) dan arsitektur *lightweight* (Xception) dapat memberikan rekomendasi implementasi yang optimal untuk klasifikasi daun sawi ?
- b. Sejauh mana perbedaan pendekatan ekstraksi fitur (konvolusi standar VGG19 vs. *depthwise separable convolution* Xception) dalam mengidentifikasi pola hama pada citra daun sawi, dan bagaimana optimasi *hyperparameter* dapat meminimalkan kekurangan inheren masing-masing arsitektur untuk mencapai akurasi maksimal?
- c. Berapa akurasi klasifikasi optimal yang dapat dicapai menggunakan VGG19 dan Xception setelah proses fine-tuning dan optimasi *hyperparameter* terbaik, serta apa rekomendasi model terbaik (VGG19 atau Xception) berdasarkan kriteria akurasi, kecepatan, dan kompleksitas model?

### 1.3. Batasan Masalah

Berikut batasan-batasan masalah pada penelitian ini:

- a. Penelitian ini dibatasi hanya pada analisis komparatif kinerja antara dua arsitektur CNN yang memiliki filosofi desain kontras: VGG19 (arsitektur *deep* klasik) dan Xception (arsitektur *lightweight* dengan *Depthwise Separable Convolution*). Metode lain di luar CNN tidak dilibatkan untuk menjaga fokus pada efisiensi fitur yang diekstrak oleh kedua arsitektur tersebut.
- b. Optimasi dan evaluasi *hyperparameter* pada kedua arsitektur dibatasi pada *learning rate*, *drop out rate* dan penyesuaian jumlah *Fully Connected Layer/output feature* (jika dilakukan *Fine-Tuning*). Parameter lain, seperti *batch size* dan *optimizer*, ditetapkan konstan dan tidak dijadikan variabel optimasi utama.
- c. Perbandingan kinerja model VGG19 dan Xception dilakukan berdasarkan tiga metrik: Akurasi Klasifikasi (sebagai indikator kinerja), Waktu Inferensi (*Inference Time*), dan Jumlah Parameter Model (sebagai indikator efisiensi komputasi).

### 1.4. Tujuan Penelitian

Bagian ini memuat tentang tujuan penelitian secara spesifik:

- a. Menganalisis dan membandingkan kinerja klasifikasi serta efisiensi komputasi antara arsitektur CNN VGG19 dan Xception dalam klasifikasi citra daun sawi berdasarkan metrik akurasi, waktu inferensi, dan jumlah parameter model.

- b. Menganalisis perbedaan pendekatan ekstraksi fitur antara VGG19 dan Xception dalam mengidentifikasi pola hama pada citra daun sawi serta pengaruh optimasi *hyperparameter* terhadap peningkatan akurasi model.
- c. Menentukan performa terbaik yang dapat dicapai oleh model VGG19 dan Xception setelah proses *fine-tuning* dan optimasi *hyperparameter*, serta merekomendasikan model yang paling optimal berdasarkan akurasi, kecepatan, dan kompleksitas model.

#### 1.5. Manfaat Penelitian

Manfaat yang dapat diambil pada penelitian ini adalah sebagai berikut:

- a. Hasil dari penelitian ini dapat menjadi referensi bagi peneliti lain yang tertarik mengembangkan lebih lanjut teknologi *deep learning* dalam mendeteksi penyakit tanaman atau aplikasi serupa di bidang pertanian, dengan model dan parameter yang lebih optimal.
- b. Bagi pengembangan teknologi kecerdasan buatan (AI), penelitian ini dapat memberikan kontribusi dalam meningkatkan performa model CNN melalui optimasi parameter, yang dapat diterapkan pada berbagai masalah klasifikasi citra, tidak hanya terbatas pada tanaman.
- c. Bagi dunia pertanian, penelitian ini dapat membantu petani meningkatkan produktivitas dengan menyediakan metode deteksi dini penyakit pada daun sawi yang lebih akurat, sehingga memungkinkan pengendalian penyakit yang lebih cepat dan efektif.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Penelitian yang dilakukan mengacu pada penelitian-penelitian sebelumnya sebagai tinjauan pustaka untuk mendukung dan sebagai penyempurna dari penelitian-penelitian sebelumnya. Berikut adalah beberapa penelitian-penelitian yang menjadi referensi bagi penelitian yang dilakukan.

(Nazalia et al., 2023) menyimpulkan bahwa algoritma (CNN) efektif dalam mendeteksi hama pada tanaman sawi. Dengan menggunakan tiga arsitektur CNN yang berbeda, penelitian ini berhasil menemukan bahwa model terbaik berasal dari Eksperimen C, yang mencapai akurasi 92%. Model tersebut dapat mengatasi masalah *underfitting* dan *overfitting* yang terjadi pada dua eksperimen sebelumnya, sehingga cocok untuk digunakan dalam mendeteksi hama pada tanaman sawi. Dataset yang digunakan dalam penelitian ini terdiri dari 1000 gambar yang diambil dari *Kaggle*. Kelebihan dari penelitian ini adalah kemampuannya dalam menggunakan arsitektur CNN secara akurat untuk mengidentifikasi hama. Dengan hasil yang signifikan, teknologi ini berpotensi membantu petani dalam mengoptimalkan hasil panen dan menerapkan pengendalian hama secara lebih efektif. Namun, kekurangannya terletak pada keterbatasan ukuran dataset, yang hanya terdiri dari 1000 gambar. Jumlah dataset yang terbatas ini dapat mempengaruhi performa model dan meningkatkan risiko terjadinya *overfitting* pada data baru yang belum pernah dilihat oleh model. Penulis menyarankan agar penelitian selanjutnya memperbesar ukuran dataset untuk mendapatkan hasil yang lebih akurat dan menghindari masalah *overfitting*.

Penelitian berikutnya yang dilakukan oleh (Sood et al., 2023) ini berfokus pada klasifikasi biner untuk membedakan antara tanaman yang sehat dan sakit, serta klasifikasi multiclass untuk mendeteksi lima jenis penyakit utama pada tanaman sawi. Peneliti menggunakan CNN sebagai model utama dalam klasifikasi, yang terbukti memberikan kinerja yang baik dalam deteksi otomatis dengan akurasi rata-

rata 94,1% untuk klasifikasi biner dan 93,84% untuk klasifikasi multiclass. Dataset yang digunakan dalam penelitian ini diperoleh dari *Kaggle (PlantVillage Dataset)* dan beberapa lokasi pertanian di daerah Punjab dan Haryana, India dengan jumlah dataset adalah 13.00 gambar. Kelebihan dari penelitian ini terletak pada penggunaan model CNN yang secara otomatis mengekstraksi fitur dari gambar daun, sehingga mengurangi ketergantungan pada teknik ekstraksi fitur manual yang kompleks. Namun, penelitian ini memiliki beberapa kekurangan. Salah satu keterbatasannya adalah ketergantungan pada kualitas dan kuantitas dataset yang digunakan. Tantangan lain yang dihadapi adalah potensi kesalahan klasifikasi pada penyakit yang memiliki karakteristik serupa, yang dapat menyebabkan penurunan akurasi dalam beberapa kasus tertentu. Penulis juga menyarankan penggunaan dataset yang lebih besar dan lebih beragam, serta eksplorasi teknik-teknik lain untuk mengatasi kesalahan klasifikasi. Penelitian mendatang juga dapat fokus pada pengembangan model yang lebih kuat terhadap variabilitas lingkungan di lapangan, seperti perubahan cuaca atau kondisi pencahayaan, yang dapat mempengaruhi akurasi model dalam mendeteksi penyakit.

Penelitian berikutnya oleh (Sakshi et al., 2024) mengusulkan metode deteksi penyakit pada daun mustard menggunakan jaringan saraf konvolusi CNN. Penelitian ini memanfaatkan dataset dari *Plant Village Databank* yang diambil dari *Kaggle.com*, berisi 3806 gambar yang terbagi dalam kategori daun sehat dan daun yang terinfeksi penyakit. Arsitektur CNN yang digunakan, seperti Sequential CNN, AlexNet, ResNet-50, dan VGG19, menunjukkan performa tinggi, dengan akurasi deteksi penyakit mencapai lebih dari 99%, terutama pada model ResNet-50 dan AlexNet. Kelebihan dari penelitian ini adalah kemampuan model CNN untuk secara akurat mendeteksi penyakit pada daun mustard, yang berpotensi meningkatkan produktivitas pertanian. Namun, penelitian ini masih memiliki keterbatasan, seperti ukuran dataset yang kecil, yang dapat menyebabkan overfitting, dan tantangan dalam penerapan teknologi ini di lapangan. Peneliti menyarankan agar penelitian selanjutnya memperbesar dataset dan mengeksplorasi metode yang lebih kompleks, seperti ensemble learning, untuk meningkatkan kinerja model dalam kondisi dunia nyata.

Penelitian selanjutnya yang dilakukan oleh (Hridoy et al., 2022) menyajikan solusi untuk mengatasi masalah utama yang dihadapi dalam produksi minyak sawi, yakni sulitnya mengidentifikasi penyakit tanaman dengan cepat karena keterbatasan infrastruktur identifikasi ahli. Dalam studi ini, penulis mengembangkan dataset khusus yang terdiri dari sembilan kelas penyakit tanaman sawi dengan menggunakan teknik augmentasi gambar sebanyak sebelas metode, sehingga menghasilkan 47.760 gambar daun, batang, dan polong tanaman sawi. Dataset ini dikumpulkan dari beberapa lahan pertanian sawi, memastikan variasi dan kesulitan latar belakang gambar. Salah satu kelebihan penelitian ini adalah arsitektur CNN yang baru, yang disebut MPNet, yang dirancang dari awal menggunakan *deep separable convolutional layers* (DSCL) dan *Inception modules* (IM). Dengan desain ini, MPNet mampu mencapai akurasi pengenalan sebesar 97,11% pada 2.388 gambar uji, yang melebihi kinerja beberapa model CNN canggih lainnya seperti MobileNetV2, DenseNet121, VGG19, dan ResNet50. Namun, terdapat beberapa kelemahan dalam penelitian ini. Misalnya, meskipun MPNet menunjukkan performa yang sangat baik dalam mengenali sebagian besar kelas penyakit, masih ada kelas tertentu seperti *leaf spot* yang memiliki akurasi dan sensitivitas yang sedikit lebih rendah dibandingkan kelas lainnya. Hal ini disebabkan oleh kemiripan fitur geometris antara gejala penyakit yang menyebabkan CNN mengalami kesulitan dalam melakukan pengenalan yang rinci. Selain itu, karena penelitian ini berfokus pada dataset yang terbatas pada sembilan kelas penyakit tanaman sawi, penerapannya pada kondisi yang lebih luas dan lebih beragam belum dapat diuji. Penulis menyarankan agar penelitian di masa mendatang memperluas dataset dengan mengumpulkan lebih banyak sampel dari berbagai kondisi dan menambahkan kelas penyakit lainnya. Dengan dataset yang lebih besar dan lebih beragam, model pengenalan seperti MPNet diharapkan dapat lebih dioptimalkan dan diaplikasikan dalam situasi yang lebih kompleks di lapangan.

Penelitian selanjutnya dilakukan oleh (Abd Algani et al., 2023) membahas sebuah metode baru untuk mendeteksi dan mengklasifikasikan penyakit daun tanaman menggunakan teknik optimasi koloni semut (ACO) yang dikombinasikan

dengan jaringan saraf konvolusi (CNN). Metode ini menunjukkan peningkatan yang signifikan dalam akurasi deteksi penyakit dibandingkan teknik lain, dengan mencapai tingkat akurasi hingga 99,98%. Penggunaan CNN dalam klasifikasi citra daun memungkinkan ekstraksi fitur yang lebih baik dari gambar daun, sementara algoritma ACO membantu menemukan jalur optimal untuk mengidentifikasi bagian daun yang terinfeksi. Kelebihan utama dari penelitian ini adalah kemampuannya untuk memberikan hasil yang lebih akurat dalam waktu yang lebih singkat dibandingkan metode tradisional lainnya, seperti *Generative Adversarial Networks* (GAN) atau *Stochastic Gradient Descent* (SGD). Metode ini juga dapat diandalkan dalam kondisi lingkungan yang berbeda dan pada variasi bentuk daun yang berbeda, menjadikannya fleksibel untuk berbagai jenis penyakit daun tanaman. Namun, penelitian ini juga memiliki kekurangan, terutama terkait kebutuhan akan jumlah data yang besar untuk pelatihan model yang optimal, serta kompleksitas perhitungan yang lebih tinggi dibandingkan beberapa metode konvensional lainnya. Selain itu, meskipun metode ini efisien dalam mengenali gejala penyakit pada daun, penelitian ini masih bergantung pada kualitas gambar yang baik, yang mungkin tidak selalu tersedia di lingkungan pertanian yang sebenarnya. Peneliti memberikan saran agar penelitian selanjutnya dapat memperluas metode ini dengan menggunakan dataset yang lebih besar dan lebih bervariasi.

Dan penelitian selanjutnya dilakukan oleh (Paymode & Malode, 2022) ini berfokus pada penerapan *Transfer Learning* menggunakan model CNN berbasis VGG19 untuk klasifikasi penyakit daun pada tanaman multikultur, khususnya daun tanaman tomat dan anggur. Dalam penelitian ini, dataset yang digunakan berasal dari *PlantVillage*, yang merupakan salah satu dataset terkenal untuk gambar penyakit tanaman. Hasil penelitian menunjukkan bahwa model CNN berbasis VGG19 yang digunakan mampu mencapai akurasi tinggi, yaitu 98,40% untuk daun tanaman anggur dan 95,71% untuk daun tanaman tomat. Keunggulan utama dari penelitian ini adalah peningkatan performa deteksi penyakit daun melalui penggunaan *Transfer Learning* dan model CNN, yang terbukti efektif dalam mengklasifikasikan daun yang sakit dan sehat dengan presisi tinggi. Selain itu,

penggunaan model yang telah dilatih sebelumnya (*pre-trained models*) membantu mempercepat proses pelatihan dan menghasilkan akurasi yang optimal. Namun, penelitian ini juga memiliki beberapa keterbatasan. Salah satunya adalah perbedaan antara gambar dalam dataset *PlantVillage*, yang diambil di laboratorium, dengan gambar nyata dari lapangan yang digunakan dalam pengujian. Hal ini berpotensi menyebabkan penurunan performa model ketika dihadapkan dengan data dari kondisi lapangan yang lebih bervariasi. Selain itu, meskipun akurasi model cukup tinggi, ada tantangan dalam hal implementasi di dunia nyata karena faktor-faktor seperti variasi pencahayaan, kondisi lingkungan yang dinamis, dan kualitas gambar yang dapat mempengaruhi kinerja model secara keseluruhan. Sebagai saran untuk penelitian selanjutnya, penulis menyarankan pengembangan database lapangan yang lebih komprehensif dan nyata, yang dapat digunakan untuk melatih model. Selain itu, penelitian di masa depan dapat memanfaatkan model CNN yang lebih kompleks, seperti Inception V3 dan ResNet, untuk analisis yang lebih mendalam dan akurasi yang lebih tinggi dalam mendeteksi penyakit tanaman.



## 2.2. Keaslian Penelitian

Tabel 2. 2 Matriks literatur review dan posisi penelitian  
Tuliskan Judul Proposal Tesis di Baris ini

| No | Judul                                                                                         | Peneliti, Media Publikasi, dan Tahun | Tujuan Penelitian                                                                                                                                                                                                                              | Kesimpulan                                                                                                                                                                                                        | Saran atau Kelemahan                                                                                                                                      | Perbandingan                                                                                                                                                                                                               |
|----|-----------------------------------------------------------------------------------------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Implementation of Convolutional Neural Network Algorithm for Pest Detection in Caisim Plants  | (Nazalia et al., 2023) IEEE          | Penelitian ini bertujuan untuk menerapkan algoritma <i>Convolutional Neural Network</i> (CNN) untuk mendeteksi hama pada tanaman sawi, membantu petani mengidentifikasi hama secara dini agar pengendalian hama dapat dilakukan lebih efektif. | Kesimpulan dari penelitian ini adalah bahwa model CNN, khususnya dari Eksperimen C, mencapai akurasi terbaik sebesar 92%, dan model tersebut dapat secara efektif digunakan dalam deteksi hama pada tanaman sawi. | Saran dari penelitian ini adalah dengan memperbesar ukuran dataset untuk mendapatkan hasil yang lebih akurat dan menghindari masalah <i>overfitting</i> . | Penelitian ini mengeksplorasi model <i>CNN from scratch</i> dengan 3 kali percobaan, sedangkan penelitian ini menggunakan model yang sudah disediakan oleh CNN, yaitu Xception dan VGG19 untuk mencari metode CNN terbaik. |
| 2  | AI-Driven Mustard Disease Identification: A Multiclass and Binary Classification Approach for | (Sood et al., 2023) IEEE             | Tujuan penelitian ini untuk mengembangkan metode berbasis <i>deep learning</i> yang mampu mendeteksi                                                                                                                                           | Kesimpulan dari penelitian ini adalah bahwa metode CNN yang diterapkan berhasil mencapai akurasi tinggi dalam klasifikasi                                                                                         | Kelemahan utama dari penelitian ini adalah ketergantungan pada kualitas dan kuantitas <i>dataset</i> , serta potensi kesalahan klasifikasi                | Penelitian ini berfokus pada deteksi penyakit pada tanaman sawi menggunakan satu metode CNN untuk klasifikasi biner dan multiclass, dengan model                                                                           |

| No | Judul                                                                          | Peneliti, Media Publikasi, dan Tahun | Tujuan Penelitian                                                                                                                                                                                                                                      | Kesimpulan                                                                                                                                                                                                                                    | Saran atau Kelemahan                                                                                                                                                                                                                                                                                                                                                       | Perbandingan                                                                                                                                                                                                                                                 |
|----|--------------------------------------------------------------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | Advanced Crop Health Monitoring                                                |                                      | penyakit pada tanaman sawi melalui klasifikasi gambar daun, baik secara biner (sehat atau sakit) maupun <i>multiclass</i> (lima jenis penyakit utama).                                                                                                 | penyakit sawi, yaitu 94,1% untuk klasifikasi biner dan 93,84% untuk klasifikasi <i>multiclass</i> . Model ini terbukti efektif dalam mendeteksi dan mengklasifikasikan lima penyakit utama pada tanaman sawi.                                 | pada penyakit yang memiliki karakteristik mirip. Sedangkan saran dari peneliti ini adalah memperluas <i>dataset</i> yang digunakan.                                                                                                                                                                                                                                        | dasar CNN. Sedangkan, penelitian yang diajukan akan menggunakan beberapa metode CNN yang lebih kompleks seperti <i>Xception</i> dan <i>VGG19</i> . Dan juga penelitian ini berfokus pada optimasi parameter seperti learning rate, dropout rate, dan filter. |
| 3  | Green Intelligence: A Sequential CNN Odyssey in Mustard Leaf Disease Detection | (Sakshi et al., 2024) IEEE           | Tujuan dari penelitian ini untuk mengembangkan model <i>deep learning</i> berbasis CNN yang dapat mendeteksi dan mengklasifikasikan penyakit pada daun sawi secara akurat, untuk mendukung pertanian dan meningkatkan produktivitas pada tanaman sawi. | Kesimpulan dari penelitian ini adalah bahwa model CNN, terutama ResNet-50 dan AlexNet, berhasil mendeteksi penyakit daun sawi dengan akurasi tinggi, mencapai lebih dari 99%, menunjukkan potensi besar <i>deep learning</i> dalam pertanian. | Kelemahan dari penelitian ini terletak pada ukuran <i>dataset</i> yang kecil, yang dapat menyebabkan <i>overfitting</i> , serta tantangan penerapan teknologi di lingkungan pertanian yang mungkin tidak memiliki akses ke infrastruktur yang dibutuhkan. Dan saran untuk peneliti selanjutnya adalah memperluas <i>dataset</i> agar menghasilkan akurasi yang lebih baik. | Penelitian ini dan penelitian yang diajukan memiliki beberapa persamaan, dari mencari parameter yang optimal menggunakan Bayesian Optimization, tetapi beda dengan model yang digunakan.                                                                     |

