

**PERBANDINGAN MODEL *CONVOLUTIONAL AUTOENCODER* DAN
VARIATIONAL AUTOENCODER UNTUK DETEKSI DINI POHON
KELAPA SAWIT YANG TERINFEKSI *GANODERMA* MENGGUNAKAN
CITRA PENGINDERAAN JAUH**

TUGAS AKHIR



UNIVERSITAS
MA CHUNG
NATANAEL MISSIONDAY GLORYANT
NIM. 312110010

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2025**

LEMBAR PENGESAHAN
TUGAS AKHIR

**PERBANDINGAN MODEL CONVOLUTIONAL AUTOENCODER DAN
VARIATIONAL AUTOENCODER UNTUK DETEKSI DINI POHON
KELAPA SAWIT YANG TERINFEKSI GANODERMA MENGGUNAKAN
CITRA PENGINDERAAN JAUH**

Oleh:

Natanael Missionday Gloryant

NIM. 312110010

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI dan DESAIN
UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer

Dosen Pembimbing I,


Dr. Kestrilia Rega Prilianti, M.Si.
NIP. 20120035

Dosen Pembimbing II,


Hendry Setiawan, S.T., M.Kom.
NIP. 20100006

Dekan Fakultas Teknologi dan Desain,



Prof. Dr. Eng. Romy Budhi Widodo
NIP. 20070035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Skripsi saya dengan “*Perbandingan Model Convolutional Autoencoder Dan Variational Autoencoder Untuk Deteksi Dini Pohon Kelapa Sawit Yang Terinfeksi Ganoderma Menggunakan Citra Penginderaan Jauh*” adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 26 September 2025



Natanael Missionday Gloryant

NIM. 312110010

UNIVERSITAS
MA CHUNG

**PERBANDINGAN MODEL *CONVOLUTIONAL AUTOENCODER* DAN
VARIATIONAL AUTOENCODER UNTUK DETEKSI DINI POHON
KELAPA SAWIT YANG TERINFEKSI *GANODERMA* MENGGUNAKAN
CITRA PENGINDERAAN JAUH**

Natanael Missionday Gloryant, Kestrilia Rega Prilianti, Hendry Setiawan
Universitas Ma Chung

Abstrak

Penyakit busuk pangkal batang akibat jamur *Ganoderma Boninense* merupakan ancaman signifikan bagi industri kelapa sawit di Indonesia, menyebabkan kerugian ekonomi substansial karena sulitnya untuk dideteksi. Gejala visualnya seringkali tidak terlihat hingga infeksi mencapai tahap lanjut, membuat metode inspeksi konvensional tidak efektif. Penelitian ini bertujuan mengatasi keterbatasan tersebut dengan memanfaatkan citra udara dari UAV yang dikombinasikan dengan metode *deep learning* untuk mendeteksi anomali.

Pada penelitian ini, digunakan model *Autoencoder*, ideal untuk deteksi anomali di mana data pohon sakit berlabel terbatas. Model dilatih pada citra pohon sehat dan mengidentifikasi anomali dari hasil *reconstruction error* yang tinggi. Secara spesifik, penelitian ini membandingkan arsitektur *Convolutional Autoencoder* (CAE) dan *Convolutional Variational Autoencoder* (CVAE), menganalisis performa keduanya terhadap dataset pohon sakit yang terakumulasi selama lima bulan.

Hasil penelitian menunjukkan bahwa CVAE terbukti lebih unggul dan konsisten dalam mendeteksi anomali. Analisis kuantitatif berdasarkan metrik *Bhattacharyya Distance* (BD) dan *Jensen-Shannon Divergence* (JSD) menunjukkan bahwa CVAE mampu menjaga nilai BD yang stabil (sekitar 0.4 hingga 0.6) dan nilai JSD yang rendah (di bawah 0.2) sepanjang periode lima bulan. Stabilitas ini memungkinkan CVAE mendeteksi anomali hingga lima bulan sebelum pohon tumbang. Di sisi lain, CAE menunjukkan performa yang tidak konsisten dengan fluktuasi nilai BD yang signifikan (misalnya, melonjak dari 0.419 pada bulan ketiga ke 0.740 pada bulan kelima), dan nilai JSD yang lebih tinggi. Performa CAE hanya efektif hingga dua bulan sebelum pohon tumbang. Dengan demikian, penelitian ini menyimpulkan bahwa CVAE adalah solusi optimal untuk deteksi dini Ganoderma karena konsistensinya yang terbukti lebih unggul.

Kata kunci: *Convolutional Autoencoder*, *Convolutional Variational Autoencoder*, Deteksi Dini, *Ganoderma Boninense*, Penginderaan Jauh

COMPARISON OF CONVOLUTIONAL AUTOENCODER AND VARIATIONAL AUTOENCODER MODELS FOR EARLY DETECTION OF GANODERMA-INFECTED OIL PALM TREES USING REMOTE SENSING IMAGERY

Natanael Missionday Gloryant, Kestrilia Rega Prilianti, Hendry Setiawan

Ma Chung University

Abstract

Basal stem rot disease caused by the Ganoderma Boninense fungus poses a significant threat to Indonesia's oil palm industry, leading to substantial economic losses due to the difficulty of early detection. Its visual symptoms are often not apparent until the infection reaches an advanced stage, making conventional inspection methods ineffective. This study aims to overcome this limitation by utilizing aerial imagery from UAVs combined with deep learning methods to detect anomalies.

This research employs an Autoencoder model, which is ideal for anomaly detection where labeled data for sick trees is limited. The model is trained on images of healthy trees and identifies anomalies from a high reconstruction error. Specifically, this study compares the architectures of the Convolutional Autoencoder (CAE) and the Conditional Variational Autoencoder (CVAE), analyzing the performance of both on a dataset of diseased trees accumulated over five months.

The research findings show that CVAE is proven to be superior and more consistent in detecting anomalies. Quantitative analysis based on the Bhattacharyya Distance (BD) and Jensen-Shannon Divergence (JSD) metrics shows that CVAE maintains stable BD values (around 0.4 to 0.6) and low JSD values (below 0.2) throughout the five-month period. This stability allows CVAE to detect anomalies up to five months before a tree collapses. In contrast, CAE shows inconsistent performance with significant fluctuations in BD values (for example, jumping from 0.419 in the third month to 0.740 in the fifth month), and higher JSD values. CAE's performance is only effective up to two months before a tree collapses. Therefore, this study concludes that CVAE is the optimal solution for early Ganoderma detection due to its superior consistency.

Keywords: Convolutional Autoencoder, Convolutional Variational Autoencoder, Early Detection, Ganoderma Boninense, Remote Sensing

KATA PENGANTAR

Puji syukur penulisan panjatkan kehadirat Tuhan Yang Maha Esa, atas anugerah dan penyertaan-Nya, sehingga penulis bisa mengerjakan dan menyelesaikan Tugas Akhir ini dengan lancar. Tak lupa juga pada kesempatan ini, penulis mengucapkan terima kasih kepada pihak-pihak yang telah memberikan dukungan dan bantuan dalam proses penggerjaan dan penyelesaian Tugas Akhir ini:

1. Dr. Kestrilia Rega Prilianti, M.Si., selaku Dosen Pembimbing Akademik sekaligus Pembimbing Tugas Akhir, atas segala bimbingan, arahan, serta motivasi yang sangat berarti selama proses penelitian dan Tugas Akhir ini.
2. Bapak Hendry Setiawan, S.T., M.Kom., selaku Dosen Pembimbing Tugas Akhir dan Kepala Program Studi Teknik Informatika Universitas Ma Chung, atas segala bantuan, dukungan, dan masukan konstruktif yang telah banyak membantu dalam Tugas Akhir ini.
3. Orang tua terkasih yang telah mendukung dan mendoakan penulis dalam menyelesaikan penelitian dan Tugas Akhir dengan baik.
4. Satu nama tak tertulis yang senantiasa menemani dan mendoakan dalam proses penelitian dan Tugas Akhir ini.
5. Serta rekan-rekan dan seluruh pihak lain yang tidak dapat penulis sebutkan satu per satu, namun telah memberikan kontribusi dan dukungan dalam bentuk apapun selama proses penelitian dan Tugas Akhir ini.

Penulis menyadari bahwa penelitian ini masih jauh dari kata sempurna, namun penulis berharap penelitian ini dapat memberikan kontribusi yang bermanfaat bagi pembaca, khususnya bagi mereka yang tengah menjalani studi serupa.

Malang, 26 September 2025



Nataael Missionday Gloryant
NIM. 312110010

DAFTAR ISI

JUDUL	i
LEMBAR PENGESAHAN	1
PERNYATAAN KEASLIAN SKRIPSI	2
Abstrak	3
KATA PENGANTAR	5
DAFTAR ISI	6
DAFTAR SINGKATAN	9
DAFTAR GAMBAR	10
DAFTAR TABEL	14
BAB I PENDAHULUAN	15
1.1 Latar Belakang Masalah	15
1.2 Identifikasi Masalah	17
1.3 Batasan Masalah	17
1.4 Perumusan Masalah	18
1.5 Tujuan Penelitian	18
1.6 Luaran	18
1.7 Manfaat Penelitian	18
1.8 Sistematika Penulisan	19
BAB II TINJAUAN PUSTAKA	21
2.1 Industri Kelapa Sawit	21
2.1.1 Penyakit Busuk Pangkal Batang	22
2.1.2 Deteksi Dini <i>Ganoderma</i>	23
2.2 Penginderaan Jauh	24
2.3 Infrengsi citra pengindraan jauh dengan AI	25
2.4 Implementasi <i>Maching Learning</i> (ML)	26

2.5	Implementasi <i>Deep Learning</i> (DL)	26
2.6	Citra Spasial	31
2.7	Pra-pemrosesan Citra Spasial (UAV)	31
2.8	Ruang warna RGB	32
2.9	Python	33
2.9.1	NumPy	34
2.9.2	Matplotlib	34
2.10	QGIS	35
2.11	Roboflow	35
2.12	Google Colaboratory	36
2.13	TensorFlow	37
2.14	Ultralytics	37
2.15	Penelitian terdahulu	38
BAB III ANALISIS DAN PERANCANGAN SISTEM		40
3.1	Alur Penelitian	40
3.2	Identifikasi Masalah	40
3.3	Analisis Kebutuhan	41
3.3.1	Kebutuhan Pengguna	41
3.3.2	Kebutuhan Peneliti	41
3.4	Studi Pustaka	42
3.5	Desain Penelitian	44
3.5.1	Penyusunan <i>Dataset</i>	46
3.5.2	Desain Model	51
3.5.3	Evaluasi Kinerja	57
BAB IV HASIL DAN PEMBAHASAN		60
4.1	<i>Dataset</i>	60

4.2 Pelatihan Model	60
4.2.1 Pelatihan Model YOLO	61
4.2.2 Pelatihan Model <i>Convolutional Autoencoder</i> (CAE)	64
4.2.3 Pelatihan Model <i>Convolutional Variational Autoencoder</i> (CVAE)	68
4.3 Evaluasi Model	74
4.3.1 Evaluasi Model YOLO	74
4.3.2 Evaluasi Model <i>Convolutional Autoencoder</i> (CAE)	75
4.3.3 Evaluasi Model <i>Convolutional Variational Autoencoder</i> (CVAE)	80
4.4 Perbandingan Hasil Rekonstruksi Citra	83
4.5 Perbandingan Kinerja CAE dan CVAE	84
4.5.1 Satu bulan sebelum tumbang	85
4.5.2 Dua bulan sebelum tumbang	86
4.5.3 Tiga bulan sebelum tumbang	88
4.5.4 Empat bulan sebelum tumbang	89
4.5.5 Lima bulan sebelum tumbang	91
4.6 Ringkasan Perbandingan Hasil	93
BAB V KESIMPULAN DAN SARAN	95
5.1. Kesimpulan	95
5.2. Saran	95
DAFTAR PUSTAKA	97

DAFTAR SINGKATAN

AE	<i>Autoencoder</i>
AI	<i>Artificial Intelligence</i>
CAE	<i>Convolutional Autoencoder</i>
CNN	<i>Convolutional Neural Network</i>
CVAE	<i>Convolutional Variational Autoencoder</i>
DL	<i>Deep Learning</i>
E	<i>Epochs</i>
LR	<i>Learning Rate</i>
MAE	<i>Mean Absolute Error</i>
ML	<i>Machine Learning</i>
MSE	<i>Mean Squared Error</i>
ReLU	<i>Rectified Linear Unit</i>
ROI	<i>Region of Interest</i>
SSIM	<i>Structural Similarity Index Measure</i>
UAV	<i>Unmanned Aerial Vehicle</i>
YOLO	<i>You Only Look Once</i>

UNIVERSITAS
MA CHUNG

DAFTAR GAMBAR

Gambar 2.1 Tanaman Kelapa Sawit	21
Gambar 2.2 Tanaman kelapa sawit terinfeksi Ganoderma	23
Gambar 2.3 Arsitektur <i>Autoencoder</i>	27
Gambar 2.4 Arsitektur <i>Convolutional AutoEncoder</i>	28
Gambar 2.5 Ruang warna RGB	33
Gambar 3.1 Alur penelitian	40
Gambar 3.2 Alur desain penelitian	45
Gambar 3.3 Alur Penyusunan <i>Dataset</i>	46
Gambar 3.4 Kebun kelapa sawit milik PTPN	47
Gambar 3.5 Penentuan ROI	48
Gambar 3.6 Pembagian citra ROI	49
Gambar 3.7 Anotasi pohon kelapa sawit sehat tanpa label	49
Gambar 3.8 Anotasi pohon kelapa sawit dengan label sakit dan sehat	50
Gambar 3.9 Contoh citra <i>timeseries</i> pohon sawit	51
Gambar 3.10 Pemilihan Dataset dan Pelatihan Model di Ultralytics	52
Gambar 3.11 CAE satu layer (a), dua layer (b), dan tiga layer (c)	53
Gambar 3.12 <i>Sampling Layer</i>	54
Gambar 3.13 <i>KL Divergence Layer</i>	54
Gambar 3.14 CVAE satu sampai tiga <i>layer</i>	55
Gambar 4.1 Grafik performa model <i>pretrain</i> pada 100 <i>epochs</i>	61
Gambar 4.2 Grafik performa model tanpa <i>pretrain</i> pada 100 <i>epochs</i>	62
Gambar 4.3 Grafik performa model tanpa <i>pretrain</i> pada 1000 <i>epochs</i>	63
Gambar 4.4 Grafik performa model <i>pretrain</i> pada 1000 <i>epochs</i>	63
Gambar 4.5 Grafik performa model CAE satu <i>layer</i> dengan konfigurasi (a) 100 <i>Epoch</i> dan <i>Learning Rate</i> 0.0005, (b) 100 <i>Epoch</i> dan <i>Learning Rate</i> 0.001,(c) 100 <i>Epoch</i> dan <i>Learning Rate</i> 0.01, (d) 150 <i>Epoch</i> dan <i>Learning Rate</i> 0.0005, (e) 150 <i>Epoch</i> dan <i>Learning Rate</i> 0.001, (f) 150 <i>Epoch</i> dan <i>Learning Rate</i> 0.01, (g) 50 <i>Epoch</i> dan <i>Learning Rate</i> 0.0005, (h) 50 <i>Epoch</i> dan <i>Learning Rate</i> 0.001, (i) 50 <i>Epoch</i> dan <i>Learning Rate</i> 0.01.	65

Gambar 4.6 Grafik performa model CAE dua *layer* dengan konfigurasi
(a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

67

Gambar 4.7 Grafik performa model CAE tiga *layer* dengan konfigurasi
(a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

69

Gambar 4.8 Grafik performa model CVAE satu *layer* dengan konfigurasi
(a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

70

Gambar 4.9 Grafik performa model CVAE dua *layer* dengan konfigurasi
(a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

71

Gambar 4.10 Grafik performa model CVAE tiga *layer* dengan konfigurasi
(a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150

<i>Epoch</i> dan <i>Learning Rate</i> 0.01, (g) 50 <i>Epoch</i> dan <i>Learning Rate</i> 0.0005, (h) 50 <i>Epoch</i> dan <i>Learning Rate</i> 0.001, (i) 50 <i>Epoch</i> dan <i>Learning Rate</i> 0.01.	73
Gambar 4.11 Gambar data pengujian YOLO (10%)	74
Gambar 4.12 Grafik Perbandingan Nilai MAE, MSE, dan SSIM CAE	79
Gambar 4.13 Grafik perbandingan Nilai MAE, MSE, dan SSIM CVAE	83
Gambar 4. 14 Citra Sehat (a), Citra Rekonstruksi CAE (b) dan Citra Rekonstruksi CVAE (c)	84
Gambar 4. 15 Citra Sehat (a), Citra Rekonstruksi CAE (b) dan Citra Rekonstruksi CVAE (c)	84
Gambar 4.16 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE satu bulan sebelum tumbang	85
Gambar 4.17 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE satu bulan sebelum tumbang	85
Gambar 4. 18 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE	86
Gambar 4.19 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE dua bulan sebelum tumbang	87
Gambar 4.20 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE dua bulan sebelum tumbang	87
Gambar 4.21 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE	87
Gambar 4.22 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE tiga bulan sebelum tumbang	88
Gambar 4.23 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE tiga bulan sebelum tumbang	88
Gambar 4. 24 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE	89
Gambar 4.25 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE empat bulan sebelum tumbang	90
Gambar 4.26 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE empat bulan sebelum tumbang	90

Gambar 4. 27 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE	91
Gambar 4.28 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE lima bulan sebelum tumbang	91
Gambar 4.29 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE lima bulan sebelum tumbang	92
Gambar 4. 30 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE	92
Gambar 4. 31 Grafik progresif nilai JSD dan BD metrik MAE, MSE, dan SSIM pada CAE untuk diskriminasi sawit sehat dan sakit.	93
Gambar 4. 32 Grafik progresif nilai JSD dan BD metrik MAE, MSE, dan SSIM pada CVAE untuk diskriminasi sawit sehat dan sakit.	94



UNIVERSITAS
MA CHUNG

DAFTAR TABEL

Tabel 3. Ringkasan hasil penelitian terdahulu	42
Tabel 3.2 Lanjutan	43
Table 4.1 Rincian <i>dataset</i>	60
Table 4.2 Hasil pelatihan model YOLO pada 100 <i>epochs</i>	62
Tabel 4.3 Perbandingan hasil pelatihan model YOLO pada 100 dan 1000 <i>epochs</i>	64
Table 4.4 Hasil pengujian model YOLO pada 100 <i>epochcs</i>	75
Table 4.5 Perbandingan hasil pengujian model YOLO pada 100 dan 1000 <i>epochs</i>	75
Table 4.6 Hasil metrik CAE satu <i>layer</i>	76
Table 4.7 Hasil metrik CAE dua <i>layer</i>	77
Table 4.8 Hasil metrik CAE tiga <i>layer</i>	78
Table 4.9 Hasil metrik CVAE satu <i>layer</i>	80
Table 4.10 Hasil metrik CVAE dua <i>layer</i>	81
Table 4.11 Hasil metrik CVAE tiga <i>layer</i>	82

**UNIVERSITAS
MA CHUNG**

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Industri kelapa sawit berperan signifikan dalam perekonomian Indonesia, berkontribusi besar terhadap pendapatan nasional dan penyediaan lapangan kerja bagi jutaan penduduk. Sebagai penghasil devisa utama, sektor ini menjadi tulang punggung ekonomi nasional, terutama dalam ekspor minyak sawit mentah dan turunannya (PSI Perkebunan, 2023). Luasnya lahan perkebunan kelapa sawit di Indonesia juga mencerminkan potensi besar dalam meningkatkan kesejahteraan masyarakat dan pembangunan daerah. Namun, tantangan teknis dan lingkungan yang kompleks menuntut perhatian serius untuk menjamin keberlanjutan sektor ini (PSI Perkebunan, 2023).

Penyakit busuk pangkal batang (BPB) yang disebabkan oleh jamur *Ganoderma boninense* merupakan ancaman serius bagi perkebunan kelapa sawit di Indonesia (Santoso, 2020). *Ganoderma* dapat menyebabkan penurunan produktivitas hingga 50% dan berpotensi menimbulkan kerugian ekonomi yang signifikan. *Ganoderma* menyerang tanaman kelapa sawit pada semua umur, menyebabkan pembusukan pada pangkal batang dan akhirnya kematian tanaman (Rahmana *et al.*, 2024). Penyebaran *Ganoderma* yang cepat dan luas ini menuntut adanya upaya pengendalian yang efektif dan efisien.

Deteksi dini pohon kelapa sawit yang terinfeksi *Ganoderma* sangatlah penting. Langkah ini tidak hanya penting untuk mengurangi kerugian ekonomi signifikan akibat pohon yang harus ditebang, tetapi juga krusial untuk menjaga keberlanjutan produksi kelapa sawit di lahan perkebunan yang terbatas (Kurihara dkk., 2020). Namun, tantangannya adalah gejala awal infeksi seringkali tidak jelas terlihat, menyulitkan petani untuk mendekripsi penyakit secara visual hingga mencapai tahap akhir. Secara konvensional, identifikasi kelapa sawit yang terinfeksi *Ganoderma* dapat dilakukan dengan mengamati gejala fisik seperti perubahan warna daun, daun tombak yang tidak terbuka, serta pembusukan pada pangkal batang. Namun, penelitian ini akan berfokus secara spesifik pada gejala

perubahan warna daun. Penulis berasumsi bahwa infeksi *Ganoderma* mengganggu sistem transportasi air dan nutrisi tanaman ke bagian atas pohon, yang pada akhirnya menyebabkan daun menguning. Oleh karena itu, dalam penelitian ini, daun kuning akan diinterpretasikan sebagai indikator adanya infeksi *Ganoderma*.

Untuk mengatasi keterbatasan deteksi visual, teknologi penginderaan jauh, seperti penggunaan *Unmanned Aerial Vehicle* (UAV) atau drone misalnya, telah terbukti efektif dalam memantau kesehatan tanaman (Santoso, 2020). Studi oleh Wicaksono dkk. (2022) menunjukkan bahwa citra UAV yang diproses menggunakan algoritma *Convolutional Neural Network* (CNN) dapat mendeteksi pokok pohon kelapa sawit dengan akurasi yang baik. Selanjutnya, mereka berhasil menggunakan model *Artificial Neural Network* (ANN) untuk memprediksi status infeksi *Ganoderma* pada tiap pokok pohon dengan akurasi mencapai 94% untuk data uji. Meskipun akurasi belum mencapai 100%, metode ini sudah dianggap cukup memadai bagi pengelola perkebunan untuk mengambil tindakan pencegahan dini terhadap serangan *Ganoderma*.

Meskipun CNN telah menunjukkan performa yang sangat baik dalam tugas deteksi dan klasifikasi gambar, seperti yang terlihat pada penelitian sebelumnya, salah satu kelemahan utama CNN adalah kebutuhan akan *dataset* pelatihan yang sangat besar untuk mencapai akurasi yang tinggi. Mengingat tantangan dalam mengumpulkan data anomali dalam jumlah besar yang bervariasi di lapangan, pendekatan alternatif yang lebih efisien data sangat dibutuhkan. Oleh karena itu, dalam penelitian ini, penulis akan menganalisis penggunaan *Autoencoder* (AE) untuk mengidentifikasi *Ganoderma* pada kelapa sawit. AE menawarkan keunggulan signifikan karena dapat dilatih menggunakan data tanpa label, menjadikannya ideal saat data berlabel terbatas atau tidak tersedia. Dengan demikian, AE mampu menemukan pola tersembunyi dalam data dengan merekonstruksi *input*, sekaligus mengidentifikasi anomali melalui *error* rekonstruksi yang tinggi pada data yang menyimpang dari pola normal (S.P., 2023).

Secara spesifik, penulis akan mengeksplorasi dan membandingkan dua variasi arsitektur AE, yaitu *Convolutional Autoencoder* (CAE) dan *Convolutional Variational Autoencoder* (CVAE). CAE adalah AE yang menggunakan lapisan

konvolusi untuk efektif mempelajari representasi data spasial, menjadikannya pilihan kuat dalam deteksi anomali berbasis citra (Jana *et al.*, 2022). Di sisi lain, VAE memperkenalkan pendekatan probabilistik ke ruang latennya, memungkinkan model ini menghasilkan sampel data baru dari distribusi yang dipelajari serta lebih baik dalam menangkap variasi anomali (Neloy dan Turgeon, 2024). *Conditional Variational Autoencoder* (CVAE) sendiri merupakan variasi dari VAE yang menggabungkan VAE dengan lapisan konvolusi, membuatnya efektif untuk data gambar.

1.2 Identifikasi Masalah

Berdasarkan latar belakang, masalah utama yang akan dibahas adalah potensi penyebaran *Ganoderma* yang dapat menyebabkan kerugian signifikan bagi industri kelapa sawit jika tidak ditangani secara tepat. Infeksi yang tidak terdeteksi sejak dini akan menyebar ke pohon sehat lainnya, yang pada akhirnya menyebabkan menurunnya produktivitas hingga tumbangnya pohon dan menimbulkan kerugian ekonomi besar.

