

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tanaman padi *Oryza sativa* L. merupakan salah satu bahan pangan utama yang memiliki peran penting dalam mendukung ketahanan pangan. Padi telah menjadi sumber makanan pokok bagi masyarakat di berbagai negara, khususnya di Asia Tenggara, termasuk Indonesia (Nurkayah et al., 2024). Hal ini menjadikan padi sebagai komoditas strategis yang perlu mendapatkan perhatian khusus, baik dari segi produksi, pengelolaan, hingga mitigasi risiko kerusakan.

Berdasarkan data dari *United States Department of Agriculture (USDA)*, jumlah produksi beras dunia pada tahun 2024 diperkirakan mencapai sekitar 509 juta metrik ton. China menjadi negara penyumbang produksi beras terbesar di dunia dengan jumlah produksi sekitar 146 juta metrik ton atau 28% dari total produksi global. Indonesia sendiri berada di peringkat keempat dengan produksi sekitar 34 juta metrik ton, menyumbang sekitar 6% dari total produksi global.

Pemerintah Republik Indonesia telah menetapkan sektor pertanian, khususnya produksi padi, sebagai prioritas utama dalam mencapai swasembada pangan. Beberapa program strategis yang dicanangkan meliputi perluasan lahan pertanian, penyediaan sarana produksi, modernisasi pertanian, serta program makan bergizi gratis. Namun, produktivitas padi di Indonesia masih berada di bawah potensi genetiknya. Secara genetika, tanaman padi memiliki potensi untuk menghasilkan hingga 12 ton/ha gabah (WAHYUNI et al., 2022). Namun, pada kenyataannya, produktivitas padi di Indonesia hanya mencapai rata-rata 5,242 ton/ha pada tahun 2024 (BPS Indonesia).

Beberapa faktor penyebab rendahnya produktivitas tersebut meliputi bencana alam seperti banjir dan kekeringan, serta serangan hama dan penyakit, salah satunya adalah *Bacterial Leaf Blight (BLB)* atau Hawar Daun Bakteri, yang menjadi ancaman utama bagi tanaman padi.

Bacterial Leaf Blight (BLB) merupakan penyakit yang disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae* dan dapat menurunkan produktivitas padi hingga 70% pada kondisi infeksi berat (IRRI, 2023). Penyebaran *BLB* sangat cepat, terutama di lingkungan dengan kelembapan tinggi dan suhu hangat. Infeksi yang parah dapat menyebabkan kehilangan hasil panen hingga 50% dari total produksi, terutama pada fase pengisian malai (Sayekti et al., 2024).

Metode deteksi manual selama ini dianggap kurang efektif karena membutuhkan waktu, tenaga, serta akurasi yang rendah dalam memantau lahan pertanian yang luas (Sishodia et al., 2020). Oleh karena itu, diperlukan solusi berbasis teknologi untuk deteksi dini penyakit *BLB* agar langkah mitigasi dapat dilakukan lebih cepat dan tepat. Seiring berkembangnya teknologi, penggunaan *Unmanned Aerial Vehicle (UAV)* atau drone di sektor pertanian semakin diminati.

Penelitian ini menggunakan drone yang terintegrasi dengan *Jetson Orin Nano* sebagai unit pemrosesan utama.

Penggunaan drone sebagai kendaraan bagi *Jetson Orin Nano* dalam pertanian sawah memberikan berbagai manfaat signifikan. Pemrosesan *real-time* memungkinkan deteksi dini penyakit tanaman langsung di lapangan, sehingga petani dapat segera mengambil tindakan pencegahan sebelum penyakit menyebar luas. Efisiensi biaya dan waktu juga tercapai, karena drone dapat memantau area sawah dengan cepat, sementara *Jetson Orin Nano* memproses data secara lokal, mengurangi kebutuhan tenaga kerja dan penggunaan pestisida (McCarthy et al., 2023). Selain itu, kemudahan mobilitas membuat drone dapat bergerak mandiri di area yang sulit dijangkau, seperti sawah berlumpur atau bertingkat, serta menyesuaikan ketinggian dan sudut kamera untuk pengambilan data yang optimal. Terakhir, monitoring berkala dan analisis prediktif memungkinkan petani melacak perkembangan penyakit dari waktu ke waktu dan merencanakan tindakan pencegahan berdasarkan pola yang teridentifikasi, meningkatkan produktivitas dan mendukung pertanian berkelanjutan (McCarthy et al., 2023). Dengan demikian, pemanfaatan drone yang diintegrasikan dengan *Jetson Orin Nano* sebagai solusi deteksi dini penyakit *BLB* pada tanaman padi memiliki potensi besar dalam meningkatkan produktivitas pertanian, mendukung ketahanan pangan, dan mewujudkan pertanian yang lebih berkelanjutan di Indonesia.

1.2 Landasan Teori

1.2.1 Tanaman Padi



Gambar 1. Area persawahan

Tanaman padi (*Oryza sativa* L.) merupakan salah satu tanaman pangan paling penting di dunia, khususnya di kawasan Asia Tenggara. Padi berperan sebagai sumber utama karbohidrat bagi sebagian besar penduduk dunia dan menjadi komoditas strategis dalam mendukung ketahanan pangan global (Sari et al., 2025). Di Indonesia, sekitar 95% penduduk mengonsumsi beras sebagai bahan pangan

pokok. Seiring dengan pertambahan jumlah penduduk, permintaan terhadap kebutuhan beras pun terus mengalami peningkatan (Sahmanda et al., 2021).



Gambar 2. Tanaman Padi

Tanaman *Oryza sativa* L. termasuk tanaman berumur relatif pendek, yaitu sekitar 5-6 bulan dalam satu siklus tanam. Secara morfologis, padi memiliki batang beruas dengan struktur malai yang menghasilkan butir gabah sebagai hasil utama panen. Tanaman ini juga memiliki kemampuan adaptasi yang tinggi, sehingga dapat dibudidayakan pada berbagai ekosistem, baik sawah maupun rawa (Priyanto et al., 2025).

1.2.2 Penyakit *Bacterial Leaf Blight* (BLB)

Layaknya tanaman lain, padi juga rentan terhadap serangan penyakit. Salah satu penyakit utama pada padi adalah *Bacterial Leaf Blight* (BLB). Penyakit BLB merupakan penyakit yang disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae*. Patogen ini pertama kali diidentifikasi di wilayah Asia dan sejak itu menyebar ke berbagai negara penghasil padi utama, termasuk Asia Tenggara, India, Tiongkok, dan Afrika. BLB diakui secara luas sebagai salah satu penyakit paling merusak yang menyerang tanaman padi, dengan siklus infeksinya berlangsung mulai dari tahap bibit hingga periode menjelang panen. Penyakit ini menunjukkan karakteristik infeksi sistemik yang terutama menyerang jaringan pembuluh tanaman serta menyebar dengan cepat pada kondisi kelembapan tinggi atau sistem irigasi intensif (Adedibu et al., 2024).



Gambar 3. Area persawahan yang terinfeksi BLB

Di Indonesia, BLB telah menjadi salah satu penyakit padi yang paling umum dijumpai pada lahan sawah beririgasi. Penyakit ini sering muncul selama musim hujan atau pada periode dengan kelembapan udara yang tinggi. Hasil penelitian menunjukkan bahwa beberapa varietas padi memiliki tingkat kerentanan yang lebih tinggi terhadap penyakit BLB, sementara varietas tahan penyakit tetap dapat terinfeksi apabila kondisi lingkungan mendukung pertumbuhan bakteri. Patogen ini menjadi tantangan besar bagi petani karena sulit dikendalikan secara efektif menggunakan pestisida kimia. Oleh karena itu, deteksi dini serta pengelolaan varietas tahan penyakit menjadi strategi utama dalam upaya pengendalian BLB (Rehman & Atiq, 2024).

Secara umum, penyakit ini menyerang bagian daun yang ditandai dengan munculnya bercak berwarna hijau keabu-abuan yang kemudian berubah menjadi cokelat kekuningan. Infeksi yang parah dapat menyebabkan daun mengering secara menyeluruh sehingga mengganggu proses fotosintesis dan menurunkan vitalitas tanaman (Ahmed et al., 2022). Kondisi ini dikenal dengan istilah *kresek*, yaitu gejala khas pada tanaman muda yang menunjukkan daun layu dan mati mendadak akibat infeksi bakteri.



Gambar 4. Daun padi yang terinfeksi BLB

Derajat keparahan infeksi BLB diklasifikasikan berdasarkan persentase luas permukaan daun yang terinfeksi, mulai dari *Grade 0* (tidak ada lesi) hingga *Grade 9* (infeksi sangat parah dengan luas daun terinfeksi melebihi 76-100%) (Ahmad et al., 2023). Tingkat penurunan hasil panen bervariasi tergantung pada tahap infeksi. Infeksi berat dapat menyebabkan kehilangan hasil panen hingga 75% (Sabri et al., 2023). Di Indonesia, penurunan hasil panen yang dilaporkan selama musim hujan berkisar antara 21-36%, dan 18-28% selama musim kemarau (Joko et al., 2019).