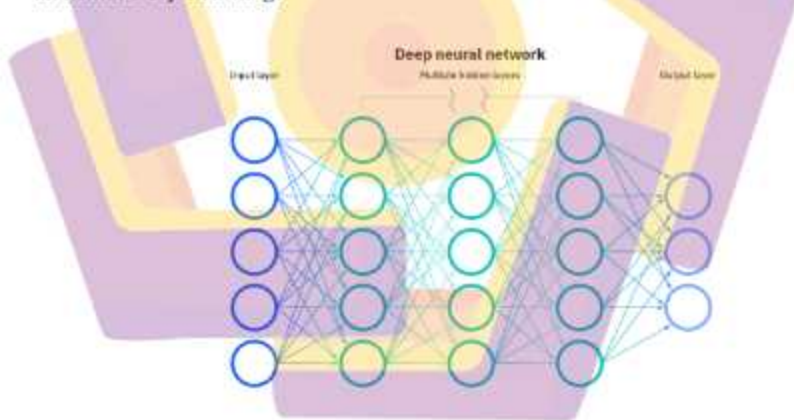
| No | Judul                                                                                      | Peneliti, Media Publikasi, dan Tahun | Tujuan Penelitian                                                                                                                                                                                                                                                                                                                                     | Kesimpulan                                                                                                                                                                                                                                                                                                                                         | Saran atau Kelemahan                                                                                                                                                                        | Perbandingan                                                                                                                                                                                                                                                                                                                                                                                               |
|----|--------------------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4  | Recognition of Mustard Plant Diseases Based on Improved Deep Convolutional Neural Networks | (Hridoy et al., 2022) IEEE           | Tujuan dari penelitian ini adalah untuk mengembangkan metode yang lebih efisien dalam mengenali penyakit tanaman sawi menggunakan CNN. Dengan membangun <i>dataset</i> khusus dari gambar daun, batang, dan polong tanaman sawi, studi ini memperkenalkan model baru bernama MPNet yang dirancang untuk meningkatkan akurasi diagnosis penyakit sawi. | Kelemahan dari penelitian ini adalah kinerja model yang sedikit kurang optimal pada kelas penyakit tertentu, seperti <i>leaf spot</i> , yang disebabkan oleh kesamaan fitur antar gejala penyakit. Selain itu, <i>dataset</i> yang digunakan terbatas hanya pada sembilan kelas penyakit, sehingga model belum diuji pada kondisi yang lebih luas. | Saran untuk penelitian selanjutnya adalah memperluas <i>dataset</i> yang akan digunakan, agar dapat meningkatkan kemampuan generalisasi model dan penerapannya pada skala yang lebih besar. | Penelitian ini memiliki tujuan yang sama dengan penelitian yang diajukan yaitu mencari metode yang paling efisien untuk klasifikasi pada daun sawi. Perbedaannya hanya metode yang digunakan, pada penelitian ini menggunakan model baru yaitu MPNet, sedangkan penelitian yang diajukan ini juga membandingkan model deep yang dibawa VGG19 dan deepwise separable convolution yang dibawa oleh Xception. |
| 5  | Leaf disease identification and classification using optimized deep learning               | (Abd Algani et al., 2023) Elsevier   | Tujuan penelitian ini adalah untuk mengembangkan metode baru yang lebih akurat dan efisien dalam mendeteksi serta                                                                                                                                                                                                                                     | Kesimpulan dari penelitian ini menunjukkan bahwa metode ACO-CNN memberikan tingkat akurasi yang sangat tinggi (99,98%) dalam                                                                                                                                                                                                                       | Kelemahan dari penelitian ini adalah kebutuhan akan <i>dataset</i> yang besar dan berkualitas tinggi untuk pelatihan, serta kompleksitas                                                    | Penelitian ini mengembangkan metode baru yaitu <b>Ant Colony Optimization (ACO)</b> dan <b>Convolutional Neural Network (CNN)</b> . Sedangkan penelitian yang diajukan akan                                                                                                                                                                                                                                |

| No | Judul                                                                                                       | Peneliti, Media Publikasi, dan Tahun   | Tujuan Penelitian                                                                                                                                                                                                                      | Kesimpulan                                                                                                                                                                                       | Saran atau Kelemahan                                                                                                                                                                                                                                                  | Perbandingan                                                                                                                                                                                                                                                                                                                        |
|----|-------------------------------------------------------------------------------------------------------------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                                                                             |                                        | mengklasifikasikan penyakit pada daun tanaman menggunakan kombinasi Ant Colony Optimization (ACO) dan Convolutional Neural Network (CNN).                                                                                              | mengidentifikasi penyakit daun dibandingkan metode lain, dan dapat diandalkan untuk berbagai kondisi lingkungan.                                                                                 | perhitungan yang lebih tinggi. Keterbatasan ini juga meliputi ketergantungan pada kualitas gambar daun, yang mungkin tidak selalu ideal dalam situasi nyata di lapangan.                                                                                              | menggunakan metode CNN yang sudah ada, tanpa mengembangkan metode baru.                                                                                                                                                                                                                                                             |
| 6  | Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG19 | (Paymode & Malode, 2022) ScienceDirect | Tujuan dari penelitian ini untuk mengembangkan model deteksi penyakit daun tanaman multikultur, khususnya pada tanaman tomat dan anggur, dengan menggunakan Convolutional Neural Network (CNN) Pembelajaran Transfer pada model VGG19. | Kesimpulan dari penelitian ini menunjukkan bahwa model CNN yang digunakan mampu mencapai akurasi yang sangat tinggi, yaitu 98,40% untuk daun tanaman anggur dan 95,71% untuk daun tanaman tomat. | Kelemahan dari penelitian ini adalah penggunaan dataset PlantVillage, yang gambar-gambarnya diambil dalam kondisi laboratorium, sehingga model mungkin kurang optimal saat diaplikasikan pada gambar yang diambil dari lapangan dengan kondisi yang lebih bervariasi. | Kedua penelitian sama-sama melakukan klasifikasi citra daun, namun berbeda pada objek dan eksplorasi parameter. Penelitian ini mengkaji penyakit daun tomat dan anggur, sedangkan penelitian yang diajukan berfokus pada daun sawi dengan eksplorasi parameter yang lebih mendalam meskipun menggunakan arsitektur VGG19 yang sama. |

## 2.3. Landasan Teori

### 2.3.1. Deep Learning

*Deep learning* merupakan salah satu cabang dari *machine learning* yang menggunakan artificial neural network untuk meniru cara berpikir manusia dan terdiri dari beberapa layers. *Deep learning* sangat efektif dalam menangani data besar dan kompleks seperti gambar, suara, dan teks. *Deep learning* ini terdiri dari banyak lapisan yang terdiri dari lapisan *input* dan *output*. Setiap lapisan ini nantinya akan bertugas untuk mengekstraksi fitur dari data. Dan salah satu keunggulannya adalah kemampuannya untuk melakukan *feature learning* otomatis dari mentah tanpa memerlukan fitur manual seperti metode *machine learning* tradisional (Sharma & Chaudhary, 2023). Gambar 2.1 akan memberikan Gambaran dari arsitektur *Deep Learning*.



Gambar 2. 1 Arsitektur Deep Learning

Sederhananya, *deep learning* merupakan teknik di dalam *machine learning* yang menggunakan jaringan syaraf tiruan untuk memproses data kompleks melalui serangkaian lapisan, memungkinkan pemahaman yang lebih baik tentang informasi yang terkandung dalam data tersebut.

### 2.3.2. Convolutional Neural Network

*Convolutional Neural Network (CNN)* merupakan salah jenis algoritma dari *deep learning* yang paling sering digunakan untuk mengklasifikasi gambar. CNN sangat efektif untuk pengenalan objek dalam gambar atau video. CNN bekerja dengan sejumlah lapisan untuk mengenali pola-pola visual dari sebuah objek (Taye, 2023). CNN terdiri dari beberapa jenis lapisan utama yang bekerja secara beruntun untuk mengubah input gambar mentah menjadi output klasifikasi yang kita inginkan. Berikut komponen yang digunakan pada CNN :

#### a. Convolutional Layer

*Convolutional Layer* adalah inti dari Convolutional Neural Network (CNN) yang berfungsi untuk mengekstrak fitur dari input, seperti gambar. Lapisan ini bekerja dengan menerapkan filter atau kernel, yaitu matriks kecil seperti 3x3 atau 5x5, pada input data. Filter ini bergeser di sepanjang input dengan langkah tertentu (*stride*) untuk menghasilkan peta fitur (*feature map*). Operasi ini melibatkan perkalian elemen matriks filter dengan elemen input pada posisi tertentu, diikuti dengan penjumlahan hasilnya. Melalui proses ini, convolutional layer dapat menangkap pola atau fitur lokal seperti tepi, tekstur, atau bentuk dalam gambar, sekaligus mempertahankan hubungan spasial antar piksel (AYENI, 2022). Parameter penting pada lapisan ini mencakup ukuran filter, langkah pergeseran (*stride*), dan padding yang berfungsi menjaga ukuran peta fitur dengan menambahkan margin di tepi input.

#### b. Activation Function

*Activation Function* digunakan untuk menambahkan non-linearitas pada model sehingga mampu mempelajari hubungan yang kompleks dalam data. Fungsi

ini diaplikasikan setelah operasi pada setiap neuron untuk menentukan apakah neuron tersebut akan diaktifkan atau tidak (AYENI, 2022). Contoh activation function yang umum digunakan meliputi ReLU (Rectified Linear Unit) yang menggantikan nilai negatif dengan nol untuk mencegah masalah vanishing gradient, sigmoid untuk menghasilkan output probabilitas antara 0 dan 1, softmax untuk menghasilkan distribusi probabilitas dalam klasifikasi multi-kelas, dan tanh yang memetakan nilai ke rentang  $[-1,1]$ . Penambahan activation function ini memungkinkan jaringan mengenali pola non-linear dan meningkatkan kemampuan model dalam menyelesaikan masalah yang kompleks.

#### c. Pooling Layer

*Pooling Layer* bertujuan untuk mengurangi dimensi peta fitur, yang membantu mempercepat perhitungan dan mengurangi risiko overfitting. Proses pooling dilakukan dengan memilih nilai tertentu dari area input, seperti mengambil nilai maksimum dalam max pooling atau menghitung rata-rata dalam average pooling. Ukuran area pooling biasanya  $2 \times 2$  atau  $3 \times 3$ , dan hasilnya adalah peta fitur yang lebih kecil tetapi tetap mempertahankan informasi penting. Dengan mereduksi ukuran peta fitur, pooling layer membuat model lebih efisien tanpa kehilangan fitur dominan dari input, seperti pola atau bentuk yang signifikan (AYENI, 2022).

#### d. Fully Connected Layer

*Fully Connected Layer* adalah lapisan terakhir dalam CNN yang menghubungkan semua neuron dari lapisan sebelumnya ke setiap neuron di lapisan berikutnya. Data dari peta fitur hasil pooling diratakan menjadi vektor (flattening) sebelum dimasukkan ke dalam lapisan *fully connected*. Pada tahap ini, jaringan

mengintegrasikan semua fitur yang telah diekstrak selama proses sebelumnya untuk menghasilkan output. *Fully connected layer* biasanya digunakan untuk membuat prediksi, seperti klasifikasi gambar, di mana jumlah neuron pada lapisan output sesuai dengan jumlah kelas dalam dataset. Aktivasi softmax sering digunakan di lapisan ini untuk menghasilkan probabilitas yang mengindikasikan keyakinan model terhadap setiap kelas (AYENI, 2022).

Berbeda dengan jaringan saraf tradisional (ANN), CNN menggunakan arsitektur yang dirancang khusus untuk menangani data gambar. ANN menghubungkan setiap neuron dari satu lapisan ke lapisan berikutnya (fully connected), sementara CNN menggunakan lapisan konvolusi untuk memproses bagian gambar secara lokal dan lebih efisien dalam mengenali pola visual. Secara keseluruhan, CNN adalah salah satu metode deep learning yang paling populer dan sukses untuk aplikasi berbasis visual, dengan kemampuan untuk belajar dari data gambar dengan cara yang sangat efisien dan akurat.

### 2.3.3. VGG19

VGG (Visual Geometry Group) merupakan salah satu arsitektur *deep learning* yang populer dan banyak digunakan dalam Convolutional Neural Network (CNN) untuk tugas pengenalan dan klasifikasi citra. Arsitektur ini dikembangkan dengan tujuan meningkatkan kemampuan CNN dalam mengekstraksi fitur visual melalui struktur jaringan yang lebih dalam dan teratur. Arsitektur VGG19 secara khusus memperkenalkan paradigma baru yang menekankan bahwa kedalaman jaringan merupakan kunci utama dalam meningkatkan performa pengenalan citra. Inovasi fundamentalnya terletak pada penggunaan filter konvolusi berukuran kecil, yaitu 3x3, yang diterapkan secara konsisten di seluruh lapisan dengan *stride* 1.

Dengan menumpuk beberapa lapisan konvolusi kecil ini, VGG19 mampu meniru efek dari filter yang lebih besar (seperti 7x7) namun dengan jumlah parameter yang lebih efisien dan penambahan fungsi aktivasi non-linear (ReLU) di setiap tahap, sehingga model dapat mempelajari fitur visual yang jauh lebih kompleks dan abstrak melalui hierarki yang lebih dalam (Arge & Mage, 2015).

Secara teknis, VGG19 terdiri dari 19 lapisan berbobot yang mencakup 16 lapisan konvolusi dan 3 lapisan *fully-connected*. Struktur ini disusun dalam blok-blok konvolusi yang diakhiri oleh lapisan *max-pooling* berukuran 2x2 untuk mengurangi dimensi spasial secara bertahap. Meskipun model ini memiliki jumlah parameter yang sangat besar—sekitar 144 juta—desainnya yang seragam dan linear membuktikan bahwa peningkatan kedalaman hingga 19 lapisan secara signifikan meningkatkan akurasi pada dataset skala besar seperti ImageNet, menjadikannya fondasi penting dalam perkembangan teknologi *Convolutional Neural Networks* (CNN) modern.

#### 2.3.4. Xception

Xception merupakan salah satu arsitektur *deep learning* dalam Convolutional Neural Network (CNN) yang dikembangkan untuk meningkatkan efisiensi dan performa pada tugas pengenalan citra. Arsitektur ini diperkenalkan oleh François Chollet sebagai pengembangan dari arsitektur Inception dengan pendekatan yang lebih ekstrem dalam memisahkan proses ekstraksi fitur. Xception didasarkan pada asumsi bahwa pemodelan korelasi spasial dan korelasi antar-kanal pada citra dapat dipisahkan secara lebih efektif (Google, 2014). Oleh karena itu, Xception menggantikan *standard convolution* dengan *depthwise separable*

convolution, yang membagi proses konvolusi menjadi *depthwise convolution* dan *pointwise convolution*.

Penerapan *depthwise separable convolution* memungkinkan Xception memiliki jumlah parameter yang lebih sedikit serta biaya komputasi yang lebih rendah dibandingkan CNN konvensional, tanpa mengorbankan akurasi model. Selain itu, penggunaan *residual connection* pada arsitektur Xception membantu menjaga stabilitas proses pelatihan pada jaringan yang dalam. Berdasarkan berbagai penelitian, arsitektur Xception terbukti mampu menghasilkan performa klasifikasi yang lebih baik dibandingkan beberapa arsitektur CNN sebelumnya, sehingga banyak digunakan dalam penelitian pengenalan citra sebagai model yang efisien dan andal.

#### **2.3.5. Dataset**

*Dataset* adalah sekumpulan data yang biasanya digunakan untuk penelitian, analisis, ataupun pemodelan. Dalam konteks *machine learning* atau *deep learning* *dataset* merupakan sekumpulan data yang digunakan untuk melatih, menguji, dan memvalidasi model dari sebuah algoritma (Yu et al., 2024). Dataset sendiri terdiri dari beberapa jenis, yaitu :

- a. Dataset Numerik merupakan Data ini terdiri dari angka dan bisa berupa bilangan bulat (integer) atau bilangan pecahan (float). Contohnya adalah harga, umur, jumlah barang, dll
- b. Dataset Kategorikal adalah data yang dibagi ke dalam beberapa kategori atau kelompok. Misalnya, warna mobil bisa berupa "merah", "biru", "hijau", dll.

- c. Data teks merupakan dataset bisa terdiri dari data teks seperti kalimat atau dokumen. Data ini memerlukan pra-pemrosesan khusus sebelum digunakan dalam model.
- d. Dataset Gambar merupakan dataset yang biasanya terdiri dari gambar. Gambar-gambar ini direpresentasikan dalam bentuk matriks piksel yang digunakan oleh model untuk belajar pola visual.

Kesimpulannya *dataset* adalah komponen fundamental dalam *machine learning* dan *deep learning* yang terdiri dari kumpulan data yang disusun secara sistematis untuk keperluan analisis dan pembelajaran. Dengan adanya *dataset* yang bersih, lengkap, dan representatif, model machine learning dapat dilatih untuk membuat prediksi yang akurat dan andal.

