

**KLASIFIKASI PENYAKIT BATANG JAGUNG MENGGUNAKAN
MOBILENETV3 BERBASIS CITRA LAPANGAN**

SKRIPSI

DISUSUN OLEH

LEYDIA APRISA

2209020181



UMSU

Unggul | Cerdas | Terpercaya

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

MEDAN

2026

**KLASIFIKASI PENYAKIT BATANG JAGUNG MENGGUNAKAN
MOBILENETV3 BERBASIS CITRA LAPANGAN**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi, pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

LEYDIA APRISA

NPM. 2209020181

**PROGRAM STUDI TEKNOLOGI INFOMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

LEMBAR PENGESAHAN

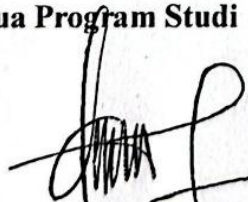
Judul Skripsi : Klasifikasi Penyakit Batang Jagung Menggunakan
MobileNetV3 Berbasis Citra Lapangan
Nama Mahasiswa : Leydia Aprisa
NPM : 2209020181
Program Studi : Teknologi Informasi

Menyetujui
Komisi Pembimbing



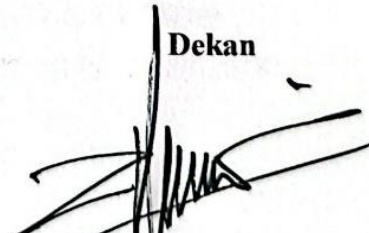
(Farid Akbar Siregar, S.Kom., M.Kom.)
NIDN. 0104049401

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom, M.Kom)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

KLASIFIKASI PENYAKIT BATANG JAGUNG MENGGUNAKAN MOBILENETV3 BERBASIS CITRA LAPANGAN

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, April 2026

Yang membuat pernyataan



Leydia Aprisa

NPM. 2209020181

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Leydia Aprisa
NPM : 2209020181
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**KLASIFIKASI PENYAKIT BATANG JAGUNG MENGGUNAKAN
MOBILENETV3 BERBASIS CITRA LAPANGAN**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, April 2026

Yang membuat pernyataan

Leydia Aprisa

NPM. 2209020181

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Leydia Aprisa
Tempat dan Tanggal Lahir : Boluk, 06 April 2004
Alamat Rumah : Huta I Boluk
Telepon/Faks/HP : 082172812703
E-mail : leydiaaprisa0601@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : TAMAT: 2016
SMP : TAMAT: 2019
SMA : TAMAT: 2022

KATA PENGANTAR



Pendahuluan

Puji syukur kehadiran Allah SWT atas segala rahmat dan kurunianya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “Klasifikasi Penyakit Batang Jagung Menggunakan MobileNetV3 Berbasis Citra Lapangan”. Pnenulis menyampaikan berterima kasih kepada berbagai pihak atas dukungan serta doa dalam proses penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim, M.Pd., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firahti Rizky, S.Kom., M.Kom. Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom. Wakil Dekan III Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. Ketua Program Studi Teknologi Informasi.
6. Bapak Okvi Nugroho, S.Kom., M.Kom. Sekretaris Program Studi Teknologi Informasi.
7. Bapak Farid Akbar Siregar, S.Kom., M.Kom. Dosen pembimbing skripsi.
8. Penulis menyampaikan terima kasih kepada ayah, Kamri Adi, yang meskipun jarang mengungkapkan secara lisan, selalu menunjukkan kasih

sayang melalui tindakan. Penulis meyakini bahwa doa dan dukungan beliau yang tidak terucapkan telah menjadi kekuatan utama hingga skripsi ini bisa dituntaskan.

9. Melalui penuh rasa rindu, penulis mempersembahkan skripsi ini kepada mama tercinta, Almarhumah Nurliana. Kepergian beliau meninggalkan duka yang mendalam, namun kasih sayang, doa, dan pengorbanan semasa hidupnya menjadi cahaya penuntun bagi penulis. Dalam setiap tantangan dan kelelahan, nama beliau selalu menjadi motivasi untuk bertahan dan menyelesaikan karya ini. Skripsi ini merupakan wujud bakti, cinta, dan kerinduan yang tulus kepada Almarhumah Nurliana, yang cintanya tetap abadi meskipun telah tiada.
10. Penulis hendak menyampaikan terima kasih yang setinggi-tingginya pada kakek dan nenek atas ketulusan dan kerja keras tanpa lelah dalam membiayai pendidikan penulis. Di usia yang seharusnya dinikmati untuk beristirahat, pengorbanan mereka menjadi penopang utama agar penulis dapat melanjutkan kuliah. Jerih payah, doa, dan dukungan yang diberikan merupakan hutang budi yang tidak akan pernah terbalaskan.
11. Akhirnya, penulis juga berterima kasih kepada diri sendiri, Leydia Aprisa, atas keteguhan dan ketabahan dalam menghadapi berbagai tantangan, tekanan, dan kehilangan selama proses penyusunan skripsi ini. Penulis berharap ke depannya dapat memberikan ruang untuk kebahagiaan sebagai bentuk penghargaan atas perjuangan yang telah dilalui.

KLASIFIKASI PENYAKIT BATANG JAGUNG MENGGUNAKAN MOBILENETV3 BERBASIS CITRA LAPANGAN

ABSTRAK

Penelitian ini bertujuan “untuk mengembangkan sistem klasifikasi penyakit batang jagung berbasis citra lapangan menggunakan metode *Deep Learning* dengan arsitektur MobileNetV3-Small.” Klasifikasi dilakukan terhadap tiga kelas, yaitu *Bacterial Stalk Rot* (BSR), *Fusarium Stalk Rot* (FSR), dan batang sehat. Dataset dikumpulkan secara langsung dari lahan pertanian dan berjumlah 666 citra, kemudian dilakukan augmentasi sehingga total dataset menjadi 7.663 citra. Tahapan pra-pemrosesan meliputi *resize* citra menjadi 224×224 piksel, normalisasi, dan augmentasi data. Dataset dipisah menjadi data latih, validasi, dan uji dengan perbandingan 70:15:15. Model MobileNetV3-Small diterapkan menggunakan pendekatan *transfer learning* dengan penambahan *Global Average Pooling*, *Batch Normalization*, *Dropout*, dan *Softmax*. Hasil pengujian menunjukkan model memperoleh akurasi yakni 83% disertai skor *precision*, *recall*, dan *F1-score* yang seimbang pada setiap kelas. Hasil tersebut menunjukkan MobileNetV3-Small mampu mengklasifikasikan penyakit batang jagung secara cukup baik dan berpotensi membantu petani dalam deteksi dini penyakit secara otomatis.

Kata Kunci: Klasifikasi Citra; Penyakit Batang Jagung; MobileNetV3-Small; Deep Learning.

CLASSIFICATION OF CORN STALK DISEASES USING MOBILENETV3 BASED ON FIELD IMAGES

ABSTRACT

This study aims to “develop a corn stalk disease classification system based on field images using the Deep Learning method with the MobileNetV3-Small architecture.” Classification was carried out on three classes, namely Bacterial Stalk Rot (BSR), Fusarium Stalk Rot (FSR), and healthy stalks. The dataset was collected directly from agricultural fields and consisted of 666 images, then augmented so that the total dataset became 7,663 images. The pre-processing stages included resizing the images to 224×224 pixels, normalization, and data augmentation. The dataset was separated into training, validation, and test data with a ratio of 70:15:15. The MobileNetV3-Small model was implemented using a transfer learning approach with the addition of Global Average Pooling, Batch Normalization, Dropout, and Softmax. The test results showed that the model obtained an accuracy of 83% with balanced precision, recall, and F1-score scores for each class. These results indicate that MobileNetV3-Small is able to classify corn stalk diseases quite well and has the potential to help farmers in early disease detection automatically.

Keywords: Image Classification; Corn Stalk Diseases; MobileNetV3-Small, Deep Learning.

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
RIWAYAT HIDUP	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	6
BAB II. LANDASAN TEORI	7
2.1 Penyakit Batang Jagung	7
2.2 Citra Digital	11
2.3 <i>Deep Learning</i>	13
2.4 <i>Convolutional Neural Network (CNN)</i>	16
2.5 MobileNetV3	22

2.6 Penelitian Terdahulu	26
2.7 Analisis GAP	31
BAB III. METODOLOGI PENELITIAN	33
3.1 Jenis Penelitian	33
3.2 Alur Penelitian	33
3.3 Pengumpulan Data	35
3.4 Pra-pemrosesan Citra	35
3.5 Pembagian Dataset	35
3.6 Perancangan Sistem	36
3.7 Arsitektur MobileNetV3	41
3.8 Evaluasi Kinerja	42
3.9 Perangkat Penelitian	43
3.10 Jadwal Penelitian	43
BAB IV HASIL DAN PEMBAHASAN	44
4.1 Dataset Penelitian	44
4.2 Pra-Pemrosesan Data.....	45
4.3 Implementasi MobileNetV3	46
4.4 Hasil Pelatihan Model	47
4.5 Evaluasi Model.....	51
4.6 Analisis Hasil Penelitian	57
4.7 Implementasi Sistem	59
BAB V KESIMPULAN DAN SARAN	67
5.1 Kesimpulan	67
5.2 Saran	69

DAFTAR PUSTAKA	70
LAMPIRAN	74

DAFTAR TABEL

Tabel 2.1	Ringkasan Penelitian Terdahulu	26
Tabel 3.2	Jadwal Penelitian	43
Tabel 4.1	Total Dataset Penelitian	44
Tabel 4.2	Pembagian Dataset	45
Tabel 4.3	Pengujian Model	47
Tabel 4.4	Hasil <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i>	56
Tabel 4.5	Hasil Evaluasi Model	57

DAFTAR GAMBAR

Gambar 2.1	Penyakit Batang Jagung <i>Bacterial Stalk Rot</i> (BSR)	8
Gambar 2.2	Penyakit Batang Jagung <i>Fusarium Stalk Rot</i> (FSR)	9
Gambar 2.3	Arsitektur CNN	16
Gambar 2.4	Arsitektur MobieNetV3	24
Gambar 3.1	Alur Penelitian	34
Gambar 3.2	Rancangan Sistem	37
Gambar 3.3	<i>Use Case Diagram</i>	38
Gambar 3.4	<i>Activity Diagram</i>	39
Gambar 3.5	<i>Design Halaman Upload</i>	40
Gambar 3.6	<i>Design Halaman Hasil</i>	40
Gambar 4.1	Grafik Akurasi Model	49
Gambar 4.2	Grafik Loss Model	50
Gambar 4.3	Hasil <i>Confusion Matrix</i>	51
Gambar 4.4	Halaman Tampilan Upload	59
Gambar 4.5	Halaman Tampilan Kamera	60
Gambar 4.6	Halaman Tampilan Pemilihan Gambar	61
Gambar 4.7	Halaman Tampilan yang Berhasil diunggah	62
Gambar 4.8	Halaman Tampilan proses Auto Crop	63
Gambar 4.9	Hasil Analisis Kelas <i>Bacterial Stalk Rot</i> (BSR)	64
Gambar 4.10	Hasil Analisis Kelas <i>Fusarium Stalk Rot</i> (FSR)	65
Gambar 4.11	Hasil Analisis Kelas Batang Jagung Sehat	66

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Jagung (*Zea mays* L.) adalah tanaman serealia yang sering ditanam oleh petani di Indonesia untuk memenuhi kebutuhan makanan dan pakan hewan. Jagung saat ini merupakan komoditas yang sangat penting bagi negara dan dalam kategori tanaman pangan di Indonesia, jagung adalah komoditas utama kedua setelah beras (Suriani et al., 2021). Berdasarkan informasi dari Badan Pusat Statistik (BPS), diperkirakan bahwa produksi Jagung pada Januari 2025 mencapai 1,24 juta ton, angka ini menunjukkan peningkatan sebanyak 0,73 juta ton atau 141,98 persen jika dibandingkan dengan produksi pada Januari 2024 yang hanya mencapai 0,51 juta ton. Peningkatan dalam produksi ini menunjukkan bahwa permintaan dan penghasilan jagung nasional terus bertambah, sehingga memerlukan upaya yang berkelanjutan untuk memelihara serta meningkatkan produktivitas jagung.

Namun, usaha untuk meningkatkan hasil jagung ini tidak terlepas dari banyak masalah lingkungan baik yang bersifat abiotik ataupun biotik, seperti perubahan iklim, kualitas benih yang kurang baik, serangan hama, serta penyakit yang disebabkan oleh jamur, dan bakteri. Faktor-faktor tersebut dapat menghalangi perkembangan tanaman dan menurunkan hasil panen. Salah satu penyakit yang menyerang tanaman jagung di Indonesia adalah busuk batang bakteri atau *Bacterial Stalk Rot* (BSR). Penyakit ini dapat mengakibatkan lunaknya dan rusaknya jaringan batang sehingga mengganggu pengangkutan nutrisi di dalam tanaman dan meningkatkan resiko tanaman roboh sebelum waktu panen (Arief et al., 2023).

Selain itu, penyakit *Fusarium Stalk Rot* yang ditimbulkan jamur *fusarium* spp. ialah sebuah penyakit batang yang umum di temukan pada tanaman jagung. Penyakit ini dapat menghalangi transportasi unsur hara ke bagian-bagian tanaman, sehingga pertumbuhan tanaman menjadi terhambat dan dapat menyebabkan gagal panen (Harish et al., 2024). Kedua penyakit batang jagung ini sering di temukan di lapangan sehingga para petani mengalami kesulitan dalam membedakan secara langsung.

Di sisi lain, kondisi batang jagung yang sehat juga perlu diperhatikan sebagai pembanding dalam proses identifikasi penyakit batang jagung. Batang jagung yang sehat adalah kondisi tanaman yang tanpa adanya serangan penyakit dan berperan penting dalam menopang tanaman serta mendukung proses penyalurann nutrisi. Oleh karena itu, kemampuan dalam membedakan antara batang jagung sehat dan batang jagung yang terserang penyakit menjadi hal yang penting dalam upaya pengendalian penyakit secara tepat. Pengelolaan tanaman yang berbasis deteksi dini diharapkan dapat membantu menjaga produktivitas jagung tetap optimal selama masa pertumbuhan (Wahyu et al., 2023)

Dari observasi awal yang dilangsungkan penulis di Desa Boluk, Kecamatan. Bosar Maligas, Kabupaten. Simalungun, diketahui bahwa petani jagung masih mengidentifikasi penyakit batang jagung berdasarkan pengamatan visual tanpa menggunakan alat atau bantuan teknologi. Kondisi ini menyebabkan petani mengalami kesulitan dalam membedakan antara penyakit batang jagung yang disebabkan oleh *Bacterial Stalk Rot* dan *Fusarium Stalk Rot*.

Dalam sisi pengembangan ilmu pengetahuan dan teknologi, banyak penelitian sebelumnya telah menggunakan teknologi *Deep Learning* terutama

Convolutional Neural Network (CNN) dalam mengklasifikasikan penyakit pada jagung. Namun, sebagian besar studi masih berfokus pada penyakit yang menyerang daun jagung. Berdasarkan studi yang dilangsungkan oleh (Iswantoro & Handayani UN, 2022) melakukan klasifikasi penyakit tumbuhan jagug berdasarkan citra daun yaitu hawar dan karat daun. Meskipun memberikan hasil yang baik, penelitian tersebut belum membahas terkait permasalahan penyakit pada batang jagung, padahal batang jagung juga berperan penting dalam menompang tanaman dan menyalurkan nutrisi.

Penyakit batang jagung adalah salah satu permasalahan yang sering di hadapi oleh petani karena menghambat pertumbuhan tanaman dan menyebabkan tanaman roboh sebelum waktu panen. Situasi ini secara langsung mempengaruhi penurunan hasil panen dan berpotensi kerugian bagi petani. Oleh karena itu di perlukan upaya dini yang lebih mudah, cepat dan akurat untuk membantu petani dalam mengenali penyakit batang jagung sejak awal. (Arief et al., 2023).

Dalam menjawab permasalahan yang ada, studi ini menawarkan pemanfaatan teknologi pengolahan citra dan *deep learning* sebagai alat bantu dalam mengidentifikasi penyakit batang jagung. Melalui pengembangan sistem klasifikasi berbasis citra lapangan menggunakan model MobileNetV3, diharapkan kondisi batang jagung dapat dikenali secara otomatis baik dalam kondisi sehat mauput terserang penyakit *Bacterial Stalk Rot* dan *Fsarium Stalk Rot*. Pendekatan ini di harapkan dapat mengambil langkah pengendalian lebih cepat dan tepat guna menjaga produkifitas tanaman jagung. Dengan demikian, studi ini bertujuan ”untuk mengembangkan dan mengevaluasi sistem klasifikasi batang jagung berbasis citra lapangan menggunakan MobileNetV3.”

1.2. Rumusan Masalah

Dari permasalahan yang sudah diuraikan di latar belakang, bisa dirumuskan bahwa identifikasi penyakit batang jagung di lapangan saat ini masih biasa dilangsungkan secara visual oleh petani, yang menjadikannya berpotensi menimbulkan kesalahan dalam membedakan kondisi batang jagung yang terserang *Bacterial Stalk Rot* dan *Fusarium Stalk Rot*. Disisi lain, pemanfaatan teknologi *Deep Learning* khususnya penggunaan MobileNetV3 untuk klasifikasi penyakit batang jagung masih relatif terbatas dan belum banyak diterapkan. Selain itu tingkat kinerja MobileNetV3 dalam klasifikasi kondisi batang jagung yang terserang *Bacterial Stalk Rot* dan *Fusarium Stalk Rot* saat ini belum di ketahui.