1.3 Batasan Masalah

Adapun batasan masalah penelitian berikut adalah:

1. Citra input diperoleh dari PT. Perkebunan Nusantara (PTPN) menggunakan UAV atau drone yang terbang pada ketinggian 500 meter.
2. Data citra yang digunakan terbatas pada periode Januari hingga Juni 2023.
3. Format warna citra input adalah RGB.
4. Penelitian ini akan berfokus pada penggunaan algoritma YOLO untuk deteksi pohon kelapa sawit, dan mengeksplorasi serta membandingkan arsitektur CAE dan CVAE untuk deteksi anomali pada pohon kelapa sawit yang sakit, guna menentukan model yang paling optimal.
5. Pengambilan data deteksi dini terbatas pada periode Juni 2023 mundur hingga Januari 2023, yaitu data yang dikumpulkan selama enam bulan sebelum dan selama masa pemantauan.

1.4 Perumusan Masalah

Berdasarkan paparan latar belakang, identifikasi permasalahan, dan batasan masalah yang telah dijelaskan sebelumnya, disusunlah rumusan masalah yang menjadi inti pembahasan dalam laporan tugas akhir ini adalah sebagai berikut:

1. Apakah deteksi dini Ganoderma pada pohon sawit melalui perubahan warna daun dapat dilakukan menggunakan model *deep learning*?
2. Seberapa jauh ke belakang (dalam hitungan bulan) model *deep learning* mampu mendeteksi anomali *Ganoderma* sebelum pohon kelapa sawit akhirnya tumbang?

1.5 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah sebagai berikut:

1. Mengembangkan model *deep learning* CAE dan CVAE yang mampu mendeteksi dini anomali infeksi *Ganoderma* pada pohon sawit.
2. Menganalisis sejauh mana batas kemampuan deteksi model CAE atau CVAE dalam mengidentifikasi anomali akibat *Ganoderma* berdasarkan urutan waktu (*timeseries*).

1.6 Luaran

Hasil yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Model deteksi dini berbasis CAE dan CVAE yang dapat mengenali anomali *Ganoderma* pada pohon sawit
2. Publikasi artikel dijurnal ilmiah berindeks SINTA.

1.7 Manfaat Penelitian

a) Bagi Penulis:

1. Kesempatan untuk belajar dan memperluas pengetahuan, pemahaman dan menajamkan kemampuan dalam bidang kecerdasan

buatan, *remote sensing*, algoritma CAE dan CVAE serta penerapannya.

b) Bagi PT. Perkebunan Nusantara, dan Petani Sawit:

1. Hasil dari penelitian ini dapat digunakan untuk mendeteksi dini *Ganoderma* pada pohon sawit agar petani sawit dan perusahaan industri kelapa sawit dapat menghindari kerugian besar.

c) Bagi Universitas dan Kelompok Riset:

1. Memperoleh acuan untuk penelitian selanjutnya untuk membuat program yang mampu mendeteksi dini *Ganoderma* pada pohon sawit.
2. Membantu menghasilkan lulusan yang kompeten melalui penelitian yang terpercaya.

1.8 Sistematika Penulisan

Penulisan laporan tugas akhir ini ditulis dengan sistematika penulisan sebagai berikut.

1. Bab I, Pendahuluan

Membahas latar belakang penelitian mengenai pentingnya deteksi dini *Ganoderma* pada pohon kelapa sawit, dampak ekonomi yang ditimbulkan, serta bagaimana penginderaan jauh dapat digunakan sebagai solusi. Selain itu, bab ini juga mencakup rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. Bab II, Tinjauan Pustaka

Berisi landasan teori yang mendukung penelitian, termasuk kajian tentang *Ganoderma* pada kelapa sawit, konsep penginderaan jauh dalam bidang pertanian, serta teori mengenai algoritma AE dan variasinya. Selain itu, penelitian terdahulu yang relevan dengan topik ini juga dibahas untuk memberikan gambaran tentang perkembangan studi terkait.

3. Bab III, Analisis dan Perancangan Sistem

Berisi penjelasan mulai dari pengumpulan data citra penginderaan jauh, proses preprocessing data, perancangan model CAE dan CVAE, serta

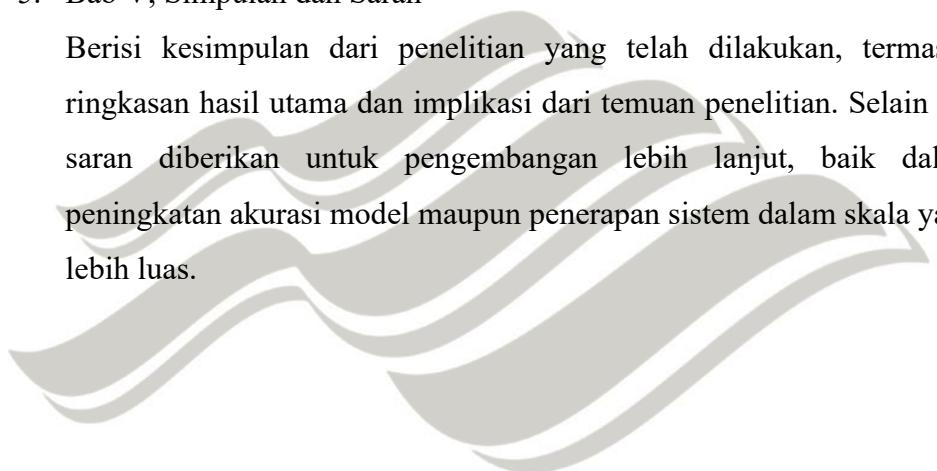
metode evaluasi yang digunakan untuk menilai kinerja model dalam mendeteksi pohon kelapa sawit yang terinfeksi *Ganoderma*.

4. Bab IV, Hasil dan Pembahasan

Membahas hasil penelitian yang diperoleh setelah pengujian model dilakukan. Hasil yang ditampilkan meliputi performa model CAE dan CVAE dalam mendeteksi pohon sawit yang terinfeksi, serta analisis terhadap efektivitas metode yang digunakan. Evaluasi dilakukan dengan membandingkan hasil model terhadap data yang telah dikumpulkan.

5. Bab V, Simpulan dan Saran

Berisi kesimpulan dari penelitian yang telah dilakukan, termasuk ringkasan hasil utama dan implikasi dari temuan penelitian. Selain itu, saran diberikan untuk pengembangan lebih lanjut, baik dalam peningkatan akurasi model maupun penerapan sistem dalam skala yang lebih luas.



UNIVERSITAS
MA CHUNG

BAB II

TINJAUAN PUSTAKA

2.1 Industri Kelapa Sawit

Kelapa sawit (Gambar 2.1) merupakan tanaman tropis penghasil minyak sawit mentah (CPO) dan minyak inti sawit (PKO) yang digunakan secara luas dalam berbagai industry, termasuk pangan (minyak goreng, margarin), kosmetik, hingga bahan bakar seperti biodiesel. Dengan produktivitasnya yang tinggi, menjadikan kelapa sawit sebagai sumber minyak nabati paling efisien dibanding tanaman penghasil minyak lainnya.



Gambar 2.1 Tanaman Kelapa Sawit

(Sumber: <https://www.sawitsetara.co/yuk-pahami-ciri-ciri-pohon-kelapa-sawit-yang-sehat/>)

Di Indonesia, kelapa sawit menjadi salah satu komoditas perkebunan yang memiliki peran yang signifikan dalam perekonomian nasional. Sebagai pemasok minyak kelapa sawit terbesar di dunia sejak tahun 2007, Indonesia telah menguasai 58% pangsa pasar global pada tahun 2020 (Xin *et al.*, 2022) dan pada tahun 2023, volume ekspor kelapa sawit telah mencapai 27,537,847 Ton dan nilai ekspor mencapai 24,008 ribu US\$ (Kementerian Pertanian Republik Indonesia, 2025). Dengan kontribusi besarnya terhadap ekspor dan perekonomian nasional, menjadi

hal yang sangat penting untuk menjaga keberlanjutan dan produktivitas kelapa sawit, sehingga baik petani maupun perusahaan perkebunan kelapa sawit harus mengambil langkah strategis guna tidak hanya menjaga, tetapi juga meningkatkan produktivitas tanaman yang mereka kelola agar hasilnya optimal dan efisien.

Salah satu tantangan utama dalam perkebunan kelapa sawit adalah serangan penyakit tanaman, khususnya penyakit busuk pangkal batang yang pada umumnya disebabkan oleh jamur *Ganoderma boninense*. *Ganoderma* umumnya menyerang tanaman kelapa sawit yang berumur lebih dari 10 tahun, terutama pada kebun yang tidak menerapkan sanitasi lahan yang baik sehingga dapat mengakibatkan penurunan produktivitas hingga kerugian besar. Upaya penanggulangan *Ganoderma* sendiri masih menjadi tantangan besar, karena belum ditemukannya metode yang efektif untuk mengatasi jamur dari tanah. Metode yang saat ini digunakan bersifat pencegahan dan pengendalian dini seperti penggunaan tanaman penutup tanah (*cover crops*), isolasi tanaman sakit, biofungisida, serta pemantauan visual dan laboratorium secara berkala.

2.1.1 Penyakit Busuk Pangkal Batang

Penyakit busuk pangkal batang merupakan salah satu penyakit pada tanaman kelapa sawit yang pada umumnya disebabkan oleh jamur *Ganoderma Boninense* (Gambar 2.2). Jamur ini menginfeksi jaringan akar lalu menyebar ke dalam batang bagian bawah tanaman. Infeksi dapat terjadi melalui kontak akar antar tanaman atau spora yang terbawa angin maupun air. Kemudian, jamur tumbuh secara perlahan tetapi progresif, menyebabkan jaringan dalam kayu mengalami pelapukan dan pembusukan. Kerusakan ini utamanya terjadi pada jaringan xilem di dalam batang, yang berfungsi sebagai pembuluh angkut utama untuk air dan nutrisi dari akar ke seluruh bagian tanaman. Ketika xilem lapuk dan membusuk, saluran transportasi ini menjadi tersumbat dan rusak, secara efektif mengganggu aliran vital air dan mineral. Akibatnya, tanaman tidak dapat memperoleh sumber daya yang cukup untuk fotosintesis dan pertumbuhan, yang pada akhirnya menyebabkan gejala fisik seperti daun menguning kemudian layu, penurunan produksi tandan

bahan segar, hingga kematian tanaman (Baharim *et al.*, 2023, cit. Daranagama & Takeuchi, 2025).



Gambar 2.2 Tanaman kelapa sawit terinfeksi Ganoderma
(Sumber: <https://gdm.id/ganoderma/>)

Menurut laporan *Palm Oil Agribusiness Strategic Policy Institute* (PASPI), tergantung tingkat keparahan infeksi, serangan *Ganoderma* dapat menurunkan produktivitas Tandan Buah Segar (TBS) hingga 50% (Sipayung, 2024). Disamping itu, masa produktif tanaman kelapa sawit juga berpotensi menurun hingga setengah dari usia ekonomis yang seharusnya (Sipayung, 2024). Dalam beberapa kasus, tanaman dapat roboh tanpa menunjukkan gejala apa pun sebelumnya (Ginting & Wiratmoko, 2021). Jika *Ganoderma* tidak segera ditangani, terutama pada perkebunan tua maka akan terjadi penurunan produktivitas. Dalam kondisi yang parah, *Ganoderma* bahkan dapat menyebabkan kematian tanaman dan menimbulkan kerugian besar bagi petani maupun pelaku industri kelapa sawit.

2.1.2 Deteksi Dini *Ganoderma*

Penyakit busuk pangkal batang yang disebabkan oleh jamur *Ganoderma boninense* merupakan salah satu ancaman serius terhadap produktivitas kelapa sawit di Indonesia. Infeksi jamur ini dapat secara signifikan menyebabkan penurunan hasil panen, bahkan berujung pada kematian tanaman, terutama pada tanaman yang berumur tua (Soetopo dkk., 2022). Kerugian yang ditimbulkan tidak hanya berdampak pada penurunan hasil produksi, tetapi juga memberikan tekanan

ekonomi yang signifikan bagi petani dan pelaku industri kelapa sawit (Azmi *et al.*, 2020). Oleh karena itu, deteksi dini terhadap infeksi *Ganoderma* menjadi langkah krusial untuk mencegah penyebaran dan memfasilitasi tindakan pengendalian yang lebih efektif sebelum penyakit meluas ke tanaman lainnya.

Sebagai respon terhadap urgensi tersebut, berbagai pendekatan deteksi dini telah dirancang guna mengidentifikasi infeksi *Ganoderma* dengan Tingkat akurasi dan efisiensi yang tinggi. Salah satu inovasi yang telah dikembangkan adalah *eNose-g*, yang merupakan teknologi sensor elektronik portabel yang membedakan tanaman sehat dan terinfeksi berdasarkan profil aroma spesifik, memberikan hasil deteksi cepat dan akurat (Badan Pengelola Dana Perkebunan Kelapa Sawit, 2021). Selain itu, metode berbasis fisika seperti Radal SIL memanfaatkan perubahan sifat dielektrik pada batang pohon untuk mendeteksi infeksi secara non-destructif dan cepat (Liaghat *et al.*, 2014). Selain itu, seiring dengan perkembangan teknologi penginderaan jauh, UAV telah dimanfaatkan untuk mengambil citra tanaman yang dari ekstraksi guna memperoleh informasi penting sebagai data analisis. Lebih lanjut, pendekatan berbasis UAV yang dikombinasikan dengan CNN telah terbukti efektif dalam mendeteksi pohon yang terinfeksi *Ganoderma* melalui analisis citra udara, bahkan sebelum munculnya gejala visual pada tanaman (Kurihara *et al.*, 2020). Seluruh pendekatan tersebut mencerminkan peran signifikan kemajuan teknologi dalam mendukung strategi pengelolaan *Ganoderma* yang lebih presisi, efisien, dan berkelanjutan, guna menjaga produktivitas serta keberlanjutan industry kelapa sawit di era modern.

2.2 Penginderaan Jauh

Penginderaan jauh adalah teknik pengumpulan data suatu objek, fenomena, atau wilayah tanpa melakukan kontak secara langsung atau jarak jauh, seperti melalui satelit, pesawat atau UAV. Dengan adanya teknologi ini, pengawasan terhadap kondisi lingkungan dapat dilakukan secara efektif dan menyeluruh. Misalnya, penelitian oleh Kosasih dkk. (2020) yang menggunakan citra satelit Lansat 8 untuk menganalisis kerapatan vegetasi dan suhu permukaan tanah serta

penelitian oleh Junarto dkk. (2020) yang menggunakan UAV untuk pemetaan kadaster.

Penginderaan jauh telah menjadi komponen vital dalam pemantauan perkebunan kelapa sawit, terkhusus dalam mendeteksi penyakit busuk pangkal batang yang disebabkan oleh jamur *Ganoderma boninense*. Dengan teknologi ini, pemantauan lahan yang sebelumnya memakan banyak waktu dan tenaga kini dapat dilakukan secara lebih efisien dan mencakup area yang lebih luas. Studi yang dilakukan oleh Kurihara *at al.* (2020) menunjukkan bahwa penggunaan citra hiperspektral dari UAV dapat mengklasifikasikan tingkat infeksi busuk pangkal batang pada pohon kelapa sawit dengan akurasi tinggi, bahkan pada tahap awal infeksi. Hal ini menunjukkan potensi besar penginderaan jauh dalam mendukung manajemen perkebunan kelapa sawit secara berkelanjutan.

Studi oleh Lee *at al.* (2022) menggunakan citra hiperspektral dari UAV untuk mendeteksi infeksi busuk pangkal batang pada pohon kelapa sawit sebelum gejala visual muncul dengan hasil efektivitas yang signifikan. Pendekatan ini memungkinkan untuk intervensi lebih awal, yang penting untuk mencegah penyebaran penyakit dan kerugian ekonomi yang lebih besar.

2.3 Infrensi citra pengindraan jauh dengan AI

Pengintegrasian penginderaan jarak jauh dengan kecerdasan buatan telah mengubah cara pemantauan dan pengelolaan lahan pertanian. Teknologi ini memungkinkan deteksi dini terhadap berbagai kondisi tanaman, seperti serangan hama, penyakit, dan stress lingkungan, melalui analisis cirta satelit atau UAV. Studi oleh Aziz *at al.* (2025) menunjukkan bahwa penggunaan kecerdasan buatan dalam penginderaan jauh dapat meningkatkan efisiensi pemantauan lahan, dan kesehatan tanaman, serta memungkinkan pengambilan keputusan yang lebih cepat dan tepat dalam manajemen pertanian. Selain itu, model *Bayesian Probabilistic Updating Model* (BPUM) yang dikembangkan oleh Shi *at al.* (2025) memanfaatkan data historis dan citra satelit secara *real-time* untuk identifikasi awal tanaman dengan akurasi tinggi, demi mendukung ketahanan pangan global.

2.4 Implementasi *Maching Learning* (ML)

Machine Learning (ML) telah menjadi alat penting dalam bidang pertanian modern. Algoritma-algoritma ML digunakan untuk mengolah data besar dari berbagai sumber, seperti sensor tanah, cuaca, dan citra tanaman, guna memprediksi hasil panen, mendekripsi penyakit, dan mengoptimalkan penggunaan sumber daya. Studi oleh Araújo *et al.* (2023) menyoroti tren terkini dalam penerapan ML di pertanian, termasuk tantangan dan perspektif masa depan. Mereka menekankan pentingnya infrastruktur siber yang mendukung pengumpulan, pemrosesan, dan pelatihan model ML dengan *dataset* multimodal untuk mempercepat inovasi di bidang pertanian.

2.5 Implementasi Deep Learning (DL)

Deep learning (DL) telah menjadi teknologi kunci dalam transformasi pertanian modern. Dengan kemampuannya dalam menganalisis data besar dan kompleks, DL memungkinkan identifikasi pola yang tidak terlihat oleh metode konvensional. Dalam bidang pertanian, DL telah digunakan untuk berbagai aplikasi seperti klasifikasi tanaman, deteksi penyakit, prediksi hasil panen, dan manajemen sumber daya. Studi oleh KesKes (2025) menyoroti bahwa integrasi DL dalam pertanian dapat meningkatkan efisiensi produksi dan mendukung pertanian berkelanjutan.

Salah satu arsitektur DL yang paling umum digunakan adalah CNN yang terdiri dari beberapa lapisan konvolusi yang menggunakan fungsi aktivasi seperti ReLU untuk mengekstrasi fitur dari data input. CNN efektif dalam menganalisis data citra, seperti citra satelit atau UAV untuk mendekripsi penyakit tanaman, klasifikasi dan klasifikasi lahan. Dalam konteks pertanian cerdas, CNN menunjukkan dapat digunakan untuk mengklasifikasikan gulma dan penyakit tanaman secara akurat (Faisal *et al.*, 2025).

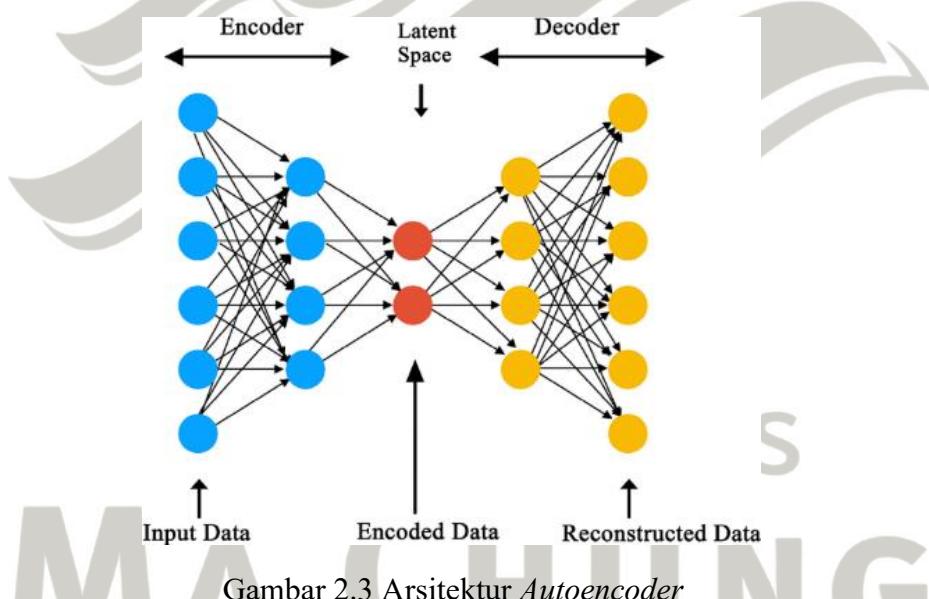
Selain CNN, arsitektur DL lainnya yang juga memiliki peran penting dalam pengolahan data citra pertanian adalah AE (Gambar 2.3). AE merupakan jenis jaringan saraf tiruan (JST) yang digunakan dalam mendekripsi anomali. AE memiliki dua bagian, yaitu:

a. Encoder

Bertujuan untuk mengubah gambar input menjadi representasi berdimensi lebih rendah yang melibatkan serangkaian lapisan terhubung yang secara bertahap mengurangi dimensi data. Kemudian informasi yang diperoleh disimpan dalam *bottleneck*, yang menjadi inti dari representasi fitur yang lebih padat dan bermakna.

b. Decoder

Bertugas untuk membalikkan proses *encoder*, yaitu merekonstruksi ulang data dari representasi laten menjadi bentuk yang menyerupai input asli dan memastikan bahwa hanya informasi penting dari data yang berhasil di pertahankan dan dipelajari. Kinerja AE dinilai dari seberapa kecil nilai *loss reconstruction*, yaitu nilai selisih antara input awal dan hasil rekonstruksi.



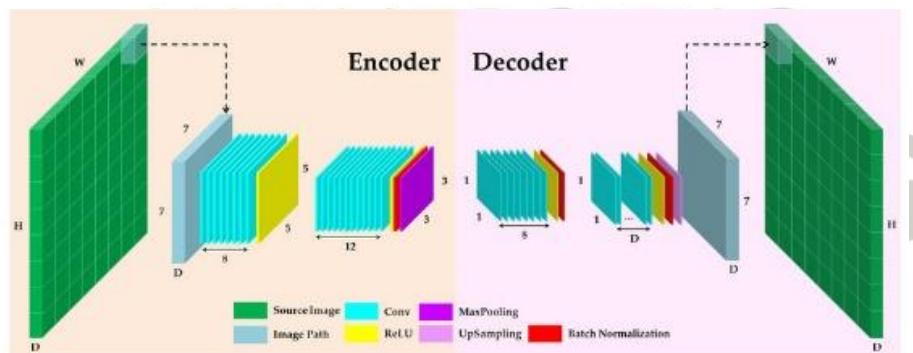
Gambar 2.3 Arsitektur Autoencoder

(Sumber: <https://pyimagesearch.com/2023/07/10/introduction-to-autoencoders/>)

Salah satu hal yang membuat AE menarik adalah pada proses pelatihan, AE membutuhkan data *training* lebih sedikit bila dibandingkan dengan CNN. CNN membutuhkan ribuan bahkan ratusan gambar, Sementara itu AE dapat dilatih menggunakan data tanpa label, menjadikannya ideal saat data berlabel terbatas atau tidak tersedia yang dalam kasus penelitian yang datanya sulit untuk dikumpulkan dalam jumlah besar menjadi daya tarik atau keunggulan tersendiri.

Dalam perkembangannya, beberapa variasi AE telah dikembangkan guna memenuhi berbagai kebutuhan dan karakteristik data. Dua diantaranya adalah, *Convolutional Autoencoder* (CAE) dan *Convolutional Variational Autoencoder* (CVAE). VAE memperluas konsep dasar dari AE dengan mengintegrasikan metode probabilistik ke dalam tahap *encoding*, yang memungkinkan dihasilkannya representasi laten dengan distribusi statistik yang lebih baik. Kemampuan ini menjadikannya sangat berguna dalam membuat model generatif atau ketika kita membutuhkan representasi data yang lebih bervariasi namun tetap informatif. Saat lapisan konvolusi ditambahkan, model ini dikenal sebagai CVAE, yang efektif untuk memproses data gambar.

Sementara itu, CAE (Gambar 2.4) dirancang secara khusus untuk menangani data berbentuk citra (*image-based data*) dengan menerapkan lapisan konvolusi (*convolutional layers*) pada bagian *encoder* dan *decoder*. Penggunaan lapisan konvolusi memungkinkan CAE untuk menangkap fitur spasial dan pola lokal dalam citra dengan lebih efektif dibandingkan AE biasa yang menggunakan lapisan *dense* atau *fully connected*. Oleh karena itu, CAE sangat cocok digunakan untuk pemrosesan citra beresolusi tinggi seperti citra satelit atau UAV dalam konteks penginderaan jauh (*remote sensing*).