#### **2.3.6. Data Preprocessing**

Pada data mentah, terdapat banyak permasalahan yang dapat mengganggu proses pengolahan data, seperti *missing value*, data *redundant*, *outliers*, atau ketidaksesuaian format data dengan sistem. Maka dari itu, peneliti melakukan proses *Data processing*. *Data processing* merupakan sebuah proses persiapan untuk mengolah data mentah menjadi data yang siap digunakan. Biasanya *preprocessing* bertujuan untuk memastikan kualitas data, menghilangkan kecacatan, dan mengatasi masalah yang mungkin akan timbul jika tidak dilakukan *preprocessing*. *Preprocessing* biasanya dilakukan karena data mentah sering tidak lengkap dan tidak konsisten (Mehrish et al., 2023). Beberapa hal yang dilakukan dalam proses *preprocessing data*, antara lain dengan membersihkan data, menangani nilai yang hilang, menormalkan atau menskalakan fitur. Proses *preprocessing data* dapat

bervariasi tergantung pada jenis data, tujuan analisis, dan alat yang digunakan. Ada 3 preprocessing yang dilakukan pada penelitian ini, yaitu :

*a. Rescaling*

Rescaling dilakukan untuk mengatur ulang nilai piksel gambar sehingga berada dalam rentang tertentu, biasanya 0–1. Hal ini membantu model CNN memproses data lebih konsisten dan efisien, karena nilai input berada dalam skala yang seragam (Bian, 2024).

*b. Resizing*

Resizing merupakan sebuah proses untuk mengubah ukuran gambar menjadi standar tertentu, contoh yang sering digunakan adalah mengubahnya menjadi 256x256 piksel. Langkah ini memastikan bahwa semua data memiliki dimensi yang seragam, yang penting untuk arsitektur CNN yang membutuhkan input berukuran tetap (Bian, 2024).

*c. Batching*

Batching mengacu pada pembagian dataset menjadi kelompok kecil atau batch untuk diproses secara bertahap. Biasanya gambar dibagi ke dalam batch yang masing-masing terdiri dari beberapa gambar per batch (Bian, 2024). Proses ini dapat mempercepat pelatihan, mengoptimalkan penggunaan memori, dan mempermudah pembaruan bobot model selama propagasi balik di setiap iterasi.

### **2.3.7. Data Augmentation**

*Data augmentation* adalah bagian penting dari ekosistem pembelajaran mesin modern, terutama dalam pembelajaran mendalam. Dengan metode yang tepat, ini akan membantu meningkatkan kemampuan model dalam memahami variasi data yang lebih luas dan meningkatkan performa keseluruhan model dalam

tugas-tugas dunia nyata (Mumuni & Mumuni, 2022). Tujuan dari *data augmentation* adalah untuk meningkatkan volume, kualitas, dan keberagaman data pelatihan sehingga menghasilkan kinerja yang lebih baik untuk pembelajaran mesin. Pada penelitian ini, peneliti menggunakan beberapa proses augmentation data, yaitu :

*a. Rotation*

Rotation merupakan teknik yang melibatkan rotasi gambar secara acak dengan sudut tertentu. Teknik ini dapat membantu model untuk menjadi lebih akurat terhadap data yang diberikan. Dengan melakukan rotasi, gambar yang miring ke satu arah dapat dimodifikasi ke arah lain, menciptakan lebih banyak variasi data yang tetap relevan dengan tugas klasifikasi (Bian, 2024).

*b. Zooming*

Zooming adalah teknik augmentasi data yang melibatkan proses memperbesar (zoom in) atau memperkecil (zoom out) gambar untuk menciptakan variasi data tanpa mengubah label. Zoom in digunakan untuk fokus pada detail tertentu dalam gambar, sedangkan zoom out memberikan pandangan lebih luas dengan menampilkan konteks keseluruhan. Teknik ini membantu model dalam menangkap pola pada berbagai skala, sehingga meningkatkan generalisasi model. Dalam aplikasi, zooming efektif untuk menangani objek yang berukuran kecil atau kompleks, seperti pada klasifikasi gambar medis atau pengenalan pola yang memerlukan analisis multi-skala (Mumuni & Mumuni, 2022).

c. *Flipping*

Flipping digunakan untuk membalik gambar secara acak, baik secara horizontal maupun vertikal. Tujuannya adalah untuk meningkatkan keragaman data pelatihan sehingga model menjadi lebih tangguh terhadap perubahan orientasi gambar (Bian, 2024). Misalnya, data yang diberikan menghadap ke kanan pada gambar asli dapat diubah menghadap ke kiri melalui *flipping horizontal*, sehingga model tetap dapat mengenali fitur yang relevan tanpa tergantung pada arah tertentu.

**2.3.8. Learning Rate**

Learning rate adalah parameter penting yang mengontrol seberapa besar pembaruan yang dilakukan terhadap bobot model pada setiap iterasi. Learning rate menentukan ukuran langkah yang diambil saat proses pembelajaran. Jika diibaratkan sebagai perjalanan, learning rate adalah panjang langkah kaki dalam menuju solusi optimal. Secara garis besar learning rate digunakan untuk mengatur seberapa cepat atau lambat model belajar dan mengonvergensi terhadap solusi optimal (Tong et al., 2022).

**2.3.9. Dropout**

Dropout adalah teknik regularisasi yang digunakan dalam pelatihan jaringan saraf tiruan, termasuk *Convolutional Neural Networks (CNN)*, yang bertujuan untuk mengurangi overfitting. Overfitting terjadi ketika model terlalu cocok dengan data pelatihan, sehingga gagal menangani data baru yang belum pernah dilihat sebelumnya (Salehin & Kang, 2023). Dropout bekerja dengan cara memilih secara acak beberapa neuron pada lapisan tertentu dan "membuangnya" sementara waktu selama proses pelatihan. Neuron-neuron ini tidak digunakan baik

pada proses forward pass (perhitungan ke depan) maupun backpropagation (proses pembaruan bobot). Hasilnya, jaringan saraf tidak dapat bergantung pada sejumlah kecil neuron untuk membuat prediksi, sehingga dipaksa untuk belajar pola yang lebih general dan tidak spesifik terhadap data pelatihan. Dalam CNN, yang dikenal efektif untuk tugas pengenalan gambar dan pemrosesan visual, *dropout* umumnya diterapkan pada lapisan *fully-connected (dense layers)*. Lapisan-lapisan ini lebih rentan terhadap *overfitting* dibandingkan dengan lapisan konvolusi. Dengan mematikan neuron secara acak pada lapisan ini, *dropout* membantu mengurangi kemungkinan model terlalu menghafal fitur-fitur tertentu dari data pelatihan. Saat model digunakan pada fase inferensi, semua neuron diaktifkan kembali, tetapi output dari masing-masing neuron akan disesuaikan berdasarkan probabilitas *dropout* yang digunakan selama pelatihan. Teknik ini juga meningkatkan kemampuan CNN dalam menangani data yang bervariasi dengan mendorong pembelajaran yang lebih robust.

### 2.3.10. Bayesian Optimization

Bayesian Optimization (BO) adalah strategi optimasi global yang dirancang untuk mencari nilai optimal dari suatu fungsi *black-box* yang mahal untuk dievaluasi dan tidak memiliki informasi gradien eksplisit. Pendekatan ini bekerja secara iteratif dengan menggabungkan dua komponen utama: *surrogate model* (umumnya *Gaussian Process*) yang berfungsi membangun model probabilistik dari fungsi objektif berdasarkan data observasi sebelumnya, serta *acquisition function* yang bertugas mengarahkan pencarian titik optimal berikutnya melalui penyeimbangan antara eksplorasi (*exploration*) pada area dengan ketidakpastian tinggi dan eksploitasi (*exploitation*) pada area yang menjanjikan (Tibaldi et al.,

2023). Karakteristik utama BO adalah efisiensi sampelnya yang tinggi, yang memungkinkan dapat menemukan solusi optimal dalam jumlah iterasi yang terbatas serta memberikan ketahanan yang unggul terhadap gangguan atau *noise* pada data observasi.

#### 2.3.11. Confussion Matrix

*Confussion matrix* adalah alat analisis yang umum digunakan dalam bidang pembelajaran mesin *machine learning* untuk mengevaluasi kinerja sebuah model klasifikasi. Ini adalah matriks dua dimensi yang merangkum hasil prediksi dari sebuah classifier dengan membandingkan label sebenarnya dengan label prediksi (Heydarian et al., 2022). **Confusion Matrix** menunjukkan jumlah prediksi yang benar dan salah dalam berbagai kategori, dan secara umum digunakan dalam klasifikasi biner dan multi-kelas. Matriks ini terdiri dari empat elemen dasar :

- a. **True Positives (TP)**: Jumlah instance yang benar-benar positif dan diprediksi positif.
- b. **True Negatives (TN)**: Jumlah instance yang benar-benar negatif dan diprediksi negatif.
- c. **False Positives (FP)**: Jumlah instance yang sebenarnya negatif tetapi diprediksi sebagai positif (sering disebut sebagai "false alarm").
- d. **False Negatives (FN)**: Jumlah instance yang sebenarnya positif tetapi diprediksi sebagai negatif (missed detection).

## BAB III

### METODE PENELITIAN

#### 3.1. Jenis, Sifat dan Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan sifat eksperimental-komparatif untuk menganalisis kinerja dua arsitektur Convolutional Neural Network (CNN) dalam klasifikasi daun sawi sehat dan terinfeksi hama. Pendekatan ini bersifat eksperimental karena melibatkan pengujian sistematis guna menemukan konfigurasi hyperparameter optimal melalui Bayesian Optimization, yang efisien dalam menentukan nilai terbaik untuk *learning rate*, *dropout rate*, dan jumlah *filter*. Penelitian ini menggunakan pendekatan komparatif dengan membandingkan dua arsitektur yang memiliki filosofi desain berbeda secara fundamental, yaitu VGG19 dan Xception. Pemilihan VGG19 didasarkan pada karakteristiknya sebagai model *deep* klasik yang mengandalkan tumpukan konvolusi linear dengan filter berukuran kecil (3 x 3). Arsitektur ini sangat efektif sebagai *baseline* dalam mengekstraksi fitur spasial dan tekstur yang sangat detail, yang krusial untuk mengidentifikasi pola penyakit pada permukaan daun sawi. Meskipun memiliki parameter yang padat, VGG19 memberikan gambaran performa dari paradigma jaringan *sequential* murni dalam menangani data citra pertanian.

Sebagai perbandingan, Xception dipilih untuk mewakili arsitektur modern yang mengedepankan efisiensi melalui mekanisme *depthwise separable convolution*. Berbeda dengan VGG19, Xception memisahkan pembelajaran korelasi spasial dan korelasi saluran (*cross-channel correlation*), sehingga

mampu menangkap fitur kompleks dengan beban komputasi yang lebih ringan. Dengan membandingkan kedua model ini, penelitian tidak hanya sekedar mencari akurasi tertinggi, tetapi juga bertujuan untuk menganalisis *trade-off* antara intensitas ekstraksi fitur (VGG19) dan efisiensi pemetaan fitur (Xception). Fokus utama analisis ini adalah menentukan apakah karakteristik morfologi penyakit daun sawi lebih baik dipetakan melalui kedalaman konvolusi tradisional atau melalui efisiensi modular, guna memberikan rekomendasi model yang paling optimal untuk diimplementasikan pada lingkungan dengan keterbatasan sumber daya komputasi (*edge devices*).

### 3.2. Metode Pengumpulan Data

Sumber data yang dibutuhkan pada penelitian ini didapatkan dari dua metode pengumpulan data sebagai berikut:

#### 3.2.1. Literature Review

Penelitian ini melibatkan pengumpulan data dengan analisis *research paper* sebelumnya yang memiliki keterkaitan dengan topik yang diangkat khususnya klasifikasi penyakit pada tanaman. Data yang didapatkan dapat berupa data publik yang tersedia pada media publikasi, *repository* publik dan lain sebagainya

### 3.2.2. *Secondary data analysis*

Ini melibatkan penggunaan data yang telah dikumpulkan oleh peneliti-peneliti sebelumnya. Data yang didapatkan dapat berupa arsip yang tersedia secara publik dan diperbolehkan untuk digunakan, biasanya terdapat pada *repository GITHUB, UCI Machine Learning, Kaggle* dan lain sebagainya.

Untuk melakukan eksperimen pada penelitian ini, peneliti menggunakan dataset yang tersedia secara *public* di *Kaggle*.

### 3.3. **Metode Analisis Data**

Untuk mencapai tujuan penelitian dan membuktikan hasilnya, maka dilakukan serangkaian tahapan sistematis yang dimulai dari Pengumpulan Data sekunder dan dilanjutkan dengan Preprocessing Data (termasuk *Rescaling, Resizing, dan Batching*) dan Data Augmentation. Dataset kemudian dipecah melalui Splitting Dataset menjadi data pelatihan, validasi, dan pengujian. Inti dari metodologi ini adalah Hyperparameter Optimization menggunakan Bayesian Optimization, yang diterapkan pada *custom classification head* dari dua arsitektur yaitu, VGG19 dan Xception. Tahapan ini bertujuan untuk menemukan model yang paling optimal dari kedua arsitektur tersebut, yang kemudian dilanjutkan dengan Implementasi Model Komparatif dan Evaluasi Kinerja Kritis. Evaluasi tidak hanya berfokus pada akurasi klasifikasi, tetapi juga pada efisiensi komputasi (waktu inferensi dan jumlah parameter) untuk menentukan trade-off optimal dan memberikan rekomendasi implementasi yang didukung data.

Analisis data dalam penelitian ini dirancang untuk mengevaluasi dampak kinerja parameter pada arsitektur Xception dan VGG19 dalam klasifikasi daun sawi sehat dan terkena hama. Tahapan pertama adalah Preprocessing Data, yang mencakup rescaling (normalisasi nilai piksel dari 0-255 menjadi 0-1 menggunakan  $\text{rescale}=1./255$ ) untuk mempercepat pelatihan dan meningkatkan stabilitas jaringan saraf. Kemudian, gambar diubah ukurannya menjadi dimensi yang seragam (resizing), yaitu (128, 128) piksel, karena model CNN memerlukan input dengan ukuran yang konsisten. Proses dilanjutkan dengan batching, yaitu pembagian data menjadi kelompok-kelompok kecil. Selanjutnya, dilakukan pembuatan model dengan menggunakan dua arsitektur CNN, Xception dan VGG19, untuk mencari konfigurasi *hyperparameter* yang optimal. Proses optimasi dilakukan secara sistematis menggunakan Bayesian Optimization. Parameter yang dicari adalah *learning rate*, *dropout rate*, dan *jumlah filter*, dengan rentang pencarian yang telah didefinisikan secara spesifik: *learning\_rate*: (1e-4, 1e-2), *dropout\_rate*: (0.1, 0.5), dan *filters*: (16, 64). Tujuan utama dari analisis data ini adalah untuk membandingkan kedua arsitektur yang telah dioptimasi tersebut, tidak hanya berdasarkan akurasi klasifikasi, tetapi juga berdasarkan efisiensi komputasi (waktu inferensi dan jumlah parameter) untuk menentukan *trade-off* terbaik dalam sistem klasifikasi.

Tahapan selanjutnya adalah Pembagian Dataset. Dataset yang telah melalui proses *preprocessing* akan dibagi menjadi tiga subset utama: 70% untuk data pelatihan (*training set*), 10% untuk data validasi (*validation set*), dan 20% untuk data pengujian (*testing set*). Pembagian ini bertujuan untuk memastikan

bahwa model yang dilatih dapat mempelajari pola dengan baik dan memiliki generalisasi yang kuat pada data baru.

Setelah pembagian data, proses Hyperparameter Optimization dilakukan menggunakan Bayesian Optimization pada *training set* dan *validation set*. Data validasi berfungsi krusial sebagai penilai independen selama optimasi untuk mencari kombinasi *hyperparameter* terbaik (*optimal*) pada kedua arsitektur (Xception dan VGG19) sambil mencegah *overfitting*. Setelah konfigurasi *hyperparameter* optimal untuk VGG19 dan Xception ditemukan, proses klasifikasi akhir dilakukan. Pada tahap ini, model yang telah dioptimasi mengambil input data dari data pengujian (*testing set*) data yang benar-benar baru dan mengklasifikasikannya ke dalam kelas sehat atau terkena hama, untuk menilai kinerja model yang sebenarnya.

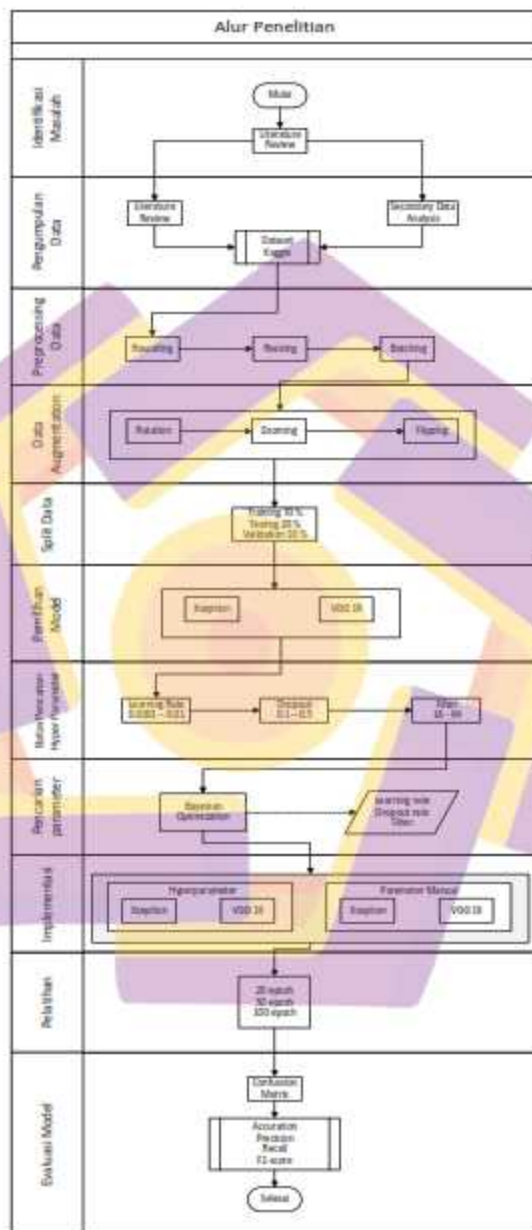
Setelah proses pelatihan dan optimasi selesai, kinerja model diuji secara menyeluruh menggunakan dataset pengujian (*testing set*), yaitu data yang belum pernah dilihat model sebelumnya, untuk menilai kemampuan generalisasi yang sebenarnya. Evaluasi ini tidak hanya menghitung akurasi klasifikasi yang dicapai, tetapi juga mengukur metrik kinerja kritis seperti waktu inferensi dan jumlah parameter model untuk mendukung analisis efisiensi komputasi. Hasil pengujian ini dihitung berdasarkan kinerja model terhadap data uji. Selain itu, Confusion Matrix digunakan untuk mendapatkan gambaran lebih rinci tentang kesalahan prediksi model, di mana setiap elemen dalam matriks menunjukkan jumlah prediksi yang benar atau salah untuk masing-masing kelas, memberikan

wawasan yang jelas tentang kesalahan klasifikasi yang sering dilakukan oleh model.