1.3. Batasan Masalah

Penelitian ini mempunyai batasan permasalahan yang dikaji supaya bisa menuntaskan permasalahan dasar. Berikut batasan yang diterapkan :

1. Penelitian ini di berfokus pada klasifikasi kondisi batang jagung, bukan bagian tanaman lain seperti daun, akar, atau tongkol.
2. Penyakit batang jagung yang diklasifikasikan dalam peneliian ini terdiri dari batang sehat, *Bacterial Stalk Rot* (BSR), dan *Fusarium Stalk Rot* (FSR).
3. Dataset yang berhasil dikumpulkan pada studi ini yakni ada 666 citra yang tersusun atas tiga kelas, yakni *Bacterial Stalk Rot* (BSR), *Fusarium Stalk Rot* (FSR), dan batang sehat.
4. Untuk meningkatkan variasi data dilakukan proses augmentasi sehingga total dataset menjadi 7.663 citra.

5. Metode klasifikasi yang diterapkan dalam penelitian ini adalah model deep learning MobileNetV3.
6. Penelitian ini membatasi pada pengembangan dan evaluasi kinerja model, tanpa membahas tindakan penanganan penyakit di lapangan.
7. Evaluasi kinerja model yang dilakukan menggunakan pengujian kuantitatif, seperti akurasi dan metrik evaluasi lainnya.
8. Sistem klasifikasi pada penelitian ini di implementasikan dalam bentuk website sederhana yang di gunakan untuk mendukung proses pengujian dan visualisai hasil klasifikasi.
9. Bahasa pemrograman yang diterapkan pada studi ini ialah Bahasa Pemrograman Python.
10. Pra-pemrosesan citra yang dilakukan mencakup proses *resize*, *normalisasi*, dan *augmentasi*.
11. Penelitian ini tidak mengklasifikasikan penyakit lainnya seperti *Antranoksa*, *Gibberella*, *Diplodia*, Arang, dan *Pythium*.

1.4. Tujuan Penelitian

Dari rumusan masalah yang sudah di jabarkan, tujuan yang hendak di capai melalui penelitian ini antara lain ialah seperti berikut :

1. Mengembangkan sistem klasifikasi penyakit batang jagung berbasis citra lapangan dengan menggunakan model MobileNetV3 sebagai alat bantu dalam mengidentifikasi kondisi batang jagung.

2. Mengidentifikasi kondisi batang jagung ke dalam tiga kelas yaitu batang sehat, *Bacterial Stalk Rot*, dan *Fusarium Stalk Rot* berdasarkan citra yang di ambil di lapangan.
3. Mengetahui kinerja MobileNetV3 dalam mengklasifikasi penyakit batang jagung.

1.5. Manfaat Penelitian

Studi ini diproyeksikan bisa menyediakan manfaat untuk banyak pihak secara langsung ataupun tidak. Manfaat penelitian ini ialah:

1. Manfaat bagi Petani

Manfaat penelitian ini bisa membantu petani jagung dalam mendeteksi penyakit batang jagung yang terkena penyakit *Bacterial Stalk Rot* dan *Fusarium Stalk Rot*.

2. Manfaat bagi peneliti

Penelitian ini dapat membantu pengetahuan penulis tentang penerapan *Deep Learning* serta pengeolahan citra untuk klasifikasi penyakit batang jagung yang masih jarang diteliti.

3. Manfaat bagi peneliti selanjutnya

Temuan studi ini bisa menyediakan manfaat bagi penelitian lainnya dengan memberikan referensi mengenai klasifikasi penyakit tanaman berbasis citra.

BAB II

LANDASAN TEORI

2.1 Penyakit Batang Jagung

2.1.1 Bacterial Stalk Rot (BSR)

Bacterial Stalk Rot (BSR) ialah penyakit mendasar pada jagung yang ditimbulkan bakteri *Dickeya zae* (sebelumnya dikenal sebagai *Erwinia chrysanthemi* pv. *zae*) dan dapat menimbulkan kerugian hingga 98% tergantung kondisi iklim, terutama pada lingkungan dengan suhu dan kelembaban tinggi yang mendukung perkembangan penyakit ini (Jha & Prajapati, 2024). Patogen ini bersifat tular tanah dan memiliki jangkauan inang yang luas, tidak hanya menyerang jagung, tetapi juga tanaman lain seperti talas, tebu, padi, dan pisang (Mirsam et al., 2025). Infeksi *Dickeya zae* menyebabkan busuk basah pada jaringan batang yang ditandai dengan pelunakan jaringan, perubahan warna menjadi cokelat, serta munculnya bau tidak sedap. Kondisi tersebut mengganggu aliran nutrisi sehingga tanaman mudah roboh, pengisian tongkol menjadi tidak sempurna, dan pada tingkat serangan berat tanaman dapat mati sebelum mencapai kematangan biologis (Suriani et al., 2021).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Bacterial Stalk Rot* (BSR) yang disebabkan oleh *Dickeya zae* merupakan penyakit serius yang berpotensi menimbulkan kerusakan besar pada tanaman jagung, terutama pada kondisi iklim dengan suhu dan kelembaban tinggi yang mempercepat penyebaran infeksi (Jha & Prajapati, 2024). Gejala seperti pembusukan batang, perubahan warna jaringan, dan bau tidak sedap mencerminkan dampak infeksi yang berat terhadap pertumbuhan dan hasil tanaman (Suriani et al., 2021). Selain itu, karena

patogen ini juga mampu menginfeksi berbagai tanaman lain, penanganan yang cepat dan tepat ialah tahapan penting dalam mencegah kerugian yang luas (Mirsam et al., 2025).



Gambar 2.1 Penyakit batang jagung *Bacterial Stalk Rot*

Sumber: Dokumentasi Peneliti (2026)

2.1.2 *Fusarium Stalk Rot (FSR)*

Fusarium Stalk Rot (FSR) merupakan penyakit yang disebabkan oleh jamur patogen dari jenis *Fusarium* yang sangat mengganggu pertumbuhan tanaman jagung, menurunkan nilai gizi, serta mengurangi hasil panen secara keseluruhan. Beberapa spesies yang terlibat antara lain "*Fusarium graminearum*, *F. culmorum*, *F. avenaceum*, *F. poae*, *F. sporotrichioides*, *F. equiseti*, *F. verticillioides*, dan *F. Proliferatum*", yang diketahui mampu menimbulkan berbagai penyakit seperti layu vaskular, hawar kepala, busuk akar, dan busuk batang hingga menyebabkan kematian tanaman (Navale et al., 2023). Gejala *Fusarium Stalk Rot (FSR)* umumnya ditandai dengan pertumbuhan tanaman yang menurun, pembusukan pelepah daun, serta kerusakan jaringan bagian dalam batang yang disertai munculnya garis coklat pada ruas bawah. Infeksi ini mengganggu penyaluran air

dan unsur hara ke daun dan tongkol, sehingga tanaman menunjukkan gejala layu, perubahan warna daun dari hijau menjadi cokelat, serta pengeringan pada bagian batang bawah akibat kerusakan dan pelapukan jaringan empulur hingga tampak robek (Tiru et al., 2022).

Dari pemaparan tersebut, bisa disimpulkan *Fusarium Stalk Rot* (FSR) yang disebabkan oleh berbagai spesies *Fusarium* seperti *F. graminearum* dan *F. culmorum* merupakan penyakit serius yang dapat menurunkan hasil panen jagung secara signifikan (Navale et al., 2023). Penyakit ini ditandai dengan pertumbuhan tanaman yang melemah, pembusukan pelepah daun, serta kerusakan jaringan batang yang tampak sebagai garis-garis cokelat pada ruas batang. Infeksi jamur mengganggu penyaluran air dan unsur hara, sehingga tanaman mengalami layu, perubahan warna daun, dan pengeringan pada batang bagian bawah. Kondisi tersebut dapat berujung pada kematian dini tanaman, yang menjadikannya diperlukan penanganan yang cepat dan tepat dalam mencegah kerugian yang lebih besar (Tiru et al., 2022)..



Gambar 2.2 Penyakit batang jagung *Fusarium Stalk Rot*

Sumber: Dokumentasi Peneliti (2026)

2.1.3 Batang Jagung Sehat

Batang jagung (*Zea mays* L.) yang sehat tidak hanya berperan sebagai penopang utama tanaman, tetapi juga memiliki berkas *vaskuler* yang mendukung distribusi air, unsur hara, dan metabolit ke seluruh bagian tanaman. Karakteristik berkas *vaskuler* ini berkaitan erat dengan struktur jaringan batang dan komponen hasil, sehingga menunjukkan bahwa integritas struktural batang sangat penting bagi fungsi fisiologis dan produktivitas tanaman (Y. Wang et al., 2025). Batang yang sehat juga mampu menyalurkan cadangan asimilat dari bagian bawah ke biji tanpa kehilangan kekuatan struktural, dengan ciri bagian bawah yang kokoh, berdiameter besar, tidak mudah patah, dan seimbang dengan bagian atas tanaman, yang dipengaruhi oleh alokasi bahan kering dan nitrogen sebagai indikator kesehatan serta fungsi mekaniknya (Ruiz et al., 2025). Selain itu, *xilem* pada batang jagung berperan penting dalam konduktivitas air dari akar ke bagian atas tanaman untuk mendukung pertumbuhan normal dan respons terhadap kondisi lingkungan (Li et al., 2024).

Dari beberapa pemaparan tersebut, bisa dijabarkan batang jagung yang sehat memegang peran penting dalam menunjang pertumbuhan dan produktivitas tanaman, karena berkas *vaskuler* yang berkembang baik memungkinkan distribusi air, unsur hara, dan metabolit secara optimal ke seluruh bagian tanaman. Batang yang kokoh, tidak mudah patah, serta memiliki alokasi bahan kering dan nitrogen yang seimbang mencerminkan kemampuannya menopang tanaman sekaligus menyalurkan cadangan nutrisi secara efektif. Kinerja *xilem* yang optimal juga memastikan pengangkutan air dari akar ke bagian atas berlangsung lancar, sehingga tanaman dapat tumbuh normal dan beradaptasi dengan kondisi lingkungan. Oleh

karena itu, kondisi batang jagung yang sehat dapat dijadikan acuan pembandingan untuk mengidentifikasi perbedaan antara tanaman normal dan tanaman yang terserang penyakit dalam penelitian ini (Li et al., 2024; Y. Wang et al., 2025)..

2.2 Citra Digital

Citra digital ialah perwujudan visual objek dunia nyata pada format digital yang tersusun dari piksel dengan nilai numerik tertentu untuk merepresentasikan warna atau tingkat kecerahan, dan dapat diperoleh melalui perangkat seperti kamera digital, pemindai, maupun hasil simulasi komputer (Dijaya, 2023). Feature extraction merupakan tahapan penting pada pengolahan citra digital yang dimaksudkan dalam mengekstrak informasi signifikan dari citra yang menjadikannya dapat digunakan dalam proses pengenalan pola dan klasifikasi. Tahap ini berperan dalam mengubah representasi data citra menjadi bentuk yang lebih terstruktur dan mudah dianalisis (Hutagalung et al., 2025). Selain menampilkan bentuk visual, citra digital juga memuat informasi penting seperti warna, tekstur, dan pola permukaan, di mana perbedaan intensitas piksel dapat menggambarkan variasi kondisi fisik objek, termasuk perubahan warna jaringan dan tekstur tanaman sebagai indikator awal serangan penyakit (Santosa et al., 2023). Oleh karena itu, citra digital dapat dimanfaatkan sebagai input utama dalam sistem klasifikasi penyakit tanaman, karena pengolahan citra secara komputasional mampu merepresentasikan kondisi kesehatan tanaman dan mendukung pengambilan keputusan otomatis, sehingga berperan penting sebagai sumber data utama dalam sistem klasifikasi penyakit berbasis teknologi (Rahma Hidayat & Jemakmum, 2025).

Dari beberapa pemaparan diatas, Pemanfaatan citra digital untuk klasifikasi penyakit tanaman telah banyak dilakukan, kebanyakan studi terdahulu masih berorientasi pada bagian daun, penelitian yang secara khusus memanfaatkan citra digital batang jagung terutama pada klasifikasi penyakit *Bacterial Stalk Rot* dan *Fusarium Stalk Rot* masih relatif terbatas. Oleh karena itu penelitian ini menggunakan citra digital batang jagung sebagai dasar dalam proses klasifikasi kondisi tanaman untuk mendukung identifikasi penyakit batang jagung secara objektif dan konsisten berbasis teknologi (Rahma Hidayat & Jemakmum, 2025; Santosa et al., 2023).

2.2.1 Pra-pemrosesan Citra

Pra-pemrosesan dimaksudkan dalam menyesuaikan format citra yang menjadikannya bisa diolah secara maksimal. Ada beberapa tahap proses yaitu :

1. *Resize* : pada proses ini citra akan di ubah ukurannya menjadi 224x224 piksel. Proses ini bertujuan untuk menyamakan seluruh citra sesuai dengan kebutuhan (Rahma Hidayat & Jemakmum, 2025).
2. *Normalisasi* : sesudah tahapan *resize*, citra di *normalisasi* sehingga rentang nilai piksel menjadi antara 0 dan 1. Hal ini bertujuan untuk mempercepat proses pelatihan, melalui *normalisasi* ini, model bisa melangsungkan pembelajaran dari citra dengan kian efisien dan menginjak angka akurasi yang juga kian besar (Rahma Hidayat & Jemakmum, 2025)
3. *Augmentasi* : adanya augmentasi ini untuk memperkaya variasi fitur, dengan beberapa teknik *augmentasi* yang di terapkan yaitu rotasi,

pergeseran, memperbesar, serta membalik secara horizontal. (Toyib et al., 2024)

2.3 Deep Learning

2.3.1 Pengertian Deep Learning

Deep Learning (DL) ialah metode dalam *Machine Learning* yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan tersembunyi untuk mempelajari pola data secara hierarkis, dari fitur sederhana sampai fitur yang lebih kompleks. Pendekatan ini dirancang untuk menangani data berskala besar dan tidak terstruktur seperti citra, video, dan teks, tanpa membutuhkan rancangan fitur manual (Mustafa, 2023; Upadhyay et al., 2025). Dalam pengolahan citra, *Deep Learning* bisa mengekstraksi fitur dengan otomatis langsung dari data mentah dan mempelajari pola visual secara bertahap, sehingga dapat mengenali karakteristik objek dengan lebih akurat. Kemampuan ini menjadikan *Deep Learning* sebagai salah satu pendekatan paling kuat dalam sistem kecerdasan buatan, khususnya untuk tugas klasifikasi dan pengenalan pola berbasis citra (S. Wang et al., 2025).

Berdasarkan uraian tersebut, bisa dijabarkan *Deep Learning* ialah metode pembelajaran mesin yang dirancang dalam mempelajari pola data secara mendalam melalui jaringan saraf berlapis. Kemampuannya mengekstraksi fitur secara otomatis membuat *Deep Learning* sangat efektif dalam menangani data tidak terstruktur seperti citra digital. Dalam pengolahan citra, pendekatan ini mampu mengenali karakteristik visual objek secara bertahap dan lebih akurat, sehingga memberikan kinerja unggul dalam klasifikasi dan pengenalan pola. Oleh karena itu, *Deep Learning* dinilai tepat dan relevan untuk diterapkan pada pembaharuan sistem

klasifikasi berbasis citra, terutama pada konteks studi ini (Mustafa, 2023; Upadhyay et al., 2025; S. Wang et al., 2025).

2.3.2 Perbedaan Deep Learning dan Machine Learning

Machine Learning umumnya memerlukan ekstraksi fitur yang dirancang secara manual berdasarkan pengetahuan manusia, sehingga keberhasilan model sangat bergantung pada kualitas fitur yang digunakan dan performanya cenderung menurun saat menghadapi data yang kompleks dengan variasi tinggi, sehingga lebih sesuai untuk data terstruktur berskala relatif kecil (Lacatusu et al., 2022). Sebaliknya, *Deep Learning* mampu mempelajari fitur secara otomatis melalui pembelajaran berlapis, sehingga lebih efektif menangani data berskala besar dan tidak terstruktur, serta mengindikasikan performa yang stabil pada citra lapangan dengan variasi pencahayaan, sudut pengambilan, dan latar belakang. Maka, *Deep Learning* dinilai lebih unggul dibandingkan *Machine Learning* tradisional dalam klasifikasi citra, mencakup di aspek pertanian dan deteksi penyakit pada tumbuhan (Das et al., 2024).