Gambar 2.4 Arsitektur *Convolutional AutoEncoder*

(Sumber: https://www.researchgate.net/figure/The-proposed-architecture-of-the-CAE_fig3_352777611)

Dalam penginderaan jauh, CAE telah banyak dimanfaatkan untuk berbagai aplikasi, seperti klasifikasi tutupan lahan, deteksi perubahan (*change detection*),

serta deteksi anomali pada lahan pertanian. Kemampuannya untuk melakukan ekstraksi fitur yang efisien dari citra beresolusi tinggi menjadikan CAE sangat relevan dalam mendeteksi pola-pola tak biasa yang muncul akibat infeksi penyakit pada tanaman, termasuk infeksi *Ganoderma* pada kelapa sawit. CAE juga memiliki keunggulan dalam melakukan rekonstruksi citra secara presisi, sehingga nilai error rekonstruksi yang dihasilkan dapat dijadikan indikator awal untuk mengenali keberadaan anomali atau gangguan pada vegetasi.

Selain itu, CAE dalam mendeteksi anomali pada infeksi *Ganoderma* hanya perlu dilatih dengan menggunakan data pohon sehat. Nantinya ketika diinputkan data terinfeksi, hasil rekonstruksinya menunjukkan nilai error yang lebih tinggi, sehingga anomali dapat dikenali berdasarkan besarnya nilai error rekonstruksi. Dalam penelitiannya Xiang *et al.* (2024) memanfaatkan pendekatan AE berbasis piksel untuk mendeteksi anomali dalam citra hiperspektral, yang menunjukkan efektivitas yang tinggi dalam mengidentifikasi area yang tidak biasa dalam data penginderaan jauh.

Dalam upaya mengevaluasi performa dari CAE dan CVAE, beberapa metrik yang umum digunakan adalah *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Structural Similarity Index Measure* (SSIM). MAE menghitung rata-rata selisih absolut antara nilai piksel input dan hasil rekonstruksi, sementara MSE menghitung rata-rata kuadrat selisih antara nilai piksel input dengan hasil rekonstruksi, yang mengukur sejauh mana informasi asli dipertahankan. Nilai MSE yang rendah menunjukkan bahwa rekonstruksi mendekati input aslinya. Namun, karena MSE mengkuadratkan kesalahan, ia sangat sensitif terhadap outlier (nilai piksel yang sangat berbeda) dan dapat mengabaikan struktur spasial penting dalam citra. Sebagai alternatif yang lebih tangguh terhadap outlier, digunakan MAE. MAE menghitung rata-rata selisih absolut antar piksel, sehingga ia memberikan bobot yang sama untuk semua kesalahan dan tidak membesarakan nilai-nilai ekstrem. Ini menjadikannya pilihan yang baik saat noise atau nilai piksel yang tidak biasa hadir dalam data.

Untuk mengatasi kelemahan metrik berbasis piksel seperti MSE dan MAE, digunakan SSIM. SSIM mengevaluasi kemiripan struktural antara dua gambar

berdasarkan luminansi, kontras, dan struktur spasial, memberikan perspektif yang lebih dekat dengan persepsi visual manusia. SSIM memberikan skor antara 0 dan 1, di mana nilai mendekati 1 menunjukkan kemiripan struktural yang tinggi (Wang *et al.*, 2004). Dengan menggunakan kombinasi metrik ini, kita bisa mendapatkan gambaran evaluasi yang lebih komprehensif, mencakup akurasi numerik dan kemiripan struktural dari hasil rekonstruksi.

Sementara itu, untuk mengevaluasi model probabilistik seperti VAE, metrik yang fokus pada perbandingan distribusi juga menjadi sangat penting. Dua di antaranya adalah *Jensen-Shannon Divergence* (JSD) dan *Bhattacharyya Distance* (BD). JSD adalah metrik yang mengukur kesamaan antara dua distribusi probabilitas dan sering digunakan untuk mengevaluasi seberapa baik distribusi laten yang dipelajari VAE mendekati distribusi target yang diinginkan (Deasy *et al.*, 2020). JSD memiliki nilai yang terikat dan simetris, menjadikannya pilihan yang kuat. Demikian pula, BD mengukur tingkat tumpang tindih antara dua distribusi probabilitas, mengukur seberapa dekat keduanya (Chawla, 2024). BD secara efektif mengukur jarak antara distribusi encoder dan distribusi prior dalam ruang laten VAE. Nilai BD yang lebih rendah menunjukkan bahwa kedua distribusi lebih mirip, menandakan bahwa VAE berhasil mempelajari representasi laten yang terdistribusi dengan baik dan efisien.

Dalam upaya meningkatkan akurasi dan efisiensi dalam pelatihan model DL, telah dikembangkan berbagai teknik optimasi. Salah satu metode optimasi paling populer dalam DL adalah *Adaptive Moment Estimation* (Adam) yang mengkombinasikan keunggulan dari momentum dan RMSProp untuk menyesuaikan *learning rate* secara adaptif terhadap parameter, sehingga mempercepat konvergensi dan menghindari jebakan pada minimum lokal (Kingma & Ba, 2014). Selain itu, *Nesterov-accelerated Adaptive Moment Estimation* (Nadam) merupakan penyempurnaan dari Adam dengan menambahkan *lookahead gradient* yang mempercepat konvergensi di banyak kasus (Dozat, 2016). Teknik lain seperti SGD dengan *Momentum*, Adagrad, dan RMSProp juga masih relevan, tergantung pada jenis *dataset* dan arsitektur jaringan yang digunakan. Pemilihan algoritma optimasi yang tepat sangat berpengaruh terhadap stabilitas pelatihan, kecepatan konvergensi, dan kualitas generalisasi model.

Selanjutnya, terdapat fungsi aktivasi dalam JST yang memegang peranan penting dalam memperkenalkan non-linearitas pada jaringan. Tanpa fungsi aktivasi, jaringan hanya akan mempelajari relasi linear yang berakibatkan terbatasnya kemampuan dalam menyelesaikan masalah kompleks seperti klasifikasi citra. Fungsi aktivasi yang umum digunakan adalah ReLU, yang sederhana namun sangat efektif untuk menghindari masalah vanishing gradient. ReLU memiliki bentuk fungsi $f(x) = \max(0, x)$ yang membuatnya sangat efisien dalam propagasi maju dan mundur. Varian dari ReLU seperti *Leaky ReLU* dan *Parametric ReLU (PReLU)* diperkenalkan untuk menangani kelemahan ReLU dalam mengabaikan nilai negatif. Selain itu, untuk arsitektur seperti CAE, *Sigmoid* atau *Tanh* juga digunakan pada lapisan output, tergantung pada skala data input.

2.6 Citra Spasial

Citra spasial merupakan representasi visual dari permukaan bumi yang diperoleh melalui sensor pada satelit atau UAV. Dalam pertanian, terkhusus pada monitoring tanaman, citra spasial berperan penting dalam mendekripsi kondisi tanaman, identifikasi penyakit dan pengelolaan lahan secara presisi. Dengan menggunakan citra spasial, memungkinkan untuk memantau area yang luas dengan efisien serta memberikan informasi yang akurat dalam suatu lahan pertanian.

2.7 Pra-pemrosesan Citra Spasial (UAV)

Sebelum citra UAV dapat digunakan untuk analisis lebih lanjut, diperlukan serangkaian proses pra-pemrosesan untuk memastikan kualitas dan akurasi data. Tahapan utama dalam pra-pemrosesan citra UAV meliputi:

a. *Stitching (Mosaicking)*

Proses penggabungan beberapa citra individu menjadi satu citra utuh (*orthomosaic*) yang mencakup area yang lebih luas. Teknik ini penting untuk menghasilkan peta yang kontinu dan konsisten, serta mempermudah analisis spasial. Angel *et al.* (2020) mengembangkan metode otomatis untuk georektifikasi

dan *mosaicking* citra hiperspektral berbasis UAV, yang meningkatkan efisiensi dalam pengolahan data citra.

b. *Georeferencing*

Penyesuaian posisi citra agar sesuai dengan koordinat geospasial di permukaan bumi. Proses ini biasanya dibantu dengan data GPS atau *Ground Control Points* (GCP) untuk memastikan akurasi spasial yang tinggi. Studi oleh Angel *et al.* (2020) menunjukkan bahwa integrasi *georeferencing* dalam *workflow* pra-pemrosesan citra UAV dapat meningkatkan akurasi spasial hingga tingkat yang signifikan.

c. *Radiometric Correction*

Koreksi nilai piksel citra untuk menghilangkan pengaruh pencahayaan, bayangan, atau atmosfer yang dapat memengaruhi hasil analisis. Proses ini penting untuk memastikan bahwa nilai reflektansi yang diperoleh mencerminkan kondisi sebenarnya di lapangan. Angel *et al.* (2020) menekankan pentingnya koreksi radiometric dalam meningkatkan kualitas citra UAV untuk aplikasi pertanian.

d. *Noise Removal*

Penghapusan data citra yang tidak relevan atau gangguan (seperti kabut, noise sensor, atau objek luar tanaman) untuk meningkatkan akurasi ekstraksi fitur. Proses ini memastikan bahwa analisis yang dilakukan hanya berdasarkan data yang valid dan relevan.

e. *Resampling dan Cropping*

Penyesuaian ukuran dan resolusi piksel agar selaras dengan skala pengolahan, serta pemotongan area yang tidak dibutuhkan untuk fokus pada area analisis yang relevan. Tahapan ini mempermudah proses analisis dan mengurangi beban komputasi.

2.8 Ruang warna RGB

Ruang warna RGB (Gambar 2.5) merupakan sistem representasi warna dalam citra digital yang menggunakan tiga warna, yaitu Merah (*Red*), Hijau (*Green*), dan Biru (*Blue*) (Pamungkas, 2023). Dengan menambahkan intensitas

cahaya, RGB dapat menghasilkan berbagai warna lainnya karena sifatnya aditif. Dalam citra digital, setiap warna memiliki intensitas antara 0 hingga 255, dimana 0 berarti tidak ada intensitas warna sedangkan 255 berarti intensitas penuh warna (Pamungkas, 2023).



Gambar 2.5 Ruang warna RGB

(Sumber: <https://www.shutterstock.com/image-vector/additive-color-mixing-scheme-rgb-colors-1439873501>)

2.9 Python

Python adalah bahasa pemrograman yang dikembangkan sebagai penerus dari bahasa pemrograman ABC oleh Guido van Rossum pada akhir tahun 1980-an dan kemudian dirilis secara publik pada tahun 1991 (Wikipedia contributors, 2025). Python mendukung berbagai paradigma pemrograman seperti objektif, prosedural dan fungsional (Python Software Foundation, 2025). Python menawarkan fitur seperti modul, pengecualian, pengetikan dan tipe data dinamis, serta kelas dengan sintaksis yang sederhana namun jelas dan kuat serta dapat diintegrasikan dengan pustaka eksternal, sistem operasi dan dapat diperluas menggunakan bahasa C atau C++ (Python Software Foundation, 2025).

2.9.1 NumPy

Numerical Python (NumPy) adalah pustaka python *open-source* yang menyediakan fungsi-fungsi matematika sampai operasi numerik dan struktur data array multidimensi. NumPy menjadi alat fundamental untuk memanipulasi data raster, seperti citra satelit atau UAV, karena kemampuannya dalam menangani array besar dengan efisien pada pengolahan citra dan penginderaan jauh. NumPy memungkinkan konversi array numerik dari citra raster sehingga, memudahkan proses seperti normalisasi nilai piksel, perhitungan indeks vegetasi (NDVI), dan deteksi perubahan tutupan lahan.

Dalam studi oleh Fuentes-Peñailillo *et al.* (2024), intergrasi antara IoT, penginderaan jauh, dan kecerdasan buatan dalam pertanian digital menunjukkan bahwa penggunaan NumPy dalam pra-pemrosesan data citra meningkatkan akurasi prediksi hasil panen. Selain itu, NumPy mampu melakukan operasi matematika kompleks, seperti transformasi *Fourier* dan dekomposisi matriks, juga mendukung analisis spasial lanjutan dalam penginderaan jauh. Dengan demikian, NumPy menjadi komponen kunci dalam *pipeline* pemrosesan data geospasial, dari tahap pra-pemrosesan hingga analisis dan visualisasi (Mulkar, H., 2024).

2.9.2 Matplotlib

Matplotlib merupakan pustaka visualisasi data 2D dan 3D dalam bahasa pemrograman Python yang sangat populer di kalangan ilmian data dan peneliti. Matplotlib digunakan dalam penginderaan jarak jauh dan pertanian presisi untuk memvisualisasikan data citra satelit hasil klasifikasi serta analisis spasial lainnya. kemampuannya dalam menghasilkan berbagai jenis grafik seperti peta sebaran histogram dan grafik *timeseries* menjadikan matplotlib alat yang esensial dalam interpretasi data geospasial.

Dalam praktiknya, Matplotlib sering digunakan bersama pustaka lain seperti NumPy dan Rasterio untuk memproses dan menampilkan citra penginderaan jauh. Misalnya, dalam analisis indeks vegetasi normalisasi (NDVI), Matplotlib dapat digunakan untuk memvisualisasikan distribusi nilai NDVI pada lahan

pertanian, membantu dalam *monitoring* kesehatan tanaman dan deteksi dini penyakit.

Selain itu, Matplotlib juga digunakan dalam pelatihan dan evaluasi model DL untuk deteksi objek dalam citra pertanian. Visualisasi hasil prediksi model, seperti *bounding box* pada objek yang terdeteksi, memungkinkan peneliti untuk menilai kinerja model secara intuitif. Dengan demikian, Matplotlib tidak hanya berfungsi sebagai alat visualisasi, tetapi juga sebagai komponen integral dalam *pipeline* analisis data penginderaan jauh dan pertanian presisi.

2.10 QGIS

Quantum geographic information system (QGIS) merupakan perangkat lunak *open-source* berbasis Sistem Informasi Geografis (SIG) yang digunakan untuk pengolahan, analisis, visualisasi dan memetakan data spasial. Dalam bidang pertanian, QGIS berperan penting dalam pemrosesan data penginderaan jauh, seperti citra dari UAV atau satelit, untuk memantau kesehatan tanaman, mendeteksi penyakit, dan mengelola lahan.

Salah satu keunggulan QGIS adalah integrasi dengan plugin tambahan yang mendukung analisis lanjutan. Misalnya, plugin *Deepness* memungkinkan pengguna menerapkan model DL langsung pada layer raster di QGIS, memfasilitasi tugas-tugas seperti segmentasi dan deteksi objek dalam citra penginderaan jauh (Aszkowski *et al.*, 2023). Selain itu, plugin *Modules for Land Use Change Simulations* (MOLUSCE) digunakan untuk menganalisis perubahan penggunaan lahan dan memprediksi scenario perubahan di masa depan, yang sangat berguna dalam perencanaan pertanian berkelanjutan (Muhammad *et al.*, 2022).

2.11 Roboflow

Roboflow adalah platform cloud komputasi visual yang menyediakan solusi *end-to-end* untuk pengembangan model deteksi objek, mulai dari anotasi data, augmentasi, pelatihan model, hingga *deployment*. Platform ini mendukung berbagai model DL, seperti YOLOv8 dan yang terbaru YOLOv11, yang memungkinkan

pengguna membangun serta menerapkan model deteksi objek secara efisien. Dalam sektor pertanian, Roboflow telah dimanfaatkan untuk mengidentifikasi penyakit tanaman, mendeteksi gulma, dan memantau perkembangan tanaman melalui citra UAV maupun satelit. Salah satu contohnya adalah ketersediaan *dataset* publik seperti *PlantDoc*, yang memuat 2.598 gambar guna membantu proses identifikasi penyakit tanaman (Roboflow, 2025).

Di samping itu, Roboflow *Universe* menyediakan beragam *dataset* dan model pra-latih yang dapat dimanfaatkan untuk meningkatkan efisiensi pertanian, menurunkan biaya produksi, serta mendorong peningkatan hasil panen (Roboflow, 2025). Platform ini juga mendukung integrasi dengan model seperti YOLOv5, yang telah terbukti memiliki tingkat presisi tinggi—mencapai 90% dalam mendeteksi penyakit pada daun padi (Kanna *et al.*, 2024). Dengan kemampuannya dalam menangani seluruh siklus pengembangan model deteksi objek, Roboflow menjadi salah satu alat penting dalam implementasi teknologi penginderaan jauh untuk mendukung pertanian presisi.

Dalam penelitian ini, Roboflow dimanfaatkan untuk mempersiapkan *dataset* melalui fitur *dataset preprocessing*, yang berperan dalam meningkatkan kualitas data sehingga dapat mendukung proses pelatihan model secara lebih optimal. Preproses yang dilakukan seperti:

1. Mengubah ukuran citra agar konsisten dengan input model dan generalisasi skala objek pada citra.
2. Menyesuaikan orientasi gambar agar memiliki orientasi yang sama semua.
3. Augmentasi data seperti membalikkan citra secara horizontal atau vertikal guna menambah variasi data, memutar citra guna membantu model mengenali objek dari berbagai sudut, dan pemotongan citra dengan tujuan memusatkan perhatian pada bagian tertentu dari citra guna meningkatkan detail analisis.

2.12 Google Colaboratory

Google Colaboratory atau singkatannya Google Colab merupakan layanan *cloud* dari Google yang dapat digunakan untuk menulis, menjalankan dan berbagi

kode Python langsung dari *web-browser* tanpa memerlukan perangkat lunak tambahan (Revou, 2025). Dengan menyediakan akses gratis ke sumber daya komputasi seperti GPU dan TPU, *Google Colab* memudahkan praktisi *data science, machine learning*, serta kalangan pendidikan dalam menjalankan berbagai kebutuhan komputasi (Google Colab, 2025).

2.13 TensorFlow

TensorFlow merupakan salah satu pustaka *open-source* yang dikembangkan oleh Google untuk kebutuhan ML dan DL. Salah satu kekuatan utama TensorFlow adalah fleksibilitasnya dalam mengatur dan menjalankan *data flow graph* yang dapat menangani operasi tensor kompleks. TensorFlow memungkinkan pengguna membangun dan menyesuaikan arsitektur model, mengatur proses pelatihan, serta melakukan evaluasi dan *deployment* model pada perangkat *edge* hingga *cloud*.

Dalam praktik pengembangan model DL modern, TensorFlow umumnya digunakan bersama dengan Keras, yang merupakan API tingkat tinggi untuk membangun dan melatih jaringan saraf tiruan. Sejak TensorFlow 2.0, Keras telah terintegrasi secara penuh sebagai *default high-level API* yang memudahkan proses pengembangan model, tanpa mengorbankan performa. Dengan menggunakan Keras, pengguna dapat membuat model DL secara lebih cepat dan efisien menggunakan antarmuka berbasis Python yang intuitif. Arsitektur seperti AE, CNN, dan LSTM dapat dirancang hanya dengan beberapa baris kode melalui pendekatan *Sequential* atau *Functional API*.

2.14 Ultralytics

Ultralytics adalah sebuah platform pengembangan yang secara khusus dikenal sebagai pengembang utama dan penyedia implementasi model-model *You Only Look Once* (YOLO) (Ultralytics, 2023). Sebagai entitas di balik versi-versi populer seperti YOLOv5 dan YOLOv8, Ultralytics menyediakan kerangka kerja yang efisien dan mudah digunakan untuk pelatihan, validasi, dan inferensi model

deteksi objek. Platform ini memfasilitasi penelitian dan aplikasi praktis dalam visi komputer dengan menawarkan kode sumber terbuka, model pra-terlatih, dan alat pendukung yang mengoptimalkan kinerja dan kecepatan deteksi objek.

2.15 Penelitian terdahulu

Dalam *deep learning*, AE telah menjadi teknik penting dalam pengolahan citra dan penginderaan jauh. Pada sektor pertanian, AE dapat digunakan untuk klasifikasi dan segmentasi citra tanaman, serta mendeteksi dini penyakit pada tanaman. Dalam penelitiannya, Isinkaye *at al.* (2025) mengembangkan model *hybrid* yang mengkombinasikan *Variational Autoencoder* (VAE) dan *Vision Transformer* (ViT) untuk meningkatkan akurasi dan ketahanan dalam mendiagnosis penyakit pada tanaman. Model tersebut mampu untuk mengklasifikasikan berbagai jenis penyakit tanaman melalui analisis citra daun. Selain itu, penelitian yang dilakukan oleh Bedi dan Gole (2021) memperkenalkan model *hybrid* yang mengkombinasikan CAE dengan CNN untuk mendeteksi penyakit pada tanaman secara otomatis. Dengan akurasi ujicoba yang mencapai 98.83%, model ini layak untuk dipertimbangkan dalam aplikasi pertanian presisi

Di sisi lain, berbagai pendekatan kecerdasan buatan dalam mendeteksi infeksi *Ganoderma* pada tanaman kelapa sawit telah memperlihatkan performa yang menjanjikan serta prospek pengembangan yang luas di masa mendatang. Penelitian yang dilakukan oleh Ahmadi *at al.* (2020) memanfaatkan citra UAV dan model *Artificial Neural Network* (ANN) dalam mendeteksi infeksi *Ganoderma* pada tanaman kelapa sawit yang berhasil mengidentifikasi tingkat keparahan infeksi dengan akurasi yang tinggi. Secara umum, berbagai penelitian yang sebelumnya telah menunjukkan bahwa AE dan teknik *deep learning* lainnya memiliki potensi dalam melakukan klasifikasi, segmentasi, serta mendeteksi dini penyakit pada tanaman, termasuk infeksi *Ganoderma* pada tanaman kelapa sawit.

Berbagai penelitian terdahulu menunjukkan bahwa model *deep learning*, khususnya *Autoencoder* (AE), menawarkan potensi dalam dalam implementasi pertanian presisi. Model AR terbukti efektif untuk klasifikasi, segmentasi, dan deteksi dini penyakit tanaman. Pendekatan *hybrid* yang mengkombinasikan AR

dengan arsitektur lain seperti *Convolutional Autoencoder* (CNN) atau *Vision Transformer* (ViT), telah mencapai akurasi tinggi dalam mendiagnosa penyakit melalui analisis citra. Sejalan dengan temuan tersebut, penelitian ini mengadopsi prinsip dasar model AE untuk mengembangkan model yang mampu mendeteksi dini infeksi *Ganoderma* pada pohon kelapa sawit. Pendekatan ini memanfaatkan kemampuan AE dalam mempelajari representasi data normal mengidentifikasi anomali visual pada citra penginderaan jauh, sebuah strategi yang didasarkan pada kesuksesan model AI sebelumnya dalam mendeteksi infeksi *Ganoderma*.



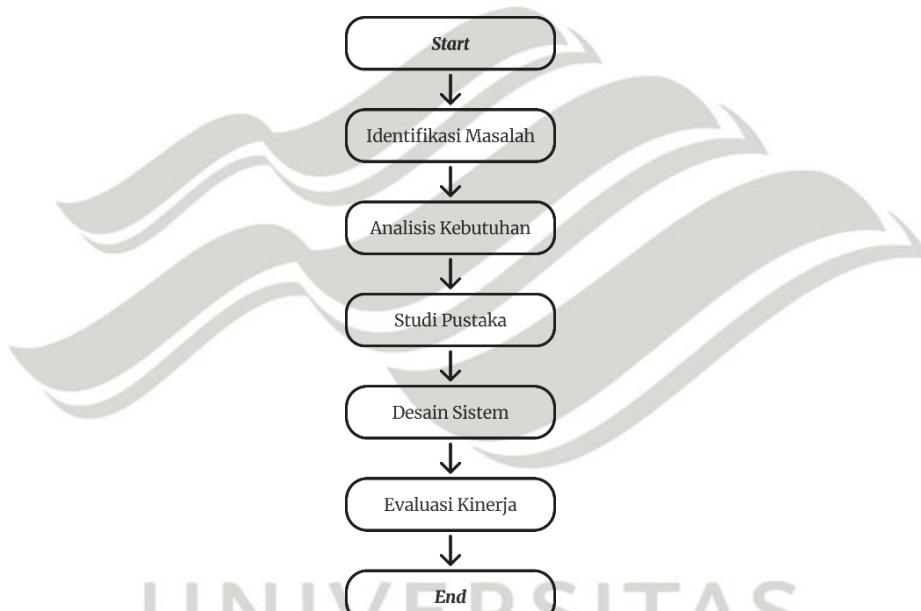
UNIVERSITAS
MA CHUNG

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Alur Penelitian

Penelitian tugas akhir ini bertujuan untuk mengembangkan dan menerapkan algoritma atau arsitektur AE guna mendeteksi infeksi *Ganoderma* secara dini pada pohon kelapa sawit dengan memanfaatkan citra yang diambil menggunakan UAV. Berikut adalah alur penelitian (Gambar 3.1) untuk penelitian tugas akhir ini.