#### **3.4. Alur Penelitian**

Untuk menyelesaikan masalah pada penelitian ini dibutuhkan langkah-langkah yang harus diselesaikan. Adapun langkah-langkah atau tahapan yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 3.1 yang dipaparkan sebagai berikut:





Gambar 3. 1 Alur penelitian

### 3.4.1. Tahapan Pengumpulan Data

Seperti yang sudah dijelaskan pada Metode Pengumpulan Data, penelitian ini menggunakan data publik yang telah digunakan dalam penelitian sebelumnya. Pendekatan ini lebih dipilih untuk menghemat waktu karena tidak memerlukan pelabelan manual maupun keterlibatan pakar. Selain itu, penggunaan data publik ini juga membantu memvalidasi hasil penelitian, karena data tersebut telah terbukti relevan dan dapat diandalkan dalam konteks penelitian serupa.

### 3.4.2. Preprocessing Data

Tahapan ini dilakukan dengan tujuan agar data yang digunakan dalam model menjadi lebih bersih, terstruktur, dan siap diolah, dengan tujuan memaksimalkan hasil dari klasifikasi daun Sawi sehat dan terkena Hama. Telah dijelaskan pada bagian "Metode Analisis Data" proses ini melibatkan *Rescaling* (normalisasi), *Resizing* (mengubah ukuran), *Batching* (Pembagian dalam Batch).

Berikut penjelasan dari langkah-langkah pada *preprocessing*:

#### 3.4.2.1. Rescaling

Salah satu tahap awal dalam praproses citra adalah normalisasi nilai piksel melalui teknik *rescaling*. Teknik ini bertujuan untuk mengubah nilai piksel dari rentang 0–255 menjadi rentang yang lebih kecil, seperti 0–1 atau -1 sampai 1. Hal ini penting karena sebagian besar model pembelajaran mendalam *deep learning* menunjukkan performa yang lebih stabil dan efisien saat menerima input dalam skala yang lebih kecil. Penyesuaian nilai piksel ini dapat membantu mempercepat proses

pelatihan, mengurangi risiko kesalahan numerik, dan meningkatkan kemampuan generalisasi model terhadap data baru. Selain itu, nilai input yang terstandarisasi juga memungkinkan operasi numerik seperti perkalian matriks berjalan dengan lebih stabil, sehingga mendukung efisiensi dalam pembaruan bobot jaringan.

#### 3.4.2.2. Resizing

Tahap *resizing* ini dilakukan untuk memastikan bahwa seluruh gambar memiliki dimensi yang seragam, misalnya  $224 \times 224$  piksel. Konsistensi ukuran ini sangat penting dalam konteks pemrosesan menggunakan jaringan saraf konvolusional (CNN), yang memerlukan input dengan ukuran tetap agar operasi seperti konvolusi dan *pooling* dapat dilakukan secara efisien dan konsisten. Selain menjaga keseragaman input, *resizing* juga berkontribusi pada efisiensi penggunaan memori dan kecepatan komputasi. Ukuran citra yang lebih kecil, seperti  $224 \times 224$  piksel, umumnya dipilih karena memberikan keseimbangan antara kualitas fitur visual dan efisiensi pemrosesan. Dengan ukuran yang seragam, model dapat lebih fokus dalam mengenali pola tanpa harus mengatasi variasi dimensi citra.

#### 3.4.2.3. Batching

Tahap *batching* ini merupakan proses pembagian dataset menjadi kelompok-kelompok kecil yang disebut *batch*. Alih-alih memproses seluruh dataset secara sekaligus, model hanya menerima satu batch pada setiap iterasi pelatihan. Pendekatan ini secara signifikan mengurangi

kebutuhan memori, sehingga memungkinkan proses pelatihan dilakukan bahkan pada perangkat dengan kapasitas memori terbatas. Selain efisiensi memori, batching memiliki peran penting dalam algoritma *Stochastic Gradient Descent* (SGD), di mana pembaruan parameter model dilakukan berdasarkan subset data per batch, bukan berdasarkan keseluruhan dataset. Hal ini memungkinkan pembaruan bobot terjadi lebih sering dan mempercepat konvergensi model. Dengan frekuensi pembaruan yang lebih tinggi, proses pelatihan menjadi lebih adaptif terhadap karakteristik data dan dapat menghasilkan model yang lebih optimal dalam waktu yang relatif singkat.

#### 3.4.3. Pembagian Dataset

Pada tahap ini dataset yang telah dipersiapkan akan dibagi menjadi tiga subset utama, yaitu data pelatihan (*training set*), data validasi (*validation set*), dan data pengujian (*testing set*). Tujuan dari pembagian ini adalah untuk memastikan bahwa proses pelatihan dan evaluasi model dilakukan secara terpisah, sehingga hasil evaluasi lebih mencerminkan kinerja sebenarnya dari model dan meminimalkan risiko *overfitting*.

Rencana proporsi pembagian data dalam penelitian ini adalah sekitar 70% untuk data pelatihan, sedangkan sisanya (30%) akan dibagi lagi antara data validasi dan data pengujian. Pembagian data dilakukan secara acak agar distribusi data tetap representatif, dan digunakan pengaturan khusus agar hasil pembagian tersebut dapat direproduksi.

Strategi ini dirancang agar model memiliki data yang cukup untuk belajar dari pola-pola yang ada, divalidasi selama pelatihan untuk menghindari *overfitting*, serta diuji secara independen guna menilai performa akhir model terhadap data yang belum pernah dilihat sebelumnya.

#### 3.4.4. Data Augmentation

Kemudian, untuk meningkatkan kemampuan generalisasi model, penelitian ini menerapkan teknik augmentasi data pada citra pelatihan. Augmentasi bertujuan untuk menghasilkan variasi sintetis dari gambar asli melalui berbagai transformasi, sehingga model dapat mempelajari pola yang lebih beragam tanpa harus menambahkan data baru secara eksplisit. Strategi augmentasi mencakup beberapa bentuk transformasi seperti rotasi, pergeseran posisi horizontal dan vertikal, perubahan skala (zoom), pembalikan gambar secara horizontal, serta pengisian ulang piksel kosong yang muncul akibat transformasi. Rotasi membantu model mengenali objek dalam berbagai orientasi, pergeseran posisi meningkatkan ketahanan model terhadap variasi letak objek, zoom memungkinkan model menangkap objek dalam ukuran yang berbeda; sedangkan pembalikan horizontal memperluas keragaman arah objek yang mungkin muncul dalam data nyata.

Secara keseluruhan, augmentasi dirancang sebagai bagian dari strategi regularisasi model, yang membantu mengurangi risiko *overfitting* dan meningkatkan kemampuan model untuk mengenali pola umum dari berbagai variasi bentuk citra. Dengan memperkaya data latih secara sintetis, model diharapkan mampu melakukan prediksi lebih akurat dan adaptif terhadap data

uji yang bersifat dinamis. Kemudian, untuk meningkatkan kemampuan generalisasi model, penelitian ini menerapkan teknik augmentasi data pada citra pelatihan. Augmentasi bertujuan untuk menghasilkan variasi sintetis dari gambar asli melalui berbagai transformasi, sehingga model dapat mempelajari pola yang lebih beragam tanpa harus menambahkan data baru secara eksplisit.

Secara keseluruhan, augmentasi dirancang sebagai bagian dari strategi regularisasi model, yang membantu mengurangi risiko *overfitting* dan meningkatkan kemampuan model untuk mengenali pola umum dari berbagai variasi bentuk citra. Dengan memperkaya data latih secara sintetis, model diharapkan mampu melakukan prediksi lebih akurat dan adaptif terhadap data uji yang bersifat dinamis.

#### 3.4.5. Model Xception, VGG19 dan pencarian Hyperparameter

Setelah proses di atas selesai dilakukan, tahap selanjutnya adalah pembuatan model untuk mencari kombinasi hyperparameter terbaik, yaitu *learning rate*, *dropout rate*, dan jumlah *filter*. Optimasi dilakukan menggunakan Bayesian Optimization dengan rentang pencarian: *learning rate* antara 0.0001 hingga 0.01, *dropout rate* antara 0.1 hingga 0.5, serta jumlah *filter* antara 16 hingga 64. Setelah rentang tersebut ditentukan, Bayesian Optimization dijalankan untuk menemukan kombinasi hyperparameter yang menghasilkan kinerja model paling optimal. Nilai hyperparameter terbaik yang diperoleh dari hasil optimasi kemudian digunakan untuk melatih model dengan arsitektur Xception dan VGG19. Proses pelatihan dilakukan dengan variasi jumlah *epoch* guna mengeksplorasi performa terbaik dari kedua arsitektur tersebut.

### 3.4.6. Implementasi

Setelah tahapan *preprocessing* dan *augmentation data* selesai, penelitian ini dilanjutkan dengan melakukan percobaan komparatif terhadap dua arsitektur CNN yang memiliki filosofi desain kontras, VGG19 dan Xception. VGG19 dikenal sebagai arsitektur *Deep Klasik* yang menggunakan lapisan konvolusi 3 x 3 yang dalam untuk mengekstrak fitur visual secara detail. Keunggulan ini harus dibayar dengan jumlah parameter yang besar dan biaya komputasi yang tinggi, sehingga menjadikan model ini memerlukan banyak memori dan waktu saat pelatihan dan inferensi. Di sisi lain, Xception mengusung konsep *Depthwise Separable Convolution* (DSC), yang secara efektif memisahkan pemrosesan spasial dan *channel*. Pendekatan DSC ini secara signifikan mengurangi jumlah parameter dan biaya komputasi tanpa mengorbankan akurasi, sehingga Xception menawarkan efisiensi yang lebih baik, terutama dalam hal pemanfaatan memori dan kecepatan komputasi, dibandingkan arsitektur tradisional seperti VGG19.

Untuk menemukan konfigurasi terbaik (optimal) bagi kedua model, dilakukan pencarian *hyperparameter* secara sistematis menggunakan Bayesian Optimization. Metode ini diterapkan untuk mencari nilai optimal pada *learning rate*, *dropout rate*, dan jumlah filter di lapisan *custom classification head*. Ruang pencarian yang telah didefinisikan adalah *learning\_rate* (0.0001 – 0.01), *dropout\_rate* (0.1 - 0.5), dan *filter* (16 - 64), yang bertujuan untuk menemukan kombinasi *hyperparameter* yang memberikan performa validasi terbaik. Setelah *hyperparameter* optimal ditemukan, pelatihan model akhir untuk Xception Optimal dan VGG19 Optimal dilakukan

### 3.4.7. Model Evaluation (*Confusion Matrix*)

Model dievaluasi menggunakan *confusion matrix*. Menggunakan *confusion matrix* untuk mendapatkan pemahaman yang lebih mendalam tentang kesalahan prediksi yang dilakukan model. Setiap elemen dalam *confusion matrix* menunjukkan berapa banyak prediksi yang benar atau salah untuk setiap kelas, sehingga memberikan wawasan yang jelas tentang di mana model sering salah dalam mengklasifikasikan gambar.



## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Pengumpulan data

Penelitian ini menggunakan data dari dua metode pengumpulan data yang dilakukan peneliti. Yang pertama melalui *literature review*, yang mana penelitian ini melibatkan pengumpulan data dengan analisis *research paper* sebelumnya yang memiliki keterkaitan dengan topik yang diangkat khususnya klasifikasi penyakit pada tanaman. Kemudian *Secondary data analysis*, yang dimana metode ini melibatkan penggunaan data yang telah dikumpulkan oleh peneliti-peneliti sebelumnya. Data yang didapatkan dapat berupa arsip yang tersedia secara publik dan diperbolehkan untuk digunakan, biasanya terdapat pada *repository GITHUB*, *Driver*, *Kaggle* dan lain sebagainya.

Untuk melakukan eksperimen pada penelitian kali ini, peneliti menggunakan dataset yang tersedia secara *public* di *Kaggle*. Gambar 4.1 memberikan sample dari *dataset* yang dipilih oleh peneliti.



Gambar 4. 1 Dataset Tanaman Sawi Ada Hama dan Sawi Tanpa Hama

Pada Gambar 4.1 di atas menunjukkan sampel gambar dari dataset yang digunakan dalam penelitian ini, yaitu dataset Caisim Dataset yang diperoleh dari Kaggle. Pemilihan dataset ini merujuk pada penelitian yang dilakukan oleh (Nazalia et al., 2023) yang berjudul *Implementation of Convolutional Neural Network Algorithm to Pest Detection in Caisim*. Dataset ini berisi total 1000 gambar yang menunjukkan daun sawi ada hama dan daun sawi tanpa hama. Dengan menggunakan dataset ini, penelitian bertujuan untuk mengembangkan model deteksi penyakit pada tanaman sawi yang lebih akurat.

Tabel 4. 1 Total gambar dan kelas dataset

| No                | Jenis Daun           | Jumlah |
|-------------------|----------------------|--------|
| 1                 | Daun Sawi Ada Hama   | 500    |
| 2                 | Daun Sawi Tanpa Hama | 500    |
| Total Keseluruhan |                      | 1000   |

Pada tabel 4.1 menampilkan total gambar dan kelas dari dataset yang digunakan dalam penelitian ini, yaitu dataset Caisim yang diperoleh dari Kaggle. Kelas daun sawi ada hama berjumlah 500 dan daun sawi tanpa hama juga ada 500.

#### 4.2. Preprocessing data

Pada tahap preprocessing data, peneliti menerapkan serangkaian proses awal yang bertujuan untuk memastikan bahwa citra masukan memenuhi standar yang diperlukan sebelum digunakan dalam pelatihan model untuk mengidentifikasi tanaman sawi sehat dan terkena hama. Preprocessing merupakan bagian krusial dalam *deep learning*, karena kualitas dan konsistensi data input sangat memengaruhi efektivitas model dalam mengenali pola dan melakukan klasifikasi secara akurat. Tiga langkah utama untuk mempersiapkan citra sebelum digunakan dalam proses pelatihan model untuk mengidentifikasi tanaman sawi sehat dan terkena hama, yaitu *rescaling* (normalisasi nilai piksel), *resizing* (penyeragaman ukuran citra), dan *batching* (pembagian data ke dalam kelompok kecil).

##### a. Rescaling

Citra digital pada umumnya disimpan dengan rentang nilai piksel 0–255. Namun, nilai sebesar ini kurang ideal untuk digunakan langsung pada model pembelajaran mendalam karena dapat memicu ketidakstabilan komputasi dan memperlambat konvergensi. Oleh karena itu, dilakukan normalisasi dengan mengubah nilai piksel menjadi skala 0–1 menggunakan

`ImageDataGenerator(rescale=1./255)`. Proses ini tidak hanya mempercepat proses pelatihan, tetapi juga membantu model melakukan pembaruan bobot secara lebih stabil serta mengurangi risiko kesalahan numerik selama pelatihan.

*b. Resizing*

Model CNN memerlukan input citra dengan dimensi tetap agar operasi konvolusi dan pooling dapat diterapkan secara konsisten di seluruh lapisan. Oleh karena itu, seluruh citra diubah ukurannya menjadi  $224 \times 224$  piksel menggunakan parameter `target_size=(224, 224)` pada `ImageDataGenerator`. Ukuran ini dipilih karena umum digunakan dalam arsitektur CNN standar, dan dianggap sebagai titik kompromi yang baik antara resolusi informasi visual dan efisiensi komputasi. Dengan ukuran input yang seragam, model dapat lebih fokus mengenali pola visual tanpa terganggu oleh variasi dimensi gambar.

*c. Batching*

Untuk mengoptimalkan efisiensi memori dan mempercepat proses pelatihan, data pelatihan dibagi menjadi batch berisi 64 citra per iterasi (`batch_size=64`). Pendekatan ini memungkinkan model membaca dan memproses data dalam kelompok kecil, daripada seluruh dataset sekaligus, yang tidak hanya lebih hemat memori tetapi juga lebih efisien dalam proses pembaruan bobot menggunakan algoritma *Stochastic Gradient Descent*. Dengan frekuensi pembaruan yang lebih sering, model dapat lebih cepat mencapai konvergensi dan menunjukkan performa pelatihan yang lebih stabil.

Ketiga proses *preprocessing* ini dilakukan secara otomatis melalui konfigurasi pada `ImageDataGenerator`, yang mengatur seluruh pipeline input model secara

efisien. Langkah-langkah ini krusial dalam memastikan kualitas data masukan serta kestabilan proses pelatihan pada model CNN yang digunakan.

#### 4.3. Pembagian dataset

Kemudian dataset ini dibagi menjadi tiga subset utama, yaitu 70% untuk data pelatihan (*training set*), 20% untuk data pengujian (*testing set*), dan 10% untuk data validasi (*validation set*). Strategi pembagian ini dirancang untuk memastikan bahwa model memiliki cukup data untuk mempelajari pola secara mendalam, sekaligus dapat dievaluasi dengan data yang tidak pernah digunakan selama pelatihan. Data pelatihan digunakan untuk membangun dan melatih model dalam mengenali pola-pola dari citra daun Sawi yang tersedia. Data validasi berfungsi untuk memantau kinerja model selama pelatihan dan menghindari risiko *overfitting*, sedangkan data pengujian digunakan sebagai evaluasi akhir untuk menilai performa model secara objektif terhadap data baru. Pembagian proporsional 70:20:10 ini dipilih karena dianggap mampu memberikan keseimbangan antara kapasitas pembelajaran model dan akurasi evaluasi. Proporsi ini juga sejalan dengan praktik umum dalam pelatihan model deep learning yang membutuhkan data dalam jumlah besar untuk menghasilkan bobot yang akurat.

Sementara itu, penelitian yang dilakukan oleh (Fraivan & Faouri, 2022) mengevaluasi performa model dengan tiga skenario pembagian data yang berbeda, yakni 50:50, 70:30, dan 90:10 antara data pelatihan dan pengujian. Hasil mereka menunjukkan bahwa peningkatan proporsi data pelatihan secara umum berkontribusi terhadap peningkatan akurasi model, meskipun hal tersebut juga bergantung pada kompleksitas model dan kualitas data. Dengan mengadopsi rasio

70:20:10 dalam penelitian ini, diharapkan model dapat dilatih secara optimal dan tetap dievaluasi secara adil menggunakan data yang belum pernah dilihat sebelumnya.