Dari beberapa pemaparan diatas, dapat di simpulkan bahwa *Machine Learning* cenderung bergantung pada fitur yang di rancang secara manual sehingga kurang fleksibel ketika menghadapi data yang kompleks. Sementara itu, pada *Deep Learning* mampu mempelajari fitur secara otomatis dan lebih adaptif terhadap variasi dan efektif untuk tugas klasifikasi citra, termasuk dalam mendeteksi penyakit tanaman (Das et al., 2024; Lacatusu et al., 2022)

2.3.3 Penerapan Deep Learning dalam Pengelolaan Citra Digital

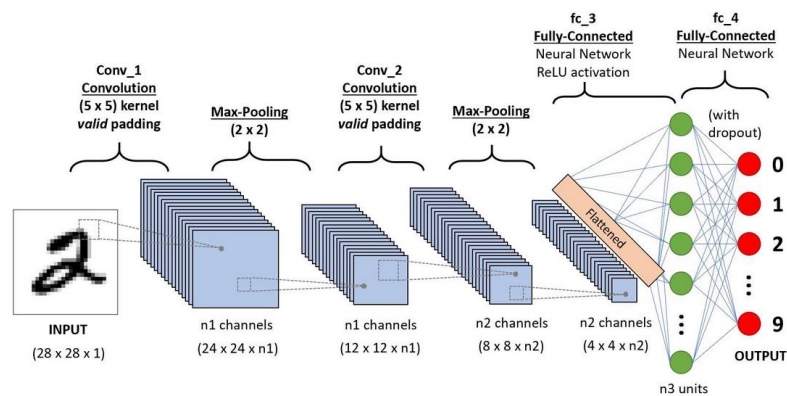
Dalam pengolahan citra digital, *Deep Learning* memproses citra sebagai input utama untuk mengekstraksi fitur visual secara otomatis, di mana lapisan awal mempelajari pola dasar seperti tepi dan tekstur, sedangkan lapisan yang lebih dalam menangkap karakteristik objek yang lebih kompleks tanpa intervensi manual (Upadhyay et al., 2025). Pendekatan ini banyak diterapkan di berbagai bidang, termasuk pertanian, dengan model seperti CNN yang mampu mengklasifikasikan kondisi tanaman dari citra lapangan dan membedakan tanaman sehat serta sakit dengan akurasi tinggi (S. Wang et al., 2025). Meski sangat efektif dan akurasinya meningkat seiring bertambahnya data latih, deep learning juga menghadapi tantangan berupa kebutuhan dataset besar dan berkualitas, risiko *overfitting*, serta komputasi yang lebih intensif dibandingkan metode tradisional (Das et al., 2024; Lacatusu et al., 2022).

Dari beberapa penelitian yang telah dijelaskan, dapat di simpulkan bahwa *Deep Learning* merupakan pendekatan yang sangat efektif dalam pengelolaan citra digital karena mampu mengekstraksi fitur visual secara otomatis dan memahami pola kompleks citra. Penerapannya khusus melalui model *Convolutional Neural Network* (CNN), terbukti mampu membedakan kondisi tanaman sehat dan terserang penyakit dengan akurasi tinggi meskipun citra diambil dalam kondisi lapangan yang beragam. Meskipun demikian, penggunaan deep learning tetap memerlukan perhatian terhadap ketersediaan data latih yang memadai serta kebutuhan komputasi yang cukup besar agar model dapat bekerja secara optimal dan menghasilkan klasifikasi yang baik (Das et al., 2024; Upadhyay et al., 2025; Wang et al., 2025).

2.4 Convolutional Neural Network (CNN)

2.4.1 Pengertian Convolutional Neural Network

CNN ialah jenis jaringan neural yang dirancang dalam klasifikasi gambar, secara otomatis dan adaptif belajar fitur spasial hierarkis dari data gambar melalui lapisan konvolusi dan pooling. Lapisan ini memungkinkan CNN mengenali fitur tingkat rendah seperti tepi dan sudut, serta pola tingkat tinggi seperti bentuk dan objek. CNN telah menjadi pusat aplikasi pengenalan dan klasifikasi gambar, memberikan hasil unggul dalam pengolahan citra dibandingkan metode sebelumnya (Taye, 2023).



Gambar 2.3 Arsitektur CNN

Sumber: <https://evbn.org/convolutional-neural-network-architecture-cnn-architecture-1678014845/>

CNN bekerja melalui susunan lapisan dasar berupa *convolutional layer*, *pooling layer*, dan *fully connected layer*, di mana lapisan konvolusi berperan sebagai pendeteksi pola visual, sedangkan *pooling layer* mereduksi dimensi data dalam meningkatkan efisiensi komputasi. Kombinasi lapisan ini memungkinkan CNN membangun representasi fitur yang hierarkis dan robust terhadap variasi skala, pencahayaan, serta sudut pengambilan citra, sehingga memberikan performa

yang unggul dibandingkan metode pembelajaran mesin konvensional, khususnya pada citra yang kompleks dan tidak terstruktur (Taye, 2023). Oleh karena itu, CNN menjadi fondasi utama dalam berbagai arsitektur *Deep Learning* modern, termasuk model ringan seperti MobileNetV3 untuk aplikasi citra lapangan dan perangkat dengan keterbatasan sumber daya, serta diakui sangat efektif dalam klasifikasi gambar karena mampu mempelajari fitur spasial dari tingkat rendah seperti tepi hingga pola tingkat tinggi seperti bentuk dan objek secara otomatis dan adaptif (Taye, 2023)

2.4.2 Arsitektur Convolutional Neural Network

CNN ialah arsitektur *Deep Learning* yang tersusun atas beberapa lapisan berurutan, dirancang untuk mengekstrak dan mempelajari fitur visual dari gambar. Setiap lapisan berkontribusi dalam membangun representasi fitur secara bertahap, mulai dari elemen sederhana yakni tepi ataupun pola kompleks pada format objek. Struktur ini membuat CNN sangat berguna untuk tugas klasifikasi gambar yang rumit, termasuk mendeteksi penyakit pada batang jagung melalui citra lapangan (Taye, 2023). Secara umum, Arsitektur CNN terdiri dari beberapa lapisan, yaitu :

1. Convolutional Layer

Lapisan konvolusi ini adalah inti CNN yang mengekstrak fitur visual dengan menerapkan filter atau kernel yang digeser seluruh gambar untuk mendeteksi pola seperti tekstur, dan bentuk. Pada lapisan awal, model yang dipelajari fitur masih dasar, sedangkan lapisan dalam fitur yang dipelajari lebih spesifik dan kompleks (Taye, 2023).

Adapun rumus pada lapisan ini ialah seperti berikut :

$$(I * K)(i, j) = \sum_{m, n} I(i + m, j + n) K(m, n)$$

Ket:

I = input (gambar),

K = filter (kernel) dan

i,j = koordinat dari output feature map

m,n = koordina dari filter (Anggara et al., 2025)

2. *Activation Layer*

Lapisan aktivasi ini berfungsi untuk memperkenalkan sifar non linear ke dalam jaringan CNN agar model dapat mempelajari hubungan kompleks. Fungsi aktivasi yang biasa diterapkan pada *convolutional layer* adalah *Rectified Linear Unit* (ReLU) sebab sederhana secara komputasi, mampu mempercepat proses pelatihan, dan mengurangi permasalahan *vanishing gradient* (Taye, 2023). Adapun rumus pada lapisan ini adalah sebagai berikut :

$$f(x) = \max(0, x)$$

Dimana, x adalah output dari lapisan konvolusi. (Anggara et al., 2025).

3. *Pooling Layer*

Lapisan *Pooling* diterapkan dalam menekan ukuran *spasial feature map* yang dikembangkan *convolutional layer* tanpa menghilangkan informasi penting. Lapisan ini membantu menurunkan kompleksitas komputasi, mengurangi jumlah parameter, dan meningkatkan ketahanan

model terhadap pergeseran kecil posisi objek pada citra (Taye, 2023).

Adapun rumus pada lapisan Pooling ini adalah sebagai berikut:

$$P(i, j) = \max_{0 \leq m < p} \max_{0 \leq n < p} I(i + m, j + n)$$

Ket:

P = hasil *pooling*

p = ukuran pooling window (Anggara et al., 2025)

4. *Fully Connected Layer*

Lapisan *fully connected* adalah tahap akhir, yang meratakan peta fitur menjadi vektor satu dimensi dan melakukan klasifikasi berdasarkan fitur yang diekstrak. Setiap *neuron* terhubung penuh dengan lapisan sebelumnya. (Taye, 2023). Adapun rumus pada lapisan ini adalah sebagai berikut:

$$y = W \cdot x + b$$

Ket:

y = output

w = matriks bobot

x = input

b = bias (Anggara et al., 2025)

5. *Softmax Layer*

Lapisan *softmax* adalah jenis aktivasi yang di gunakan dalam CNN, lapisan *softmax* ini biasanya menghitung probabilitas pada setiap kategori. Lapisan *softmax* ini saat digunakan dalam model klasifikasi, akan mengembalikan setiap kelas dan target akan mempunyai probabilitas

tinggi. (Rizki Ramadhani et al., 2022). Adapun rumus yang di gunakan pada lapisan softmax ini yaitu :

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

Dimana $\sigma(z)_i$ adalah probabilitas dari kelas i dan k adalah jumlah kelas (Anggara et al., 2025).

Selain menggunakan lapisan dasar CNN, performa model juga dapat ditingkatkan melalui penerapan teknik optimasi seperti fine tuning, batch normalization, dan dropout. Batch normalization digunakan untuk menstabilkan distribusi data selama proses pelatihan sehingga mempercepat konvergensi model, sedangkan dropout digunakan untuk mengurangi overfitting dengan menonaktifkan sebagian neuron secara acak. Studi yang dilangsungkan oleh (Wibowo et al., 2025) menjabarkan model CNN yang dioptimasi menggunakan fine tuning, batch normalization, dan dropout mampu meningkatkan akurasi klasifikasi tanaman herbal secara signifikan dibandingkan model CNN standar.

2.4.3 Cara Kerja CNN dalam Klasifikasi Citra

Dalam CNN, data gambar terlebih dahulu masuk melalui lapisan input sebagai matriks piksel (*grayscale* atau RGB) tanpa perubahan komputasi untuk menjaga informasi spasial tetap utuh (Taye, 2023). Selanjutnya, lapisan *convolutional* mengekstraksi fitur dasar dengan menerapkan filter untuk menghasilkan *feature map*, yang kemudian diproses oleh fungsi aktivasi ReLU guna menambahkan nonlinieritas agar model mampu mempelajari fitur yang lebih kompleks (Taye, 2023). *Feature map* tersebut lalu diperkecil melalui lapisan pooling untuk mengurangi dimensi data, menekan *overfitting*, dan mempercepat

komputasi tanpa menghilangkan informasi penting. Hasilnya diteruskan ke *fully connected layer* yang memediasi seluruh fitur yang sudah diekstraksi dalam menghasilkan *output* klasifikasi, yakni tiap *neuron* terhubung ke *neuron* pada lapisan sebelumnya (Rizki Ramadhani et al., 2022; Taye, 2023). Terakhir, lapisan *softmax* mengubah *output* menjadi probabilitas dan memilih nilai tertinggi sebagai hasil prediksi, sehingga CNN bisa mengetahui pola dan objek dalam gambar secara otomatis disertai akurasi tinggi (Taye, 2023).

2.4.4 Penerapan dan Keunggulan CNN dalam Citra Digital

Penerapan CNN biasa digunakan pada pemrosesan citra digital, khususnya dalam klasifikasi citra dengan kompleksitas visual tinggi, termasuk di bidang pertanian dalam mengidentifikasi kondisi tanaman dan membedakan tanaman sehat serta yang terserang penyakit berdasarkan citra lapangan. CNN mampu melakukan analisis citra secara otomatis karena dapat mengekstrak fitur visual langsung dari data tanpa perancangan fitur manual, yang menjadikannya efektif untuk citra lapangan dengan variasi pencahayaan, sudut pengambilan, dan latar belakang seperti citra batang jagung (Santosa et al., 2023; Taye, 2023; S. Wang et al., 2025).

Keunggulan CNN terletak pada kemampuannya mempelajari fitur visual secara hierarkis dalam sistem klasifikasi *end-to-end*, mengenali pola dari yang sederhana hingga kompleks terkait gejala penyakit, serta memberikan akurasi yang lebih baik dan kinerja yang stabil dibandingkan metode konvensional pada kondisi lapangan yang tidak terkontrol. Oleh karena itu, CNN menjadi fondasi bagi pengembangan arsitektur modern yang lebih ringan dan efisien seperti

MobileNetV3, yang sesuai untuk klasifikasi penyakit tanaman berbasis citra lapangan (Das et al., 2024; Taye, 2023; Upadhyay et al., 2025).

2.5 MobileNetV3

2.5.1 Pengertian MobileNetV3

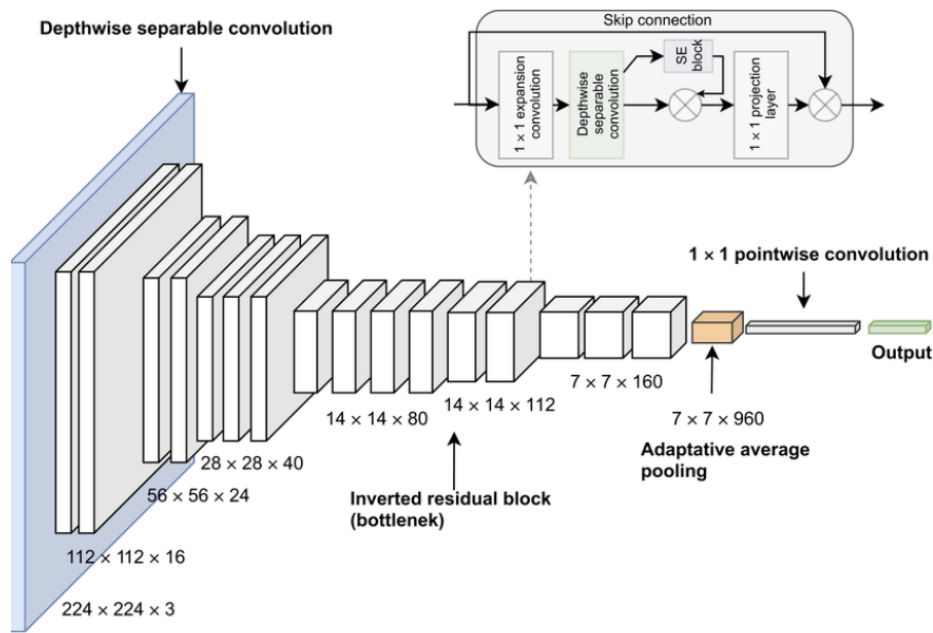
MobileNetV3 merupakan arsitektur CNN yang dikembangkan dalam menghasilkan model klasifikasi citra yang efisien dengan tetap mempertahankan akurasi yang baik, sebagai pengembangan dari MobileNetV1 dan MobileNetV2, serta hadir dalam dua varian, yaitu MobileNetV3-*Large* untuk kebutuhan sumber daya lebih tinggi dan MobileNetV3-*Small* untuk perangkat dengan keterbatasan sumber daya (Howard et al., 2019).

Aplikasi populer lain dari MobileNetV3 adalah dalam teknik *transfer learning*. Metode ini menangani berbagai masalah kategorisasi gambar dengan menggunakan model yang telah dilatih pada dataset skala besar, seperti ImageNet. Dengan menggunakan metode ini, model dapat mempercepat proses pelatihannya dan meningkatkan kinerja klasifikasinya dengan memanfaatkan pengetahuannya tentang isyarat visual dasar. Penelitian yang dilakukan oleh (Abiyyu & Rahardi, 2026) menunjukkan bahwa penggunaan MobileNetV3 berbasis *transfer learning* mampu menghasilkan performa klasifikasi penyakit tanaman yang baik serta efisien secara komputasi, khususnya di dataset yang mempunyai total citra yang terbatas.

Berdasarkan uraian tersebut. Pada penelitian ini digunakan varian MobileNetV3-*Small* karena memiliki ukuran model yang lebih ringan dan lebih relevan dalam implementasi di perangkat yang mempunyai daya terbatas.

2.5.2 Arsitektur MobileNetV3

Arsitektur MobileNetV3 dirancang melalui kombinasi *Neural Architecture Search* (NAS) yang mempertimbangkan karakteristik perangkat keras serta penyempurnaan desain secara manual untuk mencapai keseimbangan optimal antara akurasi dan latensi (Howard et al., 2019). Model ini dikembangkan dari MobileNetV2 dengan tetap menggunakan *Depthwise Separable Convolution* dan *Inverted Residual Block*, di mana *Depthwise Separable Convolution* menekan total parameter dan beban komputasi melalui tahap memisahkan konvolusi spasial dan kanal, sementara *Inverted Residual Block* memungkinkan ekspansi dimensi fitur secara efisien tanpa meningkatkan komputasi secara signifikan (Howard et al., 2019). MobileNetV3 juga mengintegrasikan modul *Squeeze and Excitation* (SE) pada beberapa *blok bottleneck* untuk memodelkan ketergantungan antar kanal melalui proses *Squeeze and Excitation*, sehingga jaringan dapat memberi bobot lebih besar pada fitur yang relevan (Howard et al., 2019). Selain itu, MobileNetV3 memperkenalkan fungsi aktivasi *Hard Swish* (*h-swish*), yaitu versi yang lebih ringan dan stabil secara komputasi dibandingkan *swish*, berbasis ReLU6, yang mampu meningkatkan akurasi tanpa menambah latensi secara signifikan, terutama pada lapisan tengah hingga akhir jaringan (Howard et al., 2019).