1.2.3 Visi Komputer

Computer vision merupakan cabang dari kecerdasan buatan (*Artificial Intelligence / AI*) yang memungkinkan komputer dan sistem untuk memahami, menganalisis, serta menginterpretasi informasi visual dari dunia nyata, seperti citra, video, dan input visual lainnya (Zhang, 2024). Bidang ini bersifat interdisipliner karena menghubungkan *artificial intelligence* dengan pemrosesan citra digital, dengan tujuan memberikan kemampuan kepada mesin untuk mengenali pola visual dan mengambil keputusan berdasarkan data tersebut (David Shamoo Excel, 2024). Sistem *computer vision* bekerja dengan cara yang menyerupai penglihatan manusia, namun memiliki keunggulan dalam kecepatan dan skala. Sistem yang telah dilatih dapat menganalisis ribuan objek atau proses dalam hitungan menit, mendeteksi cacat yang tidak terlihat oleh mata manusia, serta melampaui kemampuan analisis manusia secara signifikan (Lindroth et al., 2024).

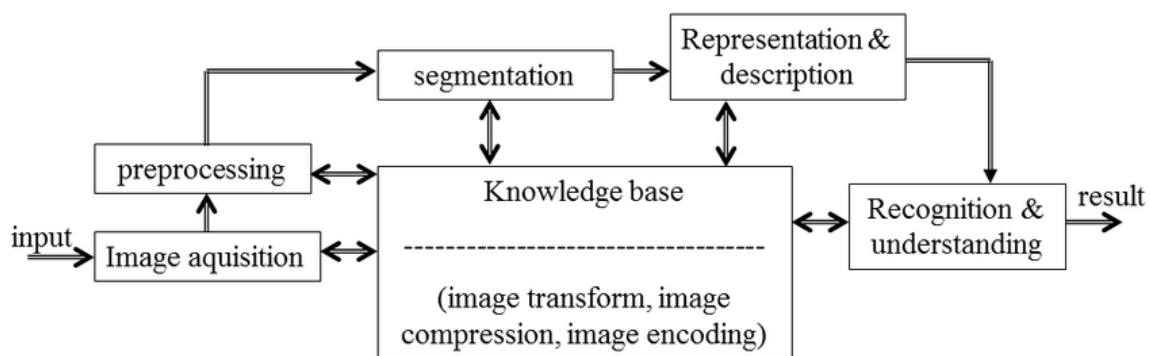
Perkembangan teknologi *computer vision* meningkat pesat dalam beberapa tahun terakhir, didorong oleh kemajuan teknik *deep learning* dan arsitektur *Convolutional Neural Networks* (CNNs). CNN bekerja dengan menggunakan operasi konvolusi untuk memecah citra menjadi bagian-bagian kecil (piksel), di mana setiap piksel diberi label untuk membantu model melakukan prediksi secara akurat. Pendekatan ini telah menunjukkan kinerja tingkat ahli dalam berbagai tugas, seperti klasifikasi citra, deteksi objek, dan segmentasi semantik di berbagai bidang, termasuk kedokteran, sistem pengawasan, serta kendaraan otonom (David Shamoo Excel, 2024). Selain CNN, teknologi mutakhir lainnya seperti *Vision Transformers* (ViTs) dan *Generative Adversarial Networks* (GANs) juga mengalami perkembangan signifikan. *Vision Transformers* mengadopsi arsitektur *transformer* yang umumnya digunakan dalam *natural language processing*, dan telah menunjukkan performa tinggi dalam *benchmark* klasifikasi citra (Lindroth et al., 2024).

Aplikasi *computer vision* sangat luas dan terus berkembang di berbagai sektor industri. Dalam bidang kesehatan, teknologi ini dimanfaatkan untuk analisis citra medis seperti *X-ray*, *MRI*, dan *CT scan* dalam membantu diagnosis penyakit dengan tingkat akurasi yang tinggi (Lindroth et al., 2024). Di bidang pertanian, kombinasi antara *deep learning* dan *computer vision* memberikan kontribusi signifikan dalam deteksi penyakit tanaman karena kemampuannya dalam mengenali pola visual dengan presisi tinggi (Yani et al., 2024). Sementara itu, di industri otomotif, *computer vision* berperan penting dalam sistem kendaraan otonom untuk mendeteksi objek, menganalisis kondisi lalu lintas, dan membuat keputusan navigasi secara *real-time*. Pada sektor manufaktur, teknologi ini digunakan untuk pembacaan

teks dan *barcode*, inspeksi kualitas produk, hingga perakitan otomatis (Lindroth et al., 2024). Kemajuan pesat dalam penerapan *computer vision* di berbagai bidang menunjukkan potensi transformasional teknologi ini. Namun demikian, tantangan masih muncul dalam hal kebutuhan pemrosesan *real-time*, skalabilitas sistem, serta isu etika terkait pengambilan keputusan otomatis yang berbasis kecerdasan buatan (David Shamoo Excel, 2024).

1.2.4 Pengolahan Citra Digital

Pemrosesan gambar digital mencakup serangkaian teknik yang memanfaatkan komputer untuk memanipulasi dan menganalisis gambar. Gambar digital diwakili sebagai matriks piksel, di mana setiap piksel menyimpan informasi mengenai kecerahan atau warna (Liu, 2024). Representasi numerik ini memungkinkan gambar diproses melalui berbagai operasi matematis, seperti transformasi, penyaringan, dan pemrosesan spasial. Tujuan pemrosesan gambar dapat mencakup peningkatan kualitas gambar, penekanan pada fitur tertentu, atau ekstraksi informasi relevan dari gambar (Liu, 2024).



Gambar 5. Proses pengolahan citra digital

Tahap fundamental dalam pemrosesan gambar digital adalah pra-pemrosesan, yang meningkatkan kualitas gambar melalui penanganan awal untuk memudahkan analisis selanjutnya. Pra-pemrosesan dapat mencakup operasi seperti standarisasi, pengurangan *noise*, pemindaian, atau peningkatan kontras. Tahap ini diikuti oleh tahap ekstraksi fitur, yang mengekstrak karakteristik penting dari gambar, seperti bentuk, tepi, tekstur, atau warna. Informasi ini menjadi dasar untuk proses analisis selanjutnya (Divya M et al., 2024).

Selain itu, pemrosesan gambar digital melibatkan teknik segmentasi, yang membagi gambar menjadi wilayah atau segmen yang lebih kecil, sehingga memisahkan objek tertentu dari latar belakangnya (Abdulateef & Salman, 2021). Segmentasi merupakan langkah kritis karena menentukan area mana yang memerlukan analisis lebih detail. Setelah segmentasi selesai, gambar biasanya masuk ke tahap klasifikasi, di mana objek dikategorikan ke dalam kelompok tertentu berdasarkan fiturnya (Cuevas et al., 2025).

Melalui tahap-tahap yang berbeda ini, pemrosesan gambar digital telah menjadi landasan utama dalam aplikasi pemrosesan visual modern. Mulai dari analisis medis dan sistem keamanan hingga pengembangan teknologi berbasis kecerdasan buatan, pemrosesan gambar digital memainkan peran kunci dalam mengubah data visual mentah menjadi informasi yang bermakna (Cuevas et al., 2025).

1.2.5 *Unmanned Aerial Vehicle (UAV)*

Unmanned Aerial Vehicle (UAV), yang lebih dikenal sebagai drone, adalah pesawat tanpa awak yang mampu terbang secara otonom melalui kendali jarak jauh atau urutan program yang telah ditentukan. Biasanya dilengkapi dengan sensor, kamera, atau instrumen lainnya, perangkat ini secara efisien mengumpulkan data dari ketinggian (Arafat et al., 2023). Dalam beberapa tahun terakhir, teknologi UAV telah mengalami kemajuan pesat, yang mengakibatkan adopsi yang semakin luas, bobot yang lebih ringan, dan kemampuan yang ditingkatkan untuk penerbangan stabil dan cakupan yang luas (Arafat et al., 2023).

Di sektor pertanian, UAV digunakan secara luas untuk mengumpulkan data lahan melalui citra udara beresolusi tinggi. Data ini digunakan untuk survei regional, pemantauan pertumbuhan tanaman, dan identifikasi hama serta penyakit (Aslan et al., 2022). Dibandingkan dengan metode tradisional yang bergantung pada pengamatan lapangan manual, drone menawarkan keunggulan signifikan dalam cakupan yang lebih luas, penghematan waktu, dan detail visual yang lebih baik. Akibatnya, drone telah menjadi teknologi modern yang vital dalam mendukung praktik pertanian presisi dan pengelolaan lahan yang efisien (García-Munguía et al., 2024).



Gambar 6. DJI Mavic 3 Classic

Salah satu UAV yang digunakan dalam penelitian ini adalah DJI Mavic 3 Classic, yang terkenal dengan kamera resolusi tinggi dan kemampuan penerbangan stabilnya, memungkinkan fotografi udara yang efisien. Sensor kamera menangkap gambar detail, dengan data visual yang dihasilkan diproses lebih lanjut untuk memantau kondisi tanaman. Fungsi penerbangan otomatis dan daya tahan yang

lebih lama menjadikannya pilihan ideal untuk pemantauan lapangan pertanian yang efisien.