Gambar 3.1 Alur penelitian

3.2 Identifikasi Masalah

Dalam perkebunan kelapa sawit saat ini masalah yang muncul adalah bagaimana mendeteksi atau memperkirakan pohon sawit mana yang kemungkinan akan terinfeksi *Ganoderma* sehingga bisa dilakukan pencegahan. Dengan luas wilayah perkebunan yang sangat luas, membutuhkan waktu yang signifikan untuk melakukan monitoring kondisi setiap pohon kelapa sawit. Oleh karena itu, dibutuhkan sebuah sistem yang akurat dalam memprediksi kemungkinan terjadinya infeksi *Ganoderma* pada citra UAV bulan selanjutnya dengan memanfaatkan citra UAV pada bulan-bulan sebelumnya.

3.3 Analisis Kebutuhan

Dalam rangka mendukung kelancaran penelitian ini, analisis kebutuhan sangat penting untuk memastikan bahwa semua aspek yang diperlukan dapat disiapkan dengan baik. Analisis kebutuhan dilakukan dengan tujuan merancang dan mengembangkan model AE yang efektif, serta menilai perangkat keras dan lunak yang diperlukan baik oleh peneliti maupun oleh pengguna yang akan menggunakan model tersebut.

3.3.1 Kebutuhan Pengguna

Kebutuhan pengguna dalam penelitian ini berfokus pada pengembangan model *deep learning* yang mampu mendeteksi dini infeksi *Ganoderma* pada tanaman kelapa sawit berdasarkan anomali visual (warna daun). Model ini dirancang sebagai inti dari sebuah sistem yang dapat mengidentifikasi pohon yang terinfeksi oleh *Ganoderma*. Meskipun demikian, pengembangan aplikasi deteksi yang akan mengintegrasikan model ini secara penuh akan menjadi fokus utama pada penelitian lanjutan.

3.3.2 Kebutuhan Peneliti

Berikut merupakan beberapa perangkat keras dan perangkat lunak yang digunakan oleh peneliti dalam melakukan penelitian ini.

1. Perangkat Keras

a. Laptop

- i. CPU : Intel i7-1165G7
- ii. GPU : NVIDIA GeForce MX350
- iii. RAM : 8 GB
- iv. SSD : 1 TB
- v. Sistem Operasi : Windows 10 Home

2. Perangkat Lunak

- a. Google Colaboratory
- b. Google Drive
- c. Python
- d. QGIS

e. Roboflow

f. Ultralytics

3.4 Studi Pustaka

Dalam tahap ini, peneliti melakukan studi pustaka terhadap penelitian-penelitian terdahulu yang berkaitan dengan penelitian ini. Penelitian terdahulu yang dijadikan referensi merupakan penelitian yang berkaitan dengan deteksi dini infeksi *Ganoderma* pada tanaman kelapa sawit khususnya menggunakan kecerdasan buatan serta penelitian yang berkaitan dengan penginderaan jauh.

Tabel 3.1 Ringkasan hasil penelitian terdahulu

No	Topik	Pengetahuan	Temuan
1.	Deteksi <i>Ganoderma</i> pada bibit sawit menggunakan hiperspektral VIS-NIR dan SVM.	Pemanfaatan data citra hiperspektral VIS-NIR dan algoritma SVM untuk deteksi infeksi <i>Ganoderma</i> pada bibit sawit.	Reflektansi spektrum NIR dari pelepah kelapa sawit menunjukkan perbedaan yang signifikan antara bibit yang tidak dan diinokulasi dengan patogen <i>G. boninense</i> . Melalui penerapan algoritma Fine Gaussian SVM, diperoleh tingkat akurasi klasifikasi sebesar 93% dengan menggunakan 35 band NIR (Azmi <i>et al.</i> , 2020).
2.	Deteksi <i>Ganoderma</i> dan hama penggerek batang pada kelapa sawit menggunakan spektrum reflektansi penginderaan jauh.	Analisis citra multispektral dan hiperspektral untuk deteksi infeksi <i>Ganoderma</i> dan hama pada kelapa sawit.	Penelitian oleh Anuar <i>et al.</i> (2021), menyatakan bahwa pencitraan multispektral mampu membedakan antara kondisi sehat, infeksi sedang, dan infeksi parah dengan akurasi keseluruhan lebih dari 90% namun tidak dengan infeksi ringan atau awal infeksi <i>Ganoderma</i> . Sementara itu, pencitraan hiperspektral mampu membedakan antara kondisi sehat, infeksi ringan, dan tingkat keparahan lainnya dari infeksi <i>Ganoderma</i> pada kelapa sawit.

Tabel 3.2 Lanjutan

No	Topik	Pengetahuan	Temuan
3.	Identifikasi penyakit pada tanaman menggunakan <i>Autoencoder</i> (AE).	Memanfaatkan AE sebagai ekstraksi fitur otomatis dalam menentukan tingkat keparahan penyakit busuk daun dini dan busuk daun akhir pada daun tomat.	Dalam penelitiannya, Ong (2021) menerapkan dua lapis arsitektur AE untuk mengekstraksi fitur dari citra, dengan menggunakan sebanyak 122 gambar untuk proses pelatihan dan pengujian. Hasilnya, model tersebut berhasil mengklasifikasikan tingkat keparahan penyakit tanaman dengan akurasi keseluruhan sebesar 72,7%.
4.	Deteksi penyakit tanaman menggunakan model <i>hybrid Convolutional Autoencoder</i> (CAE) dan <i>Convolutional Neural Network</i> (CNN).	Pendekatan model <i>hybrid CAE</i> dan CNN untuk deteksi otomatis penyakit pada tanaman.	Pendekatan hybrid yang mengombinasikan CAE dengan CNN telah dikembangkan untuk mendeteksi penyakit pada tanaman secara otomatis. Model menunjukkan akurasi pelatihan dan pengujian yang tinggi, masing-masing sebesar 99,35% dan 98,83%, dengan total parameter pelatihan yang digunakan sebanyak 9.914.

Berdasarkan berbagai penelitian terdahulu, deteksi infeksi *Ganoderma* pada kelapa sawit telah menunjukkan keberhasilan, khususnya dengan memanfaatkan citra hiperspektral yang sensitif terhadap perubahan spektrum *Near-Infrared* (NIR) pada pelepas, memungkinkan deteksi pada tahap awal infeksi (Azmi *et al.*, 2020; Anuar *et al.*, 2021). Meskipun demikian, citra multispektral yang lebih umum cenderung kurang efektif dalam mengidentifikasi gejala dini penyakit. Sebagai alternatif dari metode penginderaan jauh yang kompleks, model *deep learning* menawarkan solusi yang menjanjikan. *Autoencoder* (AE), misalnya, terbukti efektif sebagai ekstraktor fitur otomatis untuk klasifikasi penyakit tanaman, bahkan dengan jumlah data yang terbatas (Ong, 2021). Pendekatan ini diperkuat oleh

keberhasilan model *hybrid* yang mengkombinasikan CAE dengan CNN, yang menunjukkan akurasi sangat tinggi dalam deteksi penyakit tanaman secara otomatis. Hal ini menegaskan bahwa pendekatan berbasis AE adalah metode yang valid dan efektif untuk mengidentifikasi anomali yang mengindikasikan infeksi Ganoderma.

3.5 Desain Penelitian

Desain penelitian (Gambar 3.2) merupakan tahapan yang menggambarkan keseluruhan proses pengembangan model *deep learning* yang mampu mendeteksi anomali infeksi *Ganoderma* pada kelapa sawit. Pada bagian ini, proses dimulai dari pemilihan *Region of Interest* (ROI) hingga menghasilkan model yang mampu mendeteksi anomali pada pohon kelapa sawit. Desain ini dibuat untuk memastikan setiap tahapan saling terintegrasi dan mendukung pencapaian tujuan penelitian. Berikut penjelasan setiap tahapan:

- Pemilihan *Region of Interest* (ROI)

Proses diawali dengan pemilihan satu area (*region*) spesifik sebagai fokus penelitian.

- Pemotongan Citra untuk *Dataset YOLO*

Citra dari area yang dipilih kemudian dipotong ke dalam blok-blok kecil guna membentuk *Dataset YOLO*, yang dibahas pada Subbab 3.5.1.1. Tahapan ini penting untuk memperkaya jumlah data pelatihan serta memungkinkan model mendeteksi objek pohon pada skala yang lebih detail.

- Latih YOLO: Identifikasi Pohon

Dataset hasil pemotongan kemudian digunakan untuk melatih model deteksi objek YOLO yang berfungsi untuk mengidentifikasi posisi pohon kelapa sawit. Proses ini dijelaskan dalam Subbab 3.5.2.1. Hasil pelatihan berupa koordinat *bounding box* pada masing-masing pohon dalam citra.

- Pemotongan Citra untuk *Dataset AE*

Berdasarkan anotasi (*bounding box*) dari model YOLO, setiap pohon kelapa sawit dalam citra dipotong satu per satu dan digunakan untuk membentuk

dataset AE. *Dataset* ini digunakan untuk pelatihan model deteksi anomali dan dijelaskan pada Subbab 3.5.1.2.

- Persiapan dan Pengambilan *Dataset Timeseries*

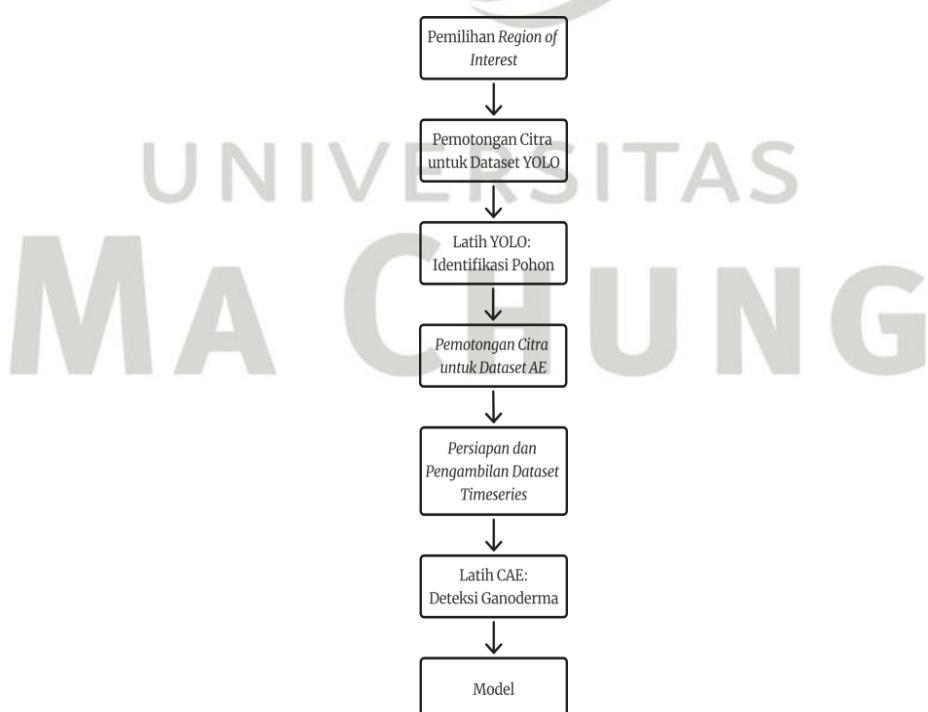
Citra diidentifikasi dan ditandai pohon-pohon kelapa sawit yang telah tumbang. Kemudian dilacak mundur pada citra dari bulan-bulan sebelumnya dan pohon-pohon yang sama diambil dari setiap bulan tersebut untuk membentuk dataset *timeseries*, yang akan digunakan untuk analisis perkembangan penyakit dari waktu ke waktu. Proses ini dibahas pada Subbab 3.5.1.3.

- Latih CAE: Deteksi *Ganoderma*

Citra pohon yang sudah terpotong digunakan untuk melatih model CAE dan VAE dalam mendeteksi anomali visual yang mengindikasikan keberadaan *Ganoderma*. Proses pelatihan model ini dibahas pada Subbab 3.5.2.2.

- Model

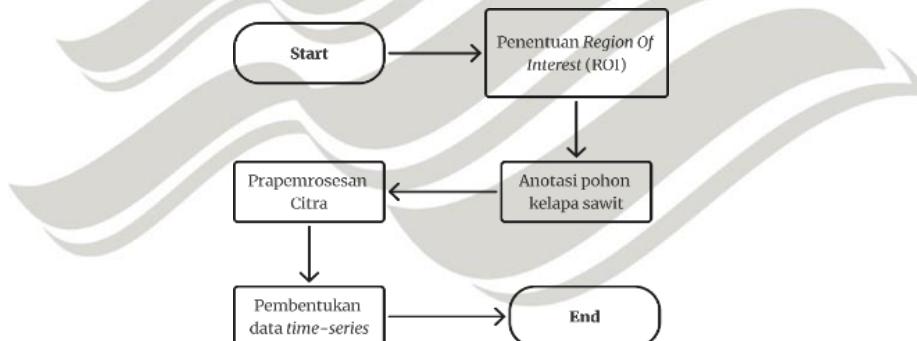
Output terakhir dari penelitian adalah model yang mampu mendeteksi anomali infeksi *Ganoderma* pada pohon kelapa sawit.



Gambar 3.2 Alur desain penelitian

3.5.1 Penyusunan *Dataset*

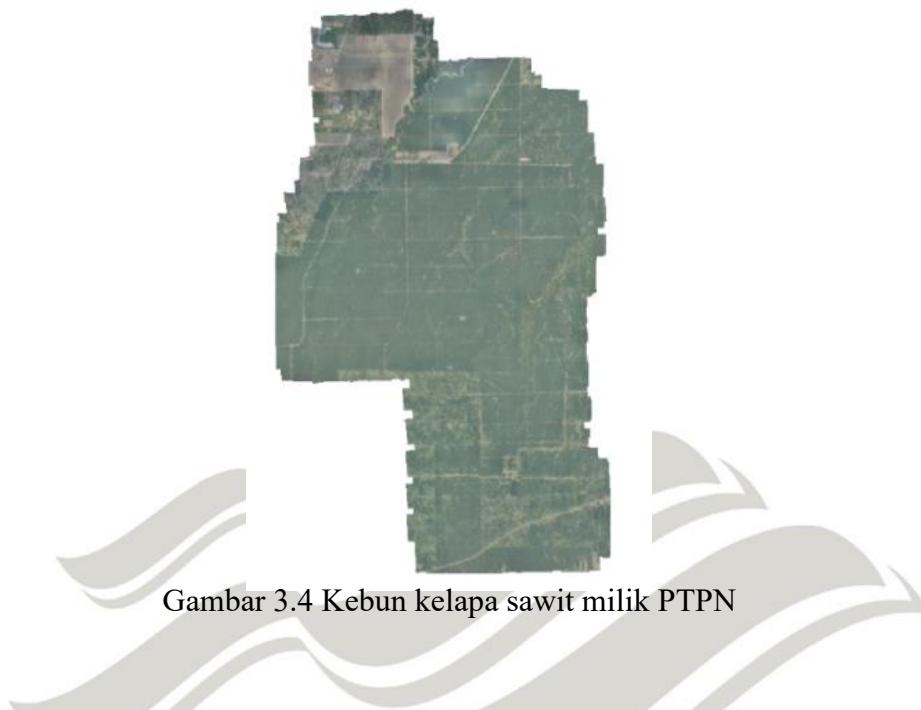
Dalam proses pengembangan model CAE guna mendeteksi dini infeksi *Ganoderma* pada pohon kelapa sawit berbasis kecerdasan buatan, tahapan penyusunan *dataset* memegang peranan yang sangat penting. *Dataset* menjadi fondasi utama dalam pelatihan dan pengujian model yang akan digunakan, baik untuk deteksi objek maupun untuk identifikasi anomali. Oleh karena itu, diperlukan pembagian *dataset* yang sistematis sesuai dengan kebutuhan dan karakteristik dari masing-masing model yang akan dikembangkan. Berikut Gambar 3.3 menjelaskan mengenai serangkaian proses penyusunan *dataset* yang akan diimplementasikan pada penelitian ini.



Gambar 3.3 Alur Penyusunan *Dataset*

Dataset yang digunakan dalam penelitian ini merupakan citra udara berukuran 50064 x 77179 piksel dengan 4 band (gambar 3.4) yang diperoleh menggunakan UAV pada ketinggian \pm 500 meter di wilayah perkebunan kelapa sawit PT. Perkebunan Nusantara (PTPN) di Gunung Bayu, Medan, Sumatera Utara. Data ini diperoleh sebagai dalam rangka kolaborasi dengan Kelompok Riset *Precision Agriculture* (KRPA). Proses pengambilan citra dilakukan secara berkala dalam rentang waktu mulai dari bulan Januari – Juni 2023 dan diproses serta dikategorikan ke dalam beberapa subset sesuai dengan tujuan analisis. Citra UAV bulan Juni 2023 akan digunakan sebagai *dataset* pelatihan untuk pengembangan dan pembelajaran model, sementara citra bulan Januari – Mei 2023 akan digunakan untuk *dataset timeseries* untuk uji coba dan evaluasi kinerja model. Model YOLO memerlukan *dataset* dengan anotasi khusus untuk mengidentifikasi objek pohon

kelapa sawit, sementara model CAE dan CVAE difokuskan untuk mendeteksi anomali dengan hanya dilatih menggunakan citra pohon sehat.

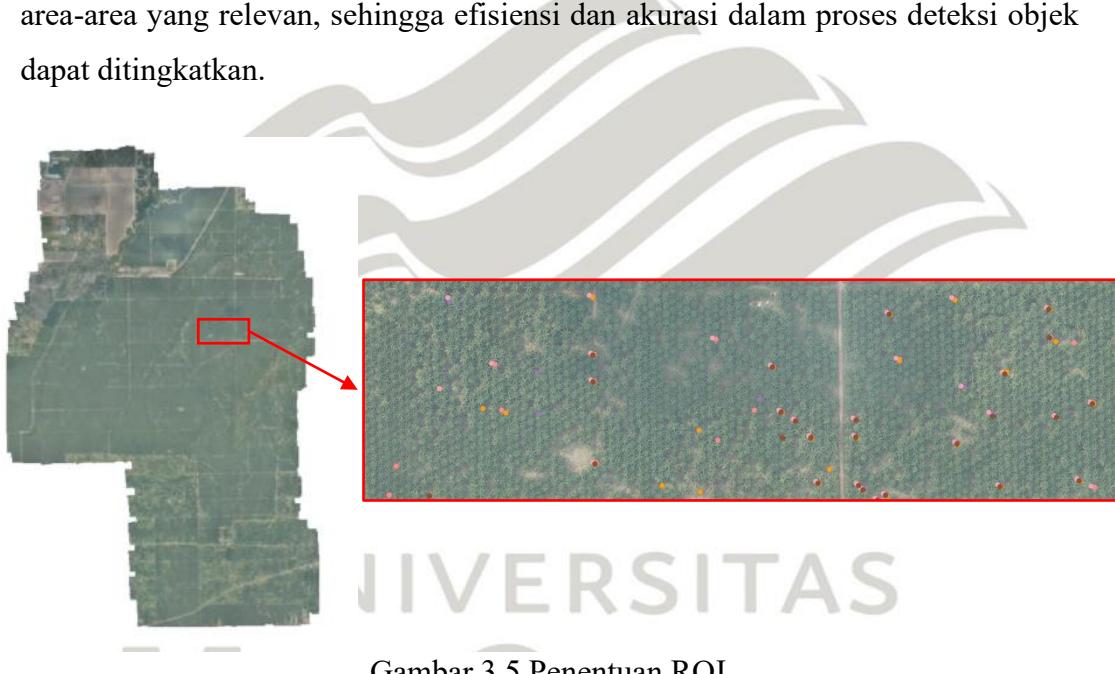


Gambar 3.4 Kebun kelapa sawit milik PTPN

Selain itu, citra UAV dari bulan Januari hingga Juni 2023 juga dimanfaatkan sebagai *dataset time-series* yang merepresentasikan dinamika kondisi tanaman dari waktu ke waktu. *Dataset* ini memiliki peran krusial dalam pengembangan sistem deteksi dini *Ganoderma* berbasis citra UAV, karena memungkinkan model untuk mempelajari dan mengenali perubahan visual yang terjadi secara bertahap pada tanaman yang mengalami infeksi. Untuk mengevaluasi batas kemampuan deteksi model, disusun pula *dataset timeseries* dengan pendekatan penelusuran mundur terhadap pohon-pohon yang telah diketahui mengalami infeksi berat atau kerusakan. Pendekatan ini bertujuan untuk mengidentifikasi titik awal perubahan visual yang dapat dikenali oleh model CAE dan CVAE, sehingga dapat diuji sejauh mana model mampu mendeteksi gejala awal sebelum munculnya tanda-tanda yang kasatmata. Pembagian dan penyusunan *dataset* dijelaskan secara rinci dalam tiga subbab berikut.

3.5.1.1 Dataset YOLO

Dataset ini disusun dengan tujuan utama untuk mendukung proses deteksi objek menggunakan model YOLO yang tugasnya mengenali dan mendeteksi objek menggunakan model YOLO yang bertugas mengenali dan mendeteksi keberadaan pohon kelapa sawit dalam citra UAV. Proses ini diawali dengan menggunakan QGIS untuk menentukan *Region of Interest* (ROI), yaitu wilayah pada kebun kelapa sawit yang diidentifikasi memiliki kepadatan pohon sawit tinggi, serta potensi adanya pohon kelapa sawit yang sakit atau tumbang (Gambar 3.5). Penentuan ROI ini bertujuan untuk memfokuskan proses anotasi dan pelatihan model hanya pada area-area yang relevan, sehingga efisiensi dan akurasi dalam proses deteksi objek dapat ditingkatkan.



Gambar 3.5 Penentuan ROI

Setelah ROI ditentukan, citra yang mengandung ROI dipotong terlebih dahulu menjadi 10 bagian atau blok yang berukuran 2283 x 1653 piksel menggunakan skrip Python kustom yang dirancang khusus untuk mempermudah proses anotasi seperti pada Gambar 3.6. Pemotongan ini bertujuan agar setiap blok mencakup bagian citra yang lebih fokus dan *manageable*.



Gambar 3.6 Pembagian citra ROI

Selanjutnya, proses penganotasi pohon kelapa sawit dilakukan menggunakan Roboflow, di mana setiap objek pohon diberi *bounding box* secara manual untuk keperluan pelatihan model deteksi. Setiap pohon kelapa sawit dalam citra ditandai menggunakan *bounding box* untuk menunjukkan posisi dan batas objek secara presisi (Gambar 3.7).

Pada tahap ini, hanya objek pohon kelapa sawit yang diberikan anotasi, sedangkan objek lain seperti bayangan, semak, maupun vegetasi selain kelapa sawit diabaikan dan tidak dianotasi. Selain itu, proses anotasi dilakukan tanpa label kondisi kesehatan pohon, sehingga klasifikasi objek dalam *dataset* ini terbatas pada identifikasi pohon kelapa sawit yang terdapat dalam citra UAV. Pendekatan ini memungkinkan model YOLO mempelajari karakteristik umum dari pohon kelapa sawit dan membedakannya dari elemen-elemen lain dalam lingkungan sekitar secara cepat dan efisien.



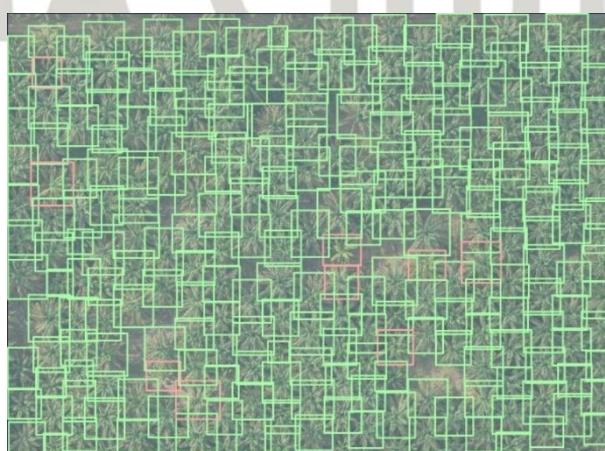
Gambar 3.7 Anotasi pohon kelapa sawit sehat tanpa label

Proses anotasi ini merupakan tahap krusial dalam penyusunan *dataset* karena secara langsung memengaruhi kualitas pembelajaran model. Anotasi yang akurat dan konsisten akan membantu model memahami pola visual objek target secara efektif dan meningkatkan performa deteksi di kondisi lapangan yang kompleks.

3.5.1.2 *Dataset Autoencoder*

Dataset AE ini merupakan kelanjutan dari *dataset YOLO* yang telah dianotasi sebelumnya. Jika pada *dataset YOLO* anotasi dilakukan tanpa memperhatikan kondisi kesehatan pohon, maka pada *dataset AE* proses penganotasi dilakukan dengan penambahan label yang mencakup informasi kesehatan pohon kelapa sawit, seperti label sehat dan terinfeksi. Informasi ini diperoleh dari data koordinat pohon yang mengalami infeksi hingga tumbang telah disediakan oleh PTPN dalam bentuk file *shapefile* (.shp).

Setiap pohon kelapa sawit dalam citra diberi label berdasarkan kondisinya, dengan *bounding box* berwarna hijau untuk pohon sehat, dan *bounding box* berwarna merah untuk pohon terinfeksi seperti yang ditampilkan pada Gambar 3.8. Label ini memudahkan proses klasifikasi dan menjadi dasar dalam tahapan prapemrosesan, yaitu *cropping* citra berdasarkan *bounding box* untuk menghasilkan potongan citra individual dari masing-masing pohon.