Gambar 2.4 Arsitektur MobileNetV3

Sumber: https://www.researchgate.net/figure/The-MobileNetV3-architecture-and-its-core-components_fig4_375462137

2.5.3 Cara Kerja MobileNetV3

Cara kerja MobileNetV3 dimulai dari lapisan konvolusi awal dalam mengekstraksi fitur dasar yakni tepi dan tekstur melalui aktivasi *Hard Swish* atau ReLU. Fitur kemudian diproses melalui *Inverted Residual Bottleneck Blocks* yang mencakup konvolusi 1×1 untuk ekspansi, *Depthwise Convolution* untuk ekstraksi spasial, serta modul *Squeeze and Excitation* pada beberapa blok guna menekankan kanal fitur yang relevan. Setelah itu, fitur diproyeksikan kembali dengan konvolusi 1×1 dan, jika dimensinya sama, diterapkan koneksi residual untuk menjaga stabilitas pelatihan. Pada bagian akhir, fitur dirangkum dengan *global average pooling* dan diteruskan ke lapisan klasifikasi yang dioptimalkan agar tetap efisien secara komputasi tanpa mengorbankan akurasi (Howard et al., 2019).

Model yang telah dilatih sebelumnya, yang telah dilatih menggunakan kumpulan data besar seperti ImageNet untuk mengekstrak fitur gambar umum, biasanya digunakan untuk melatih model dalam aplikasi MobileNetV3 yang berbasis pada *transfer learning*. Studi yang dilangsungkan oleh (Yin et al., 2022) menjabarkan MobileNetV3 digunakan sebagai model awal untuk mengekstraksi fitur daun tanaman, kemudian dilakukan modifikasi pada lapisan konvolusi terakhir dengan menambahkan *batch normalization*, *dropout*, serta *fully connected layer* untuk meningkatkan kemampuan generalisasi model. Setelah itu, model dilatih ulang menggunakan dataset penyakit daun tanaman untuk menyesuaikan parameter jaringan dengan karakteristik data baru. Pendekatan ini terbukti mampu meningkatkan akurasi klasifikasi penyakit tanaman serta menghasilkan model yang ringan dan efisien secara komputasi

2.5.4 Keunggulan MobileNetV3

Keunggulan utama MobileNetV3 terletak pada efisiensi komputasi tanpa mengorbankan akurasi, berkat penggunaan *Depthwise Separable convolution* dan *Inverted Residual Bottleneck* yang mampu menekan beban komputasi sekaligus meningkatkan performa. Dibandingkan generasi sebelumnya, MobileNetV3 menunjukkan akurasi yang lebih baik dan latensi yang juga minim, serta mampu menghasilkan representasi fitur yang lebih relevan melalui integrasi modul *Squeeze and Excitation* dan penggunaan fungsi aktivasi *Hard Swish* yang ringan dan stabil, yang menjadikannya relevan pada perangkat mobile dan sistem berbasis web (Howard et al., 2019). Dalam konteks pertanian, MobileNetV3 juga terbukti efektif untuk klasifikasi penyakit daun jagung karena mampu mencapai performa tinggi

dengan arsitektur yang ringan, tetap stabil pada citra lapangan dengan variasi pencahayaan dan latar belakang yang kompleks, serta lebih fleksibel agar bisa diterapkan pada sistem klasifikasi penyakit tanaman (Maximilliano & Rachmat, 2025).

Keunggulan MobileNetV3 juga didukung oleh kemampuannya dalam meningkatkan performa klasifikasi citra melalui penerapan teknik augmentasi data selama proses pelatihan. Penelitian menunjukkan bahwa augmentasi data seperti rotasi, perubahan warna, pembalikan citra, serta transformasi skala mampu meningkatkan variasi dataset sehingga memperkuat kemampuan model dalam mengenali fitur penyakit tanaman pada berbagai kondisi lingkungan. Pendekatan ini terbukti mampu meningkatkan akurasi serta ketahanan model terhadap gangguan latar belakang dan variasi pencahayaan pada citra penyakit tanaman (Zhang et al., 2025)

2.6 Penelitian Terdahulu

Table 2.1 Ringkasan Penelitian Terdahulu

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
1	“Klasifikasi Penyakit Tanaman Jagung Menggunakan	Penelitian ini menggunakan CNN untuk mengklasifikasikan penyakit tanaman jagung berdasarkan citra daun, seperti hawar dan karat daun.	<i>Convolutional Neural Network</i> (CNN)	Kelebihan: Akurasi tinggi dalam klasifikasi penyakit tanaman daun. Kekurangan:

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
	<p><i>Convoluti onal Neural Network (CNN)”</i> (Iswantoro & Handayani UN, 2022)</p>			<p>Tidak membahas penyakit batang jagung, yang memiliki dampak besar pada hasil panen</p>
2	<p>“Impleme ntasi Klasifikasi Citra Berbasis Tensorflo w Untuk Mendeteks i Penyakit Tanaman Pada Aplikasi Agrosan”</p>	<p>Penelitian ini mengembangkan model klasifikasi penyakit tanaman menggunakan CNN untuk mendeteksi penyakit tanaman berdasarkan citra daun. Model ini dilatih menggunakan dataset PlantVillage yang mencakup berbagai jenis tanaman dan penyakit</p>	<p><i>Convolutional Neural Network (CNN)</i></p>	<p>Kelebihan : Akurasi pelatihan yang tinggi (97,30%) dan validasi (95,02%) dengan model yang efisien dan siap diintegrasikan ke dalam aplikasi. Kekurangan :</p>

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
	(Rahma Hidayat & Jemakmu m, 2025)			Fokus pada penyakit yang menyerang daun, tidak mengatasi penyakit batang jagung secara khusus.
3	“Deteksi Penyakit pada Tanaman Padi Menggunakan Pengolahan Citra Digital dengan Metode Convolutional	Penelitian ini mengaplikasikan CNN untuk mendeteksi penyakit pada tanaman padi, dengan menggunakan dataset yang terdiri dari tiga kelas penyakit: <i>bacterial leaf blight</i> , <i>brown spot</i> , dan <i>leaf smut</i> , serta satu kelas tanaman sehat.	<i>Convolutional Neural Network</i> (CNN)	Kelebihan: Penggunaan CNN memberikan akurasi tinggi, mencapai 99,66%, dengan performa yang sangat baik dalam mendeteksi penyakit padi. Kekurangan: Fokus pada

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
	<p><i>Neural Network</i>” (Santosa et al., 2023)</p>			<p>penyakit daun tanaman padi, tidak ada pembahasan mengenai penyakit batang jagung.</p>
4	<p>“Analisa Penyakit Pada Tanaman Cabai Merah (<i>Capsicum Annum L</i>) dengan Membandingkan Tingkat menggunakan Metode</p>	<p>Penelitian ini membandingkan CNN dan KNN dalam mengklasifikasikan penyakit pada tanaman cabai merah. Hasilnya, CNN menghasilkan akurasi yang lebih tinggi (86%) dibandingkan KNN yang mencapai 81%.</p>	<p><i>Convolutional Neural Network</i> (CNN) dan <i>K-Nearest Neighbor</i> (KNN)</p>	<p>Kelebihan: CNN menunjukkan akurasi tinggi dalam klasifikasi penyakit tanaman cabai. Kekurangan: Fokus pada penyakit daun cabai, tidak mengatasi penyakit pada batang tanaman</p>

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
	<p><i>Convolutional Neural Network (CNN) dan K-Nearest Neighbor (KNN)”</i> (Anggara et al., 2025)</p>			
5	<p>“<i>An Improved MobileNet V3 Mushroom Quality Classification Model Using Images with</i></p>	<p>Penelitian ini mengembangkan model klasifikasi kualitas jamur <i>shiitake</i> menggunakan MobileNetV3 yang diperbaiki. Model ini menangani citra dengan latar belakang kompleks untuk mendeteksi kualitas jamur berdasarkan karakteristik fisiknya.</p>	MobileNetV3	<p>Kelebihan: Penggunaan MobileNetV3 yang efisien dengan peningkatan akurasi hingga 99.91% dan ukuran model yang kecil. Penggunaan</p>

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
	<i>Complex Backgroun ds”</i> (Zhu et al., 2023)			<i>PolyFocalLoss</i> mempercepat konvergensi model. Kekurangan: Fokus pada jamur <i>shiitake</i> , tidak diterapkan pada penyakit tanaman seperti pada batang jagung.

2.7 Analisis GAP

Berdasarkan ringkasan penelitian terdahulu pada Tabel 2.1, dapat diidentifikasi beberapa kesenjangan penelitian (research gap). Sebagian besar studi terdahulu, seperti yang dilangsungkan (Anggara et al., 2025; Iswantoro & Handayani UN, 2022; Rahma Hidayat & Jemakmum, 2025; Santosa et al., 2023) berfokus pada klasifikasi penyakit dari citra daun melalui metode CNN. Meskipun metode CNN terbukti mampu menghasilkan akurasi yang tinggi, penelitian-penelitian tersebut belum mengkaji penyakit yang menyerang batang tanaman,

khususnya batang jagung, yang memiliki pengaruh signifikan terhadap pertumbuhan tanaman dan hasil panen.

Selain itu, penggunaan arsitektur CNN konvensional pada penelitian sebelumnya belum mempertimbangkan efisiensi komputasi dan ukuran model, yang menjadikannya kurang optimal untuk diimplementasikan pada perangkat yang mempunyai daya terbatas, seperti perangkat mobile atau sistem berbasis web di bidang pertanian.

Studi yang dilangsungkan oleh Zhu et al. (2023) telah menjabarkan MobileNetV3 bisa memberikan akurasi yang sangat besar disertai ukuran model yang ringan dan efisien, terutama pada citra dengan latar belakang kompleks. Namun, penelitian tersebut belum diterapkan pada konteks penyakit tanaman, khususnya penyakit batang jagung, sehingga masih terdapat peluang untuk mengeksplorasi kemampuan MobileNetV3 dalam mendeteksi penyakit tanaman berbasis citra lapangan.

Oleh karena itu, kesenjangan penelitian terletak pada belum adanya penelitian yang secara khusus:

1. Mengkaji penyakit batang jagung berbasis citra lapangan,
2. Menggunakan arsitektur MobileNetV3-Small yang efisien dan ringan, serta
3. Mengintegrasikan kebutuhan akurasi tinggi dan efisiensi komputasi untuk mendukung implementasi pada sistem pertanian digital.

Penelitian ini diharapkan dapat mengisi kesenjangan tersebut dengan menerapkan MobileNetV3-Small sebagai model klasifikasi penyakit batang jagung, sehingga mampu memberikan solusi yang akurat, efisien, dan aplikatif untuk mendukung deteksi dini penyakit tanaman jagung di lapangan.

BAB III

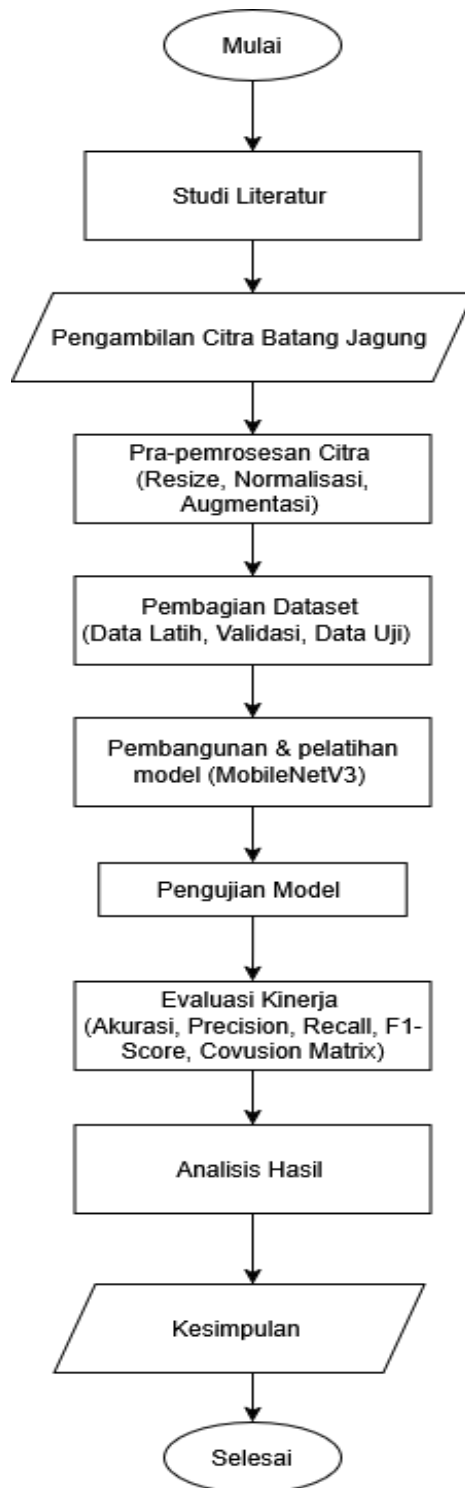
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini memakai metode eksperimen kuantitatif dalam menilai kinerja model Deep Learning MobileNetV3-Small dalam mengklasifikasi penyakit batang jagung dari citra lapangan. Evaluasi dilakukan dengan parameter akurasi, precision, recall, dan F1-score dalam mengukur performa model secara objektif.

3.2 Alur Penelitian

Penelitian dimulai dengan mengambil gambar batang jagung di lapangan untuk tiga kategori: *Bacterial Stalk Rot* (BSR), *Fusarium Stalk Rot* (FSR), dan sehat. Untuk membuat data lebih beragam dan menghindari overfitting, data tersebut diubah ukurannya, dinormalisasi, dan ditingkatkan. Selanjutnya, dataset dibagi menjadi set pengujian, validasi, dan pelatihan. Setelah membangun dan melatih model MobileNetV3-Small sesuai dengan jumlah kelas, model tersebut diuji. Terakhir, ukuran akurasi, presisi, *recall*, *F1-score*, dan matriks kebingungan digunakan untuk mengevaluasi kinerja model dalam hal keberhasilan klasifikasi.



Gambar 3.1 Alur Penelitian

3.3 Pengumpulan Data

Proses pengambilan data peneliti dilaksanakan melalui pengambilan citra langsung di lahan pertanian jagung yang bertempat di Desa Boluk, Kec. Bosar Maligas, Kab.Simalungun. Pengambilan citra dilakukan menggunakan kamera digital atau kamera ponsel untuk menangkap karakteristik visual asli tanaman dalam kondisi lingkungan alami. Selama proses ini, penulis akan mengumpulkan 666 batang jagung yang diklasifikasikan ke tiga kelas, yakni batang jagung sehat, batang terinfeksi *Bacterial Stalk Rot* (BSR), dan *Fusarium Stalk Rot* (FSR).

3.4 Pra-Pemrosesan Citra

Tahapan pra-pemrosesan dilaksanakan sebelum pelatihan model, meliputi:

1. *Resize* citra menjadi 224×224 piksel
2. *Normalisasi* piksel menggunakan skala $1/255$
3. *Augmentasi* data berupa rotasi, *zoom*, *horizontal flip*, dan *brightness*.

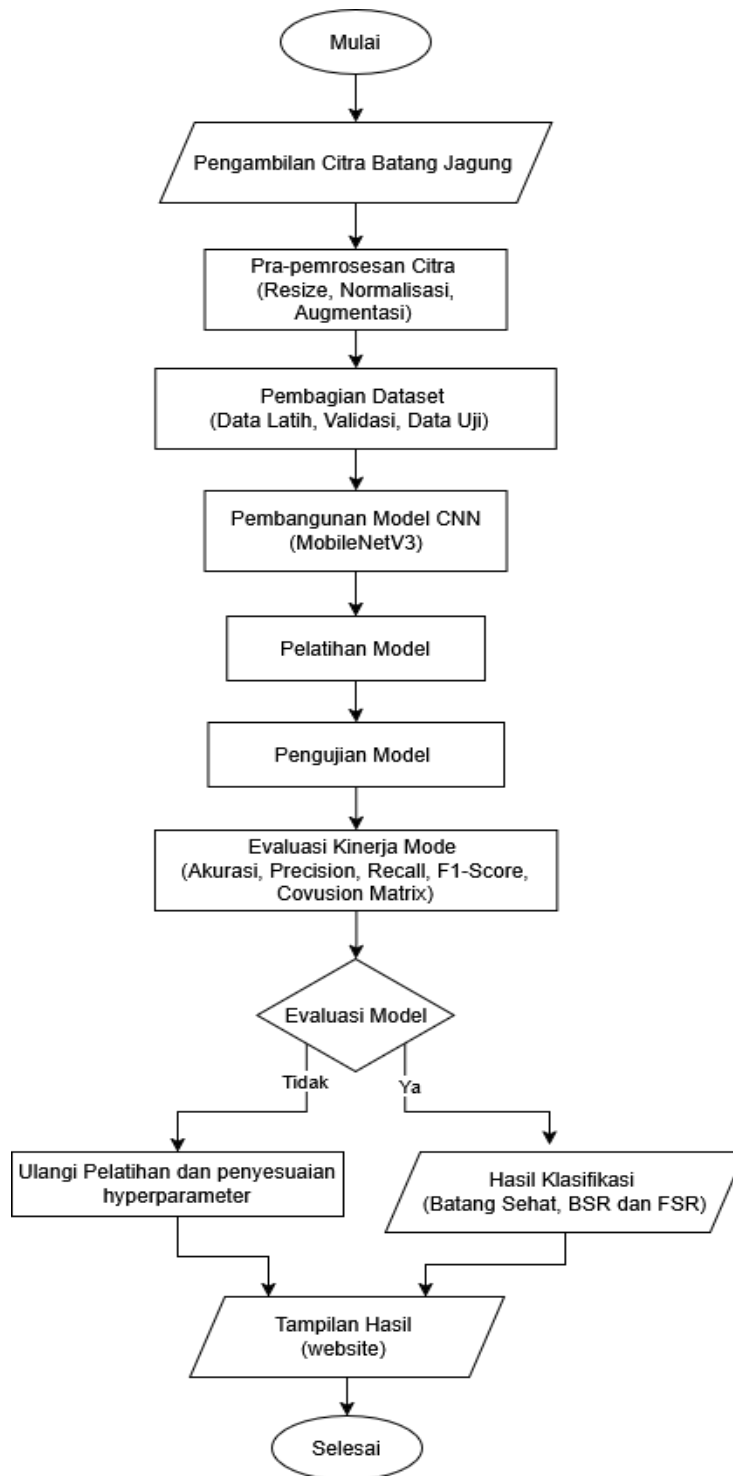
3.5 Pembagian Dataset

Setelah pra-pemrosesan, dataset dibagi secara acak menjadi tiga bagian: 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian. Pembagian ini memastikan bahwa setiap kelas terdistribusi secara merata dan evaluasi berjalan adil. Model MobileNetV3-Small dilatih menggunakan data pelatihan, disetel untuk meminimalkan overfitting dengan hyperparameter menggunakan data validasi, dan dievaluasi pada data baru yang belum pernah dilihat sebelumnya menggunakan data pengujian.