Tabel 1. Spesifikasi DJI Mavic 3 Classic

Kategori	Spesifikasi
Kamera	Sensor CMOS 4/3 Hasselblad, 20 MP
	Lensa 24 mm, FOV 84°, <i>aperture</i> f/2.8 - f/11
	ISO 100-6400 (Foto & Video), hingga 12800 (Video malam)
	Resolusi Foto Maksimum 5280 x 3956
	Format Foto JPEG, DNG (RAW)
Video	Format MP4/MOV (H.264, H.265)
	Resolusi Video Maksimum 5.1K 50fps, 4K 120fps, FHD 200fps
	<i>Bitrate</i> Video Maksimum 200 Mbps
	<i>Digital Zoom</i> hingga 3x
	10-Bit D-Log & Hasselblad NCS, Rentang Dinamis 12,8 stop
Fitur Terbang	Sensor anti-tabrak <i>omnidirectional</i> (depan, belakang, samping, atas, bawah)
	Jarak transmisi video HD hingga 15 km (O3+)
	Maksimum ketinggian terbang 6000 m
	Maksimum kecepatan 21 m/s
	Ketahanan angin Level 5 (hingga 12 m/s)
Baterai & Dimensi	Waktu terbang maksimal hingga 46 menit
	Baterai Li-ion 4S 5000 mAh
	Bobot drone 895 gram
	Dimensi tanpa baling-baling 221 x 96.3 x 90.3 mm
	Dimensi dengan baling-baling 347.5 x 283 x 107.7 mm
Penyimpanan	Penyimpanan internal 8 GB
	Slot microSD sampai 2 TB
Navigasi & Kontrol	GPS + Galileo + BeiDou
	Mode ActiveTrack 5.0, MasterShots, QuickShots, Hyperlapse, Panorama
	<i>Advanced Return to Home</i>

1.2.6 NVIDIA Jetson Orin Nano



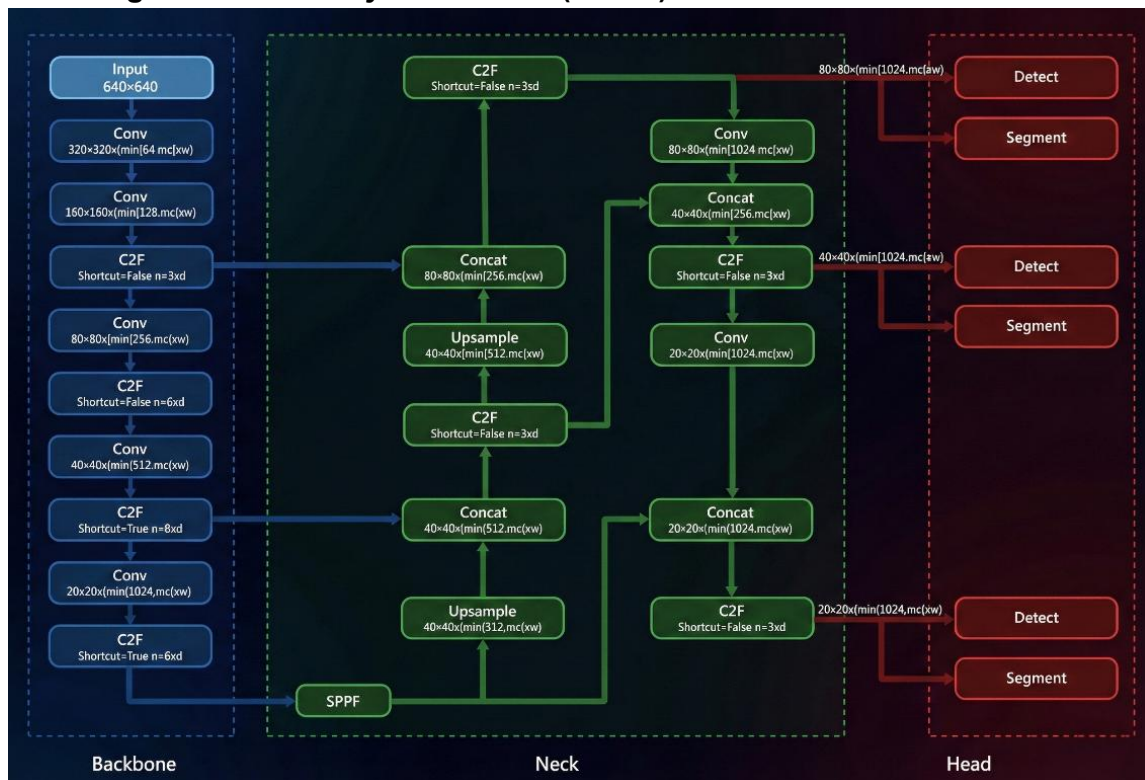
Gambar 7. NVIDIA Jetson Orin Nano

Jetson Orin Nano dirancang khusus untuk mempercepat pengembangan aplikasi AI tingkat pemula di tepi jaringan. Produk ini menetapkan standar baru dalam pembuatan robot AI tingkat pemula, drone pintar, dan kamera pintar, sambil menyederhanakan proses pengembangan untuk seri Jetson Orin Nano. Produk ini tersedia dalam varian 4GB dan 8GB. Varian 8GB dilengkapi dengan GPU berarsitektur Ampere 1024-core dan 32 Tensor Cores, beroperasi pada frekuensi clock puncak 625 MHz. Varian 4GB menggunakan GPU berarsitektur Ampere 512-core dan 16 Tensor Cores, juga beroperasi pada frekuensi clock puncak 625 MHz. Jetson Orin Nano 8GB mencapai kinerja puncak 40 TOPs, melampaui bahkan Jetson AGX Xavier 32GB. Eksperimen kami menggunakan varian memori 8GB yang beroperasi dalam mode daya 15W untuk memaksimalkan kinerja (Pham et al., n.d.).

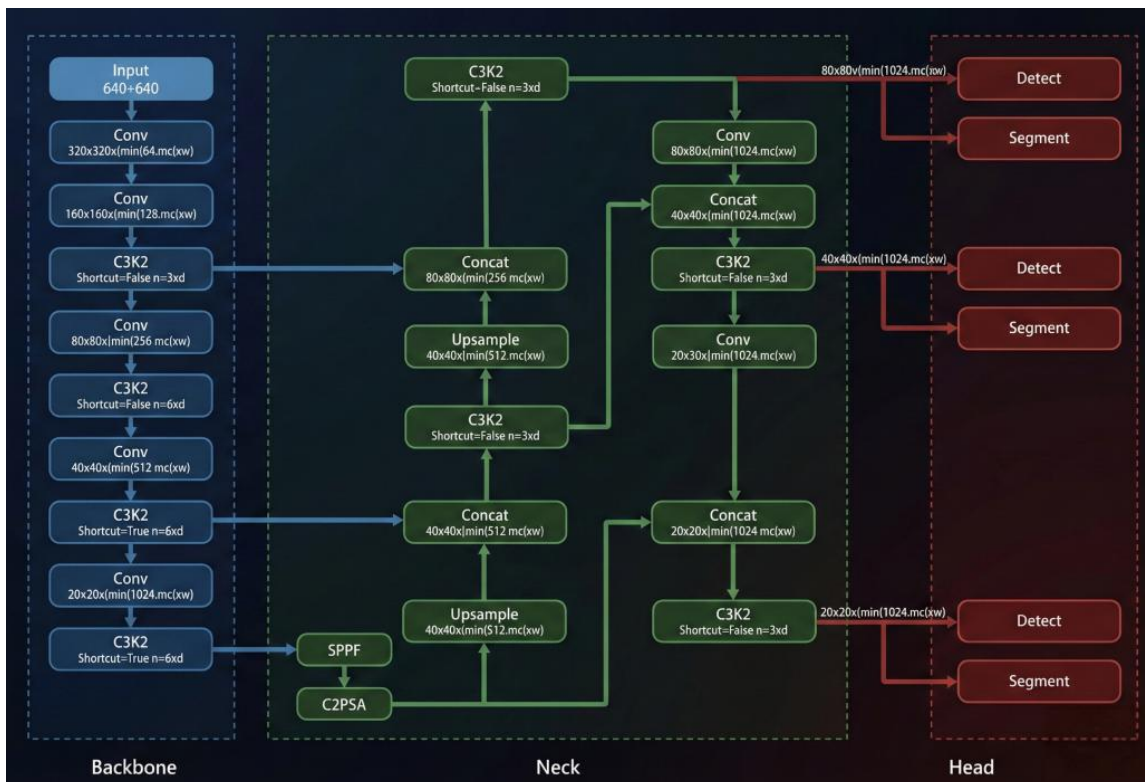
Tabel 2. Spesifikasi NVIDIA Jetson Orin Nano

<i>Publish Day</i>	January 2023
<i>AI Speed</i>	40 TOPs
<i>GPU</i>	1024-core NVIDIA Ampere GPU with 32 Tensor Cores
<i>CPU</i>	6-core Arm® Cortex®- A78AE v8.2 64-bit, 2.2GHz
<i>Memory</i>	8GB 128-bit LPDDR5 2133MHz frequency
<i>Power</i>	7W - 15W

1.2.7 Algoritma You Only Look Once (YOLO)



Gambar 8. Arsitektur YOLOv8-seg



Gambar 9. Arsitektur YOLO11-seg

YOLO (*You Only Look Once*) adalah algoritma *object detection* berbasis *deep learning* yang menandai paradigma baru pada deteksi objek dengan memproses seluruh gambar dalam satu kali pengolahan (*one-stage detector*), sehingga menghasilkan deteksi waktu-nyata dengan akurasi tinggi dan efisiensi komputasi (Widayani et al., 2024). YOLO berkembang dari versi awal yang menekankan kecepatan, hingga versi-versi lanjutan seperti YOLOv3, YOLOv4, hingga YOLOv8, yang terus meningkatkan presisi melalui modifikasi arsitektur, *backbone* seperti CSPDarknet53, teknik fusi fitur, serta *loss function* canggih dan augmentasi data. Di berbagai bidang, algoritma YOLO digunakan secara luas untuk mendeteksi dan mengklasifikasikan objek atau abnormalitas pada citra dengan berbagai metrik evaluasi seperti *precision*, *recall*, dan mAP (*mean Average Precision*) yang terus meningkat di tiap generasi algoritma, sehingga sangat berguna dalam mendukung *clinical decision making* maupun aplikasi dunia nyata lainnya (Sohan et al., 2024).