#### 4.4. Data Augmentation

Pada tahap ini, untuk meningkatkan keragaman data pelatihan dan mengurangi risiko *overfitting*, penelitian ini menerapkan teknik augmentasi data (*data augmentation*) pada citra latih menggunakan objek *ImageDataGenerator*. Teknik ini bertujuan menciptakan variasi citra secara sintesis tanpa perlu menambah jumlah data secara eksplisit. Beberapa transformasi yang digunakan dalam proses ini meliputi rotasi citra hingga 20 derajat untuk menangani variasi orientasi objek, pergeseran posisi horizontal dan vertikal hingga 10% untuk merepresentasikan objek yang tidak selalu berada di posisi tengah, serta zoom hingga 20% untuk memperbesar atau memperkecil citra. Selain itu, diterapkan juga pembalikan horizontal guna memperluas keragaman arah objek dan metode *fill mode='nearest'* untuk mengisi area kosong akibat transformasi secara alami dengan piksel terdekat.

Seluruh augmentasi dilakukan secara real-time selama proses pelatihan, sehingga data yang masuk ke dalam model merupakan hasil transformasi yang bervariasi setiap epoch. Citra hasil augmentasi tetap disesuaikan dalam ukuran  $224 \times 224$  piksel agar sesuai dengan kebutuhan input model CNN. Dengan teknik ini, model memperoleh representasi data yang lebih beragam, yang pada akhirnya berdampak positif terhadap kemampuan generalisasi dan kestabilan pelatihan. Selisih akurasi antara data latih dan validasi menjadi lebih kecil, menunjukkan bahwa augmentasi berhasil membantu mengurangi *overfitting* serta memperkuat

kemampuan model dalam mengenali objek daun sawi dengan kondisi yang bervariasi di dunia nyata.

#### 4.5. Pemilihan Model (VGG19 & Xception) dan Pencarian Hyperparameter

Pada tahap ini peneliti memilih model yang akan dipakai pada penelitian ini.

Setelah model dibuat, tahap selanjutnya adalah proses pencarian *hyperparameter* optimal yang dilakukan melalui dua pendekatan, yaitu secara otomatis menggunakan Bayesian Optimization dan secara manual. Penggunaan Bayesian Optimization bertujuan untuk mengeksplorasi ruang pencarian secara cerdas dan efisien guna menemukan kombinasi parameter yang memberikan performa terbaik. Dalam proses ini, rentang nilai yang didefinisikan untuk *learning rate* adalah ( $1e-4$ ,  $1e-2$ ) yang mendefinisikan bahwa rentang pencarian untuk *hyperparameter learning rate* adalah 0.0001 hingga 0.01, *dropout rate* antara (0.1 – 0.5, serta jumlah filter dalam rentang (16 – 64) Metode ini bekerja dengan membangun model probabilitas untuk memprediksi kombinasi parameter mana yang paling menjanjikan dalam meningkatkan akurasi model pada klasifikasi citra daun sawi.

Sebagai pembandingan terhadap hasil optimasi otomatis, peneliti juga menetapkan skenario konfigurasi secara manual dengan mengacu pada nilai-nilai standar yang umum digunakan dan dinilai stabil dalam literatur *deep learning*. Konfigurasi manual ini menggunakan parameter yang serupa. Perbandingan antara kedua metode ini dilakukan untuk mengevaluasi efektivitas optimasi sistematis dalam menghasilkan model yang lebih unggul. Seluruh kombinasi dari skenario eksperimen tersebut kemudian diterapkan pada arsitektur VGG19 dan Xception, yang selanjutnya proses pelatihan dilakukan dengan variasi jumlah epoch guna

mengeksplorasi performa terbaik dari kedua arsitektur tersebut. Tabel 4.2 dan Tabel 4.3 menunjukkan hasil penggunaan *hyperparameter learning rate, dropout rate, dan jumlah filter*.

Tabel 4. 2 Hasil Pencarian *hyperparameter learning rate, dropout rate, dan jumlah filter* Xception

| No | learning rate | dropout   | filters |
|----|---------------|-----------|---------|
| 1  | 0.0069950     | 0.2144557 | 26.89   |
| 2  | 0.0055580     | 0.3877875 | 36.31   |
| 3  | 0.0098095     | 0.3739318 | 39.08   |
| 4  | 0.0041905     | 0.4802809 | 50.74   |
| 5  | 0.0059165     | 0.1856337 | 59.98   |
| 6  | 0.0082920     | 0.10      | 45.51   |
| 7  | 0.0100000     | 0.10      | 53.48   |
| 8  | 0.0077782     | 0.3943578 | 16.02   |
| 9  | 0.0100000     | 0.50      | 64.00   |
| 10 | 0.0100000     | 0.10      | 30.85   |
| 11 | 0.0001000     | 0.50      | 56.37   |
| 12 | 0.0001000     | 0.50      | 28.98   |
| 13 | 0.0001000     | 0.50      | 42.25   |

Berdasarkan Tabel 4.2, dapat dilihat bahwa proses Bayesian Optimization menghasilkan beberapa kombinasi *hyperparameter* berupa *learning rate, dropout rate, dan jumlah filter* dengan nilai akurasi yang berbeda-beda. Pada iterasi awal, Bayesian Optimization sudah mampu menemukan kombinasi *hyperparameter* dengan performa tinggi. Sebagai contoh, pada iterasi ke-2 diperoleh akurasi sebesar 0,99 dengan *learning rate* 0,005558, *dropout rate* 0,3878, dan jumlah *filter* 36,31. Kemudian, kombinasi *hyperparameter* terbaik diperoleh pada iterasi ke-13, yang menghasilkan nilai akurasi tertinggi yaitu 1,00. Kombinasi ini terdiri dari *learning rate* 0,0001, *dropout rate* 0,5, dan jumlah *filter* sekitar 42,25. Oleh karena itu,

kombinasi hyperparameter pada iterasi ke-13 dipilih dan digunakan sebagai parameter optimal pada tahap pelatihan model selanjutnya.

Tabel 4.3 Hasil Pencarian *hyperparameter learning rate, dropout rate*, dan jumlah *filter* VGG19

| No | learning rate | dropout   | filters |
|----|---------------|-----------|---------|
| 1  | 0.0069950     | 0.2144557 | 26.89   |
| 2  | 0.0055580     | 0.3877875 | 36.31   |
| 3  | 0.0098095     | 0.3739318 | 39.08   |
| 4  | 0.0078303     | 0.1100657 | 21.02   |
| 5  | 0.0001000     | 0.50      | 29.24   |
| 6  | 0.0063177     | 0.2448284 | 64.00   |
| 7  | 0.0095382     | 0.50      | 58.20   |
| 8  | 0.0001000     | 0.10      | 48.23   |
| 9  | 0.0100000     | 0.50      | 51.36   |
| 10 | 0.0001000     | 0.10      | 46.06   |
| 11 | 0.0001000     | 0.50      | 16.00   |
| 12 | 0.0100000     | 0.10      | 31.86   |
| 13 | 0.0001000     | 0.10      | 62.01   |

Berdasarkan Tabel 4.3, dapat dilihat bahwa proses *Bayesian Optimization* menghasilkan berbagai kombinasi hyperparameter berupa *learning rate*, *dropout rate*, dan jumlah *filter* dengan nilai akurasi yang juga bervariasi pada model VGG19. Hasil ini menunjukkan bahwa perubahan nilai hyperparameter memberikan pengaruh yang signifikan terhadap performa model. Pada iterasi awal, *Bayesian Optimization* mampu menemukan kombinasi hyperparameter dengan performa yang cukup baik. Contohnya pada iterasi ke-1 diperoleh nilai akurasi sebesar 0,97 dengan *learning rate* 0,006995, *dropout rate* 0,2145, dan jumlah *filter* sekitar 26,89. Selanjutnya, kombinasi hyperparameter terbaik diperoleh pada iterasi ke-8, yang menghasilkan nilai akurasi tertinggi yaitu 0,99. Kombinasi hyperparameter tersebut terdiri dari *learning rate* 0,0001, *dropout rate* 0,1, dan jumlah *filter* sekitar 48,23. Oleh karena itu, kombinasi hyperparameter pada iterasi

ke-8 dipilih dan digunakan sebagai parameter optimal dalam proses pelatihan model VGG19 pada tahap selanjutnya.

Selain menggunakan Bayesian Optimization, peneliti juga melakukan eksperimen dengan menentukan hyperparameter secara manual berdasarkan nilai yang umum digunakan dalam penelitian *deep learning*. Rentang parameter yang digunakan meliputi *learning rate* 0,0001 hingga 0,01, *dropout rate* 0,1 hingga 0,5, serta jumlah *filter* antara 16 hingga 64. Pendekatan ini bertujuan untuk membandingkan efektivitas pemilihan parameter secara manual dengan pendekatan otomatis menggunakan Bayesian Optimization.

Pendekatan yang digunakan pada penelitian ini sejalan dengan penelitian sebelumnya. Penelitian oleh (Da Rocha et al., 2021) menggunakan Bayesian Optimization untuk memperoleh hyperparameter optimal pada klasifikasi penyakit tanaman, dengan hasil *learning rate* 0.0036931912206946992, 0.0042498507802230421, dan 0.0026763419023990384. Sementara itu, penelitian oleh (Fraivan & Faouri, 2022) menggunakan kombinasi parameter berupa *learning rate* 0,003 dan jumlah *filter* 16. Hal ini menunjukkan bahwa Bayesian Optimization merupakan metode yang efektif dalam menentukan kombinasi hyperparameter terbaik pada model klasifikasi penyakit tanaman.

#### 4.6. Pelatihan Model

Tahap selanjutnya adalah pelatihan model yang dilakukan dengan variasi jumlah epoch, yaitu 20, 50, dan 100 epoch. Tujuannya adalah untuk mengevaluasi titik optimal pelatihan, di mana model dapat mencapai kinerja terbaik terhadap data validasi. Penggunaan jumlah epoch yang terlalu sedikit dapat menyebabkan model

mengalami *underfitting*, yaitu model belum belajar pola dengan baik atau belum mampu mempelajari pola data secara maksimal. Sebaliknya, penggunaan epoch yang terlalu banyak dapat menyebabkan *overfitting*, di mana model terlalu menyesuaikan diri terhadap data latih sehingga mengurangi generalisasi terhadap data baru.

Proses evaluasi terhadap performa model dilakukan melalui serangkaian eksperimen dengan mengkombinasikan berbagai nilai *learning rate*, *dropout rate*, dan jumlah *filter* pada arsitektur jaringan konvolusional yang digunakan. Pemilihan parameter dilakukan menggunakan Bayesian Optimization dan juga dilakukan secara manual berdasarkan pendekatan *trial and error* untuk memperoleh konfigurasi *hyperparameter* yang optimal terhadap performa klasifikasi. Tabel 4.4 menunjukkan skenario-skenario yang akan dilakukan oleh peneliti.

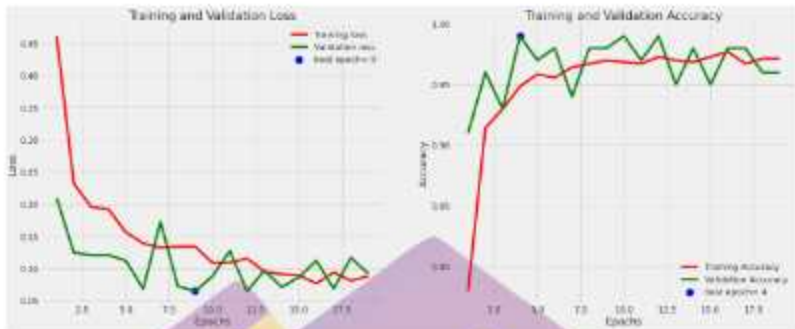
Tabel 4.4 Skenario Pelatihan yang akan dilakukan

| No | Model    | Parameter                          | Epoch       |
|----|----------|------------------------------------|-------------|
| 1  | Xception | Hyperparameter                     | 20, 50, 100 |
| 2  | VGG19    | Hyperparameter                     | 20, 50, 100 |
| 3  | Xception | LR = 0.0001, DO = 0.3, Filter = 64 | 50, 100     |
| 4  | VGG19    | LR = 0.001, DO = 0.5, Filter = 32  | 50, 100     |

Pada Tabel 4.4 ditunjukkan skenario-skenario pengujian yang dilakukan dalam penelitian ini. Skenario pertama menggunakan arsitektur Xception dengan penerapan hyperparameter hasil proses optimasi, kemudian dilakukan pelatihan dengan variasi jumlah epoch sebanyak 20, 50, dan 100 untuk menganalisis pengaruh lamanya pelatihan terhadap kinerja model. Skenario kedua yang menggunakan arsitektur VGG19 dengan hyperparameter hasil optimasi dengan variasi epoch 20, 50, dan 100 untuk mengamati perubahan kinerja model seiring peningkatan kompleksitas pelatihan. Selanjutnya, skenario ketiga menggunakan

arsitektur Xception, namun dengan kombinasi hyperparameter terbaik yang telah ditentukan secara manual, yaitu learning rate sebesar 0.0001, dropout rate sebesar 0.3, dan jumlah filter sebanyak 64, dengan variasi epoch 50 dan 100 guna mengevaluasi stabilitas performa model pada konfigurasi optimal. Skenario keempat menggunakan arsitektur VGG19 dengan kombinasi hyperparameter terbaik yang dipilih secara manual, yaitu learning rate sebesar 0.001, dropout rate sebesar 0.5, dan jumlah filter sebanyak 32, serta variasi epoch 50 dan 100 untuk mengevaluasi performa akhir dan efisiensi model setelah proses fine-tuning. Melalui keempat skenario tersebut, penelitian ini bertujuan memperoleh konfigurasi pelatihan yang optimal serta membandingkan kinerja dan efisiensi komputasi antara arsitektur Xception dan VGG19 secara komprehensif.

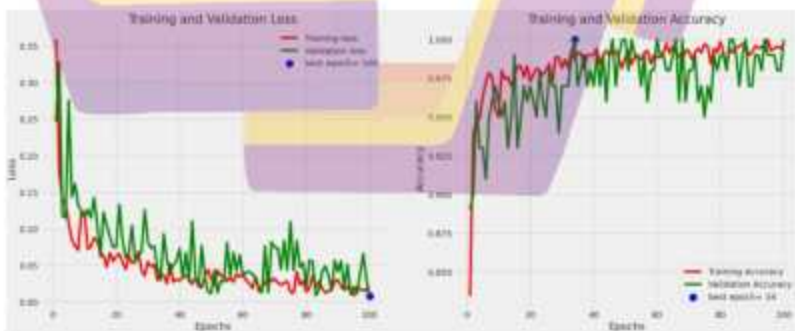
Skenario pertama yang peneliti lakukan menggunakan model Xception, yang merupakan salah satu pengembangan dari CNN dengan pendekatan *depthwise separable convolution*. Arsitektur ini dikenal unggul dalam efisiensi komputasi dan akurasi klasifikasi. Keunggulan Xception ini juga telah dibuktikan dalam penelitian yang dilakukan oleh (Jain & Singh, 2024), di mana model tersebut berhasil mencapai akurasi sebesar 93% dalam mengklasifikasi penyakit pada daun zaitun. Penggunaan parameter pada skenario pertama ini diperoleh menggunakan hasil dari pencarian hyperparameter BO. Hasil evaluasi dan representasi visual dari performa model ini disajikan pada Gambar 4.2, 4.3, dan 4.4.



Gambar 4.2 Hasil grafik *Xception* dengan 20 epoch



Gambar 4.3 Hasil grafik *Xception* dengan 50 epoch



Gambar 4.4 Hasil grafik *Xception* dengan 100 epoch

Pada gambar 4.2, penggunaan Xception dengan optimasi hyperparameter menunjukkan performa yang sangat unggul dibandingkan eksperimen sebelumnya. Pada grafik pelatihan 20 epoch, model langsung menunjukkan akselerasi akurasi yang tajam, mencapai akurasi validasi terbaik sebesar 99% hanya pada epoch ke-4. Hal ini mengindikasikan bahwa arsitektur Xception sangat efisien dalam menangkap fitur dataset sejak tahap awal pelatihan.

Selanjutnya pada gambar 4.3, pelatihan dengan 50 epoch dengan optimasi hyperparameter, model berhasil mencapai titik sempurna dengan akurasi validasi sebesar 100% pada epoch ke-29. Meskipun mencapai angka maksimal, mulai terlihat adanya fluktuasi pada kurva validasi (garis hijau) setelah melewati epoch ke-30, yang menandakan model mulai sangat sensitif terhadap variasi data. Namun, nilai *loss* validasi terendah tetap stabil di angka 0.027 pada epoch yang sama.

Kemudian eksperimen dengan 100 epoch pada gambar 4.4 dengan menggunakan optimasi hyperparameter, model menunjukkan konsistensi yang baik. Akurasi validasi tertinggi sebesar 100% dicapai pada epoch ke-34. Meskipun proses pelatihan dilanjutkan hingga epoch ke-100, model masih melakukan proses penyempurnaan dengan menurunkan nilai *loss*. Hal ini terlihat dari nilai *loss* validasi terendah yang justru dicapai pada epoch ke-100, yaitu mendekati 0,01. Kondisi ini menunjukkan bahwa setelah akurasi mencapai nilai maksimal dan tidak lagi mengalami peningkatan, model tetap memperbaiki kualitas prediksinya dengan mengurangi tingkat kesalahan pada data validasi hingga akhir pelatihan.

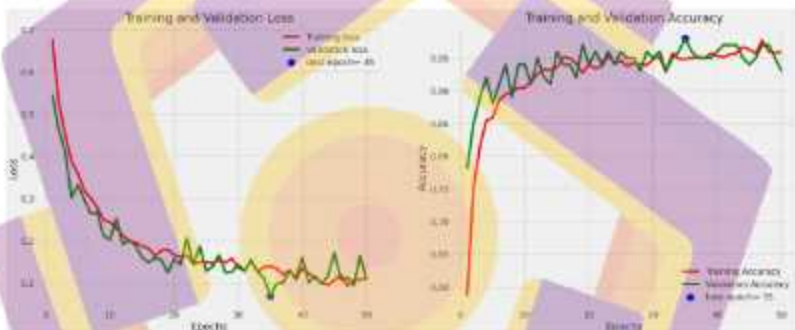
Proses pelatihan ini dilakukan dengan menerapkan *rescale* 1./255 dan *resizing* gambar. Hyperparameter yang digunakan meliputi *learning rate* 0.0001, *dropout*

rate 0.5, dan jumlah filter 42, yang dipilih berdasarkan hasil dari *Bayesian Optimization*. Jika dibandingkan dengan penelitian sebelumnya yang dilakukan oleh (Nazalia et al., 2023), model Convolutional Neural Network (CNN) yang mereka rancang dari awal (*CNN from Scratch*) digunakan untuk mendeteksi hama pada tanaman sawi. Penelitian tersebut menggunakan dataset berjumlah 1000 gambar yang terdiri atas 500 gambar tanaman dengan hama dan 500 tanpa hama. Dataset kemudian dibagi ke dalam beberapa skenario percobaan dengan variasi proporsi pelatihan, validasi, dan pengujian: 40:30:30 (Eksperimen A), 80:10:10 (Eksperimen B), dan 90:5:5 (Eksperimen C). Arsitektur CNN yang dibangun bervariasi dari 1 hingga 4 lapisan konvolusional dan menggunakan *dropout*, *early stopping*, serta model checkpoint pada eksperimen terbaik. Model CNN pada Eksperimen C mencapai akurasi tertinggi sebesar 92%, dengan precision 0.88, recall 0.96, dan F1-score 0.92. Model ini dianggap paling stabil dan tidak mengalami *overfitting* karena pelatihan dihentikan otomatis pada epoch ke-12 oleh mekanisme *early stopping*. Jika dibandingkan dengan penelitian yang dilakukan (Nazalia et al., 2023), penelitian ini menghasilkan akurasi lebih bagus dengan hasil akurasi tertinggi pada 100 epoch yaitu 99.50% dan hasil yang diperoleh tetap optimal dan tidak menunjukkan indikasi *overfitting*.