3.6 Perancangan Sistem

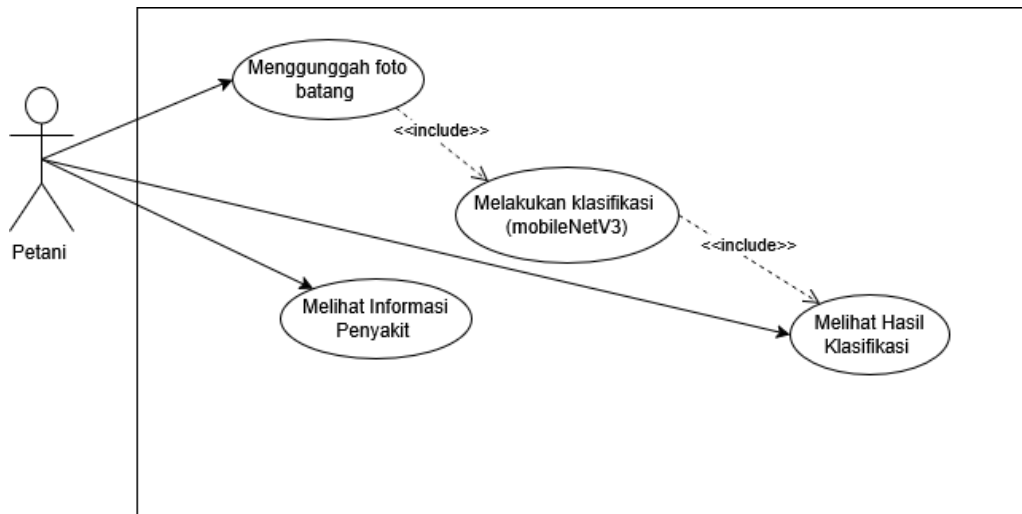
3.6.1 Rancangan Sistem

Rancangan sistem ini dimulai dari pengambilan gambar batang jagung di lapangan, lalu dilakukan pra-pemrosesan seperti *resize*, *normalisasi*, dan *augmentasi*. Terdapat tiga set data yang berbeda dalam dataset: pelatihan, validasi, dan pengujian. Akurasi, presisi, *recall*, *F1-score*, dan matriks kebingungan digunakan untuk mengevaluasi model MobileNetV3-Small setelah dikembangkan dan dilatih dengan data pelatihan. Model tersebut kemudian diuji dengan data pengujian. Jika hasilnya belum optimal, pelatihan diulang dengan penyesuaian parameter. Jika sudah optimal, sistem menampilkan hasil klasifikasi batang jagung sehat, *BSR*, atau *FSR* melalui website sederhana.



Gambar 3.2 Rancangan Sistem

3.6.2 Use Case Diagram

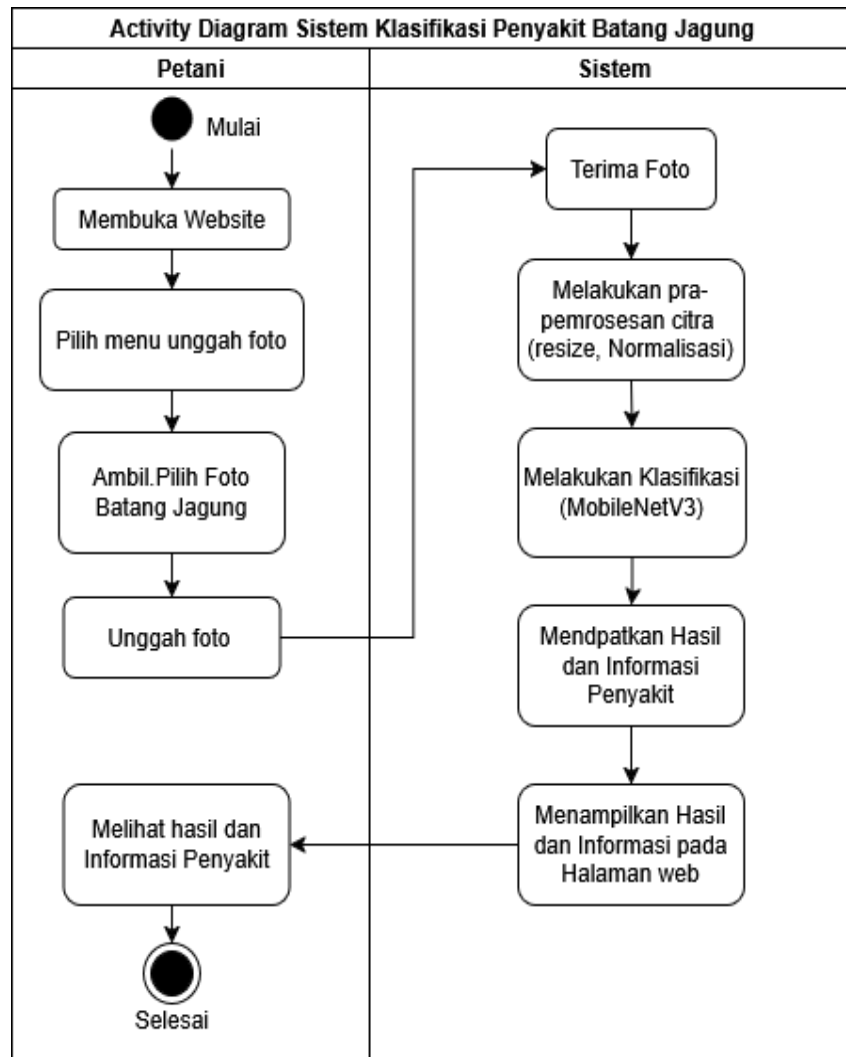


Gambar 3.3 Use Case Diagram

Berdasarkan use case diagram, petani mengunggah foto batang jagung dengan kamera ponsel untuk mengetahui kondisi kesehatannya secara otomatis. Sistem akan mengklasifikasikan penyakit dan menampilkan hasilnya, sehingga petani bisa langsung melihat informasi penyakit pada batang jagung.

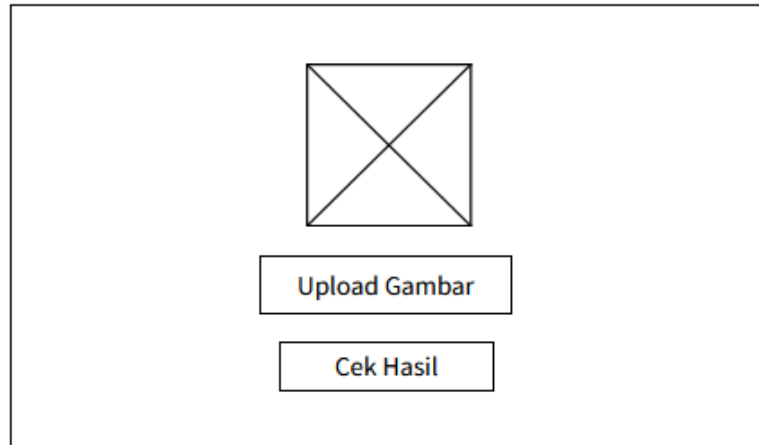
3.6.3 Activity Diagram

Alur kerja sistem pada *activity diagram* menunjukkan interaksi antara petani dan aplikasi. Proses dimulai dengan petani mengunggah gambar batang jagung, lalu sistem melakukan resize dan normalisasi agar sesuai dengan input model MobileNetV3-Small. Setelah pra-pemrosesan, sistem mengklasifikasikan penyakit dan mengambil informasi pendukung dari database. Hasil diagnosis dan penjelasan penyakit ditampilkan di website, sehingga petani bisa langsung mendapatkan informasi kesehatan tanaman di lapangan.



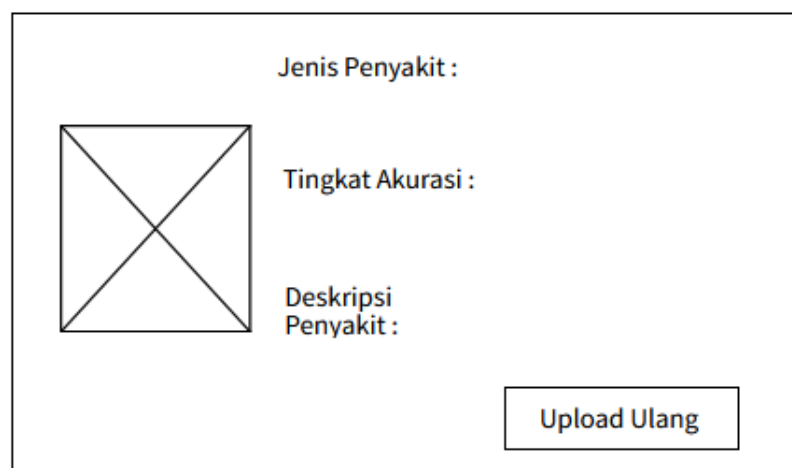
Gambar 3.4 Activity Diagram

3.6.4 Design Interface



Gambar 3.5 *Design* Halaman Upload

Halaman ini dibuat dengan tampilan sederhana agar petani mudah menggunakannya di lapangan. Di tengah halaman ada kotak untuk menampilkan preview gambar batang jagung yang akan diproses. Di bawahnya, ada dua tombol utama: satu untuk mengunggah gambar dari galeri atau kamera, dan satu lagi untuk mengirim gambar ke server agar diproses oleh model MobileNetV3-Small.



Gambar 3.6 *Design* Halaman Hasil

Setelah proses selesai, sistem mengarahkan pengguna ke halaman hasil yang menampilkan informasi tentang batang jagung. Gambar yang diunggah muncul di sisi kiri sebagai referensi, dan di sisi kanan ada panel yang menunjukkan jenis penyakit yang terdeteksi (Sehat, BSR, atau FSR) beserta nilai akurasi atau probabilitas prediksi. Desain ini memudahkan petani memahami kondisi tanaman dengan cepa

3.7 Arsitektur MobileNetV3

Arsitektur yang digunakan dalam penelitian ini adalah MobileNetV3-Small berbasis *transfer learning* dengan bobot awal dari ImageNet. Pemilihan varian Small didasarkan pada efisiensi komputasi dan ukuran model yang lebih ringan, sehingga sesuai untuk implementasi pada sistem berbasis web tanpa mengurangi performa klasifikasi secara signifikan.

Struktur MobileNetV3-Small terdiri dari *Convolutional Layer* awal untuk mengekstraksi fitur dasar dari citra berukuran 224×224 piksel, diikuti oleh rangkaian "*Inverted Residual Bottleneck Blocks*" yang memanfaatkan "*Depthwise Separable Convolution*" dan modul "*Squeeze-and-Excitation*" (*SE*) untuk mengekstraksi fitur secara efisien. Model ini menggunakan fungsi aktivasi *Hard Swish* dan *ReLU* untuk meningkatkan kemampuan non-linear.

Fitur yang dihasilkan kemudian dirangkum menggunakan *Global Average Pooling*, dilanjutkan dengan *Fully Connected Layer* sebagai lapisan klasifikasi, serta *Dropout* untuk mengurangi overfitting. Pada lapisan akhir diterapkan fungsi aktivasi *Softmax* dalam mengembangkan probabilitas pada tiga kelas, yaitu batang sehat, Bacterial Stalk Rot (BSR), dan Fusarium Stalk Rot (FSR).

Dalam penelitian ini, backbone MobileNetV3-Small dibekukan (*frozen*) pada tahap awal pelatihan dan hanya lapisan klasifikasi yang dilatih ulang agar model dapat menyesuaikan dengan karakteristik dataset penyakit batang jagung.

3.7.1 Hyperparameter

Beberapa parameter utama yang diatur meliputi *Learning Rate* untuk mengontrol kecepatan penyesuaian bobot model agar tetap stabil, *Batch Size* untuk menentukan jumlah sampel data yang diproses dalam setiap iterasi, serta jumlah *Epoch* untuk menetapkan frekuensi seluruh dataset dipelajari oleh sistem. Penentuan nilai pada parameter ini dilakukan melalui tahap eksperimen guna menemukan keseimbangan antara durasi pelatihan dan keakuratan prediksi, sehingga model dapat optimal tanpa terjebak dalam kondisi *overfitting*.

3.8 Evaluasi Kinerja

Pada Tahap ini kinerja model MobileNetV3-Small di evaluasi dalam mengklasifikasikan penyakit batang jagung secara objektif. Melalui *Confusion Matrix*, hasil prediksi sistem dipertemukan langsung dengan kondisi riil di lapangan untuk kategori Sehat, BSR, dan FSR. Ketangguhan model selanjutnya diukur melalui metrik akurasi secara menyeluruh, serta *precision*, *recall*, dan *f1-score* untuk evaluasi yang lebih spesifik pada tiap jenis penyakit. Data statistik ini memberikan landasan kuat untuk menilai efektivitas aplikasi dalam mendeteksi gejala penyakit jagung pada berbagai situasi lingkungan nyata.

3.9 Perangkat Penelitian

Pada penelitian beberapa perangkat yang di gunakan yaitu :

1. Perangkat Keras : Laptop Asus VivoBook dengan Processor Intel Core i3, RAM 8GB dan Sistem Operasi Window 11.
2. Perangkat Lunak : Bahasa Pemrograman *Python 3.13* , *TensorFlow*, *Keras*, *Scikit-learn*, *Matplotlib*, *Seaborn*, *Visual Studio Code*, dan *Streamlit*

3.10 Jadwal Penelitian

Table 3.1 Jadwal Penelitian

No	Nama Kegiatan	Bulan Ke																											
		Desember 2025				Januari 2026				Februari 2026				Maret 2026				April 2026				Mei 2026				Juni 2026			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengajuan Judul																												
2	Penulisan Bab I - Bab III																												
3	Seminar Proposal																												
4	Pengumpulan data																												
5	Pembuatan Sistem																												
6	Penulisan Bab IV - Bab V																												
7	Exhibition																												
8	Sidang																												

BAB IV

HASIL DAN PEMBAHASAN

4.1 Dataset Penelitian

Dataset yang ada ialah 3 kelas, yaitu :

1. Penyakit batang jagung Bacterial Stalk Rot
2. Penyakit batang jagung Fusarium Stalk Rot
3. Batang jagung Sehat

Dataset diambil langsung dari lapangan. Pada penelitian ini penulis berhasil mengumpulkan dataset sebanyak 666 dataset termasuk 3 kelas. Kemudian penulis melakukan proses augmentasi untuk meningkatkan jumlah data dan variasi citra. Setelah dilakukan penggabungan data asli dan data augmentasi, di peroleh jumlah total dataset berikut :

Table 4.1 Total Dataset Penelitian

Kelas	Jumlah Data
Bacterial Stalk Rot (BSR)	2228
Fusarium Stalk Rot (FSR)	2210
Batang Sehat	3225
Total	7663

Kemudian, Dataset ini dibagi menjadi:

1. Data Latih: 70%
2. Data Validasi: 15 %
3. Data Uji: 15%

Berikut adalah hasil pembagian dataset pada tabel berikut :

Table 4.2 Pembagian Dataset

Kelas	Latih	Validasi	Uji
Bacterial Stalk Rot (BSR)	1559	334	335
Fusarium Stalk Rot (FSR)	1547	331	332
Batang Sehat	2557	334	334

Total dataset yang diterapkan pada studi ini ialah 7.663 citra, yang tersusun atas 5.663 data Latih, 999 data Validasi, dan 1.001 data Uji.

4.2 Pra-Pemrosesan Data

Pada penelitian ini pra-pemrosesan dilakukan untuk menyesuaikan format citra sebelum masuk CNN. Tahapan yang dilakukan yaitu :

1. Resize citra menjadi ukuran 224 x 224 piksel
2. Normalisasi nilai piksel dilakukan untuk mengubah rentang menjadi 0-1 skla 1/255
3. Augmentasi data untuk meningkatkan variasi citra. Teknik augmentasi yang digunakan meliputi Rotasi Zoom Horizontal flip Brightness.

Setelah melakukan proses augmentasi, total dataset menjadi 7.663 citra batang jagung.

4.3 Implementasi MobileNetV3-Small

Pada penelitian ini, model yang di gunakan adalah MobileNetV3-Small berbasis *transfer learning*. Model yang di gunaan model pralatih (*pretrained model*) dengan bobot awal dari dataset ImageNet, Sehingga mampu memanfaatkan fitur visual dasar yang telah di pelajari seblumnya seperti tepi, tekstur dan pola warna. MobileNetV3-Small dipilih karena memiliki jumlah parameter yang lebih sedikit dibandingkan MobileNetV3-Large, sehingga lebih efisien secara komputasi dan tetap bisa menunjukkan performa klasifikasi yang baik di dataset berukuran menengah.