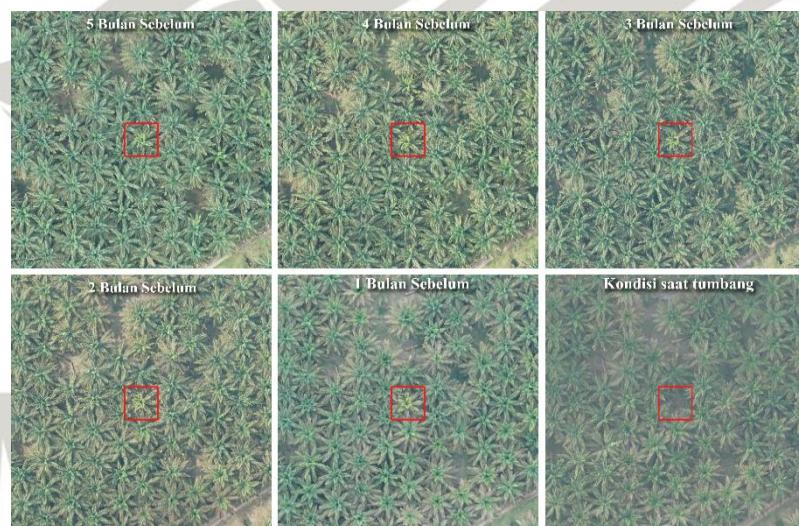


Gambar 3.8 Anotasi pohon kelapa sawit dengan label sakit dan sehat

Proses *cropping* bertujuan agar model CAE dan CVAE dapat mempelajari karakteristik visual normal pohon sawit sehat secara fokus, sehingga saat diuji dengan citra lain atau dalam kasus ini citra pohon sawit sakit, model dapat mendeteksi anomali visual yang mengindikasikan infeksi atau perubahan kondisi pohon.

3.5.1.3 Dataset Timeseries

Proses pembentukan *dataset* dilakukan dengan memilih pohon kelapa sawit yang telah teridentifikasi telah tumbang pada bulan Juni 2023. Selanjutnya, dikumpulkan citra UAV dari bulan-bulan sebelumnya (Mei-Januari 2023) yang menampilkan pohon kelapa sawit yang sama. Tujuannya adalah untuk menyusun *dataset* berurutan (*timeseries*) yang dapat menunjukkan perubahan kondisi visual pohon dari waktu ke waktu. Dengan pendekatan ini, model dapat mempelajari perubahan visual bertahap dari kondisi sakit sebelum akhirnya tumbang.



Gambar 3.9 Contoh citra *timeseries* pohon sawit

Dataset timeseries ini digunakan untuk mengevaluasi kemampuan AE mendeteksi anomali pada tahap awal infeksi dan mengukur batas deteksi (*limit detection*) model, yaitu sejauh mana model dapat mengenali tanda-tanda awal infeksi sebelum gejala muncul secara visual.

3.5.2 Desain Model

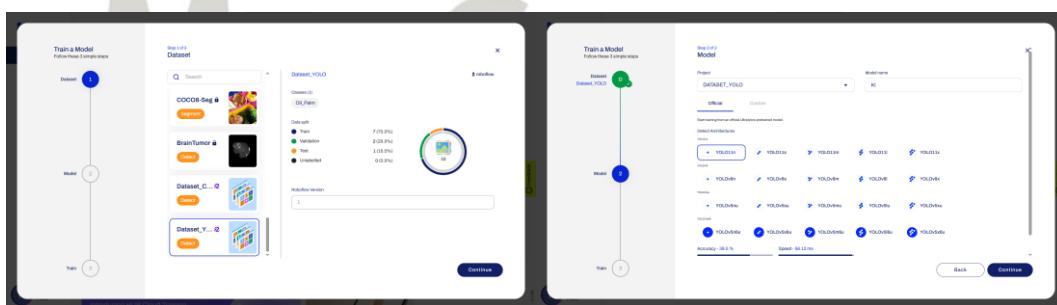
Pada penelitian ini, untuk mencapai tujuan deteksi dini infeksi ganoderma pada pohon kelapa sawit berbasis citra UAV, digunakan tiga model yaitu, model

YOLO, CAE, dan CVAE. Model YOLO berfungsi sebagai pendekripsi objek yang tujuannya mengidentifikasi dan menganotasi citra pohon kelapa sawit dari citra UAV secara otomatis. Sementara itu, dua variasi model AE akan digunakan untuk menganalisis citra hasil anotasi tersebut guna mendekripsi adanya indikasi dini infeksi *Ganoderma*. Berikut adalah penjelasan mengenai masing-masing model.

3.5.2.1 Model YOLO

Model YOLO merupakan algoritma deteksi objek berbasis DL yang dirancang untuk mendekripsi dan mengklasifikasikan objek dalam sebuah citra dengan cepat dan akurat. Dalam penelitian ini, model YOLOv11n (nano) dipilih karena arsitekturnya yang sederhana dan efisien bila dibandingkan dengan varian YOLOv11 lainnya. Model ini menawarkan keseimbangan optimal antara kecepatan, komputasi, dan akurasi, menjadikannya ideal untuk inferensi cepat serta implementasi pada perangkat dengan sumber daya terbatas. YOLOV11n nantinya akan digunakan untuk mengidentifikasi lokasi pohon kelapa sawit dalam citra UAV, dengan memanfaatkan *bounding box* sebagai penanda posisi objek.

Model YOLOv11n dibangun dan dilatih menggunakan kombinasi platform Roboflow untuk anotasi data serta Ultralytics beserta *Google Colab* untuk pelatihan. Proses pelatihan dilakukan dengan menginputkan *dataset* YOLO ke dalam model dan dengan data 70% untuk pelatihan, 20% untuk validasi, dan 10% untuk pengujian.



Gambar 3.10 Pemilihan Dataset dan Pelatihan Model di Ultralytics

Output yang diharapkan dari model ini berupa koordinat *bounding box* pohon kelapa sawit, yang kemudian digunakan sebagai input bagi model CAE dan CVAE, dengan tujuan mengevaluasi kondisi kesehatan masing-masing pohon secara individual.

3.5.2.2 Model Autoencoder (AE)

Dalam penelitian ini, penulis akan menggunakan dua variasi arsitektur AE, yaitu CAE dan CVAE. Kedua model ini dikembangkan melalui pendekatan eksperimental untuk mengidentifikasi konfigurasi *layer* yang paling optimal.

3.5.2.2.1 Convolutional Autoencoder (CAE)

CAE dikembangkan secara mandiri oleh penulis, mengacu pada pustaka TensorFlow (Tensorflow, 2024) dan repositori GitHub (aaryadev, 2024). Arsitekturnya diuji dalam tiga konfigurasi *encoder* yang berbeda, masing-masing dengan satu, dua, atau tiga *layer* Conv2D. Setiap *layer* Conv2D diikuti oleh fungsi aktivasi ReLU dan MaxPooling2D untuk mengurangi dimensi spasial gambar secara bertahap. Bagian *decoder* CAE dirancang secara simetris, menggunakan satu, dua, atau tiga *layer* Conv2DTranspose yang masing-masing diikuti oleh fungsi aktivasi ReLU dan UpSampling2D untuk mengembalikan dimensi gambar ke ukuran aslinya.

The diagram illustrates three different configurations of a Convolutional Autoencoder (CAE) architecture, labeled (a), (b), and (c). Each configuration consists of an encoder and a decoder, both of which are symmetric in structure.

Encoder: The encoder processes input images through a series of layers. In all three cases, it starts with an *InputLayer* followed by a *Conv2D* layer. Subsequent layers depend on the configuration:

- (a) One Layer:** Followed by a *ReLU* layer, a *MaxPooling2D* layer, another *Conv2DTranspose* layer, another *ReLU* layer, an *UpSampling2D* layer, and a final *Conv2D* layer.
- (b) Two Layers:** Followed by two *ReLU* layers, two *MaxPooling2D* layers, two *Conv2DTranspose* layers, two *ReLU* layers, two *UpSampling2D* layers, and a final *Conv2D* layer.
- (c) Three Layers:** Followed by three *ReLU* layers, three *MaxPooling2D* layers, three *Conv2DTranspose* layers, three *ReLU* layers, three *UpSampling2D* layers, and a final *Conv2D* layer.

Decoder: The decoder reconstructs the image through a series of layers that mirror the encoder's structure:

- (a) One Layer:** Followed by a *Conv2D* layer, a *ReLU* layer, a *MaxPooling2D* layer, another *Conv2DTranspose* layer, another *ReLU* layer, an *UpSampling2D* layer, and a final *Conv2D* layer.
- (b) Two Layers:** Followed by two *Conv2D* layers, two *ReLU* layers, two *MaxPooling2D* layers, two *Conv2DTranspose* layers, two *ReLU* layers, two *UpSampling2D* layers, and a final *Conv2D* layer.
- (c) Three Layers:** Followed by three *Conv2D* layers, three *ReLU* layers, three *MaxPooling2D* layers, three *Conv2DTranspose* layers, three *ReLU* layers, three *UpSampling2D* layers, and a final *Conv2D* layer.

Tables: Below each diagram are tables summarizing the layers, their types, output shapes, and the number of parameters (Param #).

Layer (type)	Output Shape	Param #
input_layer_17 (<i>InputLayer</i>)	(None, 128, 128, 3)	0
conv2d_30 (<i>Conv2D</i>)	(None, 128, 128, 32)	896
re_lu_27 (<i>ReLU</i>)	(None, 128, 128, 32)	0
max_pooling2d_13 (<i>MaxPooling2D</i>)	(None, 64, 64, 32)	0
conv2d_transpose_13 (<i>Conv2DTranspose</i>)	(None, 64, 64, 32)	9,248
re_lu_28 (<i>ReLU</i>)	(None, 64, 64, 32)	0
up_sampling2d_13 (<i>UpSampling2D</i>)	(None, 128, 128, 32)	0
conv2d_31 (<i>Conv2D</i>)	(None, 128, 128, 3)	867

Layer (type)	Output Shape	Param #
input_layer (<i>InputLayer</i>)	(None, 128, 128, 3)	0
conv2d (<i>Conv2D</i>)	(None, 128, 128, 8)	224
re_lu (<i>ReLU</i>)	(None, 128, 128, 8)	0
max_pooling2d (<i>MaxPooling2D</i>)	(None, 64, 64, 8)	0
conv2d_1 (<i>Conv2D</i>)	(None, 64, 64, 16)	1,168
re_lu_1 (<i>ReLU</i>)	(None, 64, 64, 16)	0
max_pooling2d_1 (<i>MaxPooling2D</i>)	(None, 32, 32, 16)	0
conv2d_2 (<i>Conv2D</i>)	(None, 32, 32, 32)	4,640
re_lu_2 (<i>ReLU</i>)	(None, 32, 32, 32)	0
max_pooling2d_2 (<i>MaxPooling2D</i>)	(None, 16, 16, 32)	0
conv2d_transpose (<i>Conv2DTranspose</i>)	(None, 16, 16, 32)	9,248
re_lu_3 (<i>ReLU</i>)	(None, 16, 16, 32)	0
up_sampling2d (<i>UpSampling2D</i>)	(None, 32, 32, 32)	0
conv2d_transpose_1 (<i>Conv2DTranspose</i>)	(None, 32, 32, 16)	4,624
re_lu_4 (<i>ReLU</i>)	(None, 32, 32, 16)	0
up_sampling2d_1 (<i>UpSampling2D</i>)	(None, 64, 64, 16)	0
conv2d_transpose_2 (<i>Conv2DTranspose</i>)	(None, 64, 64, 32)	4,640
re_lu_5 (<i>ReLU</i>)	(None, 64, 64, 32)	0
up_sampling2d_2 (<i>UpSampling2D</i>)	(None, 128, 128, 32)	0
conv2d_3 (<i>Conv2D</i>)	(None, 128, 128, 3)	867

Layer (type)	Output Shape	Param #
input_layer (<i>InputLayer</i>)	(None, 128, 128, 3)	0
conv2d (<i>Conv2D</i>)	(None, 128, 128, 16)	448
re_lu (<i>ReLU</i>)	(None, 128, 128, 16)	0
max_pooling2d (<i>MaxPooling2D</i>)	(None, 64, 64, 16)	0
conv2d_1 (<i>Conv2D</i>)	(None, 64, 64, 32)	4,640
re_lu_1 (<i>ReLU</i>)	(None, 64, 64, 32)	0
max_pooling2d_1 (<i>MaxPooling2D</i>)	(None, 32, 32, 32)	0
conv2d_transpose (<i>Conv2DTranspose</i>)	(None, 32, 32, 32)	9,248
re_lu_2 (<i>ReLU</i>)	(None, 32, 32, 32)	0
up_sampling2d (<i>UpSampling2D</i>)	(None, 64, 64, 32)	0
conv2d_transpose_1 (<i>Conv2DTranspose</i>)	(None, 64, 64, 16)	4,624
re_lu_3 (<i>ReLU</i>)	(None, 64, 64, 16)	0
up_sampling2d_1 (<i>UpSampling2D</i>)	(None, 128, 128, 16)	0
conv2d_transpose_2 (<i>Conv2DTranspose</i>)	(None, 128, 128, 3)	435

Gambar 3.11 CAE satu layer (a), dua layer (b), dan tiga layer (c)

3.5.2.2.2 Convolutional Variational Autoencoder (CVAE)

CVAE dikembangkan berdasarkan artikel Tensorflow yang mendemonstrasikan pelatihan model VAE yang telah dimodifikasi dengan menambahkan *convolutional layers* agar dapat mendukung *dataset* MNIST (Tensorflow, 2024). Arsitektur dasar CVAE yang mirip dengan CAE namun dengan dua *layer* tambahan krusial. Pertama adalah *Sampling layer* (Gambar 3.12), yang mengimplementasikan *Reparameterization Trick*. *Layer* ini memungkinkan pengambilan sampel dari distribusi laten yang dipelajari *encoder* dengan cara yang dapat di-*backpropagate*, memisahkan komponen acak dari parameter yang dipelajari untuk memungkinkan perhitungan gradien yang efektif.

```
● ● ●
@keras.saving.register_keras_serializable()
class Sampling(tf.keras.layers.Layer):
    def call(self, inputs):
        z_mean, z_log_var = inputs
        batch = tf.shape(z_mean)[0]
        dim = tf.shape(z_mean)[1]
        epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
        return z_mean + tf.exp(0.5 * z_log_var) * epsilon
```

Gambar 3. 12 *Sampling Layer*

Kedua adalah *KL Divergence Layer*, sebuah *layer* kustom yang menghitung dan menambahkan *KL Divergence Loss* ke model. *Layer* ini berfungsi sebagai regularisasi, memastikan bahwa distribusi laten yang dipelajari oleh *encoder* tetap mirip dengan distribusi normal standar, sehingga memaksa ruang laten menjadi lebih halus dan terstruktur. Ini krusial untuk kemampuan CVAE dalam menghasilkan data baru yang bermakna.

```
● ● ●
@keras.saving.register_keras_serializable()
class KLDivergenceLayer(tf.keras.layers.Layer):
    def __init__(self, input_pixels, **kwargs):
        super().__init__(**kwargs)
        self.input_pixels = tf.cast(input_pixels, tf.float32)

    def call(self, inputs):
        z_mean, z_log_var = inputs
        kl_loss_per_sample = -0.5 * tf.reduce_sum(z_log_var - tf.square(z_mean) - tf.exp(tf.exp(z_log_var)) + 1, axis=1) # Corrected exp(z_log_var)
        kl_loss_normalized = kl_loss_per_sample / self.input_pixels
        self.add_loss(tf.reduce_mean(kl_loss_normalized))
        return inputs
```

Gambar 3.13 *KL Divergence Layer*

Dalam implementasinya, model dibangun menggunakan *Functional API* Keras. Dimulai dengan *encoder*, ia menerima citra input berukuran 128x128x3 dan secara berurutan melewati tiga *layer Conv2D*. Konfigurasi ini memungkinkan fleksibilitas dalam kedalaman model: untuk satu *layer*, hanya filter 32 yang digunakan; untuk dua *layer*, filter 32 dan 64 akan aktif; sedangkan untuk tiga *layer*, ketiga filter 32, 64, dan 128 digunakan secara berurutan. Output konvolusional ini kemudian di-*flatten* dan diumpulkan ke dua *layer dense* terpisah untuk menghasilkan *z_mean* dan *z_log_var*, yang merupakan parameter distribusi *Gaussian* di ruang laten. *Sampling layer* kemudian mengambil *z_mean* dan *z_log_var* ini untuk menghasilkan sampel *z*, yang menjadi representasi laten.

Secara bersamaan, *KL DivergenceLayer* menghitung *KL Divergence Loss* dari *z_mean* dan *z_log_var* dan menambahkannya ke total *loss* model. Selanjutnya, *Decoder* mengambil sampel *z* dari ruang laten, mengembangkannya kembali menggunakan *layer dense* dan *reshape* ke dimensi spasial dan jumlah filter yang cocok dengan *bottleneck encoder* (16x16x128). Kemudian, tiga *layer Conv2DTranspose* secara bertahap memperbesar dimensi citra kembali ke 128x128x3, dan aktivasi *sigmoid* pada layer output terakhir memastikan nilai piksel berada dalam rentang 0 hingga 1.

```
● ● ●
def build_vae(input_shape=(128, 128, 3), latent_dim=8):
    print(f"\n--- Building VAE with input_shape: {input_shape} and latent_dim: {latent_dim} ---")

    # --- Encoder ---
    encoder_inputs = tf.keras.Input(shape=input_shape, name="encoder_input")
    x = tf.keras.layers.Conv2D(filters=2, kernel_size=3, strides=(2, 2), activation='relu', padding='same', name="enc_conv1")(encoder_inputs) #satu layer
    x = tf.keras.layers.Conv2D(filters=4, kernel_size=3, strides=(2, 2), activation='relu', padding='same', name="enc_conv2")(x) #dua layer
    x = tf.keras.layers.Conv2D(filters=128, kernel_size=3, strides=(2, 2), activation='relu', padding='same', name="enc_conv3")(x) #tiga layer

    output_spatial_dim = input_shape[0] // (2**3)
    filters_at_bottleneck = 128

    x = tf.keras.layers.Flatten(name="enc_flatten")(x)
    z_mean = tf.keras.layers.Dense(latent_dim, name="z_mean")(x)
    z_log_var = tf.keras.layers.Dense(latent_dim, name="z_log_var")(x)

    z = Sampling(name="z_sampling")([z_mean, z_log_var])

    input_pixels = input_shape[0] * input_shape[1] * input_shape[2]
    _ = KLDivergenceLayer(input_pixels=input_pixels, name="kl_divergence_loss")([z_mean, z_log_var])

    encoder = tf.keras.Model(encoder_inputs, [z_mean, z_log_var, z], name="encoder")

    # --- Decoder ---
    latent_inputs = tf.keras.Input(shape=(latent_dim,), name="decoder_input")

    initial_dense_units = output_spatial_dim * output_spatial_dim * filters_at_bottleneck

    x = tf.keras.layers.Dense(units=initial_dense_units, activation=tf.nn.relu, name="dec_dense")(latent_inputs)
    x = tf.keras.layers.Reshape(target_shape=(output_spatial_dim, output_spatial_dim, filters_at_bottleneck))(x)

    x = tf.keras.layers.Conv2DTranspose(filters=64, kernel_size=3, strides=2, padding='same', activation='relu', name="dec_conv1")(x) #satu layer
    x = tf.keras.layers.Conv2DTranspose(filters=32, kernel_size=3, strides=2, padding='same', activation='relu', name="dec_conv2")(x) #dua layer
    x = tf.keras.layers.Conv2DTranspose(filters=32, kernel_size=3, strides=2, padding='same', activation='relu', name="dec_conv3")(x) #tiga layer

    decoder_outputs = tf.keras.layers.Conv2DTranspose(filters=input_shape[-1], kernel_size=3, strides=1, padding='same', activation='sigmoid', name="decoder_output")(x)
    decoder = tf.keras.Model(latent_inputs, decoder_outputs, name="decoder")

    outputs = decoder(encoder(encoder_inputs)[2])
    vae = tf.keras.Model(encoder_inputs, outputs, name="vae")

    return vae, encoder, decoder
```

Gambar 3.14 CVAE satu sampai tiga *layer*

Setelah kedua model dibuat, model akan dilatih dengan tiga variasi nilai *learning rate* (0,01; 0,001; dan 0,0005) serta tiga jumlah *epochs* yang berbeda (50, 100, dan 150). Pemilihan skenario pelatihan ini didasarkan pada prinsip *hyperparameter tuning* untuk menemukan konfigurasi model yang paling optimal. *Learning rate* merupakan parameter krusial yang menentukan seberapa cepat model menyesuaikan bobotnya saat belajar dari data. Pengujian dengan nilai bervariasi (0,01; 0,001; dan 0,0005) bertujuan untuk menemukan laju pembelajaran yang ideal: nilai yang terlalu tinggi dapat menyebabkan model melewati titik optimal, sementara nilai yang terlalu rendah dapat membuat proses pelatihan menjadi sangat lambat. Di sisi lain, jumlah *epochs* menentukan seberapa banyak kali model melihat seluruh dataset pelatihan. Pengujian dengan variasi 50, 100, dan 150 epochs dilakukan untuk mengamati keseimbangan antara *underfitting* (model belum cukup belajar) dan *overfitting* (model terlalu menghafal data latihan). Oleh karena itu, pengujian kombinasi dari kedua parameter ini secara sistematis memungkinkan peneliti untuk mengidentifikasi titik optimal di mana model dapat mencapai akurasi tertinggi tanpa mengorbankan efisiensi waktu atau kemampuan generalisasinya pada data baru.

Proses pelatihan model dilakukan dengan memanfaatkan platform *Google Colaboratory*, dan seluruh hasil pelatihan disimpan secara otomatis pada layanan penyimpanan *Google Drive*. *Dataset* yang digunakan dalam tahap ini adalah data healthy yang berasal dari citra pohon kelapa sawit pada bulan Juni 2023 dan dilakukan menggunakan algoritma optimasi *Adam* dengan pembagian data sebesar 80% untuk pelatihan, 20% untuk validasi. Dalam proses deteksi anomali, kedua model memanfaatkan perbandingan antara citra asli dan citra hasil rekonstruksi dengan mengukur nilai MAE, MSE dan SSIM. Nilai-nilai tersebut kemudian dianalisis dan dibandingkan distribusi datanya antar model guna mengevaluasi efektivitas masing-masing dalam mengidentifikasi pohon kelapa sawit yang berpotensi terinfeksi *Ganoderma boninense* menggunakan dua metrik: *Jensen-Shannon Distance* (JSD) dan *Bhattacharyya Distance* (BD).

3.5.3 Evaluasi Kinerja

Pada tahap ini, dilakukan evaluasi terhadap kinerja dari tiga model yang dikembangkan, yaitu YOLO sebagai model deteksi objek dan 2 Variasi AE sebagai model deteksi anomali visual pada pohon kelapa sawit. Evaluasi dilakukan guna mengukur sejauh mana efektivitas dan akurasi dari masing-masing model dalam menjalankan tugasnya. Selain itu, dilakukan juga evaluasi terhadap kemampuan deteksi dini (*Limit Detection*) dari model CAE dan CVAE, guna mengetahui seberapa jauh model mampu mengenali gejala infeksi *Ganoderma* sebelum tanda visual terlihat secara nyata. Metriks evaluasi yang digunakan disesuaikan dengan karakteristik keluaran dari masing-masing model.

3.5.3.1 Model YOLO

Model YOLO dievaluasi dengan menggunakan tiga metrik, yaitu:

- *Precision*, mengukur sejauh mana model dapat mengidentifikasi pohon kelapa sawit dengan benar tanpa menghasilkan terlalu banyak deteksi palsu.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive + False\ Positive\ (FP)} \quad (3-1)$$

- *Recall*, menunjukkan kemampuan model dalam menangkap semua pohon kelapa sawit yang benar-benar ada dalam citra.

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive + False\ Negative\ (FN)} \quad (3-2)$$

- *F1-Score*, merupakan rata-rata harmonis dari precision dan recall, yang memberikan gambaran menyeluruh terhadap keseimbangan antara keduanya.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3-3)$$

3.5.3.2 Dataset Autoencoder

Evaluasi model CAE dan CVAE dilakukan dengan dua metrik, yaitu MSE dan SSIM.