YOLOv8 memperluas kemampuannya tidak hanya untuk deteksi objek, tetapi juga untuk segmentasi yang diimplementasikan dalam model YOLOv8-Seg. Model ini memiliki arsitektur modular yang terdiri dari CSPDarknet53 untuk ekstraksi fitur, *C2F Module* yang menggabungkan fitur level tinggi dan konteks untuk memperkaya representasi fitur, serta *Detection Head independent* yang mendukung deteksi *multi-task* secara efisien. Khusus pada segmentasi, YOLOv8-Seg mampu melakukan segmentasi semantik, menghasilkan peta mask (*mask map*) yang presisi untuk area objek, sehingga sangat relevan untuk tugas-tugas seperti segmentasi organ atau lesi pada citra medis. Berbagai studi menunjukkan, dengan praproses dan augmentasi optimal serta *hyperparameter tuning*, YOLOv8-Seg menghasilkan performa yang sangat baik, baik pada deteksi maupun segmentasi dan memberikan manfaat signifikan untuk aplikasi diagnosis medis serta bidang lain yang memerlukan segmentasi cepat dan akurat (Widayani et al., 2024).

1.2.8 Evaluasi Model

Evaluasi model merupakan komponen fundamental dalam pengembangan sistem pembelajaran mesin, khususnya untuk aplikasi pertanian seperti deteksi penyakit tanaman atau segmentasi citra lahan. Evaluasi yang komprehensif memungkinkan peneliti untuk menilai performa model secara objektif dan memastikan kesesuaian model untuk implementasi di lapangan. Dalam konteks pertanian, evaluasi yang akurat sangat penting karena keputusan agronomis bergantung pada presisi dan reliabilitas sistem otomatis yang digunakan untuk mendukung produktivitas dan keberlanjutan pertanian.

Intersection Over Union (IOU). *Intersection over Union* (IoU) adalah metrik klasik untuk mengevaluasi kinerja model dalam *object detection* dan segmentasi. IoU menghitung rasio antara *overlap* dan *union* antara *predicted mask* dan *ground truth mask*, yang mengukur *intersection* dari kedua *mask* tersebut.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Gambar 10. Perhitungan IoU

Dalam matematika ditulis sebagai,

$$\text{Intersection of Union} = \frac{A \cap B}{A \cup B} \quad (1)$$

Di mana A merupakan *predicted area* dari mask dan B merupakan *ground truth* yang berisi objek. Kinerja model meningkat seiring dengan peningkatan nilai IoU, dimana nilai IoU yang lebih tinggi mengindikasikan perbedaan yang lebih kecil antara kandidat yang dihasilkan dengan *ground truth bounding boxes*.

Dice Score. *Dice Score* atau *Dice Similarity Coefficient (DSC)* adalah metrik yang mengukur *similarity* antara dua set dengan menghitung *overlap* relatif mereka. *Dice Score* dihitung menggunakan

$$\text{Dice Score} = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (2)$$

Di mana A merupakan *predicted mask* dan B merupakan *ground truth mask*. Nilai DSC yang mendekati 1 mengindikasikan tingkat *similarity* yang lebih tinggi antara *predicted* dan *actual lesion boundaries*. Dalam tugas segmentasi citra, DSC yang tinggi menunjukkan bahwa algoritma berhasil mensegmentasi target *region* dengan *minimal errors* (*omission* atau *inclusion* dari piksel yang tidak perlu). Dalam studi segmentasi *left ventricle*, DSC digunakan untuk mengukur kesamaan antara *mask* yang diprediksi dengan *ground truth*, dimana nilai yang lebih tinggi menunjukkan keberhasilan segmentasi yang lebih baik dengan *minimal error*.

Confusion Matrix. *Confusion matrix* merupakan metode yang digunakan untuk mengukur kinerja suatu metode klasifikasi pada *data mining* atau *machine learning*. *Confusion matrix* menyajikan informasi perbandingan antara hasil prediksi yang dilakukan oleh sistem dengan hasil klasifikasi yang sebenarnya dalam bentuk tabel. Tabel ini memuat beberapa istilah penting, yaitu:

- *True Positive* (TP) yang menunjukkan jumlah data dengan kelas positif yang diklasifikasikan benar,
- *True Negative* (TN) untuk data kelas negatif yang juga terklasifikasi benar,
- *False Positive* (FP) untuk data negatif yang salah diprediksi sebagai positif,
- *False Negative* (FN) untuk data positif yang salah diprediksi sebagai negatif.

Dengan adanya *confusion matrix*, dapat dihitung berbagai metrik kinerja seperti akurasi, presisi, *recall*, dan spesifisitas dalam proses evaluasi model klasifikasi (Sathyanarayanan, 2024).

Frame per Second. *Frames per Second* (FPS) merupakan metrik penting untuk mengukur kecepatan inferensi model segmentasi, terutama pada aplikasi yang membutuhkan pemrosesan *real-time* seperti *autonomous driving*, *medical imaging*, *plant disease detection*, dan *augmented reality*.

$$FPS = \frac{\text{Jumlah frame yang diproses}}{\text{Waktu (detik)}} \quad (3)$$

FPS mengindikasikan jumlah *frame* yang dapat diproses oleh model dalam satu detik, dan nilai FPS yang lebih tinggi menunjukkan bahwa model mampu melakukan segmentasi dengan lebih cepat. Pada konteks segmentasi semantik *real-time*, model dengan FPS tinggi dapat memberikan respons langsung terhadap input visual, yang sangat krusial untuk sistem yang memerlukan *feedback* segera.

1.3 Tujuan dan Manfaat

1.3.1 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk mengetahui penerapan visi komputer dalam sistem deteksi penyakit *Bacterial Leaf Blight* (BLB) pada tanaman padi.
2. Menganalisis kinerja dari penerapan visi komputer dalam sistem deteksi penyakit *Bacterial Leaf Blight* (BLB) pada tanaman padi.

1.3.2 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mengefisienkan waktu dan tenaga dalam memantau lahan persawahan dalam jangkauan luas.
2. Membantu para petani dalam mengidentifikasi penyakit *Bacterial Leaf Blight* (BLB).

3. Mendeteksi dini serangan penyakit agar dapat segera ditindaklanjuti sehingga dapat mengurangi kerugian akibat serangan penyakit.
4. Meningkatkan manfaat teknologi kamera drone dalam bidang pertanian.

BAB II

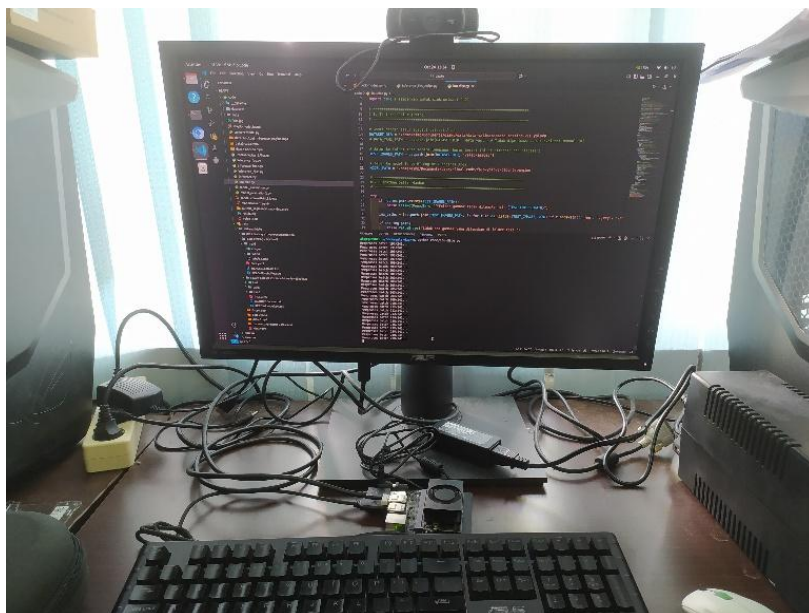
METODE PENELITIAN

2.1 Tempat dan Waktu

Penelitian dilakukan mulai sejak disetujuinya proposal penelitian pada Maret 2025 sampai Oktober 2025. Penulis melakukan pengambilan data penelitian di area persawahan yang berlokasi di Desa Baju Bodoa, Kec. Maros Baru, Kabupaten Maros, Sulawesi Selatan 90512 seperti ditunjukkan pada Gambar 11.