MobileNetV3-Small digunakan sebagai *feature extractor*, dimana lapisan *backbone* mempertahankan bobot awalnya pada tahap awal pelatihan. Hal ini bertujuan untuk mempercepat proses pelatihan serta meningkatkan kemampuan generalisasi model pada dataset dengan jumlah citra terbatas.

Adapun Arsitektur model yang di gunakan yaitu :

1. *Backbone* MobileNetV3-Small sebagai *feature extractor*
2. *Global Average Pooling* untuk merangkum fitur spasial
3. *Batch Normalization* untuk menstabilkan distribusi aktivasi
4. *Dense layer* untuk lapisan klasifikasi
5. *Dropout* untuk mengurangi risiko *Overfitting*
6. *Output Layer Softmax* untuk klasifikasi 3 kelas

Model menerima input citra yang berukuran 224 x 224 piksel sesuai dengan tahap pra-pemrosesan. Pada tahap pelatihan, model menggunakan skema *feature extraction*, di mana lapisan backbone MobileNetV3-Small dibekukan (*frozen*), sehingga hanya lapisan klasifikasi yang dilatih. Pendekatan ini memanfaatkan fitur

dasar dari ImageNet tanpa mengubah bobot awal pada backbone, sehingga pelatihan menjadi lebih stabil dan efisien.

Pada penelitian ini juga melakukan model pelatihan pada parameter. Model di latih menggunakan :

1. *Optimizer*: Adam
2. *Loss Function*: *Categorical Crossentropy*
3. *Learning Rate*: 0.0001
4. *Batch Size*: 32
5. *Epoch*: 30

4.4 Pelatihan Model

4.4.1 Pengujian Model

Tabel 4.3 Pengujian Model

Pengujian Ke-	Akurasi (%)
1	81%
2	81%
3	81%
4	82%
5	83%
6	83%
7	81%
8	81%
9	82%
10	80%

Berdasarkan tabel 4.3, penulis melakukan pengujian model sebanyak 10 kali untuk mengetahui konsistensi dari model tersebut. Nilai akurasi di peroleh berada pada rentang 80% hingga 83%.

Untuk menghitung rata-rata akurasi melalui rumus seperti berikut:

$$\bar{X} = \frac{\sum x}{n}$$

Ket :

1. \bar{X} = Rata-rata
2. $\sum x$ = Jumlah semua skor akurasi
3. n = Jumlah penhujian

Maka jumlah perhitungan rata rata nya adalah:

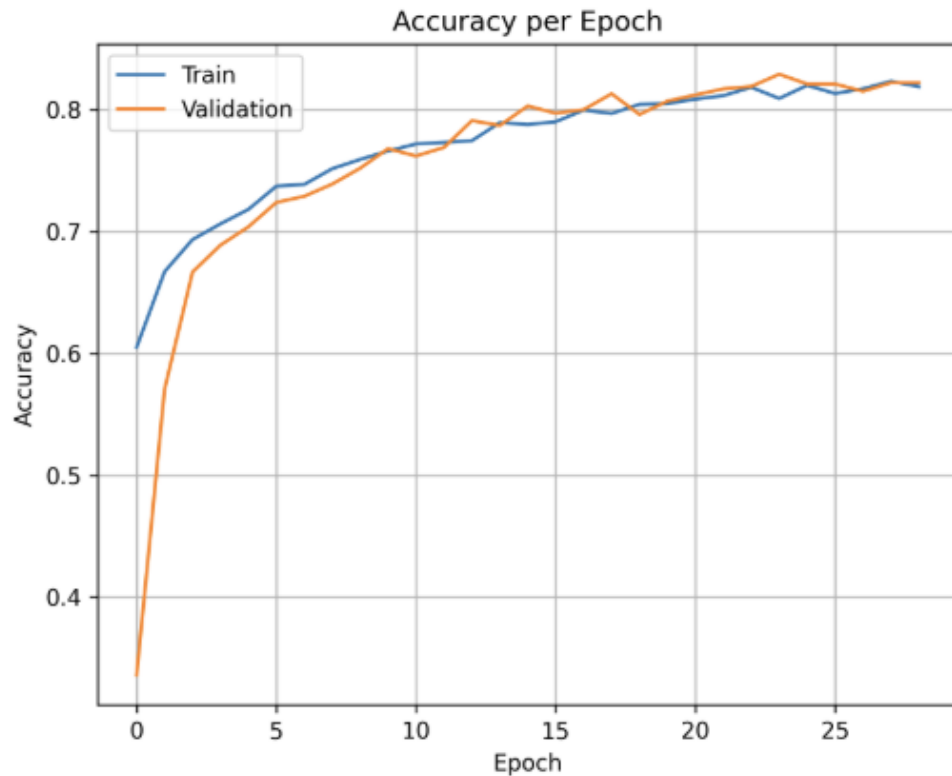
$$\bar{x} = \frac{81+81+81+82+83+83+81+81+82+80}{10}$$

$$\bar{x} = \frac{815}{10}$$

$$\bar{x} = 81,5\%$$

Rata-rata akurasi model yang di peroleh dalam 10 kali pengujian adalah sebesar 81.5%. temuan tersebut menjabarkan model mempunyai peforma yang stabil dan konsiste untuk melangsungkan klasifikasi terhadap data uji.

4.4.2 Grafik Akurasi Model



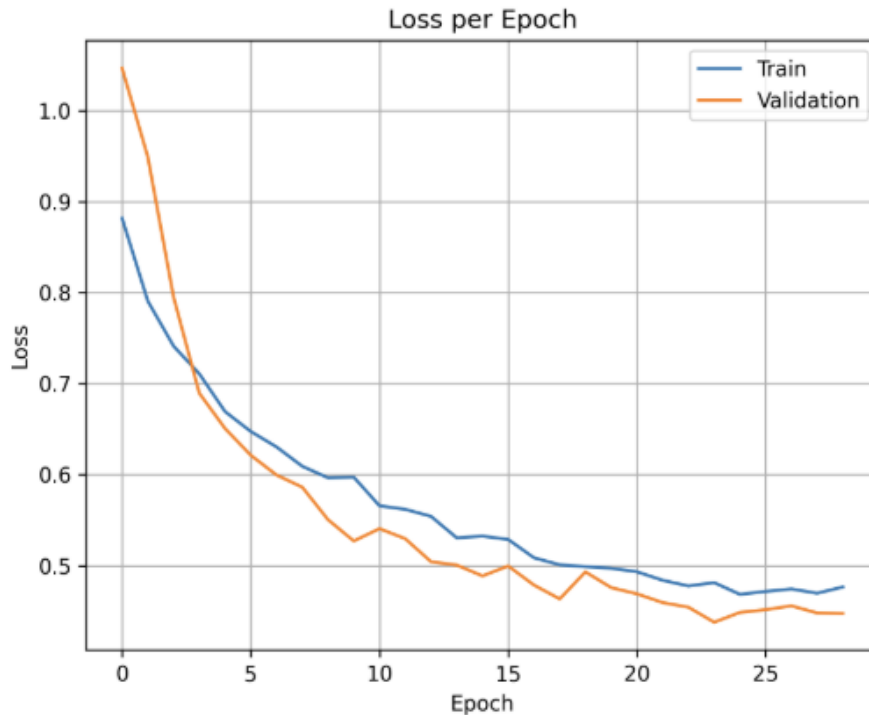
Gambar 4.1 Grafik Akurasi Model

Berdasar gambar di atas, performa model mengalami peningkatan yang konsisten selama proses pelatihan sebanyak 30 epoch. Namun, proses pelatihan berhenti sebelum mencapai 30 epoch. Hal ini disebabkan karena menggunakan EarlyStopping yang berperan dalam menghentikan pelatihan secara otomatis apabila nilai validation loss tidak mengindikasikan ada peningkatan. Akurasi training sekitar $\pm 81\%$ dan akurasi validasi sekitar $\pm 81\%$. Nilai akurasi validasi lebih tinggi dibandingkan nilai akurasi training.

Temuan tersebut menjabarkan model tidak mengindikasikan overfitting dan mempunyai kapasitas generalisasi cukup baik. Selain itu, jarak antara kurva training dan validasi relatif kecil dan bergerak searah kondisi ini menunjukkan proses pelatihan berjalan stabil dan mampu menyesuaikan parameter secara optimal tanpa

terjadi fluktuasi ekstrem. Secara keseluruhan grafik ini menjabarkan model MobileNetV3-Small yang di terapkan mampu mempelajari pola klasifikasi penyakit batang jagung dengan performa yang baik dan stabil.

4.4.3 Grafik Loss Model



Gambar 4.2 Grafik *Loss* Model

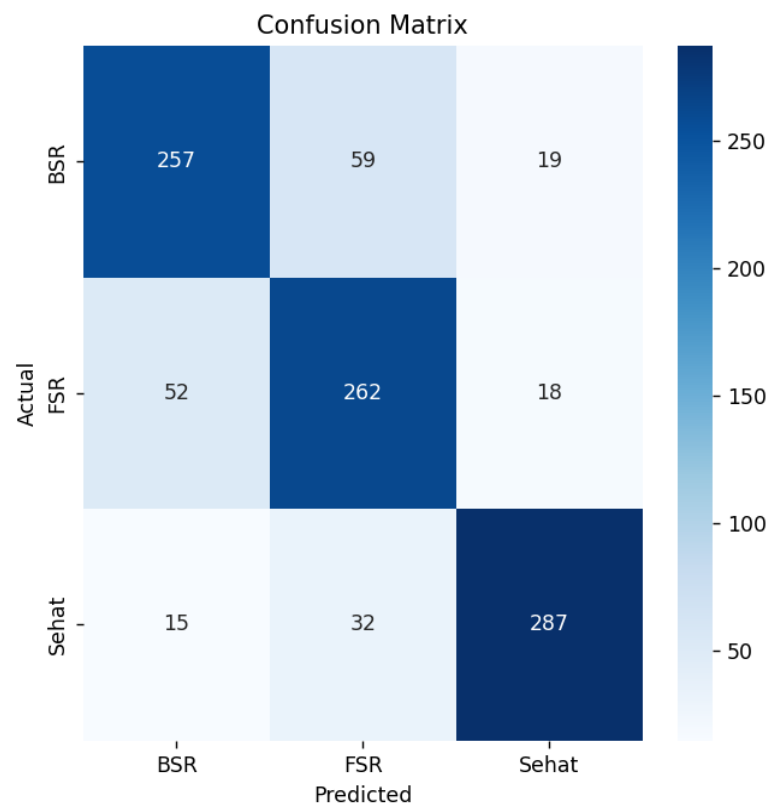
Dari grafik *loss per epoch*, nilai loss di data training dan *validation* mengalami penurunan pada pelatihan. Pada epoch awal, training loss sebesar $\pm 0,88$ dan validation loss sebesar $\pm 1,05$. Seiring bertambahnya epoch, nilai loss menurun sampai menginjak angka 0,47 pada training dan 0,45 pada validation. Penurunan loss yang stabil serta perbedaan yang tidak signifikan antara training dan validation menjabarkan model tidak mengindikasikan ada overfitting dan bisa melangsungkan generalisasi secara optimal pada data uji.

4.4.4 Analisis Training dan Validation

Berdasarkan grafik accuracy dan loss, terlihat skor training accuracy dan validation accuracy mengindikasikan ada peningkatan secara bertahap hingga mencapai kisaran $\pm 81\%$. Sementara itu, skor training loss dan validation loss menunjukkan tren penurunan yang stabil hingga mendekati 0,47 pada akhir epoch. Perbedaan antara kurva training dan validation relatif kecil serta bergerak searah, sehingga menjabarkan model tidak mengindikasikan ada overfitting dan mempunyai kapasitas generalisasi yang baik pada data validasi.

4.5 Evaluasi Model

4.5.1 Hasil Confusion Matrix



Gambar 4.3 Hasil *Confusion Matrix*

Berdasarkan temuan *confusion matrix* di atas, sebagian kesalahan klasifikasi terjadi antara kelas FSR dan BSR yang menunjukkan adanya kemiripan karakteristik visual pada dua jenis penyakit tersebut. Sementara pada kelas sehat memiliki tingkat prediksi yang stabil dan jumlah klasifikasi benar lebih tinggi. MobileNetV3-Small mampu membedakan kondisi batang jagung dengan cukup baik dan stabil terhadap data uji.

Dari gambar *confusion matrix* dapat di hitung nilai True Positif, False Negatif, False Positif dan True Negatif sebagai berikut.

Total data uji = 1001

1. Kelas BSR

$$TP = 257$$

$$FN = 59 + 19 = 78$$

$$FP = 52 + 15 = 67$$

$$TN = 1001 - 402 = 599$$

2. Kelas FSR

$$TP = 262$$

$$FN = 52 + 18 = 70$$

$$FP = 59 + 32 = 91$$

$$TN = 1001 - 423 = 578$$

3. Kelas Sehat

$$TP = 287$$

$$FN = 15 + 42 = 47$$

$$FP = 19 + 18 = 32$$

$$TN = 1001 - 371 = 630$$

Maka total prediksi benar:

$$257 + 262 + 287 = 806$$

sehingga perhitungan akurasi dapat di tulis sebagai berikut:

1. Accuracy

Adapun rumus untuk menghitung akurasi sebagai berikut:

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Total\ Data}$$

$$Accuracy = \frac{806}{1001} = 80,51\%$$

2. Precision

Adapun rumus untuk menghitung Precision sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

a. Kelas BSR

$$\begin{aligned} Precision &= \frac{257}{257 + 67} \\ &= \frac{257}{324} \\ &= 0,79 \end{aligned}$$

b. Kelas FSR

$$\begin{aligned} Precision &= \frac{262}{262 + 91} \\ &= \frac{262}{353} \\ &= 0,74 \end{aligned}$$

c. Kelas Sehat

$$\begin{aligned} \textit{Precision} &= \frac{287}{287 + 37} \\ &= \frac{287}{324} \\ &= 0,89 \end{aligned}$$

3. Recall

Adapun rumus untuk menghitung Recall sebagai berikut:

$$\textit{Recall} = \frac{TP}{TP + FN}$$

a. Kelas BSR

$$\begin{aligned} \textit{Recall} &= \frac{257}{257 + 78} \\ &= \frac{257}{335} \\ &= 0,77 \end{aligned}$$

b. Kelas FSR

$$\begin{aligned} \textit{Recall} &= \frac{262}{262 + 70} \\ &= \frac{262}{332} \\ &= 0,79 \end{aligned}$$

c. Kelas Sehat

$$\begin{aligned} \textit{Recall} &= \frac{287}{287 + 47} \\ &= \frac{287}{334} \\ &= 0,89 \end{aligned}$$

4. F1 Score

Adapun rumus untuk menghitung F1-Score sebagai berikut:

$$F1 = 2 X \frac{Precision \times Recall}{Precision + Recall}$$

a. Kelas BSR

$$Precision = 0,79$$

$$Recall = 0,77$$

$$F1 = 2 X \frac{0,79 \times 0,77}{0,79 + 0,77}$$

$$F1 = 2 X \frac{0,6083}{1,56}$$

$$F1 = 2 X 0,3899$$

$$F1 = 0,78$$

b. Kelas FSR

$$Precision = 0,74$$

$$Recall = 0,79$$

$$F1 = 2 X \frac{0,74 \times 0,79}{0,74 + 0,79}$$

$$F1 = 2 X \frac{0,5846}{1,53}$$

$$F1 = 2 X 0,382$$

$$F1 = 0,76$$

c. Kelas Sehat

$$\text{Precision} = 0,89$$

$$\text{Recall} = 0,86$$

$$F1 = 2 \times \frac{0,89 \times 0,86}{0,89 + 0,86}$$

$$F1 = 2 \times \frac{0,7654}{1,75}$$

$$F1 = 2 \times 0,4374$$

$$F1 = 0,87$$

4.5.2 Hasil Precision, Recall, dan F1-Score

Table 4.4 Hasil Precision, Recall, dan F1-Score

Kelas	Precision	Recall	F1-Score	Support
BSR	0.79	0.77	0.78	335
FSR	0.74	0.79	0.76	332
Sehat	0.89	0.86	0.87	334

Kelas Sehat memiliki performa terbaik disertai skor precision yakni 0,89, recall yakni 0,86, dan F1-score yakni 0,87. Sementara itu, kelas BSR dan FSR juga mengindikasikan performa yang cukup baik disertai skor F1-score masing-masing yakni 0,78 dan 0,76.

Kesalahan klasifikasi sebagian besar terjadi antara kelas BSR dan FSR, yang kemungkinan ditimbulkan oleh kemiripan karakteristik visual pada kedua kelas. Pada konteks menyeluruh, model mengindikasikan kapasitas generalisasi yang baik dalam melakukan klasifikasi tiga kelas penyakit.

Table 4.5 Hasil Evaluasi Model

Metode Evaluasi	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.81	1001
Macro Average	0.81	0.81	0.81	1001
Weighted Average	0.81	0.81	0.81	1001

Model menghasilkan akurasi yakni 81% disertai skor macro dan weighted average precision, recall, serta F1-score yakni 81% yang menunjukkan performa klasifikasi yang stabil dan seimbang pada seluruh kelas.