- MAE mengukur rata-rata kesalahan rekonstruksi absolut antara citra input dan hasil rekonstruksi model. Nilai MAE yang rendah menandakan bahwa

citra direkonstruksi dengan baik, sedangkan nilai tinggi mengindikasikan potensi anomali.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}|$$

- MSE mengukur rata-rata kesalahan rekonstruksi antara citra input dan hasil rekonstruksi model. Nilai MSE yang rendah menandakan bahwa citra direkonstruksi dengan baik, sedangkan nilai tinggi mengindikasikan potensi anomali.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y})^2 \quad (3-5)$$

- SSIM mengevaluasi kesamaan struktural antara dua citra, memberikan gambaran yang lebih manusiawi terhadap perbedaan visual yang penting. Nilai SSIM berkisar antara -1 hingga 1, dengan nilai mendekati 1 menunjukkan kemiripan yang tinggi.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3-6)$$

3.5.3.3 Limit Detection

Evaluasi *limit detection* bertujuan untuk menguji kemampuan model CAE dalam mendeteksi infeksi *Ganoderma* pada tahap awal, sebelum gejala visual muncul secara jelas. Untuk itu, digunakan *dataset timeseries* selama lima bulan (Januari–Mei 2023) yang memuat citra pohon sawit sebelum tumbang di bulan Juni 2023.

Untuk menilai batas kemampuan deteksi model dalam membedakan citra sawit sehat dan sakit, penulis akan menganalisis perbedaan distribusi metrik MAE, MSE dan SSIM menggunakan dua metrik: *Jensen-Shannon Distance* (JSD) dan *Bhattacharyya Distance* (BD). JSD, dengan rentang 0-1, akan menunjukkan adanya perbedaan signifikan jika nilainya mendekati 1. Sementara itu, nilai BD di atas 1

akan menjadi indikator pemisahan data yang sangat baik, karena menunjukkan tingkat tumpang tindih distribusi yang minimal.

Dengan pendekatan ini, dapat diidentifikasi batas waktu terawal (*limit detection*) di mana model pertama kali mengenali perubahan visual sebagai indikasi infeksi. Hasil evaluasi ini sangat penting dalam konteks penerapan sistem deteksi dini, karena menggambarkan sejauh mana model dapat membantu pengambilan keputusan sebelum infeksi menyebar lebih luas.



UNIVERSITAS
MA CHUNG

BAB IV

HASIL DAN PEMBAHASAN

4.1 *Dataset*

Dalam penelitian ini, penulis menggunakan serangkaian *dataset* yang dikumpulkan dan disiapkan khusus untuk melatih dan menguji model-model *deep learning* dengan peruntukannya masing-masing. Berikut rincian komposisi dari masing-masing *dataset*:

1. Dataset Pertama (YOLO): Terdiri dari 10 potongan gambar ROI dengan jumlah total 1767 gambar pohon kelapa sawit. Dataset ini secara spesifik dialokasikan untuk melatih model deteksi YOLO. Tujuannya adalah agar model YOLO mampu secara akurat mengenali dan mendeteksi lokasi pohon kelapa sawit dalam citra.
2. Dataset Kedua (AE): Berjumlah 1722 gambar pohon sehat. Dataset ini merupakan fokus utama untuk melatih dan menguji model CAE dan CVAE. Tujuan utamanya adalah untuk memungkinkan model-model ini mendeteksi secara dini adanya anomali yang mengindikasikan infeksi infeksi Ganoderma pada pohon kelapa sawit, dengan membedakan pola gambar pohon sehat dari pola gambar pohon yang terinfeksi.
3. Dataset Ketiga (*Timeseries*): Dataset ini mencakup 205 titik data, yang merupakan akumulasi pengamatan selama lima bulan sebelum pohon kelapa sawit tumbang, dengan masing-masing 41 titik data per bulan. Dataset ini digunakan untuk menguji kemampuan deteksi dini model *autoencoder* mulai dari satu bulan hingga lima bulan sebelum tumbang.

Table 4.1 Rincian *dataset*

Dataset	Jumlah Gambar
YOLO	1767
AE	1722
Time-series	205

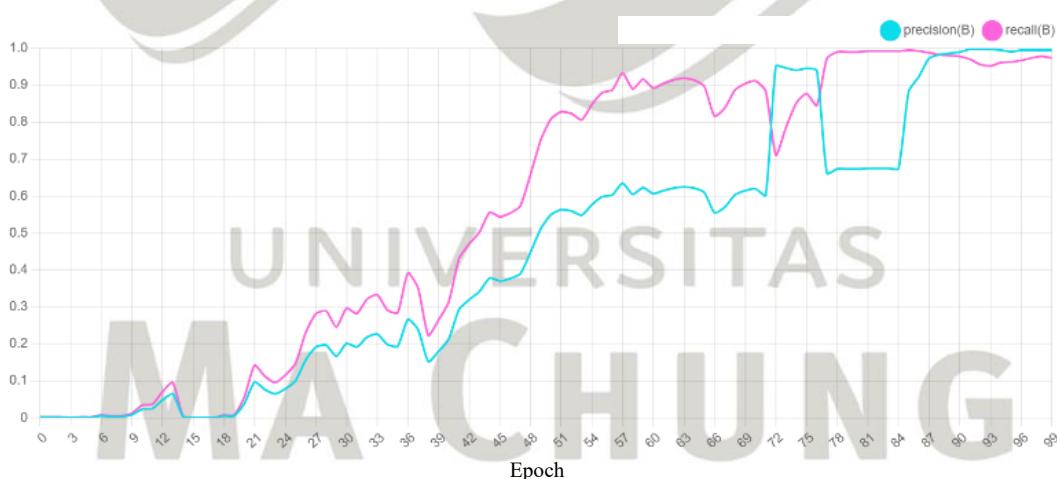
4.2 **Pelatihan Model**

Model penelitian ini terdiri dari dua komponen utama: satu arsitektur YOLO untuk deteksi objek dan dua variasi arsitektur CAE dan CVAE untuk deteksi

anomali. Proses pelatihan model YOLO menggunakan 9 potongan gambar ROI yang dibagi menjadi 80% untuk pelatihan, 20% untuk validasi. Sementara itu, untuk pelatihan CAE dan CVAE, penulis memanfaatkan 1.722 gambar pohon sawit sehat, di mana 80% digunakan sebagai data pelatihan dan 20% sebagai data validasi.

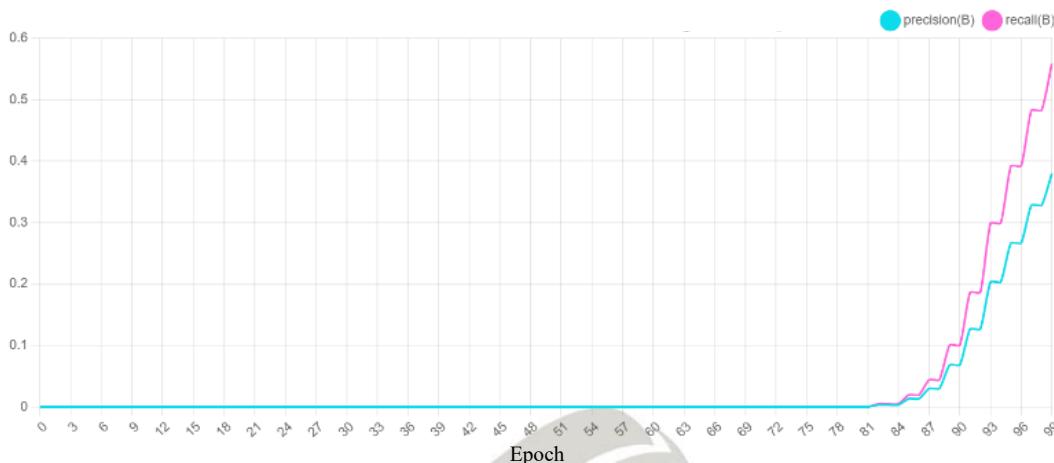
4.2.1 Pelatihan Model YOLO

Berdasarkan hasil pelatihan model YOLO, penggunaan model *pretrain* menunjukkan performa yang jauh lebih baik dibandingkan dengan pelatihan tanpa *pretrain*. Hal ini terbukti pada Gambar 4.1 yang memvisualisasikan proses pembelajaran model. Pada awal pelatihan (*epoch* 0-8) nilai *precision* dan *recall* menunjukkan model masih dalam tahap adaptasi. Kinerja model sempat membaik pada *epoch* 9-13, namun kemudian menurun kembali pada *epoch* ke-14 yang mungkin menandakan model sementara kehilangan bobot optimal akibat penyesuaian bobot. Setelah itu, nilai *precision* dan *recall* kembali naik secara signifikan dengan fluktuasi pada *epoch* 19-99.



Gambar 4.1 Grafik performa model *pretrain* pada 100 *epochs*

Sebaliknya, model yang dilatih tanpa *pretrain* mengalami kesulitan mencapai performa yang baik pada jumlah *epochs* yang sama. Seperti yang ditunjukkan pada Gambar 4.2, model ini memerlukan waktu yang jauh lebih lama untuk belajar. Model tidak menunjukkan adanya perkembangan atau stagnan hingga *epoch* 80 dan baru mulai menunjukkan perbaikan atau perkembangan pada *epoch* 81-99.



Gambar 4.2 Grafik performa model tanpa *pretrain* pada 100 *epochs*

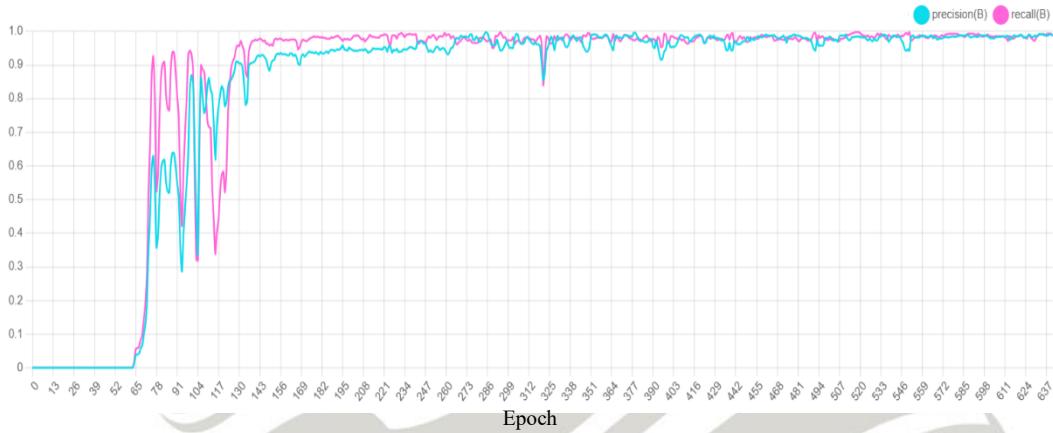
Berdasarkan hasil keduanya pada 100 *epochs* (Tabel 4.2), model dengan *pretrain* mendapatkan nilai *precision*, *recall*, dan F1 *score* yang sangat tinggi dan stabil pada data pelatihan dan validasi, dengan F1 *score* di atas 0.97. Hal ini menunjukkan kemampuan deteksi yang akurat, dengan kesalahan prediksi yang sangat minim. Sementara itu, model tanpa *pretrain* mendapat nilai F1 *score* pada data validasi hanya 0.45, dengan *precision* dan *recall* yang relatif rendah, menunjukkan bahwa model sering salah mendekripsi objek atau melewatkannya sebenarnya.

Table 4.2 Hasil pelatihan model YOLO pada 100 *epochs*

Pretrain	Epochs	Train			Validation		
		P	R	F1s	P	R	F1s
Yes	100	0.9777	0.9640	0.9078	0.9950	0.9740	0.9844
No		0.4624	0.6293	0.5331	0.3800	0.5588	0.4524

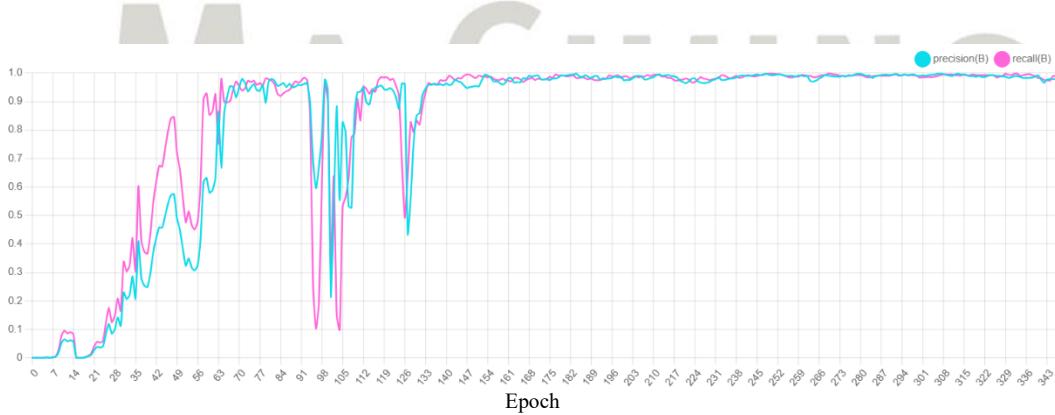
Untuk mengatasi masalah tersebut, penulis mencoba meningkatkan performa model tanpa *pretrain* dengan menaikkan jumlah *epoch* menjadi 1000 untuk kedua model. Tujuannya adalah memberi kesempatan bagi model tanpa *pretrain* untuk belajar dari awal dan menyesuaikan bobot sepenuhnya dengan karakteristik *dataset* kelapa sawit tanpa dipengaruhi pola dari *dataset* umum.

Berkat *early stopping* dengan *patience* 100, model tanpa *pretrain* berhenti di *epoch* ke-641 (Gambar 4.3). Model tanpa *pretrain* menunjukkan perkembangan yang lambat diawal dengan tidak adanya perkembangan signifikan pada *epoch* 0-63. Namun pada *epoch* 65-76 terjadi konvergensi yang diikuti oleh fluktuasi panjang hingga *epoch* 553, sebelum akhirnya berhenti



Gambar 4.3 Grafik performa model tanpa *pretrain* pada 1000 *epochs*

Sebaliknya, model *pretrain* berhenti jauh lebih cepat yaitu pada *epoch* 343 (Gambar 4.4), hampir separuh waktu pelatihan model tanpa *pretrain*. Menariknya, meskipun model ini tampak tidak stabil diawal dengan fluktuasi acak hingga *epoch* 134, model berhasil mencapai nilai *precision* dan *recall* yang tinggi pada *epoch* 92 dan 118. Setelah fluktuasi tersebut, model mulai stabil dan mempertahankan kinerja optimalnya hingga *epoch* ke-343.



Gambar 4.4 Grafik performa model *pretrain* pada 1000 *epochs*

Kemudian dengan naiknya *epoch*, model tanpa *pretrain* berhasil mencapai F1 Score sebesar 0.9864 pada data validasi, berhasil melampaui F1 Score model

pretrain yang dilatih pada 100 *epochs* (0.9844). Namun demikian, peningkatan jumlah *epochs* hingga 10 kali lipat pada model tanpa *pretrain* menunjukkan kebutuhan waktu pelatihan yang jauh lebih lama dan sumber daya komputasi yang lebih besar. Sementara itu, model *pretrain* hanya mengalami peningkatan performa yang relatif kecil saat *epoch* dinaikkan 10 kali lipat, mengindikasikan bahwa bobot *pretrain* sudah sangat optimal bahkan pada jumlah *epoch* yang lebih kecil.

Tabel 4.3 Perbandingan hasil pelatihan model YOLO pada 100 dan 1000 *epochs*

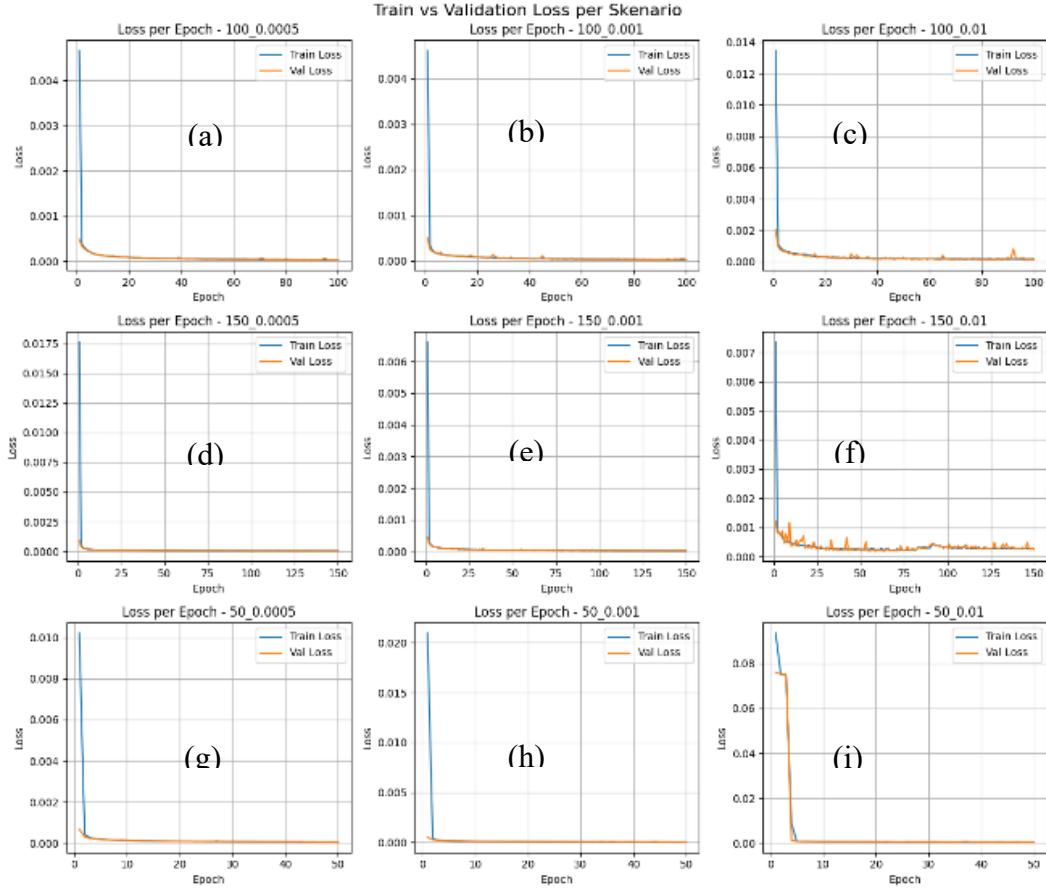
Pretrain	Epochs	Train			Validation		
		P	R	F1s	P	R	F1s
Yes	100	0.9777	0.9640	0.9078	0.9950	0.9740	0.9844
No	100	0.4624	0.6293	0.5331	0.3800	0.5588	0.4524
Yes	1000	0.9768	0.9827	0.9798	0.9947	0.9926	0.9937
No	1000	0.9692	0.9870	0.9781	0.9874	0.9853	0.9864

Kesimpulannya, meskipun model *pretrain* menunjukkan fluktuasi yang besar diawal, model tetap berhasil mencapai bobot optimal jauh lebih cepat dibandingkan model tanpa *pretrain* yang membutuhkan waktu pelatihan dan *epoch* yang lebih banyak untuk belajar dan stabil. Perbedaan ini menegaskan bahwa dengan *pretrain* dapat membantu model memanfaatkan pengetahuan sebelumnya untuk mengenali fitur penting pada citra kelapa sawit, sehingga proses pelatihan lebih cepat konvergen, lebih akurat dan stabil. Selain itu, penggunaan *pretrained* secara signifikan mempersingkat waktu pengembangan model dan memaksimalkan hasil deteksi meskipun dengan keterbatasan sumber daya pelatihan.

4.2.2 Pelatihan Model *Convolutional Autoencoder* (CAE)

Untuk arsitektur CAE, hasil eksperimen dengan satu *layer*, menunjukkan penurunan *loss* dan *validation loss* yang sangat cepat di *epoch* 0-20, mencapai konvergensi dengan baik dalam waktu singkat. Penurunan ini konsisten di berbagai konfigurasi *epoch* dan *learning rate*. Misalnya, dengan 100 *epochs* dan *learning rate* 0.0005, *loss* turun dari 0.0046 menjadi 0.00014 hanya dalam 8 *epochs* (*epoch* 0-7). Meskipun konvergensi terjadi dengan sangat cepat, penambahan *epoch* lebih

dari 50 *epochs* tidak memberikan peningkatan yang signifikan karena model telah mencapai performal optimalnya.



Gambar 4.5 Grafik performa model CAE satu *layer* dengan konfigurasi (a) 100 Epoch dan Learning Rate 0.0005, (b) 100 Epoch dan Learning Rate 0.001, (c) 100 Epoch dan Learning Rate 0.01, (d) 150 Epoch dan Learning Rate 0.0005, (e) 150 Epoch dan Learning Rate 0.001, (f) 150 Epoch dan Learning Rate 0.01, (g) 50 Epoch dan Learning Rate 0.0005, (h) 50 Epoch dan Learning Rate 0.001, (i) 50 Epoch dan Learning Rate 0.01.

Secara umum, model satu *layer* menunjukkan performa yang sangat stabil dan akurat. *validation loss* terbaik tercatat sebesar 0.000026 pada *learning rate* 0.0005 dengan 150 *epochs*, yang membuktikan kemampuan rekonstruksi yang luar biasa. Semakin kecil *learning rate* yang digunakan, semakin rendah pula *validation loss* yang dicapai. Namun, *learning rate* yang terlalu besar (seperti 0.01) menyebabkan performa menurun secara signifikan dan tidak stabil, yang

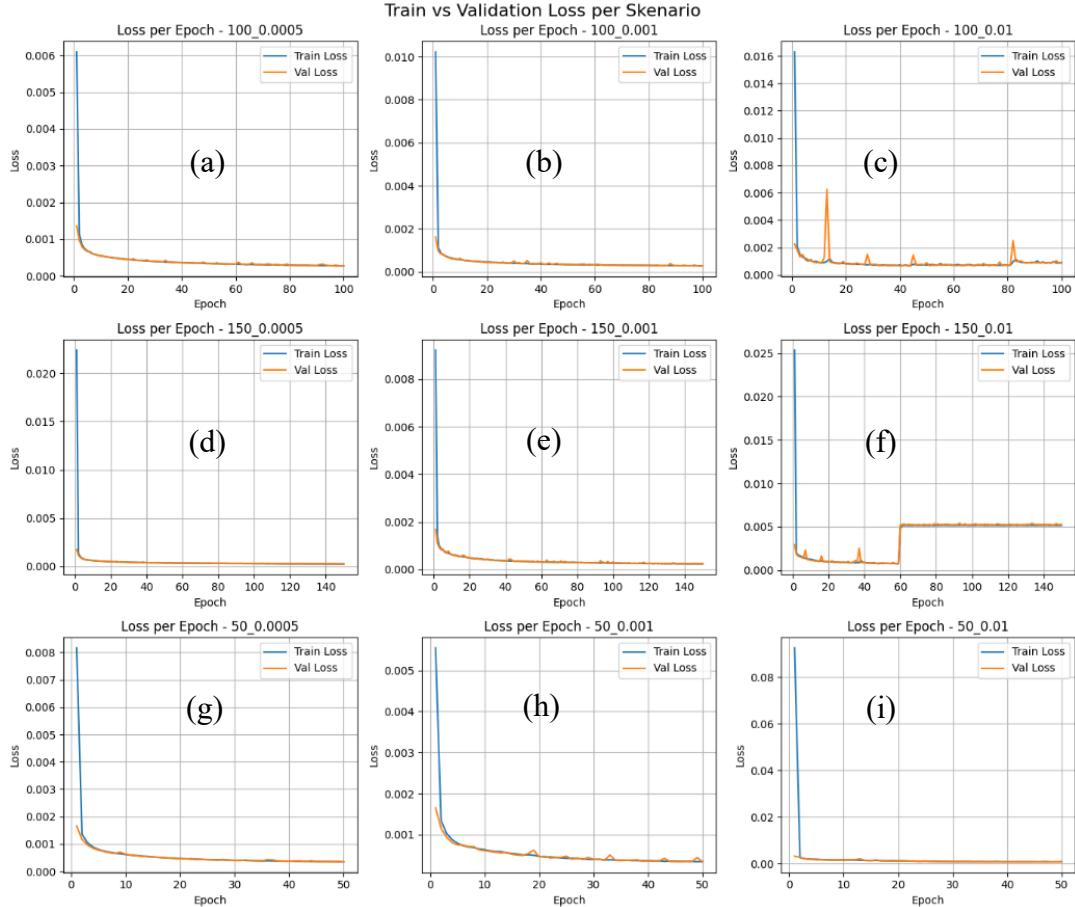
ditunjukkan oleh *validation loss* yang lebih tinggi (sekitar 0.000501 - 0.000197). Hal ini mengindikasikan bahwa model satu *layer* sangat sensitif terhadap *learning rate*.

Selanjutnya, pada dua *layer*, hasil menunjukkan pola penurunan *loss* dan *validation loss* yang serupa dengan model satu *layer*, yaitu cepat di awal dan kemudian melambat. Meskipun penambahan *layer* kedua bertujuan untuk menangkap representasi data yang lebih kompleks, performa model ini tidak selalu lebih baik. Nilai *loss* dan *validation loss* akhir memang sedikit lebih rendah dibandingkan model satu *layer*, yang menunjukkan bahwa model ini mampu belajar fitur yang lebih akurat. Sebagai contoh, dengan 100 *epochs* dan *learning rate* 0.0005, nilai *loss* dan *validation loss* sudah berada di bawah 0.00056 pada *epoch* ke-8, dan penurunan terus berlanjut secara perlahan hingga paling rendah mencapai 0.00027.