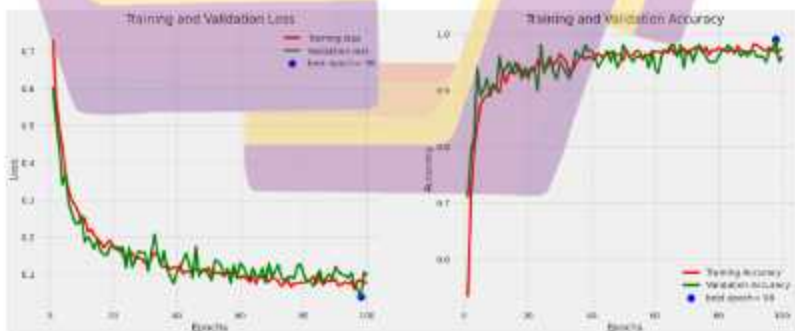
Skenario kedua yang dilakukan oleh peneliti adalah menggunakan model VGG19 dengan hyperparameter menggunakan Bayesian Optimization. Hasil evaluasi dan representasi visual dari performa model ini disajikan pada Gambar 4.5, 4.6, dan 4.7.



Gambar 4.5 Hasil grafik *VGG19* dengan 20 epoch



Gambar 4.6 Hasil grafik *VGG19* dengan 50 epoch



Gambar 4.7 Hasil grafik *VGG19* dengan 100 epoch

Pada gambar 4.5, pelatihan yang dilakukan dengan 20 epoch, model menunjukkan performa yang sangat stabil. Akurasi validasi terbaik sebesar 96% berhasil dicapai pada epoch ke-18, dengan nilai *loss* validasi terendah di angka 0,137. Grafik menunjukkan bahwa kurva *training* dan *validation* saling berhimpitan secara konsisten, yang menandakan bahwa model VGG sangat stabil dalam memproses dataset ini sejak tahap awal tanpa menunjukkan gejala *underfitting* maupun *overfitting*.

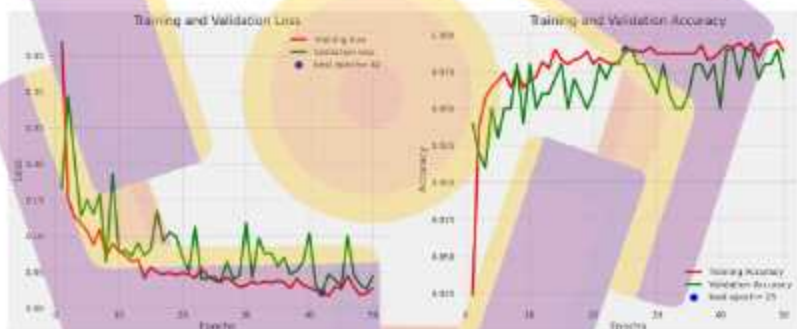
Kemudian pada gambar 4.6, peningkatan jumlah pelatihan menjadi 50 epoch memberikan kemajuan yang signifikan. Model mencapai titik optimal pada epoch ke-35 dengan akurasi validasi yang meningkat menjadi 98%. Seiring dengan peningkatan akurasi, nilai *loss* validasi berhasil ditekan hingga angka 0,068. Hal ini membuktikan bahwa penambahan jumlah epoch pada model VGG secara efektif membantu model untuk mengenali fitur-fitur yang lebih kompleks pada citra.

Dan pada gambar 4.7, eksperimen dilakukan selama 100 epoch, model VGG mencapai performa puncaknya. Akurasi validasi tertinggi mencapai angka 99% yang diperoleh pada epoch ke-98. Menariknya, nilai *loss* validasi terus menurun hingga mencapai titik terendah sebesar 0,038 pada akhir pelatihan. Berbeda dengan model Xception yang cenderung mengalami fluktuasi tajam pada epoch tinggi, model VGG menunjukkan kurva yang lebih halus dan stabil hingga epoch ke-100. Hal ini menunjukkan bahwa arsitektur VGG memiliki ketahanan yang sangat baik terhadap gangguan (*noise*) data selama proses pelatihan jangka panjang.

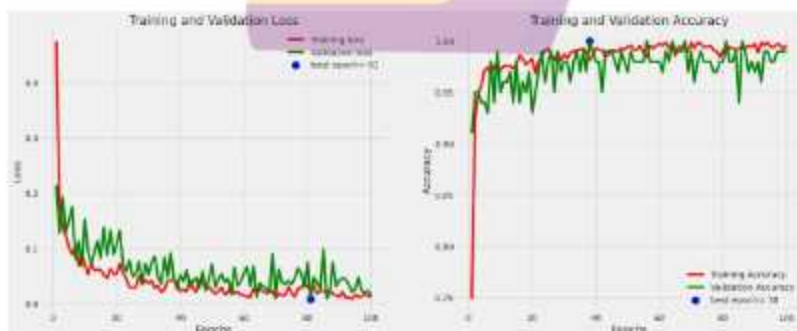
Jika dibandingkan dengan penelitian sebelumnya yang dilakukan oleh (Fraivan & Faouri, 2022) dengan beberapa model yang dipilih dalam penelitiannya, mereka

mencapai akurasi terbesar 82,9%. Hasil ini diperoleh setelah melalui proses *preprocessing* data, termasuk penerapan teknik augmentasi untuk menyeimbangkan dataset. Selain itu, dataset dibagi menjadi 90% untuk *training* dan 10% untuk *testing*. Model dilatih menggunakan learning rate 0.003 dan jumlah filter 16, dengan durasi pelatihan hanya selama 10 *epoch*. Jadi jika dibandingkan dengan penelitian ini, hasil akhirnya lebih optimal tanpa indikasi overfitting.

Kemudian pada *skenario* ketiga, peneliti menggunakan model Xception dengan parameter yang dicari secara manual dan telah dilakukan beberapa kali *trial and error*. Hasil evaluasi dan representasi visual dari performa model ini disajikan pada Gambar 4.8 dan 4.9.



Gambar 4.8 Hasil grafik Xception dengan 50 epoch



Gambar 4.9 Hasil grafik *Xception* dengan 100 *epoch*

Pada Gambar 4.8, pelatihan model *Xception* menggunakan parameter manual selama 50 epoch menunjukkan performa yang sangat kompetitif sejak awal proses dimulai. Model ini berhasil mencapai akurasi validasi terbaik sebesar 99% yang dicapai pada epoch ke-25, di mana kurva akurasi terlihat meningkat secara signifikan pada 10 epoch pertama dan mulai menunjukkan tren mendatar setelah melewati epoch ke-20. Dari sisi tingkat kerugian, nilai *loss* validasi terbaik diperoleh pada epoch ke-42 dengan nilai yang sangat rendah sebesar 0,02, yang mengindikasikan bahwa model memiliki tingkat kesalahan yang sangat minim dalam mengklasifikasikan data. Meskipun terdapat fluktuasi kecil pada kurva validasi hijau, tren keseluruhannya tetap terjaga stabil dan menunjukkan bahwa model mampu mempelajari fitur-fitur penting dalam dataset secara efektif tanpa adanya indikasi overfitting yang menonjol pada rentang waktu pelatihan ini.

Selanjutnya, pada Gambar 4.9 adalah hasil pelatihan selama 100 epoch dengan parameter manual, model ini berhasil meningkatkan performanya hingga mencapai titik maksimal. Model ini mencatatkan akurasi validasi sempurna sebesar 100% yang dicapai pada epoch ke-38, menunjukkan bahwa durasi pelatihan yang lebih lama memberikan ruang bagi model untuk lebih mengoptimalkan bobot internalnya dalam mengenali pola citra secara presisi. Peningkatan ini juga tercermin secara nyata pada nilai *loss* validasi yang mencapai titik terendah sebesar 0,01 pada epoch ke-81. Berdasarkan grafik tersebut, terlihat bahwa walaupun akurasi sudah mencapai batas maksimal di pertengahan sesi, model tetap terus melakukan optimasi pada nilai kerugian hingga mendekati akhir pelatihan, yang menghasilkan

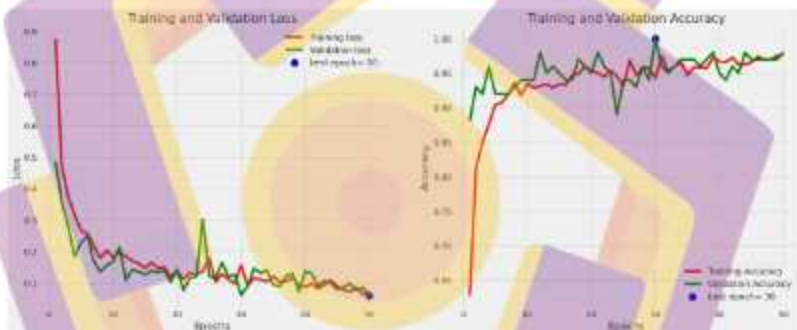
model dengan tingkat kepercayaan prediksi yang sangat tinggi serta stabilitas yang lebih matang dibandingkan skenario sebelumnya.

Secara keseluruhan, hasil dari pelatihan menggunakan parameter manual ini menunjukkan bahwa arsitektur Xception memiliki kapabilitas yang sangat tinggi untuk mengenali pola data secara akurat. Meskipun tanpa bantuan optimasi Bayesian, model tetap mampu menghasilkan akurasi yang sempurna, walaupun memerlukan waktu (jumlah epoch) yang sedikit lebih lama untuk mencapai titik jenuh akurasi dibandingkan dengan versi yang dioptimalkan.

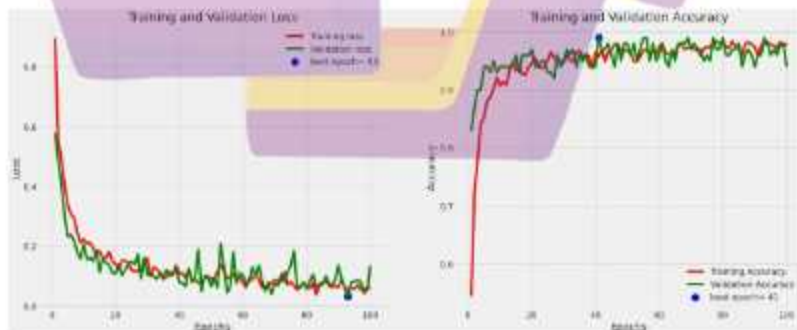
Jika dibandingkan dengan penelitian yang dilakukan oleh (Sood et al., 2023), performa model Xception dalam penelitian ini tergolong sangat kompetitif. Penelitian tersebut menerapkan pendekatan CNN untuk klasifikasi penyakit daun sawi dengan akurasi rata-rata sebesar 93.84% pada skenario multi-klasifikasi dan 94.1% pada klasifikasi biner. Meskipun model CNN yang digunakan oleh (Sood et al., 2023) juga menunjukkan hasil yang solid, capaian akurasi validasi sebesar 97.50% oleh model Xception dalam penelitian ini juga menunjukkan peningkatan performa yang cukup signifikan, terutama dalam konteks generalisasi terhadap data validasi dan kestabilan konvergensi. Hal ini mempertegas bahwa penggunaan arsitektur Xception yang memanfaatkan *depthwise separable convolution* tidak hanya unggul dari sisi akurasi, tetapi juga mampu menangkap fitur spasial dengan efisiensi tinggi, menjadikannya lebih unggul dalam klasifikasi citra berbasis penyakit tanaman. Selain itu, penelitian yang dilakukan oleh (Jain & Singh, 2024) dan memperoleh akurasi 93% dalam klasifikasi penyakit daun zaitun menggunakan Xception, maka akurasi 97.50% yang diperoleh pada penelitian ini juga

menunjukkan peningkatan yang signifikan dalam konteks pengklasifikasian daun sawi. Hal ini mengindikasikan bahwa penyesuaian arsitektur dan pemilihan parameter yang tepat sangat menentukan keberhasilan penerapan model Xception pada domain klasifikasi citra tanaman.

Dan pada skenario terakhir peneliti menggunakan model VGG19 dengan parameter yang dicari secara manual dan telah dilakukan beberapa kali *trial and error*. Hasil evaluasi dan representasi visual dari performa model ini disajikan pada Gambar 4.10 dan 4.11.



Gambar 4.10 Hasil grafik VGG19 dengan 50 epoch



Gambar 4.11 Hasil grafik VGG19 dengan 100 epoch

Pada Gambar 4.10 di atas, hasil pelatihan model VGG19 menggunakan parameter manual selama 50 epoch menunjukkan performa yang bagus dan optimal. Model ini berhasil mencapai titik akurasi validasi maksimal sebesar 100% pada epoch ke-30, yang mengindikasikan bahwa arsitektur VGG19 mampu mengenali fitur-fitur pada dataset dengan tingkat presisi yang sempurna melalui penyesuaian parameter secara manual. Seiring dengan pencapaian akurasi tersebut, nilai *loss* validasi juga menunjukkan tren penurunan yang konsisten hingga mencapai titik terbaiknya pada epoch ke-50 dengan nilai sebesar 0,06. Grafik menunjukkan bahwa kurva pelatihan dan validasi bergerak selaras tanpa ada jarak lebar yang memisahkan keduanya, mencerminkan stabilitas model yang sangat baik serta kemampuan generalisasi yang tinggi tanpa menunjukkan gejala *overfitting* yang berarti selama 50 epoch tersebut.

Dan pada gambar 4.11 menunjukkan hasil pelatihan model VGG19 selama 100 epoch menggunakan parameter manual. Hasil pelatihan menunjukkan bahwa model mampu belajar secara stabil dan konsisten. Akurasi validasi tertinggi sebesar 99% dicapai pada epoch ke-41, yang menandakan performa model sudah sangat baik sejak pertengahan pelatihan. Selain itu, penambahan jumlah epoch memberikan dampak positif terhadap penurunan nilai *loss*. *Loss* validasi terendah berhasil dicapai pada epoch ke-93 dengan nilai 0,03, menunjukkan bahwa model semakin mampu meminimalkan kesalahan prediksi. Kurva validasi yang relatif stabil hingga akhir pelatihan menandakan bahwa parameter manual yang digunakan mampu menjaga keseimbangan performa model tanpa menimbulkan *overfitting* yang signifikan.

Sedangkan yang dilakukan oleh (Sakshi et al., 2024) pada penelitiannya, model VGG mencapai akurasi sebesar 97.50%. Hasil ini diperoleh setelah melalui proses *preprocessing* data, yang mencakup augmentasi untuk menyeimbangkan dataset dengan teknik *rotasi*, *flip vertikal*, dan *flip horizontal*, serta mengubah ukuran gambar menjadi  $224 \times 224$  piksel. Selain itu, penelitian ini menggunakan metode *5-Fold Cross-Validation*, di mana dataset dibagi menjadi 5 bagian (*folds*) dengan proporsi kelas yang setara. Pada setiap eksperimen, 4 bagian (80%) digunakan untuk pelatihan, sementara 1 bagian (20%) digunakan untuk validasi. Model dilatih menggunakan *learning\_rate* 0.00424, *dropout\_rate* 0.4755, dan jumlah filter 32, yang diperoleh melalui *Bayesian Optimization*.

Penerapan teknik regularisasi seperti *dropout* dan penggunaan strategi *data augmentation* pada tahap awal pelatihan juga memberikan kontribusi terhadap kestabilan pelatihan dan pencapaian akurasi tinggi ini. Hasil ini menunjukkan bahwa model mampu mempelajari fitur visual secara efektif dan mempertahankan performa optimal pada data validasi, menjadikannya layak untuk digunakan dalam sistem klasifikasi berbasis citra pada domain yang relevan. Keberhasilan model ini dalam mencapai performa ini tidak terlepas dari penerapan strategi pelatihan yang tepat, seperti penggunaan teknik *dropout* dan strategi *data augmentation* (*rotasi*, *zoom*, dan *flipping*) pada tahap *preprocessing*. Kombinasi ini memungkinkan model untuk menghindari *overfitting*, memperluas keberagaman data pelatihan, dan meningkatkan ketangguhan dalam mengenali fitur visual yang bervariasi.

Pada eksperimen dengan jumlah maksimum 100 epoch, model VGG menunjukkan performa yang sangat baik dan stabil selama proses pelatihan.

Akurasi validasi tertinggi yang dicapai sebesar 99% terjadi pada epoch ke-98, yang menandakan bahwa model mampu mempelajari pola data secara optimal. Selain itu, nilai loss validasi terus mengalami penurunan hingga mencapai nilai terendah sebesar 0,038 pada akhir proses pelatihan. Jika dibandingkan dengan model Xception yang pada epoch-epoch akhir cenderung mengalami fluktuasi performa, kurva pelatihan model VGG terlihat lebih halus dan konsisten hingga epoch ke-100. Kondisi ini menunjukkan bahwa arsitektur VGG memiliki stabilitas yang baik selama pelatihan jangka panjang serta relatif lebih tahan terhadap variasi atau noise pada data.

Berdasarkan hasil pengujian, perbedaan performa antara kedua model dapat dijelaskan melalui karakteristik arsitektur masing-masing. Model Xception menunjukkan kecepatan inferensi yang lebih tinggi dengan rata-rata 27,58 ms per citra dibandingkan VGG19 yang memerlukan 33,37 ms. Hal ini terjadi karena Xception menerapkan mekanisme *depthwise separable convolution* yang secara efisien memisahkan proses ekstraksi fitur spasial dan korelasi antar saluran, sehingga beban komputasi pada operasi matriks berkurang drastis tanpa kehilangan informasi fitur yang signifikan. Sebaliknya, VGG19 menggunakan konvolusi standar yang padat parameter; meskipun proses komputasinya lebih lambat, arsitektur yang lebih sederhana dan linear ini memberikan kurva akurasi yang lebih stabil dan halus selama 100 *epoch*, yang menunjukkan ketangguhan model dalam menjaga konsistensi bobot jaringan.