(a)



(b)

Gambar 11. Lokasi penelitian: (a) Area persawahan tempat pengambilan data; (b) Laboratorium Artificial Intelligence (AI)

Adapun pelaksanaan penelitian yang dilakukan di Laboratorium AI (*Artificial Intelligence*), Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin Kabupaten Gowa.

2.2 Benda Uji dan Alat

Benda dan alat yang digunakan dalam penelitian ini meliputi perangkat lunak dan perangkat keras berikut:

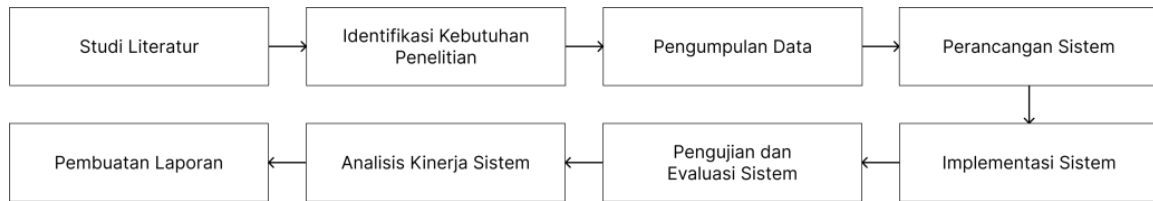
1. Perangkat lunak
 - a. *Operating System*: Windows 11 64-bit
 - b. Visual Studio Code
 - c. Kaggle, *accelerator* GPU P100
 - d. Microsoft Excel
 - e. Microsoft Word
 - f. Microsoft Powerpoint
 - g. Google Chrome
 - h. Notepad
 - i. Mendeley
 - j. Figma
2. Perangkat keras
 - a. HP Pavilion 14-bf193TX (Core i5-8250U, RAM 12GB, 1TB HDD + 256GB SSD, Win 11, 14 Inch, Silver)
 - b. Drone DJI Mavic 3 Classic
 - c. Jetson Orin Nano Super Developer Kit 8GB
3. Bahasa pemrograman
 - a. Python v3.10.12
4. Library
 - a. JetPack 6.2 oleh Tim NVIDIA
Berguna untuk *Deployment* dan Pengembangan AI di *Edge* yang menyediakan *software stack* terintegrasi (OS, driver, CUDA, TensorRT) khusus untuk perangkat Jetson Orin Nano/NX.
 - b. torch (PyTorch) v2.5.0a0+... oleh Tim PyTorch (Meta AI)
Berguna untuk *Deep Learning* yang menyediakan *framework* yang fleksibel, berkinerja tinggi (terutama di GPU/CUDA), dan berbasis graf komputasi dinamis. Digunakan sebagai *backend* utama untuk menjalankan model Jaringan Saraf Tiruan.
 - c. ultralytics v8.3.202 oleh Tim Ultralytics
Berguna untuk Deteksi Objek *Real-Time* yang menyediakan implementasi model YOLO (*You Only Look Once*) yang sederhana dan efisien untuk tugas visi komputer seperti klasifikasi, deteksi, dan segmentasi.
 - d. opencv-python v4.11.0.86 oleh Tim OpenCV
Berguna untuk Visi Komputer (*Computer Vision*) yang mencakup fungsi untuk membaca dan menulis *frame* video/gambar,

memanipulasi gambar (misalnya, *resize*, transformasi), dan menampilkan hasil inferensi deteksi objek.

- e. *numpy* v1.23.5 oleh Travis E. Oliphant et al.
Berguna untuk Komputasi Numerik yang menyediakan dukungan array dan matriks besar multidimensi, serta fungsi-fungsi matematis yang efisien untuk operasi pada data, termasuk data gambar.
- f. *pandas* v1.3.5 oleh Tim The Pandas Development
Berguna untuk Manipulasi dan Analisis Data yang menyediakan struktur data (seperti *DataFrame*) dan operasi yang efisien untuk membersihkan, memproses, dan menganalisis data terstruktur.
- g. *matplotlib* v3.5.1 oleh John D. Hunter & Michael Droettboom et al.
Berguna untuk Visualisasi Data yang digunakan untuk menggambar grafik statis, interaktif, dan visualisasi kompleks lainnya di Python.
- h. *scikit-learn* (*sklearn*) v1.7.2 oleh Tim Scikit-learn
Adalah pustaka Machine Learning untuk Python. Berguna menyertakan algoritma untuk klasifikasi, regresi, pengelompokan, pemrosesan pra-data, pemilihan dan evaluasi model, dan lain-lain.
- i. *tensorflow* v2.15.1 oleh Tim Google
Berguna untuk Deep Learning yang menyediakan framework untuk pembangunan dan pelatihan model Jaringan Saraf Tiruan, sering digunakan bersama Keras sebagai antarmuka tingkat tinggi.
- j. *onnx* v1.18.0 oleh Tim ONNX Community (Microsoft, Facebook, dll.)
Berguna untuk Interoperabilitas Model AI yang menyediakan format terbuka untuk merepresentasikan model *machine learning*. Memungkinkan model dipertukarkan antara framework yang berbeda (*torch* ke *onnx* ke *runtime*).
- k. *onnxruntime-gpu* v1.20.0 oleh Tim Microsoft
Berguna untuk Akselerasi Inferensi yang menyediakan *runtime* berkinerja tinggi, terutama dioptimasi untuk komputasi GPU, untuk menjalankan model yang disimpan dalam format ONNX.
- l. *tensorrt* v10.3.0 oleh Tim NVIDIA
Berguna untuk Optimasi Inferensi pada perangkat NVIDIA. Melakukan optimasi grafik jaringan saraf untuk memaksimalkan *throughput* dan meminimalkan latensi saat melakukan inferensi, umumnya pada perangkat Jetson atau server dengan GPU NVIDIA.

2.3 Tahapan penelitian

Penelitian dilakukan dengan beberapa tahapan seperti yang ditunjukkan pada gambar berikut:



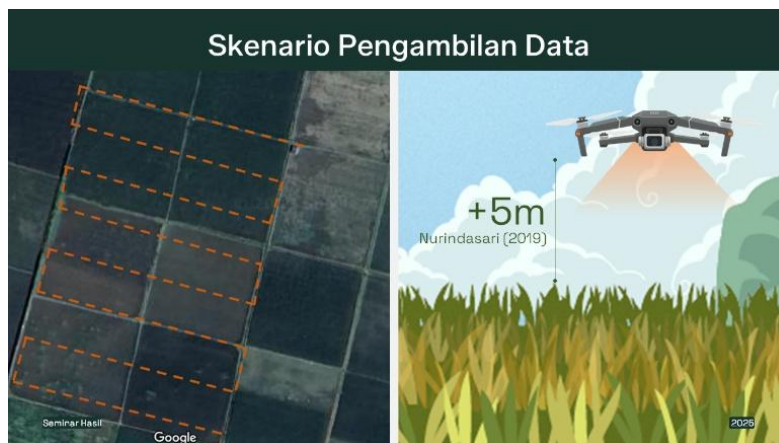
Gambar 12. Tahapan penelitian

2.4 Teknik Pengambilan Data

Data yang digunakan dalam penelitian ini merupakan kumpulan data primer.

2.4.1 Data primer

Data primer yang digunakan dalam penelitian ini adalah data yang diperoleh langsung dari area persawahan yang merupakan objek utama penelitian. Proses pengambilan data diawali dengan melakukan koordinasi dengan Balai Proteksi Tanaman Pangan & Hortikultura (UPT BTPH) Provinsi Sulawesi Selatan sebagai instansi yang memiliki kompetensi dalam mengidentifikasi area persawahan yang terinfeksi penyakit BLB pada tanaman padi. Keterlibatan pihak BTPH sangat penting untuk memastikan akurasi dalam penentuan lokasi pengambilan sampel penelitian.