Namun, model dua *layer* ini tampaknya lebih sensitif terhadap *learning rate*. Nilai *validation loss* terbaik (0.000234) tercapai pada kombinasi *learning rate* 0.001 dan 150 *epochs*, yang sedikit lebih tinggi dibandingkan model satu *layer*. Performa memburuk secara signifikan saat *learning rate* dinaikkan menjadi 0.01, dengan *validation loss* yang jauh lebih tinggi (0.000800 - 0.000712). Ini menunjukkan bahwa *learning rate* yang terlalu besar membuat model dua *layer* lebih sulit beradaptasi. Meskipun penambahan *epochs* membantu menurunkan *validation loss*, penurunannya tidak setajam pada model satu *layer*, menegaskan adanya peningkatan kompleksitas yang tidak selalu berbanding lurus dengan peningkatan performa optimal.

Terakhir, hasil eksperimen dengan tiga *layer* (Gambar 4.7) menunjukkan performa yang paling stabil dan nilai *loss* terendah dibandingkan model lain, dengan penurunan *loss* dan *validation loss* yang konsisten bahkan pada *epochs* yang lebih tinggi. Hal ini mengindikasikan kemampuannya untuk mengekstrak fitur yang lebih mendalam, yang menghasilkan rekonstruksi data paling akurat. Sebagai contoh, dengan 100 *epochs* dan *learning rate* 0.0005, nilai *loss* dan *validation loss* sudah mencapai 0.0018 dan 0.0017 pada *epoch* ke-7. Model ini juga menunjukkan kemampuan untuk terus memperbaiki diri di *epoch* selanjutnya dengan 0.0009 pada

loss dan *validation loss* pada *epoch* ke-99, menjadikannya pilihan ideal untuk tugas rekonstruksi yang lebih kompleks.



Gambar 4.6 Grafik performa model CAE dua *layer* dengan konfigurasi (a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

Namun, model ini tampak paling rentan terhadap *overfitting* atau kesulitan konvergensi, terutama pada *learning rate* yang tinggi. Saat *learning rate* diatur ke 0.01, *training loss* tidak mengalami perubahan di 0.257887 pada 50 *epochs*, dan *validation loss* yang stagnan. Bahkan dengan *learning rate* yang lebih kecil (0.0005) dan 150 *epochs*, *validation loss* terbaik hanya mencapai 0.000829, yang masih lebih tinggi dari arsitektur dengan *layer* lebih sedikit. Kompleksitas arsitektur tiga *layer* tampaknya tidak sebanding dengan rezim pelatihan atau jumlah data yang

ada, dan bahkan dengan *learning rate* kecil, peningkatannya terbatas. Hal ini menunjukkan bahwa tiga *layer* mungkin tidak optimal untuk tugas yang dihadapi model ini.

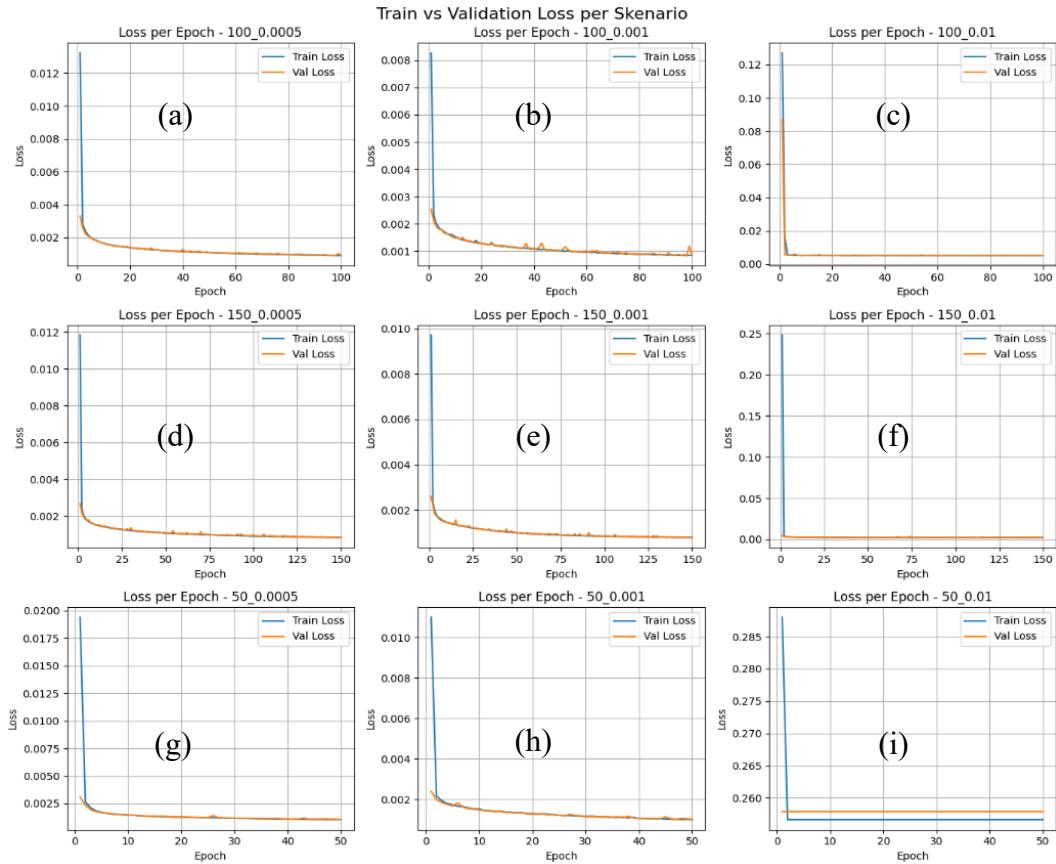
Secara keseluruhan, hasil eksperimen pada CAE dengan variasi satu, dua, dan tiga *layer*, dapat disimpulkan bahwa tingkat *learning rate* memiliki pengaruh yang paling signifikan terhadap konvergensi dan performa model. Secara konsisten, penggunaan *learning rate* yang terlalu tinggi (0.01) mengakibatkan model gagal konvergen atau menunjukkan *training loss* yang tidak stabil atau bahkan tidak ada peningkatan, sementara tingkat yang lebih rendah (0.001 dan 0.0005) terbukti jauh lebih efektif. Dari ketiga konfigurasi *layer*, *learning rate* 0.001 secara umum menunjukkan performa yang paling optimal dalam pelatihan model, dengan hasil *validation loss* terendah dan paling stabil di setiap eksperimen. Hal ini menunjukkan bahwa pemilihan *learning rate* yang tepat adalah kunci untuk mencapai kinerja terbaik, terlepas dari jumlah *layer* konvolusi yang digunakan.

4.2.3 Pelatihan Model *Convolutional Variational Autoencoder* (CVAE)

Eksperimen yang dilakukan pada arsitektur CVAE dengan konfigurasi satu *layer* menunjukkan sensitivitas tinggi terhadap pemilihan *learning rate*. Hasil pengujian secara jelas mengindikasikan bahwa nilai *learning rate* yang lebih rendah (0.0005 dan 0.001) secara signifikan lebih unggul dibandingkan nilai yang lebih besar (0.01). Secara konsisten, *learning rate* yang lebih kecil ini menghasilkan minimal *validation loss* yang jauh lebih rendah dan stabil, berada pada kisaran optimal 1.1483 hingga 1.1484. Kontrasnya, penggunaan *learning rate* 0.01 menyebabkan kinerja yang stagnan, di mana *validation loss* cenderung tertahan pada level yang lebih tinggi, yaitu sekitar 1.151 hingga 1.152.

Secara spesifik, konfigurasi dengan *learning rate* = 0.0005 yang dilatih selama 100 *epochs* terbukti menjadi titik optimal, mencatatkan nilai *validation loss* terendah sebesar 1.1472. Menariknya, ketika proses pelatihan diperpanjang hingga 150 *epochs*, terjadi fenomena peningkatan *validation loss* menjadi sekitar 1.152–1.153. Peningkatan ini, meskipun kecil, merupakan indikasi adanya gejala *overfitting* ringan. Hal ini dikonfirmasi oleh tren *training loss* yang terus menurun

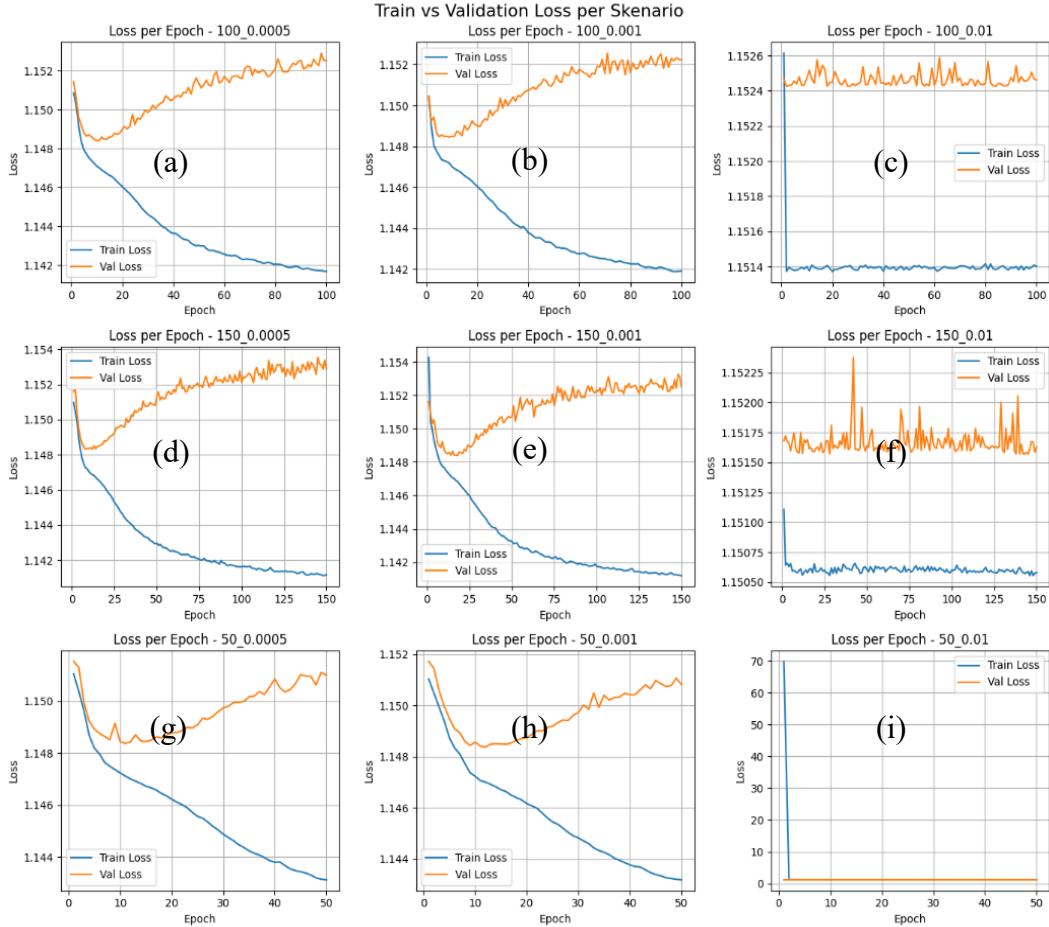
(mencapai minimum ≈ 1.1411), menunjukkan bahwa model terus menghafal data pelatihan namun mulai kehilangan kemampuan untuk menggeneralisasi secara efektif pada data validasi yang belum pernah dilihat sebelumnya.



Gambar 4.7 Grafik performa model CAE tiga *layer* dengan konfigurasi (a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

Adapun *learning rate* 0.01 menunjukkan hasil yang paling buruk. Bahkan setelah 150 *epochs*, model ini gagal menunjukkan perbaikan signifikan dan mencatat *validation loss* tertinggi (≈ 1.1524). Kinerja yang buruk ini menyiratkan bahwa nilai 0.01 terlalu besar, menyebabkan model gagal melakukan konvergensi yang efektif atau bahkan terperangkap dalam *local minima* yang *suboptimal* dan tinggi di permukaan energi (fungsi *loss*). Oleh karena itu, untuk arsitektur CVAE

ini, *learning rate* yang rendah sangat krusial untuk memastikan proses pembelajaran yang akurat dan terhindar dari perilaku yang tidak stabil.

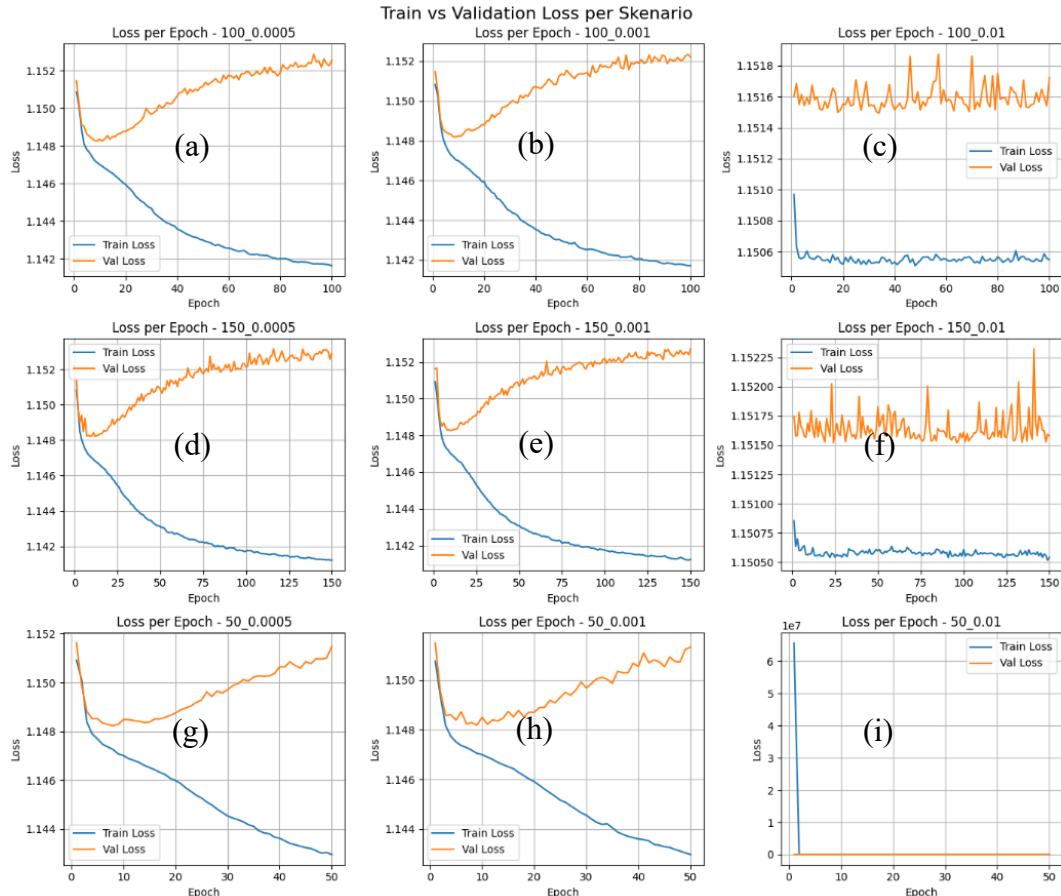


Gambar 4.8 Grafik performa model CVAE satu *layer* dengan konfigurasi (a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

Sedangkan pada dua *layer*, menguatkan temuan yang didapat dari model satu *layer*, yakni adanya ketergantungan kritis pada pemilihan *learning rate*. Kinerja terbaik dari model dua *layer* ini tercapai pada *learning rate* 0.0005. Pada konfigurasi ini, *validation loss* mencapai nilai terendah sekitar 1.1475 setelah 100 *epochs*, menandakan bahwa model telah mencapai konvergensi yang optimal. Peningkatan jumlah *epochs* menjadi 150 ternyata tidak menghasilkan perbaikan kinerja yang signifikan. Sebaliknya, hal ini justru memicu gejala *overfitting* ringan,

yang terlihat dari sedikit kenaikan pada final *validation loss* meskipun minimum *training loss* menunjukkan penurunan yang lebih substansial. Ini menggarisbawahi pentingnya *early stopping* untuk mencegah model menghafal data pelatihan.

Secara garis besar, model CVAE dua *layer* menunjukkan kinerja yang sedikit lebih unggul dibandingkan model satu *layer*. Hal ini terlihat dari minimum *validation loss* yang sedikit lebih rendah, berkisar antara 1.1481–1.1482. Konsisten dengan temuan sebelumnya, *learning rate* yang rendah (0.0005 dan 0.001) secara konsisten menghasilkan performa terbaik. Sebaliknya, penggunaan *learning rate*=0.01 kembali terbukti menjadi penghalang utama bagi proses pelatihan. Nilai yang terlalu tinggi ini menyebabkan model gagal berkonvergensi secara efektif, dengan *validation loss* yang tetap stagnan pada level tinggi di sekitar 1.1524. Hasil ini menegaskan bahwa, meskipun penambahan *layer* pada model dua *layer*



Gambar 4.9 Grafik performa model CVAE dua *layer* dengan konfigurasi (a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

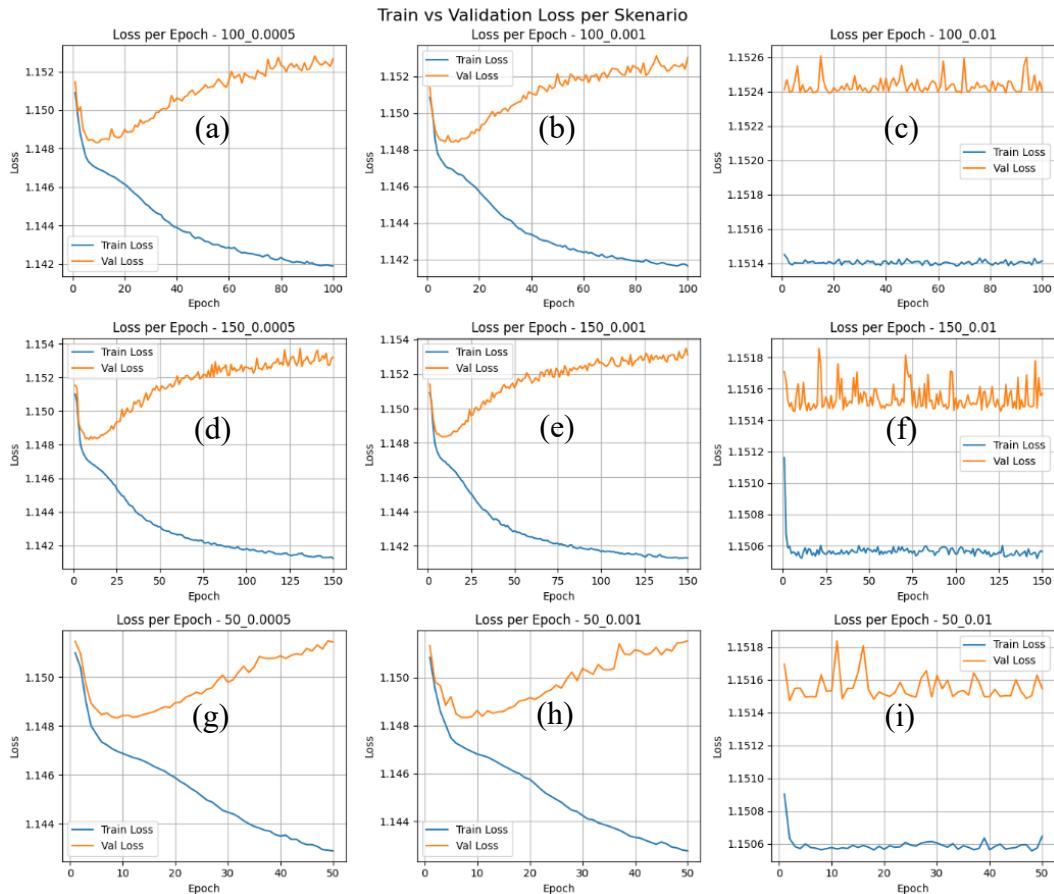
memberikan kapasitas representasional tambahan yang sedikit meningkatkan kemampuan generalisasi, model tersebut tetap sangat sensitif terhadap parameter *learning rate* dan rentan terhadap *overfitting* jika dilatih terlalu lama.

Terakhir pada tiga *layer*, tren perilaku model yang rentan terhadap *overfitting* semakin jelas terlihat. Meskipun peningkatan kapasitas model ini memungkinkan minimum *validation loss* mencapai nilai yang sangat rendah—berkisar antara 1.1483 hingga 1.1484, yang setara dengan kinerja model dua *layer*—*validation loss* akhir setelah pelatihan yang diperpanjang justru cenderung lebih tinggi. Gejala *overfitting* ini menjadi lebih nyata, terutama ketika dilatih menggunakan jumlah *epochs* yang lebih banyak dan *learning rate* yang kecil. *Learning rate* yang kecil memungkinkan model untuk menyesuaikan bobotnya secara *fine-tuning*, namun ketika dikombinasikan dengan kedalaman lapisan yang lebih besar, hal ini meningkatkan risiko penghafalan data pelatihan secara berlebihan.

Kinerja terbaik dari model tiga *layer* ini dicapai pada *learning rate* 0.0005, di mana *validation loss* terendah tercatat pada nilai 1.1470 setelah 100 *epochs*. Nilai ini merupakan *validation loss* terendah di antara semua konfigurasi lapisan yang diuji (satu, dua, dan tiga *layer*), mengindikasikan bahwa model tiga *layer* memiliki kapasitas representasional tertinggi untuk belajar fitur-fitur yang kompleks. Namun, pencapaian *loss minimum* yang sangat rendah ini disertai dengan peringatan: struktur tiga *layer* menuntut pengawasan *hyperparameter* yang lebih ketat. Karena kerentanan yang lebih tinggi terhadap *overfitting*, strategi seperti *early stopping* menjadi sangat krusial untuk mencegah degradasi kinerja generalisasi pada data yang tidak terlihat, yang mana ditunjukkan dengan kenaikan *validation loss* pada akhir pelatihan.

Seperti pola yang sudah terlihat pada model-model sebelumnya, *learning rate* 0.001 juga menunjukkan konvergensi yang baik dengan *validation loss* sedikit lebih tinggi (sekitar 1.1479). Namun, *learning rate* 0.01 kembali gagal berfungsi secara efektif, menyebabkan *validation loss* tetap tinggi dan tidak stabil (~1.1522). Penambahan *epochs* hingga 150 pada *learning rate* kecil juga tidak memberikan keuntungan signifikan dan justru menunjukkan tanda-tanda *overfitting*, di mana

final training loss turun drastis (~ 1.1412) namun *final validation loss* malah naik (~ 1.153). Ini menunjukkan bahwa arsitektur tiga *layer* terlalu kompleks sehingga model mulai "menghafal" data pelatihan, yang berdampak buruk pada performa validasi.



Gambar 4.10 Grafik performa model CVAE tiga *layer* dengan konfigurasi (a) 100 *Epoch* dan *Learning Rate* 0.0005, (b) 100 *Epoch* dan *Learning Rate* 0.001, (c) 100 *Epoch* dan *Learning Rate* 0.01, (d) 150 *Epoch* dan *Learning Rate* 0.0005, (e) 150 *Epoch* dan *Learning Rate* 0.001, (f) 150 *Epoch* dan *Learning Rate* 0.01, (g) 50 *Epoch* dan *Learning Rate* 0.0005, (h) 50 *Epoch* dan *Learning Rate* 0.001, (i) 50 *Epoch* dan *Learning Rate* 0.01.

Secara keseluruhan, hasil eksperimen pada CVAE menunjukkan pola yang konsisten di semua jumlah *layer* (satu, dua, dan tiga). Berbeda dengan CAE, variasi *learning rate* dan peningkatan *epochs* tidak secara signifikan memperbaiki *validation loss*, yang cenderung stagnan pada nilai tinggi di sekitar 1.15. Meskipun *training loss* menunjukkan sedikit penurunan, *validation loss* yang tidak bergeming mengindikasikan bahwa model CVAE dengan konfigurasi yang diuji kesulitan

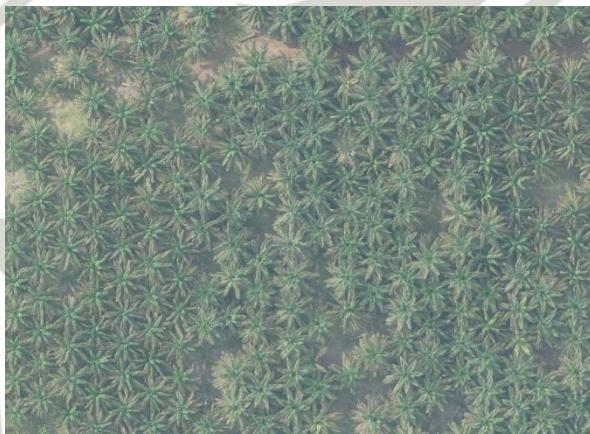
dalam mempelajari representasi yang efektif untuk generalisasi data, atau mungkin terjebak dalam *local minima*. Penambahan kedalaman lapisan juga tidak memberikan kontribusi positif yang signifikan terhadap kinerja validasi.