Tingginya akurasi yang mencapai rentang 99-100% pada kedua model dipengaruhi secara signifikan oleh penerapan *Bayesian Optimization* dalam

menentukan kombinasi *hyperparameter* secara presisi. Penggunaan *learning rate* yang sangat kecil (0,0001) memungkinkan model melakukan penyesuaian bobot secara inkremental dan halus, sehingga proses pelatihan dapat mencapai titik konvergensi optimal tanpa mengalami fenomena *overshooting*. Selain itu, teknik augmentasi data seperti rotasi dan *flip* memperkuat kemampuan generalisasi model dalam mengenali objek hama meskipun posisinya tidak berada di tengah citra. Hal ini membuat model tetap mampu mempertahankan performa tinggi pada berbagai variasi data uji yang diberikan.

Meskipun model menunjukkan performa yang sangat baik, analisis kesalahan klasifikasi menunjukkan bahwa model masih memiliki keterbatasan dalam membedakan karakteristik morfologi daun tertentu. Kesalahan prediksi umumnya terjadi pada citra daun sawi yang memiliki lubang, namun sebenarnya tidak disebabkan oleh serangan hama. Hal ini disebabkan karena model cenderung mengenali adanya lubang atau ketidakteraturan permukaan daun sebagai ciri utama serangan hama, padahal kondisi tersebut juga dapat muncul akibat kerusakan fisik non-biotis atau robekan mekanis. Keterbatasan ini menunjukkan bahwa model masih sangat sensitif terhadap fitur visual yang mirip secara bentuk namun berbeda secara etiologi, yang menyebabkan nilai *loss* validasi pada Xception cenderung fluktuatif setelah *epoch* ke-30 dibandingkan dengan VGG19 yang lebih kaku dan stabil dalam mengabaikan *noise* tersebut.

Sebagai bentuk validasi terhadap keterbaruan penelitian, hasil akurasi model Xception (99%) dan VGG19 (99,00%) dalam penelitian ini menunjukkan peningkatan yang signifikan dibandingkan dengan penelitian-penelitian terdahulu

yang menjadi *State-of-the-Art*. Jika dibandingkan dengan penelitian (Nazalia et al., 2023) yang menggunakan CNN *from scratch* dengan akurasi 92%, atau penelitian (Sood et al., 2023) yang mencapai 94,1% pada klasifikasi biner daun sawi, hasil penelitian ini terbukti lebih unggul. Pencapaian ini menegaskan bahwa penggunaan model *transfer learning* yang dikombinasikan dengan optimasi sistematis seperti *Bayesian Optimization* jauh lebih efektif daripada penggunaan arsitektur standar atau pencarian parameter secara manual. Keunggulan hasil ini memberikan kontribusi nyata dalam pengembangan sistem deteksi hama yang lebih akurat dan efisien untuk diimplementasikan pada sektor pertanian digital. Dibawah ini merupakan table perbandingan hasil dengan penelitian lain.

Tabel 4.5 Perbandingan dengan Penelitian Terdahulu

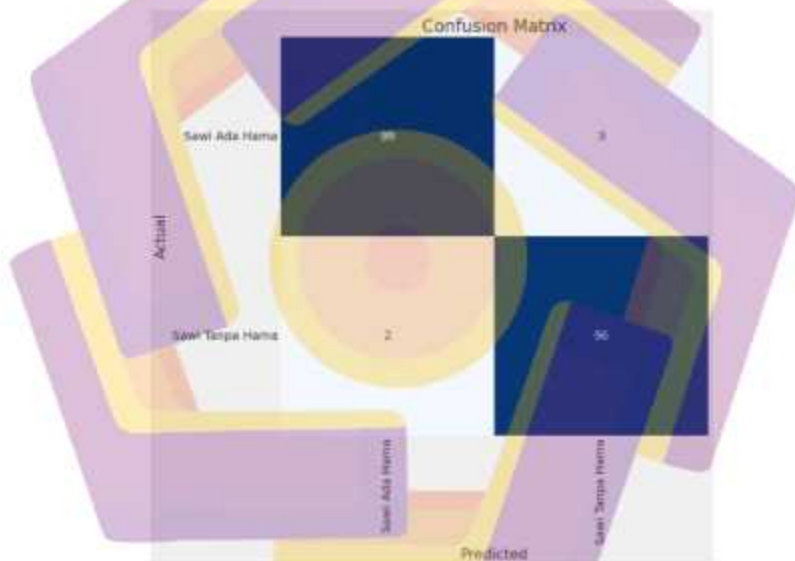
| Peneliti (Tahun)        | Metode/Model             | Dataset (Jumlah Citra)            | Akurasi Terbaik |
|-------------------------|--------------------------|-----------------------------------|-----------------|
| Nazalia et al. (2023)   | CNN from Scratch         | Daun Caisim (1000)                | 92,00%          |
| Fraihan & Faouri (2022) | CNN Standar              | Daun Tanaman Jagung (Multi-class) | 82,90%          |
| Sood et al. (2023)      | CNN Base Model           | Daun Sawi (Binary)                | 94,10%          |
| Sakshi et al. (2024)    | VGG16/AlexNet            | Penyakit Daun Sawi                | 97,50%          |
| Penelitian Ini (2024)   | Xception + Bayesian Opt. | Daun Sawi (1000)                  | 99,00%          |
| Penelitian Ini (2024)   | VGG19 + Bayesian Opt.    | Daun Sawi (1000)                  | 99,00%          |

#### 4.7. Evaluasi Model

Setelah model klasifikasi pada daun sawi berhasil diimplementasikan, tahap akhir dalam penelitian ini adalah melakukan evaluasi kinerja model menggunakan

*confusion matrix*. Evaluasi ini bertujuan untuk memperoleh pemahaman yang lebih komprehensif terhadap pola kesalahan prediksi yang dilakukan oleh model dalam membedakan dua kelas kondisi daun sawi.

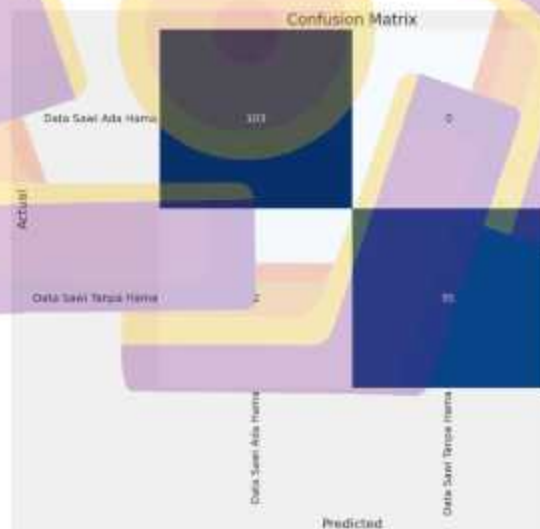
Evaluasi model yang pertama Adalah hasil dari model Xception yang menggunakan *hyperparameter* dari hasil pencarian Bayesian Optimization dengan *learning rate* = 0.0001, *dropout rate* = 0.5, dan *filter* 42. Hasil dari *confussion matrix* ditunjukkan pada gambar 4.12, 4.13, dan 4.14.



Gambar 4.12 Hasil *confussion matrix* hyperparameter Xception 20 epoch



Gambar 4.13 Hasil *confussion matrix* hyperparameter *Xception* 50 epoch



Gambar 4.14 Hasil *confussion matrix* hyperparameter *Xception* 100 epoch

Pada Gambar 4.12 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur Xception dengan 20 epoch. Pada kategori Sawi Ada Hama, sebanyak 99 gambar berhasil diprediksi dengan benar, sementara terdapat kesalahan prediksi yaitu 3 gambar salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, sebanyak 96 gambar berhasil diprediksi dengan benar, dengan kesalahan prediksi mencakup 2 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 195 benar dan 5 prediksi salah.

Pada Gambar 4.13 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur Xception dengan 50 epoch. Pada kategori Sawi Ada Hama, sebanyak 101 gambar berhasil diprediksi dengan benar, sementara terdapat kesalahan prediksi yaitu 2 gambar salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 94 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 3 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 195 benar dan 5 prediksi salah.

Pada Gambar 4.14 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur Xception dengan 100 epoch. Pada kategori Sawi Ada Hama, sebanyak 103 gambar berhasil diprediksi dengan benar, sementara tidak ada gambar yang salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat

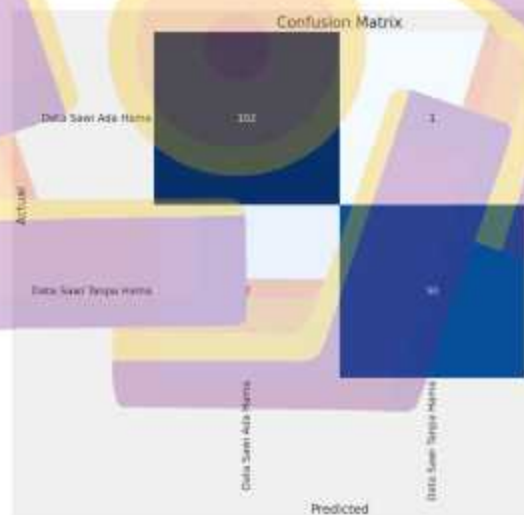
95 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 2 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 198 benar dan 2 prediksi salah.

Berdasarkan data di atas, terlihat adanya peningkatan performa yang signifikan ketika model dilatih hingga 100 epoch. Pada 20 dan 50 epoch, model memiliki tingkat kesalahan yang sama (5 prediksi salah), meskipun distribusi kesalahannya sedikit bergeser antar kategori. Pada 100 epoch, model mencapai performa optimal dengan hanya 2 kesalahan prediksi. Menariknya, pada tahap ini model berhasil memprediksi seluruh data kategori Sawi Ada Hama dengan sempurna (103 benar, 0 salah). Hal ini menunjukkan bahwa penambahan jumlah *epoch* pada arsitektur Xception efektif dalam membantu model mempelajari fitur-fitur yang lebih detail untuk membedakan antara sawi yang terkena hama dan yang sehat.

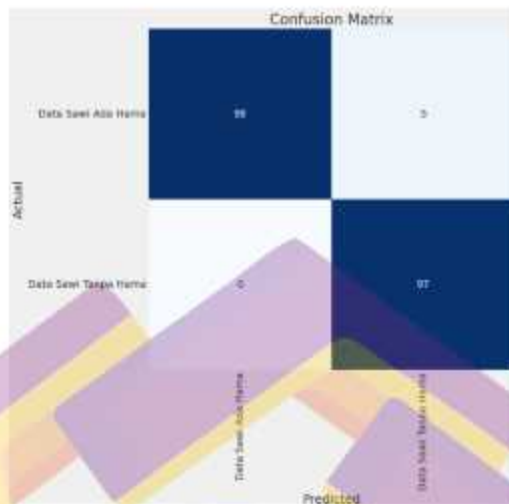
Selanjutnya hasil dari evaluasi model yang kedua menggunakan model VGG19 dan menggunakan *hyperparameter* dari hasil pencarian Bayesian Optimization dengan *learning rate* = 0.0001, *dropout rate* = 0.5, dan *filter* 42. Hasil dari *confusion matrix* ditunjukkan pada gambar 4.15, 4.16, dan 4.17.



Gambar 4.15 Hasil *confussion matrix* hyperparameter *VGG19* 20 epoch



Gambar 4.16 Hasil *confussion matrix* hyperparameter *VGG19* 50 epoch



Gambar 4.17 Hasil *confussion matrix* hyperparameter VGG19 100 epoch

Pada Gambar 4.15 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur VGG19 dengan 20 epoch. Pada kategori Sawi Ada Hama, sebanyak 101 gambar berhasil diprediksi dengan benar, sementara terdapat kesalahan prediksi yaitu 2 gambar salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 90 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 7 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 191 benar dan 9 prediksi salah.

Pada Gambar 4.16 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur VGG19 dengan 50 epoch. Pada kategori Sawi Ada Hama, sebanyak 102 gambar berhasil diprediksi dengan benar, sementara terdapat kesalahan prediksi yaitu 1

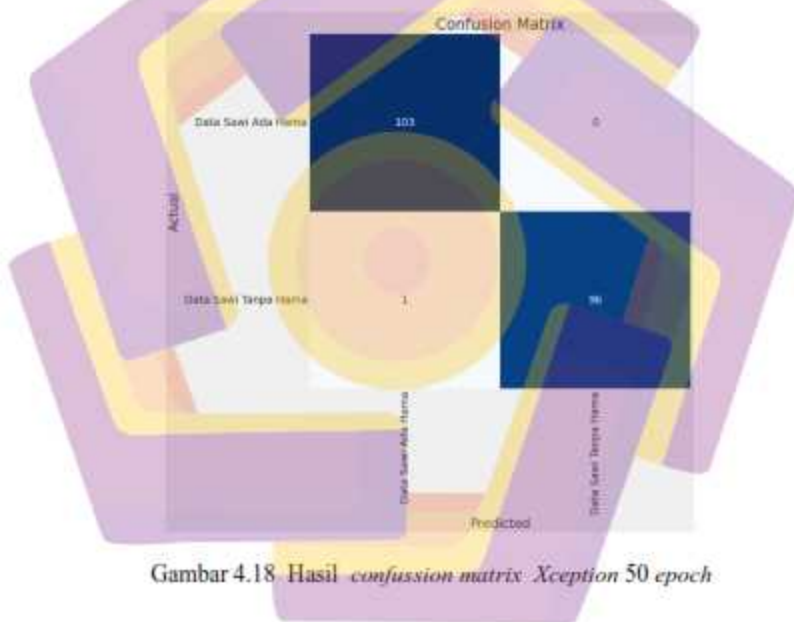
gambar salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 90 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 7 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 192 benar dan 8 prediksi salah.

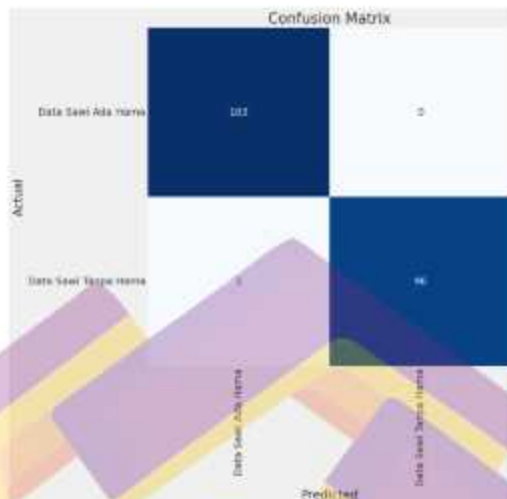
Pada Gambar 4.17 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur VGG19 dengan 100 *epoch*. Pada kategori Sawi Ada Hama, sebanyak 98 gambar berhasil diprediksi dengan benar, sementara terdapat kesalahan prediksi yaitu 5 gambar salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 97 gambar yang berhasil diprediksi dengan benar, dan tidak ada gambar yang diprediksi salah sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 195 benar dan 5 prediksi salah.

Dari data hasil pengujian menggunakan arsitektur VGG19 dengan optimasi hyperparameter di atas, dapat disimpulkan bahwa model VGG19 menunjukkan tren peningkatan akurasi yang stabil seiring dengan bertambahnya jumlah *epoch*. Dari 20 ke 50 *epoch* terdapat peningkatan 1 data benar, dan melonjak cukup baik saat mencapai 100 *epoch* dengan total 195 prediksi benar. Hasil terbaik dicapai pada 100 *epoch* dengan tingkat akurasi 97,5%. Pada titik ini, model menunjukkan kemampuan yang sangat baik dalam mengenali kategori "Sawi Tanpa Hama" dengan nol kesalahan (0 gambar salah diprediksi sebagai Ada Hama). Pada pengujian 20 dan 50 *epoch*, model cenderung lebih sering salah memprediksi "Sawi

Tanpa Hama" sebagai "Sawi Ada Hama" (7 gambar salah). Namun, pada 100 *epoch*, pola kesalahan berubah di mana seluruh kesalahan (5 gambar) terjadi pada kategori "Sawi Ada Hama" yang terdeteksi sebagai "Sawi Tanpa Hama".

Kemudian hasil dari evaluasi model skenario ketiga menggunakan model Xception yang parameternya dicari secara manual dengan *learning rate* = 0.0001, *dropout rate* = 0.3, dan *filter* 64. Hasil dari *confussion matrix* ditunjukkan pada gambar 4.18 dan 4.19.





Gambar 4.19 Hasil *confusion matrix* Xception 100 epoch

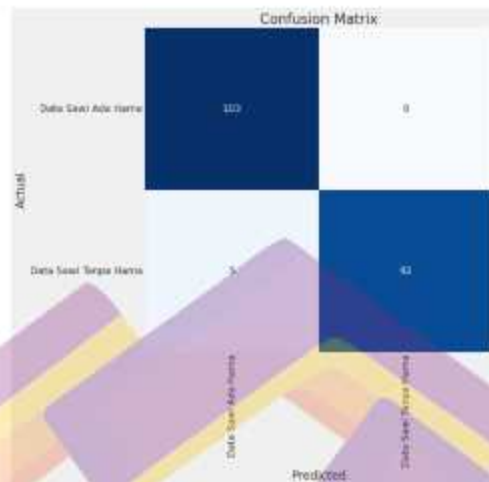
Pada Gambar 4.18 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur Xception dengan parameter manual selama 50 *epoch*. Pada kategori Sawi Ada Hama, sebanyak 103 gambar berhasil diprediksi dengan benar, sementara tidak ada gambar yang salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 96 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup hanya 1 gambar yang salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 199 benar dan 1 prediksi salah.

Pada Gambar 4.19 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model menggunakan arsitektur Xception dengan parameter manual selama 100 *epoch*. Pada kategori Sawi Ada Hama, sebanyak 103 gambar berhasil diprediksi dengan benar, sementara tidak ada gambar yang salah diprediksi

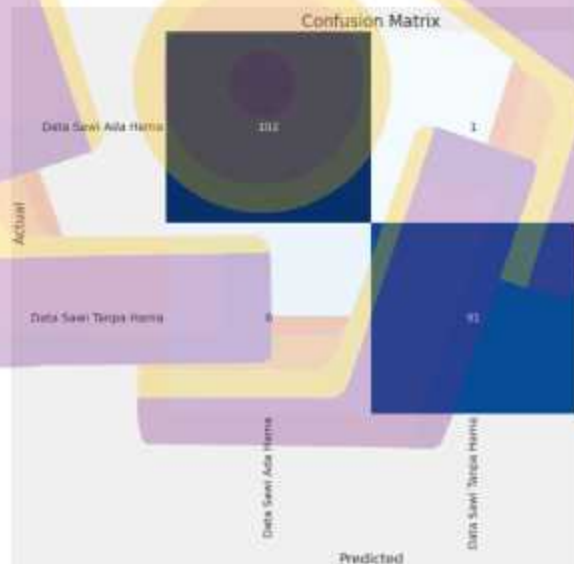
sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 96 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 1 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 199 data dengan benar dan hanya 1 prediksi yang salah.