4.6 Analisis Hasil Penelitian

Dari temuan studi yang sudah dilangsungkan, model MobileNetV3-Small mampu mengklasifikasikan penyakit batang jagung ke dalam tiga kelas, yaitu *Bacterial Stalk Rot* (BSR), *Fusarium Stalk Rot* (FSR), dan batang sehat. Berdasarkan hasil pengujian terhadap 1001 data uji, diperoleh nilai accuracy sebesar 80,51% atau sekitar 81%.

Skor precision, recall, dan F1-score pada tiap kelas mengindikasikan performa yang cukup seimbang. Kelas BSR mendapat precision yakni 0,79, recall yakni 0,77, dan F1-score yakni 0,78. Kelas FSR memperoleh precision yakni 0,74, recall yakni 0,79, dan F1-score yakni 0,76. Sementara itu, kelas Sehat menunjukkan

performa terbaik dengan precision yakni 0,89, recall yakni 0,86, dan F1-score yakni 0,87. Skor yang ada tersebut menjabarkan model memiliki kemampuan klasifikasi yang stabil pada setiap kelas.

Berdasarkan confusion matrix, masih terdapat beberapa kesalahan klasifikasi terutama antara kelas BSR dan FSR. Hal tersebut bisa ditimbulkan oleh kemiripan karakteristik visual pada citra lapangan, seperti perubahan warna batang dan tekstur jaringan yang serupa pada kedua jenis penyakit tersebut. Meskipun demikian, model tetap mampu membedakan ketiga kelas dengan performa yang baik.

Selain itu, proses augmentasi data yang diterapkan selama pelatihan terbukti membantu meningkatkan variasi citra dan memperbaiki kemampuan generalisasi model pada data uji. Hal ini ditunjukkan dengan perbedaan yang relatif minim antara nilai training dan validation accuracy, sehingga model tidak mengindikasikan ada overfitting secara signifikan.

Penggunaan arsitektur MobileNetV3-Small juga memberikan keunggulan dari sisi efisiensi komputasi dan ukuran model yang ringan. Disertai total parameter yang terbilang kecil, model ini tetap bisa menghasilkan performa klasifikasi yang baik, sehingga sesuai untuk implementasi berbasis web maupun perangkat dengan sumber daya terbatas.

Berdasarkan hasil penelitian ini, rumusan masalah yang dikaji di Bab I telah terjawab, yakni:

1. Sistem berbasis MobileNetV3-Small berhasil dikembangkan untuk melakukan klasifikasi penyakit batang jagung secara otomatis dan dapat menjadi alternatif dalam membantu identifikasi penyakit di lapangan.

2. Arsitektur MobileNetV3-Small dapat diterapkan untuk klasifikasi penyakit batang jagung berbasis citra lapangan dengan performa yang baik.
3. Kinerja model telah diketahui melalui evaluasi kuantitatif dengan nilai accuracy sebesar 80,51% serta skor precision, recall, dan F1-score yang stabil pada setiap kelas.

4.7 Implementasi Sistem


Sistem Klasifikasi Penyakit Batang Jagung

Pilih Sumber Gambar

Sumber gambar:

- Upload Galeri
- Kamera

Upload Gambar

 Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG

Gambar 4.4 Halaman Tampilan Upload

Gambar 4.4 menjabarkan halaman utaman web untuk klasifikasi penyakit batang jagung. Pada halaman ini, para petani dapat memilih sumber gambar melalui galeri atau kamera sebelum melakukan proses klasifikasi. Jika petani memilih gambar dari upload galeri maka petani dapat klik tombol browse files.

Sistem Klasifikasi Penyakit Batang Jagung

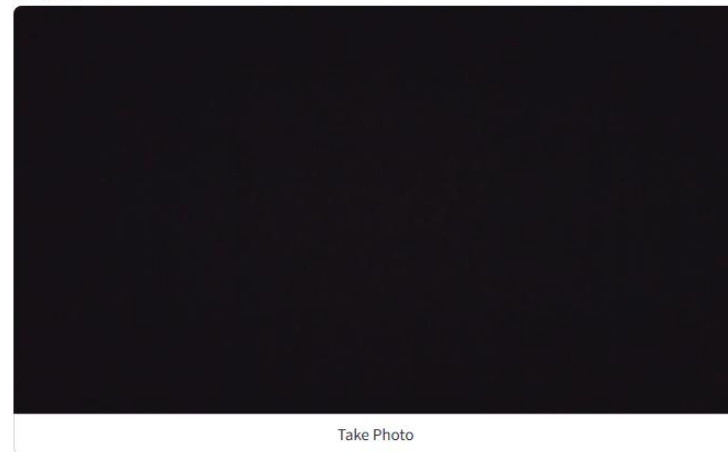
Pilih Sumber Gambar

Sumber gambar:

Upload Galeri

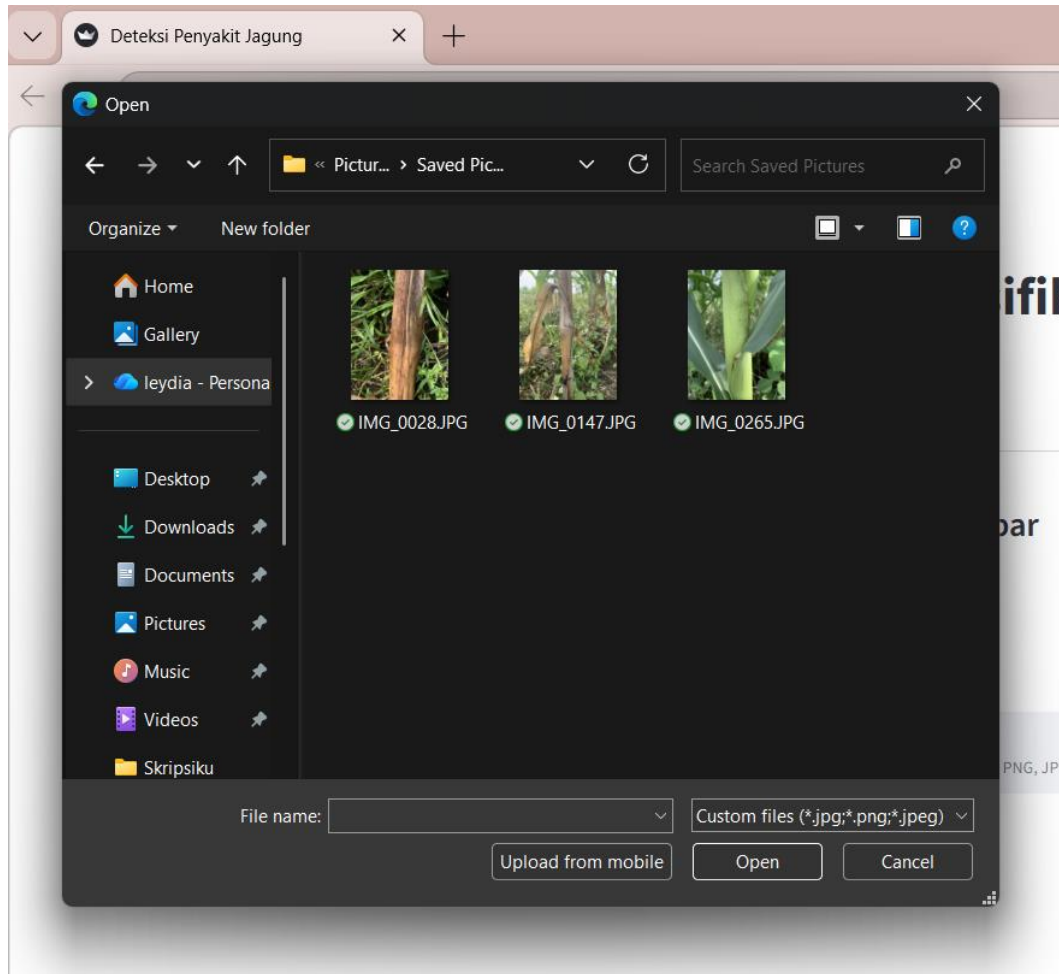
Kamera

Ambil Foto



Gambar 4.5 Halaman Tampilan Kamera

Kemudian, pada gambar 4.5 jika petani memilih untuk analisis penyakit batang jagung menggunakan kamera, petani dapat klik tombol kamera lalu otomatis akan muncul kamera. selanjutnya, petani bisa langsung klik take photo untuk pengambilan gambar. Fitur ini memudahkan para petani untuk melakukan deteksi penyakit secara langsung di lapangan.



Gambar 4.6 Halaman Tampilan pemilihan gambar

Pada gambar 4.6 ini adalah proses pemilihan gambar dari galeri. Petani dapat memilih gambar dengan format jpg, png dan jpeg. Gambar yang dipilih petani kemudian di proses oleh sistem menggunakan mobileNetV3-Small untuk dilakukan klasifikasi kedalam 3 kelas yaitu *Bacterial Stalk Rot (BSR)*, *Fusarium Stalk Rot (FSR)*, dan batang sehat.

Sistem Klasifikasi Penyakit Batang Jagung

Pilih Sumber Gambar

Sumber gambar:

- Upload Galeri
- Kamera

Upload Gambar:

Drag and drop file here
Limit: 20MB per file • JPG, PNG, JPEG

Browse files

IMG_0028.JPG 2.9MB

×

Gambar Input

The `use_culture_sites` parameter has been deprecated and will be removed in a future release. Please utilize the `sites` parameter instead.



Gambar 4.7 Halaman Tampilan Input yang Berhasil Diunggah

Gambar 4.7 menampilkan gambar batang jagung yang telah berhasil diunggah ke dalam sistem. Setelah petani memilih gambar, sistem akan menampilkan gambar sebagai input sebelum melakukan proses klasifikasi. Pada fase ini dilaksanakan dalam memastikan gambar yang dipilih relevan dan siap diproses .

Auto Crop Fokus Batang

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `width` parameter instead.



Cek Hasil

Gambar 4.8 Halaman Tampilan Proses Auto Crop

Gambar 4.8 menampilkan proses auto crop untuk memfokuskan area batang jagung. Tahap ini merupakan bagian preprocessing yang dimaksudkan dalam menekan area latar belakang yang tidak relevan yang menjadikan model lebih fokus pada karakteristik visual batang jagung. Dengan adanya fitur ini gambar akan dapat di analisis secara lebih optimal oleh model MobileNetV3-Small sebelum dilakukan klasifikasi. Kemudian pada tahap ini petani dapat klik tombol cek hasil untuk melihat hasil klasifikasi.



Cek Hasil

Model sangat yakin dengan hasil prediksi.

Jenis Penyakit: BSR

Tingkat Keyakinan Model: 99.30%

Deskripsi:

Bacterial Stalk Rot (BSR) merupakan penyakit penting pada tanaman jagung yang disebabkan oleh bakteri *Dickeya zeae*. Penyakit ini menyebabkan pembusukan basah pada jaringan batang di tandai dengan pelunakan jaringan, perubahan warna menjadi coklat dan di tandai dengan bau yang tidak sedap. Kondisi tersebut mengganggu aliran nutrisi sehingga tanaman mudah roboh.

Gambar 4.9 Hasil Analisis kelas *Bacterial Stalk Rot* (BSR)

Pada gambar 4.9 menampilkan hasil analisis gambar yang sudah di input sebelumnya. Sistem mengidentifikasi gambar sebagai penyakit *Bacterial Stalk Rot* (BSR). Berdasarkan gambar tersebut sistem mengindikasikan jenis penyakit yang terdeteksi disertai taraf keyakinan model yakni 99.30% dan menampilkan deskripsi singkat mengenai karakteristik penyakit untuk membantu petani memahami hasil identifikasi.



Cek Hasil

Model cukup yakin. Gejala penyakit kemungkinan sedang berkembang.

Jenis Penyakit: FSR

Tingkat Keyakinan Model: 74.50%

Deskripsi:

Fusarium Stalk Rot (FSR) disebabkan beberapa jenis jamur fusarium yang sangat mengganggu pertumbuhan tanaman jagung, menurunkan nilai gizi, serta mengurangi hasil panen secara keseluruhan. tanaman mengalami layu, perubahan warna daun, dan pengeringan pada batang bagian bawah

Gambar 4.10 Hasil Analisis kelas *Fusarium Stalk Rot* (FSR)

Pada gambar 4.10 menampilkan hasil Sistem dalam klasifikasi sebagai *Fusarium Stalk Rot* (FSR) dan menampilkan tingkat keyakinan model sebesar 74,50% serta deskripsi singkat mengenai penyakit tersebut. Hal ini bertujuan untuk memudahkan petani untuk mengetahui penyakit batang jagung .



Cek Hasil

Model sangat yakin dengan hasil prediksi.

Jenis Penyakit: Sehat

Tingkat Keyakinan Model: 95.58%

Deskripsi:

Tanaman tidak menunjukkan gejala penyakit.

Gambar 4.11 Hasil Analisis Kelas Batang Jagung Sehat

Sama seperti gambar sebelumnya, gambar 4.11 ini menampilkan hasil klasifikasi batang jagung sehat yang tidak terkena penyakit, dengan tingkat keyakinan model sebesar 95.58% serta sistem juga menampilkan deskripsi mengenai batang jagung tersebut.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari temuan penelitian terkait klasifikasi penyakit batang jagung menggunakan MobileNetV3-Small berbasis citra lapangan, maka bisa disimpulkan bahwa:

1. Sistem klasifikasi bisa dikembangkan dan diimplementasikan. Penelitian ini berhasil membangun sistem klasifikasi berbasis web melalui penerapan arsitektur MobileNetV3-Small melalui pendekatan transfer learning. Dataset yang diterapkan ialah 666 citra asli yang kemudian ditingkatkan melalui proses augmentasi menjadi 7.663 citra. Seluruh citra diproses melalui tahapan resize 224×224 piksel, normalisasi skala $1/255$, serta augmentasi sebelum dilakukan pelatihan model.
2. Model mampu mengidentifikasi tiga kelas penyakit secara otomatis. Model yang dibangun dapat mengklasifikasikan batang jagung ke dalam tiga kategori, yaitu *Bacterial Stalk Rot* (2.228 citra), *Fusarium Stalk Rot* (2.210 citra), dan batang sehat (3.225 citra). Dataset dibagi menjadi 70% data latih (5.663 citra), 15% data validasi (999 citra), dan 15% data uji (1.001 citra), sehingga proses evaluasi dilakukan secara objektif pada data yang belum dilatih.
3. Kinerja MobileNetV3-Small menunjukkan performa yang stabil dan cukup baik. Berdasarkan 10 kali pengujian, model memperoleh rentang akurasi antara 80% hingga 83%, dengan rata-rata akurasi sebesar 81,5%. Selama

proses pelatihan sebanyak maksimal 30 epoch (menggunakan EarlyStopping), nilai training accuracy dan validation accuracy mencapai sekitar $\pm 81\%$, dengan penurunan training loss dari $\pm 0,88$ ke angka $\pm 0,47$ serta validation loss dari $\pm 1,05$ menjadi $\pm 0,45$. Perbedaan yang minim antara kurva training dan validation menunjukkan model tidak mengindikasikan ada overfitting dan mempunyai kapasitas generalisasi yang cukup baik.

4. MobileNetV3-Small efektif dan efisien untuk klasifikasi citra lapangan. Dengan penggunaan optimizer Adam, learning rate 0.0001, batch size 32, dan fungsi loss categorical crossentropy, model mampu mencapai performa yang stabil dengan arsitektur yang ringan. Hal ini menunjukkan bahwa MobileNetV3-Small sesuai untuk diterapkan pada sistem berbasis web maupun perangkat yang disertai sumber daya terbatas tanpa mengorbankan performa klasifikasi secara signifikan.

Dengan demikian, MobileNetV3-Small dapat dinyatakan mampu melakukan klasifikasi penyakit batang jagung dengan performa yang cukup baik dan berpotensi mendukung deteksi dini penyakit di bidang pertanian digital.

5.2 Saran

Dari temuan penelitian dan keterbatasan yang masih terdapat dalam pengembangan sistem klasifikasi penyakit batang jagung ini, maka berbagai saran yang bisa dipertimbangkan bagi studi selanjutnya ialah seperti berikut:

1. Pengembangan Dataset dengan Penambahan Jenis Penyakit Lain

Untuk meningkatkan cakupan sistem klasifikasi, penelitian selanjutnya disarankan menambahkan jenis penyakit batang jagung lainnya seperti Antraknosa, Gibberella, Diplodia, Arang, maupun Pythium.

2. Peningkatan Kualitas dan Keaslian Dataset

Penambahan dataset asli dengan jumlah yang lebih besar tetap diperlukan sehingga dapat membantu model mempelajari pola visual penyakit secara lebih akurat dibandingkan hanya mengandalkan data hasil augmentasi.

3. Penggunaan Metode Evaluasi yang Lebih Komprehensif

Evaluasi pada penelitian ini menggunakan pembagian data 70:15:15 serta metrik akurasi, precision, recall, dan F1-score. Penelitian lainnya bisa menerapkan metode evaluasi tambahan seperti K-Fold Cross Validation untuk memperoleh hasil evaluasi yang lebih stabil dan mengurangi potensi bias pembagian data.

4. Perbandingan dengan Arsitektur Model Lain

Model yang diterapkan pada penelitian ini ialah MobileNetV3-Small melalui pendekatan feature extraction. Penelitian selanjutnya dapat membandingkan model ini dengan arsitektur lain seperti ResNet, EfficientNet, DenseNet, atau MobileNetV3-Large.