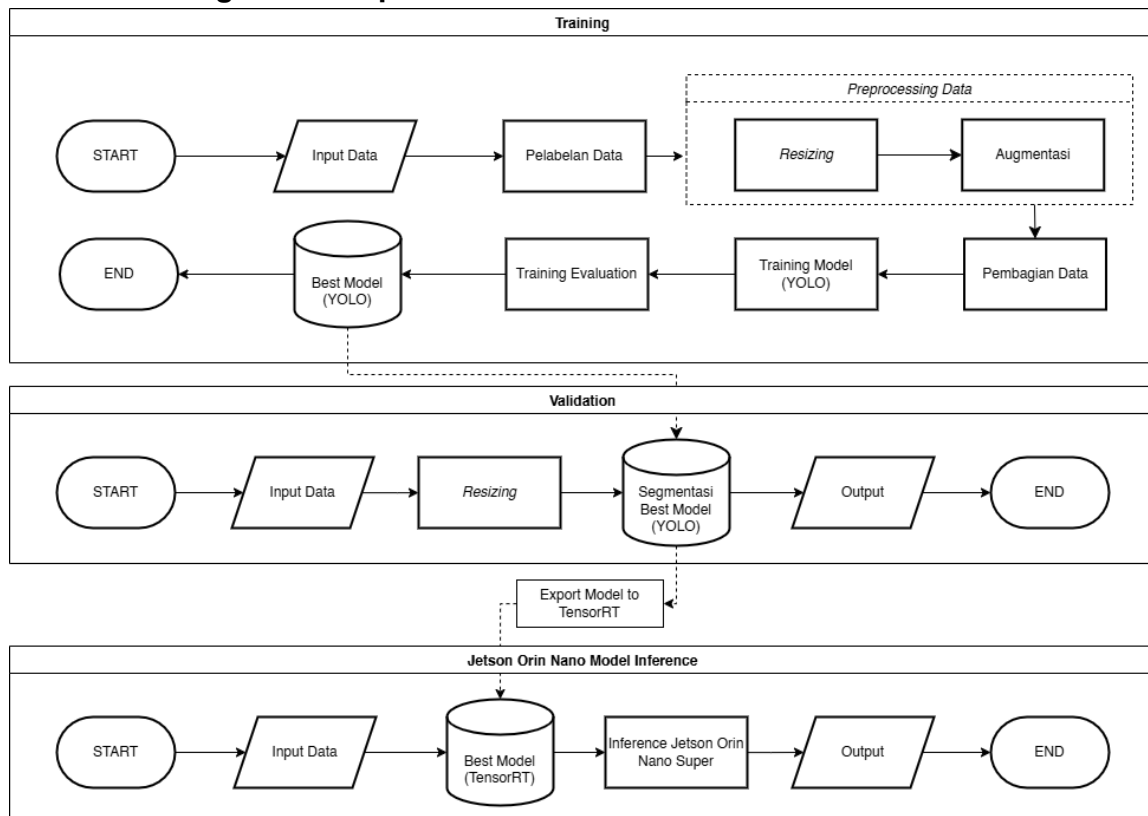


Gambar 13. Skenario pengambilan dataset

Setelah mengetahui area yang terinfeksi, dilakukan perekaman menggunakan drone DJI Mavic 3 Classic dengan ketinggian terbang sekitar 5 meter di atas permukaan tanah. Posisi kamera drone berada di sudut 45 derajat.

Adapun waktu pengambilan data dilakukan pada pagi hari sekitar pukul 08.00 - 10.00 waktu setempat dengan kondisi cuaca cerah, waktu ini dipilih bertujuan untuk mengurangi *noise* bayangan drone yang disebabkan oleh cahaya sinar matahari.

2.5 Perancangan dan Implementasi Sistem



Gambar 14. Perancangan alur sistem

2.5.1 Tahap Training Model Segmentasi

1. Input Data

Proses pertama dalam pengembangan sistem adalah pengumpulan data citra tanaman padi yang menjadi dasar dalam pelatihan model segmentasi. Data yang digunakan merupakan data primer seperti yang telah dijelaskan pada sub-bab 2.4.1 Data Primer.

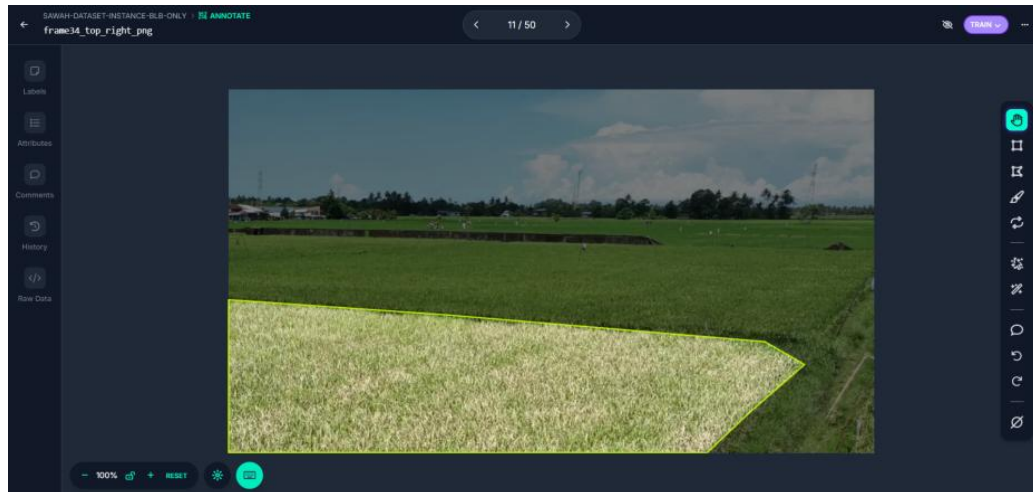


Gambar 15. Contoh dataset

Dataset yang terkumpul berjumlah 491 citra, terdiri atas campuran area daun padi sehat dan terinfeksi. Variasi ini penting agar model dapat membedakan karakteristik visual antara daun normal dan daun yang menunjukkan gejala BLB.

2. Pelabelan Data

Tahap berikutnya adalah pelabelan (annotasi) citra untuk menandai area daun yang terinfeksi penyakit BLB. Proses ini dilakukan secara manual menggunakan *platform* Roboflow, yang menyediakan antarmuka visual untuk menggambar area segmentasi pada setiap citra.



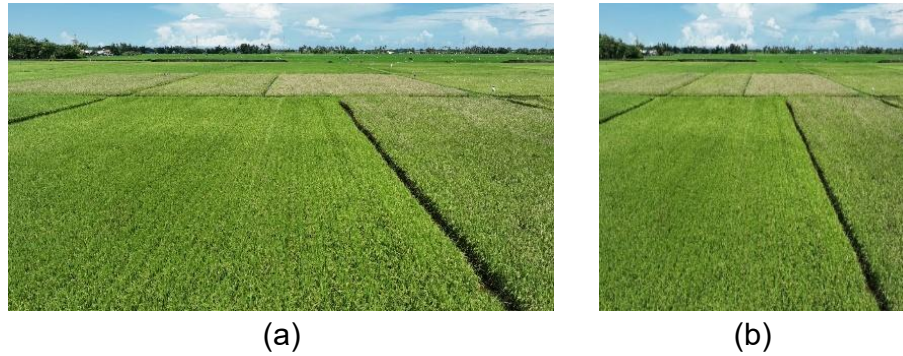
Gambar 16. Proses pelabelan data di Roboflow.

Setiap citra diberi label sesuai dengan area daun yang terinfeksi, menghasilkan mask biner yang menunjukkan batas antara daun sehat dan daun terinfeksi. Hasil pelabelan ini digunakan sebagai ground truth pada tahap pelatihan model segmentasi.

Setelah proses pelabelan selesai, dataset dibagi menjadi dua bagian dengan rasio 75:25, yaitu data latih (*training set*) sebanyak 333 citra, data validasi (*validation set*) sebanyak 96 citra. Adapun untuk data uji coba (*testing set*) sebanyak 240 citra. Pembagian ini bertujuan agar model dapat belajar dari sebagian besar data sekaligus diuji kemampuannya pada data yang belum pernah dilihat sebelumnya.

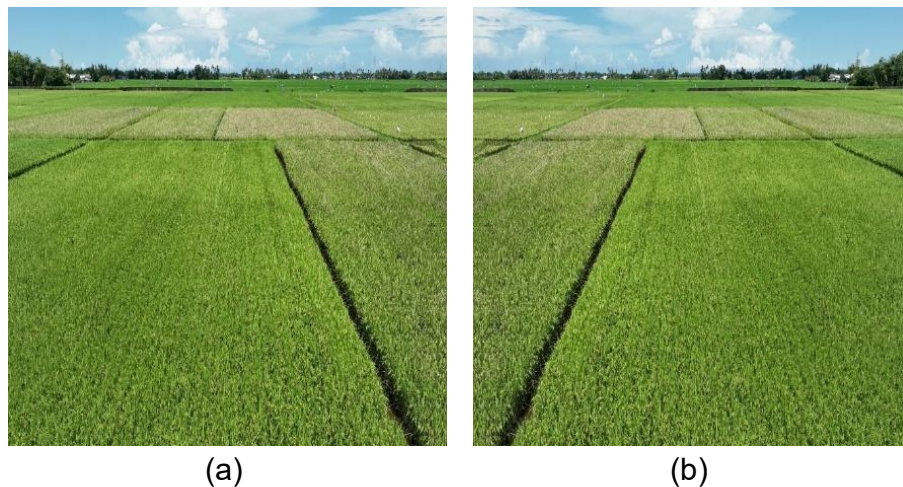
3. Preprocessing

Tahap pra-pemrosesan (*preprocessing*) dilakukan untuk memastikan keseragaman dimensi dan meningkatkan variasi data citra sebelum proses pelatihan. Terdapat dua langkah utama dalam tahap ini, yaitu *resize* dan augmentasi data.