Pengujian dengan parameter manual dengan model xception ini menunjukkan hasil yang stabil, di mana pada 50 *epoch* maupun 100 *epoch*, model mencapai jumlah prediksi benar yang sama yaitu 199 dari 200 data. Model berhasil memprediksi seluruh gambar pada kategori Sawi Ada Hama dengan sempurna (103 benar dan 0 salah) pada kedua pengujian tersebut. Hanya terdapat 1 kesalahan prediksi pada kategori Sawi Tanpa Hama, di mana satu gambar salah diidentifikasi sebagai Sawi Ada Hama. Karena hasil pada 50 *epoch* sudah identik dengan 100 *epoch*, hal ini menunjukkan bahwa model Xception dengan parameter manual sudah mencapai titik optimalnya lebih cepat tanpa memerlukan iterasi tambahan yang lebih banyak.

Dan hasil dari evaluasi model pada skenario keempat menggunakan model VGG19 yang parameternya dicari secara manual dengan *learning rate* = 0.001, *dropout rate* = 0.5, dan *filter* 32. Hasil dari *confussion matrix* ditunjukkan pada gambar 4.20 dan 4.21.



Gambar 4.20 Hasil *confussion matrix* VGG19 50 epoch



Gambar 4.21 Hasil *confussion matrix* VGG19 100 epoch

Pada Gambar 4.20 ditampilkan *confussion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur

VGG19 dengan parameter manual pada 50 *epoch*. Pada kategori Sawi Ada Hama, sebanyak 103 gambar berhasil diprediksi dengan benar, sementara tidak ada gambar yang salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 92 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 5 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 195 benar dan 5 prediksi salah dari total 200 data uji.

Pada Gambar 4.21 ditampilkan *confusion matrix* yang menunjukkan performa prediksi model untuk masing-masing kategori menggunakan arsitektur VGG19 dengan parameter manual pada 100 *epoch*. Pada kategori Sawi Ada Hama, sebanyak 102 gambar berhasil diprediksi dengan benar, sementara terdapat kesalahan prediksi yaitu 1 gambar salah diprediksi sebagai Sawi Tanpa Hama. Selanjutnya, pada kategori Sawi Tanpa Hama, terdapat 91 gambar yang berhasil diprediksi dengan benar, sedangkan kesalahan prediksi mencakup 6 gambar salah diprediksi sebagai Sawi Ada Hama. Secara keseluruhan, hasil ini menunjukkan bahwa model memprediksi 193 benar dan 7 prediksi salah dari total 200 data uji.

Hasil akhir dari model VGG19 dengan parameter manual ini mencapai performa terbaiknya pada 50 *epoch* dengan akurasi 97,5%. Pada pengujian 50 *epoch*, model berhasil mengklasifikasikan seluruh gambar kategori Sawi Ada Hama dengan akurasi 100% (103 benar, 0 salah). Lalu, terdapat penurunan jumlah prediksi benar dari 195 menjadi 193 saat *epoch* ditingkatkan menjadi 100. Kesalahan prediksi ini didominasi oleh kategori Sawi Tanpa Hama yang terdeteksi

sebagai Sawi Ada Hama, yaitu sebanyak 5 gambar pada 50 *epoch* dan 6 gambar pada 100 *epoch*.

Jika dibandingkan dengan yang dilakukan oleh (Sakshi et al., 2024) pada penelitiannya, model *VGG* mencapai akurasi yang sama, yaitu sebesar 97.50%. Namun, keunggulan skenario ini terletak pada kestabilan dan konsistensi model dalam klasifikasi, yang tercermin dari sebaran kesalahan prediksi yang relatif kecil dan merata tanpa dominasi kesalahan yang mencolok pada kategori tertentu. Kelebihan lain dari penelitian ini adalah adanya visualisasi confusion matrix dan grafik pelatihan, yang memungkinkan analisis mendalam terkait performa model dan indikasi *overfitting*. Sebaliknya, penelitian sebelumnya tidak menampilkan grafik pelatihan secara eksplisit, sehingga validitas generalisasi model terhadap data baru kurang dapat dianalisis secara menyeluruh.

Lebih lanjut, jika dibandingkan dengan penelitian terdahulu seperti yang dilakukan oleh (Jain & Singh, 2024), yang melaporkan akurasi sebesar 93% untuk klasifikasi penyakit daun zaitun menggunakan arsitektur *Xception*, maka hasil pada penelitian ini memiliki keunggulan tersendiri. Performa tinggi ini tidak hanya ditunjukkan oleh nilai akurasi, tetapi juga oleh distribusi kesalahan klasifikasi yang rendah dan tidak terfokus pada satu kelas saja, yang menjadi indikator kestabilan model.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Setelah melakukan serangkaian eksperimen pada klasifikasi Sawi ada Hama dan Sawi Tanpa Hama menggunakan dua arsitektur Convolutional Neural Network (CNN) yang berbeda, yaitu VGG19 dan Xception, serta menerapkan strategi *tuning hyperparameter* menggunakan Bayesian optimization dan secara manual, evaluasi berdasarkan berbagai skenario pelatihan, maka diperoleh beberapa kesimpulan sebagai berikut :

- a. Kinerja dari akurasi dan efisiensi komputasi pada pemilihan arsitektur model CNN dan konfigurasi parameter berpengaruh signifikan terhadap performa klasifikasi daun sawi. Arsitektur Xception (*lightweight*) menunjukkan performa terbaik dengan akurasi tertinggi mencapai 99,50% pada skenario parameter manual, serta mampu mencapai akurasi validasi 99% hanya pada *epoch* ke-4. Model VGG19 (*deep*) juga menunjukkan performa tinggi dengan akurasi puncak 99% pada 100 *epoch* (BO), namun membutuhkan waktu pelatihan lebih lama untuk mencapai hasil maksimal dibandingkan Xception. Dari sisi efisiensi, Xception lebih unggul dalam menangkap fitur sejak tahap awal pelatihan, sementara VGG19 memberikan keunggulan pada kestabilan kurva pelatihan yang lebih halus dan konsisten.
- b. Perbedaan pendekatan antara ekstraksi fitur dan optimasi hyperparameter menunjukkan bahwa penggunaan *depthwise separable convolution* pada arsitektur Xception lebih efisien dalam mengidentifikasi pola hama. Pendekatan ini mampu menurunkan beban komputasi dibandingkan konvolusi

standar yang digunakan pada VGG19. Optimasi *hyperparameter* melalui *Bayesian Optimization* (BO) berhasil meminimalkan kekurangan inheren masing-masing model, pada xception, kombinasi *learning rate* 0.0001 dan *dropout* 0.5 berhasil meredam fluktuasi pada *epoch* tinggi. Sedangkan pada VGG19, optimasi membantu model mengenali fitur kompleks secara bertahap hingga mencapai titik optimal pada 100 *epoch*. Penerapan augmentasi data (rotasi, *zoom*, *shift*) dan preprocessing (*rescaling* 1./255) terbukti krusial dalam meningkatkan kemampuan generalisasi dan mencegah *overfitting* pada kedua arsitektur.

- c. Akurasi klasifikasi optimal yang dicapai adalah 99,50% pada model Xception (menggunakan 50 dan 100 epoch dengan pengaturan manual) serta 99% pada model VGG19 (menggunakan 100 epoch dengan Bayesian Optimization). Berdasarkan hasil tersebut, model terbaik direkomendasikan adalah Xception karena mampu mencapai akurasi tertinggi dengan efisiensi komputasi yang lebih baik dan proses ekstraksi fitur yang lebih cepat sejak awal pelatihan.

## 5.2. Saran

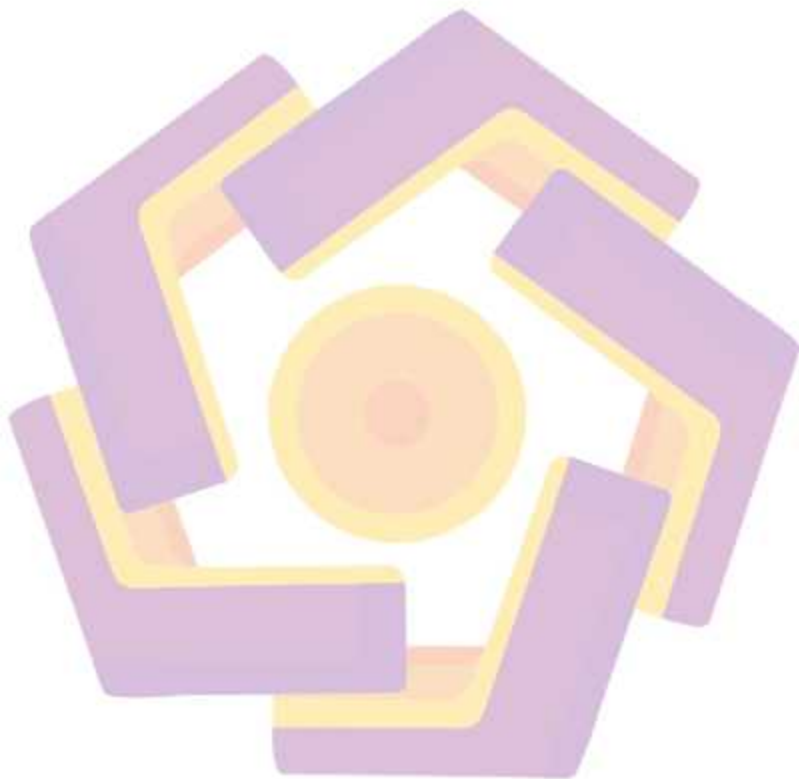
Berdasarkan kesimpulan yang telah kami dipaparkan, penelitian ini masih memiliki kekurangan dan beberapa kendala, maka akan lebih baik jika pada penelitian selanjutnya dapat memperbaiki kekurangan yang ada. Berikut beberapa saran untuk penelitian selanjutnya :

1. Disarankan untuk melakukan peningkatan kualitas citra pada tahap preprocessing, khususnya melalui teknik penajaman atau peningkatan kontras, guna memperjelas gejala hama yang masih samar. Kondisi ini umumnya terjadi pada daun yang baru mulai terserang hama, di mana ciri awal serangan sering

kali menyerupai daun sehat sehingga menyulitkan model dalam melakukan klasifikasi. Dengan penerapan teknik penajaman atau peningkatan kontras citra, bercak-bercak kecil yang sebelumnya kurang terlihat dapat menjadi lebih menonjol, sehingga membantu model dalam membedakan daun sehat dan daun yang terinfeksi secara lebih akurat. Hal ini penting untuk meminimalkan kesalahan klasifikasi, khususnya pada tahap awal serangan hama.

2. Penelitian selanjutnya disarankan untuk mengeksplorasi arsitektur *lightweight* terkini, seperti EfficientNet-B0 dan MobileNetV3, guna memperoleh model dengan keseimbangan terbaik antara akurasi dan efisiensi komputasi untuk perangkat bersumber daya terbatas.
3. Pengembangan klasifikasi multi-kelas dan identifikasi jenis hama disarankan sebagai arah penelitian selanjutnya. Penelitian ini dapat dikembangkan dengan memperluas label klasifikasi dari dua kelas ("Ada Hama" dan "Tanpa Hama") menjadi klasifikasi multi-kelas. Dengan menambahkan kategori jenis hama atau penyakit secara spesifik, seperti serangan ulat, kutu daun, atau bercak daun, sehingga model diharapkan mampu memberikan informasi yang lebih rinci dan aplikatif. Informasi tersebut dapat membantu pengguna dalam menentukan langkah pengendalian yang lebih tepat dan efektif.
4. Pengujian pada data lapangan secara real-time disarankan untuk dilakukan pada penelitian selanjutnya. Model yang telah dihasilkan perlu diuji menggunakan dataset mandiri yang diperoleh langsung dari kondisi lapangan nyata, dengan memperhatikan variasi latar belakang, perubahan intensitas cahaya matahari, serta perbedaan sudut pengambilan gambar. Pengujian ini bertujuan untuk

mengevaluasi kemampuan generalisasi dan ketangguhan model sebelum diimplementasikan dalam bentuk aplikasi deteksi hama berbasis perangkat mobile.



## DAFTAR PUSTAKA

- Abd Algani, Y. M., Marquez Caro, O. J., Robladillo Bravo, L. M., Kaur, C., Al Ansari, M. S., & Kiran Bala, B. (2023). Leaf disease identification and classification using optimized deep learning. *Measurement: Sensors*, 25(September 2022), 100643. <https://doi.org/10.1016/j.measen.2022.100643>
- Al-Lami, H. F. D., You, M. P., Banga, S. S., & Barbetti, M. J. (2023). Novel Resistances Provide New Avenues to Manage Alternaria Leaf Spot (*Alternaria brassicae*) in Canola (*Brassica napus*), Mustard (*B. juncea*), and Other Brassicaceae Crops. *Plant Disease*, 107(2), 372–381. <https://doi.org/10.1094/PDIS-05-22-1153-RE>
- Arge, F. O. R. L., & Mage, C. I. (2015). *V d c n l - s i r*. 1–14.
- AYENI, J. A. (2022). Convolutional Neural Network (CNN): The architecture and applications. *Applied Journal of Physical Science*, 4(4), 42–50. <https://doi.org/10.31248/ajps2022.085>
- Bian, W. (2024). *Real-Fake Face Detection Based on Joint Multi-Layer CNN Structure and Data Augmentation*. *Daml* 2023, 25–29. <https://doi.org/10.5220/0012800300003885>
- Da Rocha, E. L., Rodrigues, L., & Mari, J. F. (2021). *Maize leaf disease classification using convolutional neural networks and hyperparameter optimization*. *October*, 104–110. <https://doi.org/10.5753/wvc.2020.13489>
- Fraiwani, M., & Faouri, E. (2022). On the Automatic Detection and Classification of Skin Cancer Using Deep Transfer Learning. *Sensors*, 22(13). <https://doi.org/10.3390/s22134963>
- Google, C. (2014). *Xception: Deep Learning with Depthwise Separable Convolutions*.
- Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, 10, 19083–19095. <https://doi.org/10.1109/ACCESS.2022.3151048>
- Hridoy, R. H., Armi, A. D., & Hassan, M. A. (2022). Recognition of Mustard Plant Diseases Based on Improved Deep Convolutional Neural Networks. *2022 IEEE Region 10 Symposium, TENSYP 2022*, 1–6. <https://doi.org/10.1109/TENSYP54529.2022.9864487>
- Jain, E., & Singh, A. (2024). Olive Leaf Disease Classification Using the Xception Model: A Deep Learning Approach. *2024 International Conference on Cybernation and Computation, CYBERCOM 2024*, 715–719. <https://doi.org/10.1109/CYBERCOM63683.2024.10803182>

- Lu, J., Tan, L., & Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture (Switzerland)*, 11(8), 1–18. <https://doi.org/10.3390/agriculture11080707>
- Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R., & Poria, S. (2023). A review of deep learning techniques for speech processing. In *Information Fusion* (Vol. 99). <https://doi.org/10.1016/j.inffus.2023.101869>
- Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, 16(November), 100258. <https://doi.org/10.1016/j.array.2022.100258>
- Nazalia, C. L., Palupiningsih, P., Prayitno, B., & Purwanto, Y. S. (2023). Implementation of Convolutional Neural Network Algorithm to Pest Detection in Caisim. *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, 609–614. <https://doi.org/10.1109/ICCoSITE57641.2023.10127792>
- Paymode, A. S., & Malode, V. B. (2022). Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG. *Artificial Intelligence in Agriculture*, 6, 23–33. <https://doi.org/10.1016/j.aiaa.2021.12.002>
- Sakshi, Sharma, C., Sharma, S., Sharma, T., & Gochhait, S. (2024). Green Intelligence: A Sequential CNN Odyssey in Mustard Leaf Disease Detection. *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems, ICETIS 2024, March*, 877–882. <https://doi.org/10.1109/ICETIS61505.2024.10459509>
- Salehin, I., & Kang, D. K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics (Switzerland)*, 12(14). <https://doi.org/10.3390/electronics12143106>
- Samiha, Y. T. (2023). Strategi Pemanfaatan Media Air (Hidroponik) Pada Budidaya Tanaman Kangkung, Pakcoy dan Sawi Sebagai Alternatif Urban Farming. *Journal on Education*, 06(01), 5835–5848.
- Sharma, S., & Chaudhary, P. (2023). Machine learning and deep learning. *Quantum Computing and Artificial Intelligence: Training Machine and Deep Learning Algorithms on Quantum Computers*, 71–84. <https://doi.org/10.1515/9783110791402-004>
- Sood, A., Kumar Sarangi, P., Kumar Sahoo, A., Rani, L., Bajaj, K., & Kumar Agrawal, A. (2023). AI-Driven Mustard Disease Identification: A Multiclass and Binary Classification Approach for Advanced Crop Health Monitoring.

- Proceedings - International Conference on Technological Advancements in Computational Sciences, ICTACS 2023*, 256–261. <https://doi.org/10.1109/ICTACS59847.2023.10390082>
- Srinidhi, C. L., Ciga, O., & Martel, A. L. (2021). Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67, 101813. <https://doi.org/10.1016/j.media.2020.101813>
- Syech Ahmad, M. T. A., & Sugiarto, B. (2023). Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Ikan Cupang Berbasis Mobile. *Digital Transformation Technology*, 3(2), 712–723. <https://doi.org/10.47709/digitech.v3i2.3245>
- Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3). <https://doi.org/10.3390/computation11030052>
- Tian, L. G., Liu, C., Liu, Y., Li, M., Zhang, J. Y., & Duan, H. L. (2020). Research on plant diseases and insect pests identification based on CNN. *IOP Conference Series: Earth and Environmental Science*, 594(1). <https://doi.org/10.1088/1755-1315/594/1/012009>
- Tibaldi, S., Vodola, D., & Tignone, E. (2023). *uantum Engineering Bayesian Optimization for QAOA uantum Engineering*. July.
- Tong, Q., Liang, G., & Bi, J. (2022). Calibrating the adaptive learning rate to improve convergence of ADAM. *Neurocomputing*, 481, 333–356. <https://doi.org/10.1016/j.neucom.2022.01.014>
- Trihardianingsih, L., Sunyoto, A., & Hidayat, T. (2023). Classification of Tea Leaf Diseases Based on ResNet-50 and Inception V3. *Sinkron*, 8(3), 1564–1573. <https://doi.org/10.33395/sinkron.v8i3.12604>
- Yu, R., Liu, S., & Wang, X. (2024). Dataset Distillation: A Comprehensive Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1), 150–170. <https://doi.org/10.1109/TPAMI.2023.3323376>

## LAMPIRAN