DAFTAR PUSTAKA

- Abiyyu, A. N., & Rahardi, M. (2026). *Comparison of Transfer learning Models MobileNetV3-Large and EfficientNet-B0 for Rice Leaf Disease Classification*. *10(1)*, 818–828.
- Anggara, A. A., Ridho, A., & Mutia, C. (2025). Analisa Penyakit Pada Tanaman Cabai Merah (*Capsicum annuum* L) Dengan Membandingkan Tingkat Akurasi Menggunakan Metode Convolutional Neural Network (CNN) dan K-Nearest Neighbor (KNN). *Jurnal Teknologi Informasi*, *4(1)*, 52. <https://doi.org/10.35308/jti.v4i1.11816>
- Arief, N. A., Wijayanti, R. L., Chairunnisa, R. A., Tsabita, N. A., Malang, U. N., Bioteknologi, P. S., Malang, U. N., Barat, N. T., Barat, J., Tengah, J., & Utara, S. (2023). *Kolaborasi Menuju Akselerasi Inovasi Sains untuk Kemandirian Kesehatan dan Kemajuan Ekonomi Hijau 1. 4*, 1–5.
- Das, P. K., Rupa, S. S., Pumrin, S., Das, U. C., & Hossen, M. K. (2024). Deep Learning for Plant Disease Detection and Classification: A Systematic Analysis and Review. *Current Applied Science and Technology*, *24(4)*. <https://doi.org/10.55003/cast.2024.259016>
- Harish, J., Jambhulkar, P. P., Bajpai, R., R, K., R, L., Vamsidharreddy, N., & H. B, S. (2024). Evaluation of Inbreds for Resistance to Post-Flowering Stalk Rot Disease in Maize (*Zea mays*). *Journal of Advances in Biology & Biotechnology*, *27(5)*, 666–675. <https://doi.org/10.9734/jabb/2024/v27i5827>
- Hutagalung, F. S., Siregar, F. A., & Al-Khowarizmi. (2025). *JITE (Journal of Informatics and Telecommunication Engineering)*. *9(July)*, 1–12.

- Iswantoro, D., & Handayani UN, D. (2022). Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Universitas Batanghari Jambi*, 22(2), 900. <https://doi.org/10.33087/jiubj.v22i2.2065>
- Lacatusu, Florin, Ionita, A. D., Lăcătușu, M., & Olteanu, A. (2022). Performance Evaluation of Information Gathering from Edge Devices in a Complex of Smart Buildings. *Sensors*, 22(3). <https://doi.org/10.3390/s22031002>
- Rahma Hidayat, J., & Jemakmum. (2025). Implementasi Klasifikasi Citra Berbasis Tensorflow Untuk Mendeteksi Penyakit Tanaman Pada Aplikasi Agrosan. *Jurnal Fasilkom*, 15(1), 124–130. <https://doi.org/10.37859/jf.v15i1.8536>
- Rizki Ramadhani, I., Nilogiri, A., & A'yun, Q. (2022). Klasifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network Classification Of Plants Based On Leaf Image Using Convolutional Neural Network Method. *Jurnal Smart Teknologi*, 3(3), 2774–1702. <http://jurnal.unmuhjember.ac.id/index.php/JST>
- Santosa, A. A., Fu'adah, R. Y. N., & Rizal, S. (2023). Deteksi Penyakit pada Tanaman Padi Menggunakan Pengolahan Citra Digital dengan Metode Convolutional Neural Network. *Journal of Electrical and System Control Engineering*, 6(2), 98–108. <https://doi.org/10.31289/jesce.v6i2.7930>
- Suriani, Patandjengi, B., Junaid, M., & Muis, A. (2021). The presence of bacterial stalk rot disease on corn in Indonesia: A review. *IOP Conference Series: Earth and Environmental Science*, 911(1). <https://doi.org/10.1088/1755-1315/911/1/012058>
- Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network:

- Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3).
<https://doi.org/10.3390/computation11030052>
- Toyib, M., Pratama, T. D. K., & Aqil, I. (2024). Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data. *Algoritma : Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumihan Dan Angkasa*, 2(3), 108–120.
<https://journal.arimsi.or.id/index.php/Algoritma/article/view/69>
- Upadhyay, A., Chandel, N. S., Singh, K. P., Chakraborty, S. K., Nandede, B. M., Kumar, M., Subeesh, A., Upendar, K., Salem, A., & Elbeltagi, A. (2025). Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artificial Intelligence Review*, 58(3). <https://doi.org/10.1007/s10462-024-11100-x>
- Wahyu, M., Nurlina, N., & Irawan, D. (2023). Multitek Indonesia : Jurnal Ilmiah Multitek Indonesia : Jurnal Ilmiah. *Multitek Indonesia: Jurnal Ilmiah*, 17(1), 60–68.
- Wang, S., Xu, D., Liang, H., Bai, Y., Li, X., Zhou, J., Su, C., & Wei, W. (2025). Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review. *Remote Sensing*, 17(4), 1–30. <https://doi.org/10.3390/rs17040698>
- Wibowo, A., Zulpani, R., Windarto, A. P., Wanto, A., & Andani, S. R. (2025). ENHANCING HERBAL PLANT LEAF IMAGE DETECTION ACCURACY. *10(4)*, 859–867. <https://doi.org/10.33480/jitk.v10i4.6498>
- Yin, X., Li, W., Li, Z., & Yi, L. (2022). Recognition of grape leaf diseases using MobileNetV3 and deep transfer learning. *15(3)*, 184–194.
<https://doi.org/10.25165/j.ijabe.20221503.7062>

Zhang, J., Yang, X., Fu, X., Wang, B., & Li, H. (2025). *LDL-MobileNetV3S: an enhanced lightweight MobileNetV3-small model for potato leaf disease diagnosis through multi-module fusion*. *October*, 1–19.
<https://doi.org/10.3389/fpls.2025.1656731>

Zhu, F., Sun, Y., Zhang, Y., Zhang, W., & Qi, J. (2023). An Improved MobileNetV3 Mushroom Quality Classification Model Using Images with Complex Backgrounds. *Agronomy*, *13*(12).
<https://doi.org/10.3390/agronomy13122924>

LAMPIRAN

```
#save_augmented.py

import os

import glob

from tensorflow.keras.preprocessing.image import ImageDataGenerator,
img_to_array, load_img

# AUGMENTATION SETTING

datagen = ImageDataGenerator(

    rescale=1./255,

    rotation_range=40,

    zoom_range=0.1,

    horizontal_flip=True,

    brightness_range=[0.7, 1.3],

    fill_mode='nearest'

)

# PATH DATA

data_folders = [

    'Dataset/RawData/BSR',

    'Dataset/RawData/FSR',

    'Dataset/RawData/Sehat'

]

save_base_dir = 'Dataset/AugmentedData'

os.makedirs(save_base_dir, exist_ok=True)
```

```

# AUGMENTASI

for folder in data_folders:

    # Ambil semua format gambar

    image_paths = glob.glob(os.path.join(folder, '*.*'))

    class_name = os.path.basename(folder)

    save_dir = os.path.join(save_base_dir, class_name)

    os.makedirs(save_dir, exist_ok=True)

    for image_path in image_paths:

        img = load_img(image_path, target_size=(256, 256))

        x = img_to_array(img)

        x = x.reshape((1,) + x.shape)

        i = 0

        for batch in datagen.flow(

            x,

            batch_size=1,

            save_to_dir=save_dir,

            save_prefix='aug',

            save_format='jpeg'):

            i += 1

            if i >= 10:

                break

```

```

#train_model.py

import os

import numpy as np

import tensorflow as tf

import matplotlib.pyplot as plt

from tensorflow.keras.applications import MobileNetV3Small

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras import layers, models

from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau,
ModelCheckpoint

# PATH DATASET

train_dir = "DatasetFinal/Train"

valid_dir = "DatasetFinal/Validation"

test_dir = "DatasetFinal/Test"

IMG_SIZE = (224,224)

BATCH_SIZE = 32

EPOCHS = 30

os.makedirs("Models", exist_ok=True)

os.makedirs("Reports", exist_ok=True)

# DATA GENERATOR

train_datagen = ImageDataGenerator(

    rescale=1./255,

    rotation_range=15,

    zoom_range=0.1,

```

```

        horizontal_flip=True,
        brightness_range=[0.9,1.1]
    )
    valid_datagen = ImageDataGenerator(rescale=1./255)
    test_datagen = ImageDataGenerator(rescale=1./255)
    train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical'
    )
    valid_generator = valid_datagen.flow_from_directory(
        valid_dir,
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical'
    )
    test_generator = test_datagen.flow_from_directory(
        test_dir,
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical',
        shuffle=False
    )

```

```

# MODEL MOBILENETV3 SMALL

base_model = MobileNetV3Small(

    weights='imagenet',

    include_top=False,

    input_shape=(224,224,3)

)

# Freeze

base_model.trainable = False

model = models.Sequential([

    base_model,

    layers.GlobalAveragePooling2D,

    layers.BatchNormalization,

    layers.Dense(128, activation='relu'),

    layers.Dropout(0.5),

    layers.Dense(3, activation='softmax')

])

model.compile(

    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),

    loss='categorical_crossentropy',

    metrics=['accuracy']

)

model.summary

# CALLBACK

```

```

callbacks = [
    EarlyStopping(patience=5, restore_best_weights=True),
    ReduceLROnPlateau(patience=3, factor=0.3, verbose=1),
    ModelCheckpoint('Models/best_model.keras', save_best_only=True)
]

# TRAINING

history = model.fit(
    train_generator,
    validation_data=valid_generator,
    epochs=EPOCHS,
    callbacks=callbacks
)

# SAVE FINAL MODEL

model.save("Models/model_final.keras")

# EVALUASI TEST

test_loss, test_acc = model.evaluate(test_generator)

print("\nTest Accuracy :", test_acc)

print("Test Loss :", test_loss)

# GRAFIK TRAINING

plt.figure(figsize=(12,5))

# Accuracy

plt.subplot(1,2,1)

plt.plot(history.history['accuracy'], label='Train')

plt.plot(history.history['val_accuracy'], label='Validation')

```

```
plt.title('Accuracy per Epoch')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.legend

plt.grid(True)

# Loss

plt.subplot(1,2,2)

plt.plot(history.history['loss'], label='Train')

plt.plot(history.history['val_loss'], label='Validation')

plt.title('Loss per Epoch')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend

plt.grid(True)

plt.tight_layout

plt.savefig("Reports/training_curve.png", dpi=300)

plt.show
```

```
#split_dataset.py

import os

import random

import shutil

# PATH DATASET SUMBER

source_dir = "Dataset/ProcessedData"
```

```

# OUTPUT DATASET BARU

output_dir = "DatasetFinal"

train_ratio = 0.7

val_ratio = 0.15

test_ratio = 0.15

random.seed(42)

classes = os.listdir(source_dir)

# BUAT FOLDER OUTPUT

for split in ["Train", "Validation", "Test"]:

    for cls in classes:

        os.makedirs(os.path.join(output_dir, split, cls), exist_ok=True)

# SPLIT DATASET

for cls in classes:

    cls_path = os.path.join(source_dir, cls)

    images = [

        f for f in os.listdir(cls_path)

        if f.lower.endswith((".jpg", ".jpeg", ".png"))

    ]

    random.shuffle(images)

    total = len(images)

    train_end = int(train_ratio * total)

    val_end = int((train_ratio + val_ratio) * total)

    train_imgs = images[:train_end]

    val_imgs = images[train_end:val_end]

```

```

test_imgs = images[val_end:]

print(f"\n📁 Kelas {cls}")

print("Train:", len(train_imgs))

print("Validation:", len(val_imgs))

print("Test:", len(test_imgs))

for img in train_imgs:

    shutil.copy(

        os.path.join(cls_path, img),

        os.path.join(output_dir, "Train", cls, img)

    )

for img in val_imgs:

    shutil.copy(

        os.path.join(cls_path, img),

        os.path.join(output_dir, "Validation", cls, img)

    )

for img in test_imgs:

    shutil.copy(

        os.path.join(cls_path, img),

        os.path.join(output_dir, "Test", cls, img)

    )

print(" Dataset berhasil di-split 70:15:15")

#evaluate_model.py

import os

```

```

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from sklearn.metrics import classification_report, confusion_matrix

os.makedirs("Reports", exist_ok=True)

# Load Model

model = load_model("Models/best_model.keras")

# Generator Test

test_datagen = ImageDataGenerator(rescale=1./255)

test_generator = test_datagen.flow_from_directory(

    "DatasetFinal/Test",

    target_size=(224,224),

    batch_size=32,

    class_mode='categorical',

    shuffle=False

)

class_names = list(test_generator.class_indices.keys)

# Evaluasi

loss, acc = model.evaluate(test_generator)

print("\nTest Accuracy:", acc)

```

```

print("Test Loss:", loss)

# Prediksi

y_pred = model.predict(test_generator)

y_pred_classes = np.argmax(y_pred, axis=1)

y_true = test_generator.classes

# Classification Report

report = classification_report(

    y_true,

    y_pred_classes,

    target_names=class_names

)

print("\nClassification Report:")

print(report)

# Simpan classification report

with open('Reports/classification_report.txt', 'w') as f:

    f.write("=== Classification Report ===\n\n")

    f.write(report)

# Confusion Matrix

cm = confusion_matrix(y_true, y_pred_classes)

# Tambahkan ke file TXT

with open('Reports/classification_report.txt', 'a') as f:

    f.write("\n\n=== Confusion Matrix ===\n\n")

    f.write(np.array2string(cm))

# Simpan Confusion Matrix sebagai GAMBAR

```

```
plt.figure(figsize=(6,6))

sns.heatmap(

    cm,

    annot=True,

    fmt='d',

    cmap='Blues',

    xticklabels=class_names,

    yticklabels=class_names

)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.savefig('Reports/confusion_matrix.png', dpi=300)

plt.show
```

```
#app.py
```

```
import streamlit as st
```

```
import numpy as np
```

```
import tensorflow as tf
```

```
from PIL import Image
```

```
import pandas as pd
```

```
# Load Model
```

```
model = tf.keras.models.load_model("Models/best_model.keras")
```

```

class_names = ['BSR', 'FSR', 'Sehat']

# Informasi Penyakit

info_penyakit = {

    "BSR": "Bacterial Stalk Rot (BSR) merupakan penyakit penting pada tanaman jagung yang disebabkan oleh bakteri Dickeya zeae. penyakit ini menyebabkan pembusukan basah pada jaringan batang di tandai dengan pelunakan jaringan, perubahan warna menjadi coklat dan di tandai dengan bau yang tidak sedap. Kondisi tersebut mengganggu aliran nutrisi sehingga tanaman mudah roboh.",

    "FSR": "Fusarium Stalk Rot (FSR) disebabkan beberapa jenis jamur fusarium yang sangat mengganggu pertumbuhan tanaman jagung, menurunkan nilai gizi, serta mengurangi hasil panen secara keseluruhan. tanaman mengalami layu, perubahan warna daun, dan pengeringan pada batang bagian bawah",

    "Sehat": "Tanaman tidak menunjukkan gejala penyakit."

}

# Auto Crop Tengah Gambar

def auto_crop_center(image):

    width, height = image.size

    crop_size = int(min(width, height) * 0.8)

    left = (width - crop_size) // 2

    top = (height - crop_size) // 2

    right = left + crop_size

```

```

    bottom = top + crop_size

    return image.crop((left, top, right, bottom))

# Fungsi Prediksi
def predict(image):

    image = image.resize((224,224))

    image = np.array(image) / 255.0

    image = np.expand_dims(image, axis=0)

    prediction = model.predict(image)[0]

    return prediction

# Session History
if "history" not in st.session_state:

    st.session_state.history = []

# WEBSITE

st.set_page_config(page_title="Deteksi Penyakit Jagung", layout="centered")

st.title("Sistem Klasifikasi Penyakit Batang Jagung")

st.divider

# Pilih Sumber Gambar

st.subheader("Pilih Sumber Gambar")

option = st.radio(

    "Sumber gambar:",

```

```

        ("Upload Galeri", "Kamera")
    )

    image = None

    if option == "Upload Galeri":
        uploaded_file = st.file_uploader("Upload Gambar", type=["jpg", "png", "jpeg"])

        if uploaded_file is not None:
            image = Image.open(uploaded_file).convert("RGB")

    elif option == "Kamera":
        camera_file = st.camera_input("Ambil Foto")

        if camera_file is not None:
            image = Image.open(camera_file).convert("RGB")

# Prediksi

if image is not None:
    st.subheader("Gambar Input")

    st.image(image, use_column_width=True)

# Auto Crop

cropped = auto_crop_center(image)

st.subheader("Auto Crop Fokus Batang")

st.image(cropped, use_column_width=True)

if st.button("Cek Hasil"):

```

```

pred = predict(cropped)

predicted_class = class_names[np.argmax(pred)]

confidence = np.max(pred) * 100

if confidence > 80: tingkat_keyakinan = "Model sangat yakin dengan hasil
prediksi."

elif confidence > 60: tingkat_keyakinan = "Model cukup yakin. Gejala
penyakit kemungkinan sedang berkembang."

else: tingkat_keyakinan = "Tingkat keyakinan rendah. Gejala penyakit
kemungkinan masih ringan atau citra kurang jelas."

st.warning(tingkat_keyakinan)

st.success(f"Jenis Penyakit: {predicted_class}")

st.info(f"Tingkat Keyakinan Model: {confidence:.2f}%")

st.write("Deskripsi:")

st.write(info_penyakit[predicted_class])

```