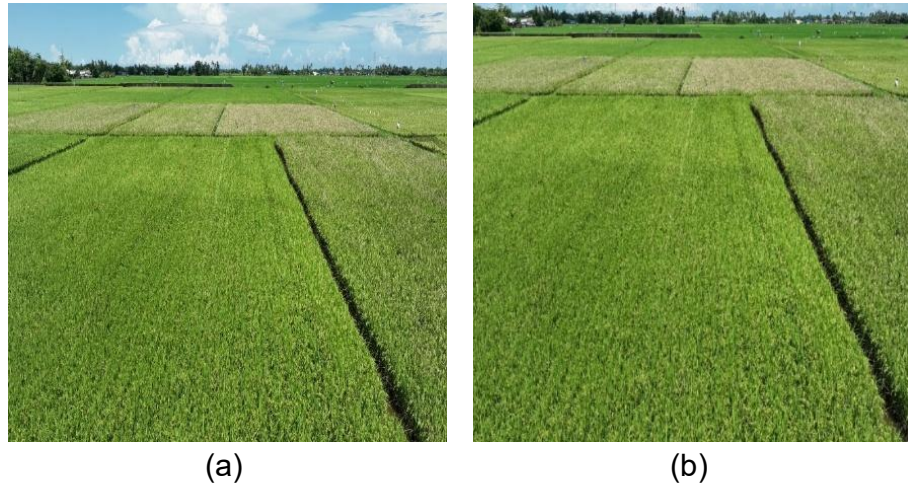
Gambar 17. Contoh *preprocessing resize*: (a) sebelum *resize*:
(b) setelah *resize*

Resize. Semua citra diubah ukurannya menjadi 640×640 piksel sesuai dengan ukuran masukan yang diterima oleh arsitektur YOLOv8n-seg. Proses *resize* bertujuan untuk menyeragamkan resolusi citra agar model dapat memproses setiap gambar dengan efisien tanpa kehilangan proporsi penting dari objek daun padi. Selain itu, penyeragaman ukuran input juga mempermudah optimalisasi model saat dikonversi ke TensorRT.



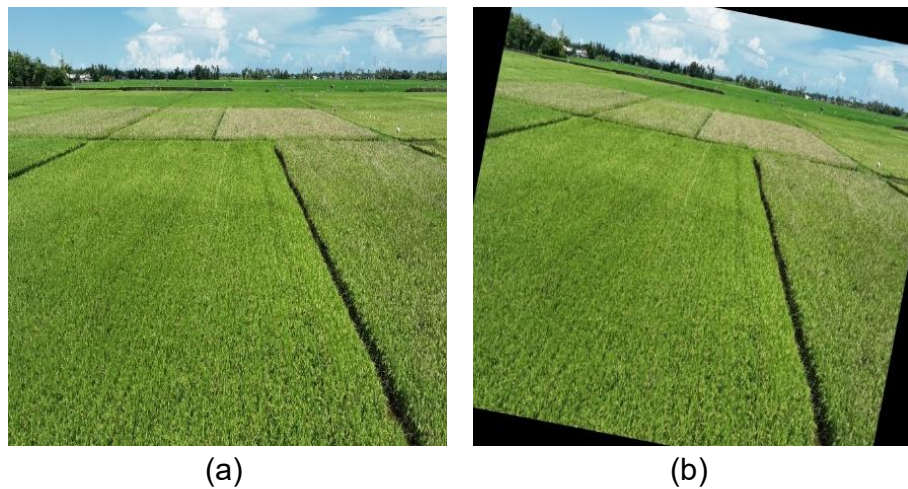
Gambar 18. Contoh hasil *horizontal flip*: (a) sebelum *horizontal flip*;
(b) setelah *horizontal flip*

Horizontal Flip. Teknik ini mencerminkan citra secara horizontal terhadap sumbu vertikal. Metode ini membantu model belajar mengenali gejala penyakit BLB dari arah yang berbeda, tanpa terpengaruh orientasi daun atau posisi kamera UAV.



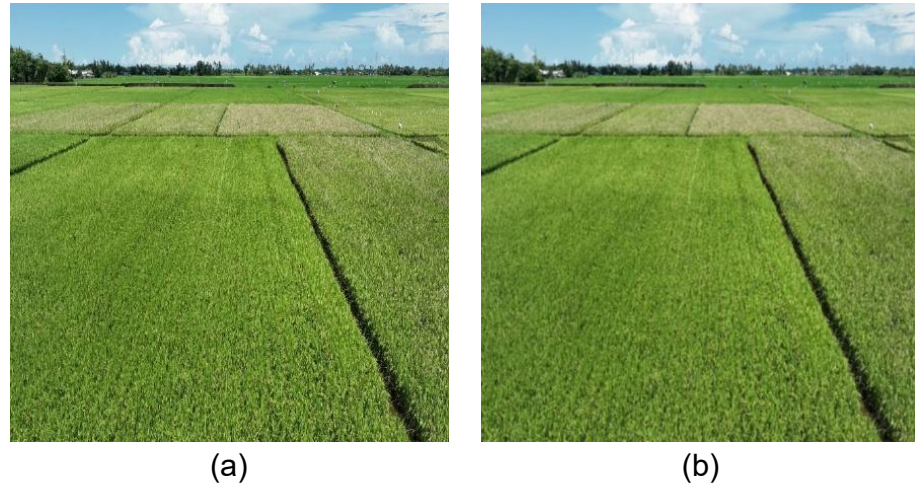
Gambar 19. Contoh hasil *crop*: (a) sebelum *crop*; (b) setelah *crop*

Crop. Augmentasi crop dilakukan dengan pengaturan 0% minimum zoom dan 30% maksimum *zoom*. Teknik ini meniru kondisi pengambilan citra pada jarak yang bervariasi oleh UAV, sehingga model menjadi lebih adaptif terhadap perubahan skala objek daun padi.



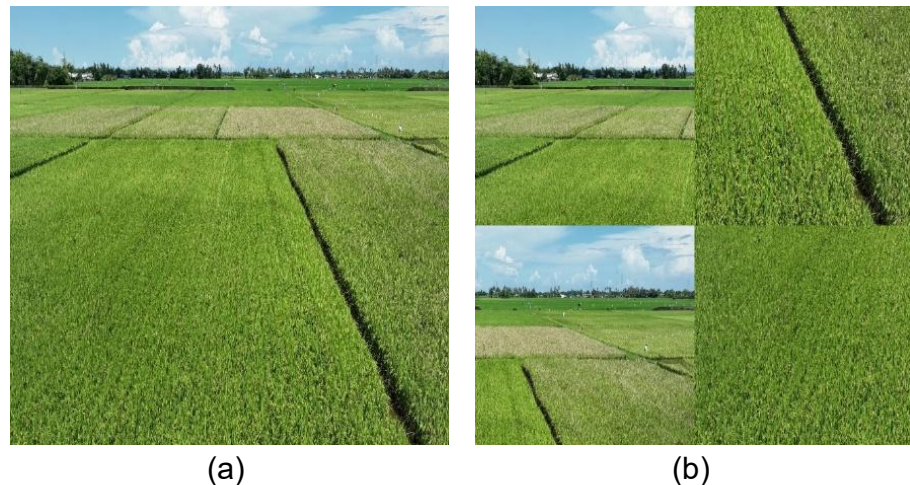
Gambar 20. Contoh *preprocessing rotation*: (a) sebelum *rotation*; (b) setelah *rotation*

Rotation. Setiap citra diputar secara acak antara -15° hingga $+15^\circ$ untuk meniru variasi sudut pandang yang terjadi saat UAV terbang. Variasi rotasi ini membantu model mengenali objek daun yang difoto dari berbagai arah dan kemiringan.



(a) (b)
Gambar 21. Contoh *preprocessing blur*: (a) sebelum *blur*;
(b) setelah *blur*

Blur. Efek blur dengan intensitas hingga 1 piksel diterapkan untuk meniru kondisi citra yang sedikit buram akibat getaran UAV atau pergerakan daun. Teknik ini memperkuat ketahanan model terhadap ketidaksempurnaan visual yang sering muncul pada pengambilan citra lapangan.



(a) (b)
Gambar 22. Contoh *preprocessing image tiling*: (a) sebelum *image tiling*; (b) setelah *image tiling*

Image Tiling. Selain *resize* dan augmentasi, penelitian ini juga menerapkan teknik *image tiling* sebagai bagian dari proses *preprocessing*. Image tiling adalah metode pemotongan citra beresolusi tinggi menjadi beberapa potongan (*tile*) berukuran lebih kecil dengan resolusi seragam, sehingga setiap tile dapat dianalisis secara independen oleh model.

4. Pembagian Data

Setelah dilakukan augmentasi data seperti *horizontal flip*, *crop*, *rotation*, *blur*, dan *image tiling* didapatkan data latih (training set) sebanyak 1002 citra, data validasi (*validation set*) sebanyak 96 citra, dan data uji coba (*testing set*) sebanyak 240 citra.

5. Melatih Model YOLOv8n-seg dan YOLOv11n-seg

Pada penelitian ini, dilakukan proses pelatihan terhadap dua varian model segmentasi, yaitu YOLOv8n-seg dan YOLOv11n-seg, untuk mendeteksi dan mensegmentasi area tanaman padi yang terdampak penyakit *Bacterial Leaf Blight* (BLB) pada citra aerial persawahan. Pemilihan kedua model ini didasarkan pada kebutuhan sistem untuk mendukung inferensi *real-time* pada perangkat *edge*, sekaligus mengevaluasi pengaruh peningkatan kompleksitas arsitektur terhadap akurasi segmentasi.

Dataset yang telah melalui tahap praproses dan pembagian data digunakan sebagai input pelatihan untuk kedua model. Data latih digunakan untuk memperbarui bobot model, sedangkan data validasi dimanfaatkan untuk memantau performa selama proses pelatihan dan mencegah terjadinya *overfitting*. Pelatihan dilakukan menggunakan bahasa pemrograman Python dengan framework PyTorch dan implementasi dari Ultralytics.

Model YOLOv8n-seg dilatih sebagai *baseline model* dengan fokus pada efisiensi komputasi dan kecepatan inferensi. Arsitektur YOLOv8n-seg menggunakan pendekatan *end-to-end instance segmentation* yang mengintegrasikan tugas deteksi objek dan pembentukan *segmentation mask* dalam satu kerangka kerja. Struktur *Backbone*, *Neck*, dan *Head* pada YOLOv8n-seg dirancang relatif ringan, sehingga memungkinkan proses pelatihan dan inferensi berjalan lebih cepat dengan konsumsi sumber daya yang lebih rendah. Pendekatan ini menjadikan YOLOv8n-seg sesuai untuk skenario pemantauan *real-time* yang sensitif terhadap *latency*.

Sementara itu, YOLOv11n-seg dilatih menggunakan dataset dan konfigurasi parameter yang sama untuk memastikan perbandingan performa yang adil dan terkontrol. YOLOv11n-seg merupakan pengembangan lanjutan dari generasi YOLO sebelumnya dengan penyempurnaan pada proses ekstraksi fitur dan penggabungan multi-skala. Arsitektur YOLOv11n-seg tetap mempertahankan pendekatan *end-to-end*, namun dengan *backbone* dan *neck* yang lebih kaya representasi spasial dan semantik. Penyempurnaan ini bertujuan meningkatkan kualitas segmentasi, khususnya pada batas area objek yang kompleks dan tidak tegas.

Konfigurasi parameter pelatihan untuk kedua model ditetapkan sama, sebagaimana ditunjukkan pada Tabel 3, dengan jumlah *epoch* sebanyak 1000, ukuran citra input 640×640, batch size 16, serta mekanisme *early stopping* dengan nilai *patience* 100. Penyeragaman konfigurasi ini bertujuan untuk memastikan bahwa perbedaan performa yang dihasilkan

pada tahap evaluasi terutama dipengaruhi oleh perbedaan arsitektur model, bukan oleh variasi parameter pelatihan. Pelatihan dilakukan menggunakan Python dan framework PyTorch, dengan konfigurasi parameter sebagai berikut:

Tabel 3. Konfigurasi parameter *training* model YOLO

Parameter	YOLO
<i>Epoch</i>	1000
<i>Image Size</i>	640x640
<i>Batch Size</i>	16
<i>Early Stopping (patience)</i>	100

Selama proses pelatihan, performa model dipantau menggunakan data validasi untuk mengamati konvergensi dan stabilitas pembelajaran. Pendekatan ini memungkinkan identifikasi perbedaan karakteristik pelatihan antara YOLOv8n-seg dan YOLOv11n-seg, yang selanjutnya tercermin pada hasil evaluasi akurasi segmentasi dan waktu inferensi. Dengan strategi pelatihan yang seragam, analisis perbandingan kedua model dapat dilakukan secara objektif, sehingga *trade-off* antara akurasi dan efisiensi komputasi dapat dievaluasi dengan lebih jelas.

6. Evaluasi Model

Setelah proses pelatihan selesai, dilakukan tahap evaluasi model untuk mengukur performa deteksi dan segmentasi area daun padi yang terinfeksi penyakit BLB. Evaluasi dilakukan menggunakan empat metrik utama, yaitu *Intersection over Union* (IoU), *Dice Score*, *Confusion Matrix*, dan *Frame Per Second* (FPS). Nilai hasil pengujian metrik akan dijelaskan pada BAB III, bagian Hasil dan Pembahasan.

7. Menyimpan *Best Model* (YOLOv8n-seg)

Setelah dilakukan evaluasi terhadap model yang telah dilatih sebelumnya, model dengan performa terbaik (*best model*) disimpan sebagai hasil akhir dari tahap pelatihan.

2.5.2 Tahap Validasi Model Segmentasi

1. Input Video Evaluasi

Pada tahap validasi, data yang digunakan merupakan data yang benar-benar baru dan tidak pernah disentuh pada proses *training*, sehingga dapat berfungsi untuk menguji kemampuan generalisasi model terhadap data *out-of-distribution*. Proses seleksi data validasi harus memperhatikan beberapa aspek seperti variasi citra, distribusi kelas, serta potensi kondisi ekstrem yang dapat menguji *robustness* model.

Dalam penelitian ini digunakan video hasil rekaman UAV digunakan sebagai masukan (input) untuk menguji kinerja model YOLOv8n-seg. Video

diambil menggunakan UAV pada ketinggian 5 meter dengan resolusi 1920×1080 piksel dan durasi rata-rata 20 detik per video.

Setiap video menampilkan area persawahan dengan kondisi terdapat penyakit BLB. Penggunaan video sebagai data validasi memungkinkan sistem diuji dalam kondisi dinamis, termasuk gerakan kamera (*camera panning*) dan variasi jarak pengambilan gambar.

2. Preprocessing

Setiap video yang dimasukkan ke dalam proses evaluasi, dilakukan preprocessing berupa *resizing* dari ukuran 1920×1080 piksel menjadi 640×640 piksel. Ini dilakukan karena menyesuaikan ukuran inputan dari model YOLOv8n-seg yaitu 640×640 piksel.

3. Segmentasi

Validasi model segmentasi dilakukan dengan penerapan model YOLO terbaik hasil *training* pada data validasi yang telah disediakan. Model secara langsung menghasilkan *mask* segmentasi pada tiap *frame* citra validasi, di mana area target yang terdeteksi ditandai melalui overlay mask pada citra asli. Proses ini menggambarkan kapasitas model dalam mengidentifikasi area sesuai kebutuhan segmentasi.

4. Export Model to TensorRT

Tabel 4. Konfigurasi parameter *export model* ke TensorRT

Parameter	Nilai untuk FP32 (Full Precision)	Nilai untuk FP16 (Half Precision)	Nilai untuk INT8 (Integer 8-bit)
format	“engine”	“engine”	“engine”
half	False	True	False
int8	False	False	True
data	/path/to/data.yaml	/path/to/data.yaml	/path/to/data.yaml
dynamic	True	True	True
batch	1	1	1
Workspace (gb)	4	4	4

Pada tahap ini, model YOLOv8 hasil pelatihan dikonversi ke dalam format TensorRT (.engine) untuk diimplementasikan pada perangkat *edge* NVIDIA Jetson Orin Nano. Proses konversi dilakukan menggunakan pustaka Ultralytics dengan fungsi ekspor bawaan YOLO. Tujuan utama dari konversi ini adalah memperoleh model yang dapat berjalan lebih cepat dan efisien saat melakukan inferensi, khususnya pada sistem dengan sumber daya terbatas seperti Jetson.

Model hasil pelatihan awal disimpan dalam format PyTorch (.pt), kemudian diekspor menjadi format TensorRT dengan pengaturan tertentu. Selama proses ekspor, model dikonversi dalam presisi FP32, FP16 (*half*

precision) dan INT 8 untuk mengoptimalkan performa inferensi sekaligus mengurangi kebutuhan memori GPU. Pengaturan ukuran *workspace* GPU sebesar 4 GB dan ukuran batch sebesar 1 digunakan untuk menyesuaikan dengan kapasitas perangkat serta kebutuhan inferensi *real-time*. Selain itu, parameter input dinamis diaktifkan agar model dapat menerima variasi ukuran citra selama pengujian. File konfigurasi dataset dilibatkan dalam proses ini untuk memastikan integritas label dan struktur data yang digunakan tetap konsisten dengan model selama pelatihan.

Hasil dari proses konversi menghasilkan file dengan ekstensi .engine yang telah teroptimasi oleh TensorRT, siap digunakan untuk proses inferensi pada Jetson Orin Nano. Dengan demikian, tahap ini berperan penting dalam menjembatani model hasil pelatihan pada lingkungan komputasi desktop dengan sistem penerapan berbasis edge yang menuntut efisiensi tinggi.

5. Output

Hasil segmentasi yang diperoleh selanjutnya divisualisasikan dalam bentuk video, dengan mask segmentasi yang di-overlay pada tiap frame citra validasi secara berurutan. Visualisasi ini difungsikan sebagai bentuk representasi konkret dari prediksi model, sehingga performa segmentasi dapat diamati dengan jelas pada data validasi yang digunakan.