4.3 Evaluasi Model

Setelah pelatihan model YOLO, CAE dan CVAE selesai, setiap model akan dievaluasi dengan menggunakan data pengujian untuk mengukur kinerjanya sesuai dengan tujuan perancangannya masing-masing.

4.3.1 Evaluasi Model YOLO

Model YOLO akan diuji menggunakan data pengujian (10%) yang sebelumnya sudah dipisahkan dan belum pernah dilihat oleh model. Model dievaluasi menggunakan tiga metrik yakni, *precision*, *recall* dan *F1 Score* guna melihat seberapa baik model dalam mendekripsi objek pohon kelapa sawit dalam sebuah citra.



Gambar 4.11 Gambar data pengujian YOLO (10%)

Pada 100 *epochs*, model dengan *pretrain* mendapatkan nilai *precision*, *recall*, dan *F1 score* yang sangat tinggi diatas 0.98. Hal ini menunjukkan kemampuan deteksi yang akurat, dengan kesalahan prediksi yang sangat minim. Sementara itu, model tanpa *pretrain* mendapat nilai *F1 score* pada data uji hanya 0.59, dengan *precision* dan *recall* yang relatif rendah, menunjukkan bahwa model sering salah mendekripsi objek atau melewatkannya.

Table 4.4 Hasil pengujian model YOLO pada 100 epochs

Pretrain	Epochs	Train			Validation			Test		
		P	R	F1s	P	R	F1s	P	R	F1s
Yes	100	0.9777	0.9640	0.9078	0.9950	0.9740	0.9844	0.9870	0.9912	0.9891
No		0.4624	0.6293	0.5331	0.3800	0.5588	0.4524	0.5233	0.6856	0.5936

Kemudian dengan bertambahnya jumlah *epoch*, model tanpa *pretrain* berhasil mencapai F1 Score sebesar 0.9866 pada data pengujian, mendekati F1 Score model *pretrain* yang dilatih pada 1000 epochs (0.9885) maupun 100 epochs (0.9891). Namun demikian, peningkatan jumlah epochs hingga 10 kali lipat pada model tanpa *pretrain* menunjukkan kebutuhan waktu pelatihan yang jauh lebih lama dan sumber daya komputasi yang lebih besar. Sementara itu, model *pretrain* hanya mengalami peningkatan performa yang relatif kecil saat *epoch* dinaikkan 10 kali lipat, mengindikasikan bahwa bobot *pretrain* sudah sangat optimal bahkan pada jumlah *epoch* yang lebih kecil.

Table 4.5 Perbandingan hasil pengujian model YOLO pada 100 dan 1000 epochs

Pretrain	Epochs	Train			Validation			Test		
		P	R	F1s	P	R	F1s	P	R	F1s
Yes	100	0.9777	0.9640	0.9078	0.9950	0.9740	0.9844	0.9870	0.9912	0.9891
No		0.4624	0.6293	0.5331	0.3800	0.5588	0.4524	0.5233	0.6856	0.5936
Yes	1000	0.9768	0.9827	0.9798	0.9947	0.9926	0.9937	0.9858	0.9913	0.9885
No		0.9692	0.9870	0.9781	0.9874	0.9853	0.9864	0.9820	0.9913	0.9866

Dengan demikian, dapat disimpulkan bahwa penerapan *pretrained weight* pada YOLO sangat direkomendasikan karena mampu meningkatkan efisiensi pelatihan, mengurangi risiko *underfitting*, bahkan dengan data terbatas mampu mencapai performa yang tinggi dengan jumlah *epoch* yang lebih relatif sedikit.

4.3.2 Evaluasi Model *Convolutional Autoencoder* (CAE)

Model akan diuji pada 41 citra yang diambil satu bulan sebelum pohon kelapa sawit tumbang. Pengujian ini bertujuan melihat kemampuan model dalam mendeteksi anomali pada citra sakit. Evaluasi dilakukan menggunakan metrik MAE, MSE, dan SSIM. Nilai MAE dan MSE yang tinggi akan menunjukkan

kesulitan model dalam merekonstruksi citra sakit, sebab model hanya dilatih dengan citra sehat. Sementara itu, nilai SSIM yang rendah akan mengonfirmasi adanya perbedaan signifikan antara citra asli dan citra hasil rekonstruksi, yang menjadi indikator keberhasilan deteksi anomali.

Pada CAE satu *layer*, model dengan *learning rate* 0.01 konsisten menghasilkan nilai MAE dan MSE yang tinggi (rata-rata MAE 0.0159–0.0291) dan SSIM yang rendah (rata-rata 0.928–0.969), yang mengindikasikan model kesulitan merekonstruksi citra yang tidak sesuai dengan pola citra sehat. Sementara itu, model dengan *learning rate* 0.001 dan 0.0005 menghasilkan nilai MAE dan MSE yang jauh lebih rendah dan SSIM yang sangat tinggi (di atas 0.993) yang mengindikasikan model merekonstruksi citra dengan sangat baik.

Table 4.6 Hasil metrik CAE satu *layer*

E ^a	LR ^b	MAE			MSE			SSIM		
		MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
50	1.0.E-02	0.0291008	0.0508933	0.0190448	0.0014968	0.0039860	0.0006137	0.9280846	0.9626345	0.8628805
	1.0.E-03	0.0053026	0.0097783	0.0034054	0.0000541	0.0001780	0.0000202	0.9946886	0.9964128	0.9887602
	5.0.E-04	0.0064393	0.0167661	0.0040190	0.0000864	0.0004897	0.0000277	0.9931487	0.9955962	0.9776033
100	1.0.E-02	0.0159856	0.0374636	0.0100322	0.0005767	0.0023411	0.0001679	0.9697638	0.9869571	0.9249511
	1.0.E-03	0.0059284	0.0162965	0.0035723	0.0000814	0.0005377	0.0000227	0.9938686	0.9965344	0.9775697
	5.0.E-04	0.0050853	0.0109028	0.0033226	0.0000521	0.0002280	0.0000198	0.9948239	0.9965324	0.9821800
150	1.0.E-02	0.0237413	0.0436353	0.0136480	0.0011587	0.0033291	0.0003303	0.9482757	0.9742580	0.8713512
	1.0.E-03	0.0064293	0.0230489	0.0036246	0.0001123	0.0009735	0.0000231	0.9935313	0.9966439	0.9736260
	5.0.E-04	0.0053652	0.0143994	0.0033661	0.0000615	0.0003620	0.0000194	0.9951870	0.9969813	0.9832013
Mean		0.0114864	0.0247983	0.0071150	0.0004089	0.0013806	0.0001383	0.9790413	0.9891723	0.9491248
Max		0.0291008	0.0508933	0.0190448	0.0014968	0.0039860	0.0006137	0.9951870	0.9969813	0.9887602
Min		0.0050853	0.0097783	0.0033226	0.0000521	0.0001780	0.0000194	0.9280846	0.9626345	0.8628805

^aEpochs

^bLearning Rate

Hal ini menunjukkan bahwa model dengan *learning rate* tinggi lebih efektif dalam mendeteksi anomali pada citra sakit dibandingkan *learning rate* rendah. Meskipun demikian, rata-rata nilai SSIM yang tetap sangat tinggi untuk citra sakit (sekitar 0.98) menunjukkan bahwa dengan satu *layer*, model masih cenderung merekonstruksi citra sakit dengan **terlalu baik**, sehingga kemampuan dalam mendeteksi anomali terbatas. Dengan kata lain, dengan hanya satu *layer*, model **belum mampu** membedakan secara signifikan antara citra sehat dan sakit. Namun, dari keseluruhan sembilan model satu *layer* yang diuji, model dengan 50 *epochs* dan *learning rate* 0.01 menunjukkan performa unggul dalam mendeteksi anomali.

Di sisi lain, pada dua *layer*, hasil yang diperoleh kurang lebih sama dengan pada satu *layer*. Model dengan *learning rate* yang tinggi (0.01) menunjukkan performa yang baik, di mana nilai MAE dan MSE rata-rata tinggi (0.1064998 dan 0.0180233) pada 150 *epochs*. Sebaliknya, nilai SSIM rendah (0.2654537), menunjukkan ketidakmampuannya untuk merekonstruksi citra anomali yang menyimpang dari pola normal. Sementara itu, model dengan *learning rate* yang lebih rendah (0.0005 dan 0.001) secara konsisten menghasilkan performa yang buruk dengan nilai MAE dan MSE rata-rata jauh lebih rendah, sementara SSIM rata-rata meningkat signifikan hingga di atas 0.95.

Table 4.7 Hasil metrik CAE dua *layer*

E ^a	LR ^b	MAE			MSE			SSIM		
		MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
50	1.0.E-02	0.0368840	0.0606399	0.0229426	0.0023627	0.0057163	0.0008997	0.8487958	0.9238936	0.7532483
	1.0.E-03	0.0173975	0.0315913	0.0094228	0.0005764	0.0018948	0.0001612	0.9437362	0.9746487	0.8968823
	5.0.E-04	0.0181260	0.0387363	0.0102792	0.0006559	0.0029929	0.0001929	0.9433996	0.9715394	0.9033556
	1.0.E-02	0.0401871	0.0683397	0.0248747	0.0027846	0.0071091	0.0010365	0.8404465	0.9148577	0.7436457
100	1.0.E-03	0.0161503	0.0276971	0.0089204	0.0004896	0.0013015	0.0001409	0.9524089	0.9787521	0.9163683
	5.0.E-04	0.0150350	0.0238504	0.0083971	0.0004131	0.0009609	0.0001296	0.9547492	0.9782998	0.9265554
	1.0.E-02	0.1064998	0.1883111	0.0706209	0.0180233	0.0476277	0.0080314	0.2654537	0.4236528	0.1463703
	1.0.E-03	0.0161065	0.0367055	0.0080597	0.0005546	0.0024037	0.0001212	0.9550238	0.9805025	0.9209028
150	5.0.E-04	0.0160282	0.0352326	0.0091700	0.0005136	0.0023404	0.0001483	0.9545258	0.9778696	0.9106013
	Mean	0.0313794	0.0567893	0.0191875	0.0029304	0.0080386	0.0012068	0.8509488	0.9026685	0.7908811
	Max	0.1064998	0.1883111	0.0706209	0.0180233	0.0476277	0.0080314	0.9550238	0.9805025	0.9265554
	Min	0.0150350	0.0238504	0.0080597	0.0004131	0.0009609	0.0001212	0.2654537	0.4236528	0.1463703

^aEpochs

^bLearning Rate

Ini menunjukkan bahwa model dengan *learning rate* tinggi lebih efektif dalam mendeteksi pola anomali pada citra sakit. Secara keseluruhan, model CAE dua *layer* menunjukkan perkembangan yang signifikan dalam mendeteksi anomali bila dibandingkan dengan satu *layer*. Khususnya pada 150 *epochs* dengan *learning rate* 0.01 dengan nilai SSIM yang rendah (sekitar 0.2654537) yang menunjukkan bahwa ketidakmampuan model untuk merekonstruksi citra anomali yang menyimpang dari pola normal. Meskipun demikian, rata-rata nilai SSIM yang diperoleh masih relatif tinggi (sekitar 0.902) untuk citra sakit menunjukkan bahwa model tiga layer ini masih merekonstruksi citra sakit dengan **cukup baik**,

Terakhir, pada tiga *layer*, sama seperti *layer-layer* sebelumnya, model dengan *learning rate* yang tinggi (0.01) menunjukkan kinerja yang sangat baik dalam mendeteksi anomali. Dengan nilai rata-rata MAE dan MSE pada 50 *epochs*

mencapai 0.537 dan 0.658, sementara SSIM anjlok drastis hingga 0.0001174. Hasil ini mengindikasikan bahwa model gagal total dalam merekonstruksi citra anomali yang menyimpang dari pola normal sesuai dengan harapan. Akan tetapi, performa buruk ini secara langsung disebabkan oleh stagnasi *validation loss* selama pelatihan (Gambar 4.7 (i)), yang menunjukkan bahwa model tidak belajar secara efektif sehingga tidak dapat diandalkan.

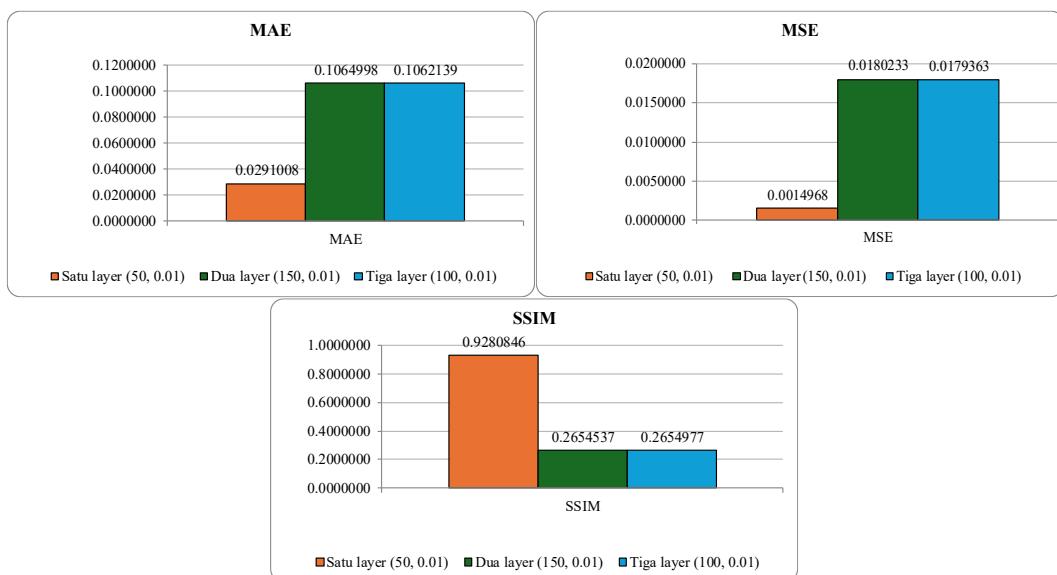
Table 4.8 Hasil metrik CAE tiga *layer*

E ^a	LR ^b	MAE			MSE			SSIM		
		MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
50	1.0.E-02	0.5377582	0.6582488	0.4503308	0.3072069	0.4580136	0.2243897	0.0001174	0.0002042	0.0000638
	1.0.E-03	0.0354195	0.0544404	0.0219398	0.0021722	0.0046068	0.0008537	0.7541202	0.8711751	0.6666439
	5.0.E-04	0.0368844	0.0581810	0.0240467	0.0023766	0.0053795	0.0010347	0.7393722	0.8549247	0.6531300
	1.0.E-02	0.1062139	0.1882612	0.0702753	0.0179363	0.0475009	0.0079693	0.2654977	0.4237297	0.1464336
100	1.0.E-03	0.0332566	0.0530898	0.0205093	0.0019104	0.0042883	0.0007535	0.7873200	0.8829123	0.7024419
	5.0.E-04	0.0336915	0.0513413	0.0212084	0.0019623	0.0041900	0.0008139	0.7716227	0.8758828	0.6875070
	1.0.E-02	0.0648296	0.1034472	0.0430323	0.0069161	0.0155034	0.0030778	0.4738924	0.6715034	0.3486413
	1.0.E-03	0.0316384	0.0514826	0.0189021	0.0017545	0.0041015	0.0006716	0.8025794	0.8909420	0.7369564
150	5.0.E-04	0.0317475	0.0494779	0.0196195	0.0017504	0.0038490	0.0006923	0.7949409	0.8840991	0.7167462
	Mean	0.1012711	0.1408856	0.0766516	0.0382206	0.0608259	0.0266952	0.5988292	0.7061526	0.5176182
	Max	0.5377582	0.6582488	0.4503308	0.3072069	0.4580136	0.2243897	0.8025794	0.8909420	0.7369564
	Min	0.0316384	0.0494779	0.0189021	0.0017504	0.0038490	0.0006716	0.0001174	0.0002042	0.0000638

^aEpochs
^bLearning Rate

Oleh karena itu, dipilih model alternatif yang menunjukkan performa terbaik berikutnya dalam mendeteksi anomali. Model dengan 100 *epochs* pada *learning rate* yang sama dipilih karena memberikan keseimbangan optimal dengan nilai MAE rata-rata 0.106, MSE rata-rata 0.018, dan SSIM rata-rata 0.265. Angka-angka ini tetap menunjukkan tingkat *error* rekonstruksi yang tinggi dan kemiripan struktural yang rendah, yang merupakan indikator kuat untuk deteksi anomali.

Analisis menunjukkan bahwa *learning rate* tinggi secara konsisten menghasilkan performa terbaik untuk deteksi anomali. Menariknya, rata-rata nilai SSIM secara keseluruhan menurun secara signifikan menjadi 0.59, jauh dari rata-rata 0.85 yang tercatat pada model dua *layer*. Penurunan ini didorong oleh model-model dengan learning rate 0.01, yang memiliki nilai MAE rata-rata melonjak hingga 0.101 dan MSE rata-rata 0.038, sementara rata-rata SSIM-nya anjlok hingga 0.59. Hal ini menunjukkan bahwa dengan tiga *layer*, CAE akhirnya mampu untuk mengidentifikasi anomali pada citra sakit.



Gambar 4.12 Grafik Perbandingan Nilai MAE, MSE, dan SSIM CAE

Berdasarkan hasil dari ketiga *layer* pada citra pohon sawit sakit satu bulan sebelum tumbang, dapat disimpulkan bahwa model dengan dua *layer* (150 epochs, learning rate 0.01) merupakan model yang paling efektif dalam mendeteksi anomali. Hal ini terlihat dari nilai MAE (0.1065) dan MSE (0.0180) yang jauh lebih tinggi dibandingkan model lainnya, serta nilai SSIM yang sangat rendah (0.2654537).

Nilai MAE dan MSE yang tinggi ini menunjukkan bahwa model kesulitan besar dalam merekonstruksi citra yang terinfeksi, menandakan perbedaan signifikan antara citra asli dan citra yang direkonstruksi. Sebaliknya, model satu *layer* (50 epochs, learning rate 0.01) dan tiga *layer* (100 epochs, learning rate 0.01) menunjukkan performa yang sangat berbeda. Model satu *layer* memiliki nilai MAE dan MSE yang sangat rendah, serta SSIM yang sangat tinggi (sekitar 0.9281). Kondisi ini mengindikasikan bahwa model tersebut merekonstruksi citra dengan sangat baik, sehingga kurang efektif dalam mendeteksi anomali karena tidak mampu membedakan citra sehat dan sakit secara signifikan.

4.3.3 Evaluasi Model *Convolutional Variational Autoencoder* (CVAE)

Sama seperti CAE, CVAE akan diuji menggunakan 41 citra satu bulan sebelum tumbang pada data *timeseries* yang telah disiapkan sebelumnya. Model akan dievaluasi menggunakan tiga metrik yang sama yakni, MAE, MSE dan SSIM guna melihat seberapa baik model dapat mengidentifikasi anomali pada pohon kelapa sawit sakit sebelum pohon tersebut tumbang.

Hasil pengujian pada model CVAE satu *layer* menunjukkan bahwa model ini dengan sangat efektif dalam mendekripsi anomali pada citra pohon sawit sakit. Secara umum, model konsisten menghasilkan nilai MAE dan MSE yang tinggi (rata-rata MAE 0.1038; rata-rata MSE 0.0173), yang mengindikasikan bahwa model kesulitan merekonstruksi citra yang tidak sesuai dengan pola normal. Terlihat juga bahwa SSIM rata-rata pada seluruh pengujian berada pada angka yang sangat rendah (0.2288), jauh dari nilai 1.

Table 4.9 Hasil metrik CVAE satu *layer*

E ^a	LR ^b	MAE			MSE			SSIM		
		MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
50	1.0.E-02	0.1074842	0.1914445	0.0703306	0.0184337	0.0494772	0.0079601	0.2653537	0.4236035	0.1462796
	1.0.E-03	0.0994282	0.1575642	0.0661049	0.0158465	0.0364041	0.0071531	0.2289199	0.3550258	0.1132856
	5.0.E-04	0.1002738	0.1613902	0.0695336	0.0162404	0.0388016	0.0078244	0.2263245	0.3253515	0.1117504
	1.0.E-02	0.1070666	0.1905993	0.0702375	0.0182764	0.0489064	0.0079524	0.2654072	0.4236671	0.1463388
100	1.0.E-03	0.1000467	0.1532561	0.0671255	0.0160090	0.0353326	0.0073812	0.2138735	0.3129877	0.0995519
	5.0.E-04	0.1038975	0.1669004	0.0698052	0.0175081	0.0414243	0.0079254	0.2012646	0.2851507	0.0975272
	1.0.E-02	0.1069838	0.1945314	0.0692525	0.0183778	0.0512632	0.0077089	0.2665539	0.4278922	0.1483714
	1.0.E-03	0.1049213	0.1532674	0.0724759	0.0178502	0.0361477	0.0085587	0.1990285	0.2816541	0.0935445
150	5.0.E-04	0.1044065	0.1577373	0.0746399	0.0177221	0.0380221	0.0089699	0.1933260	0.2843523	0.0852789
	Mean	0.1038343	0.1696323	0.0699451	0.0173627	0.0417532	0.0079371	0.2288947	0.3466316	0.1157698
	Max	0.1074842	0.1945314	0.0746399	0.0184337	0.0512632	0.0089699	0.2665539	0.4278922	0.1483714
	Min	0.0994282	0.1532561	0.0661049	0.0158465	0.0353326	0.0071531	0.1933260	0.2816541	0.0852789

^aEpochs

^bLearning Rate



Analisis lebih mendalam menunjukkan bahwa jumlah *epochs* dan tingkat *learning rate* tidak memberikan dampak sebesar pada model CAE. Meskipun terdapat sedikit perbedaan, seluruh konfigurasi menunjukkan nilai MAE dan MSE yang tinggi dan SSIM yang rendah. Dengan nilai rata-rata MAE (0.1074) dan MSE (0.0184) tertinggi, model dengan 50 *epochs* dan *learning rate* 0.01 menunjukkan tingkat *reconstruction error* yang paling ekstrem serta salah satu nilai rata-rata SSIM terendah (0.2653). Hal ini mengindikasikan bahwa arsitektur CVAE memiliki kapasitas bawaan yang kuat untuk membedakan anomali, bahkan dengan

konfigurasi yang paling sederhana (satu *layer*, *epochs* sedikit, dan *learning rate* tinggi).

Sementara itu, bila jumlah *layer* ditambahkan, data pengujian menunjukkan kinerja yang sama efektifnya seperti satu *layer* dalam mendeteksi anomali. Model ini secara konsisten menghasilkan nilai MAE dan MSE yang tinggi, mengindikasikan ketidakmampuannya untuk merekonstruksi citra anomali yang menyimpang dari pola normal. Mirip dengan CVAE satu *layer*, model CVAE dua *layer* ini juga tidak terlalu sensitif terhadap jumlah *epochs* dan *learning rate* dibandingkan dengan model CAE. Hasilnya, seluruh konfigurasi model CVAE dua *layer* menunjukkan pola yang sama: nilai MAE dan MSE yang tinggi sejalan dengan nilai SSIM yang rendah, mengonfirmasi kemampuannya untuk mendeteksi anomali. Akan tetapi, bila diteliti lebih lanjut, rata-rata nilai MAE (0.10333) dan MSE (0.0172) sedikit menurun bila dibandingkan rata-rata nilai MAE (0.10383) dan MSE (0.0173) dari CVAE satu *layer*. Di antara semua konfigurasi tersebut, model dengan 50 *epochs* dan *learning rate* 0.01 terbukti menjadi yang paling efektif.

Table 4.10 Hasil metrik CVAE dua *layer*

E ^a	LR ^b	MAE			MSE			SSIM		
		MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
50	1.0.E-02	0.1078510	0.1929928	0.0703332	0.0185969	0.0501858	0.0079837	0.2653363	0.4236583	0.1462958
	1.0.E-03	0.1003229	0.1585554	0.0711501	0.0162984	0.0377217	0.0081676	0.2135366	0.3309862	0.1047667
	5.0.E-04	0.0992959	0.1566563	0.0692126	0.0158965	0.0354711	0.0076943	0.2207192	0.3363079	0.1276968
	1.0.E-02	0.1073630	0.1954307	0.0694064	0.0185340	0.0516901	0.0077708	0.2656059	0.4257325	0.1481489
100	1.0.E-03	0.1003623	0.1585258	0.0679566	0.0162211	0.0363438	0.0075075	0.2116430	0.3253886	0.0847475
	5.0.E-04	0.1032886	0.1623113	0.0722683	0.0172550	0.0390203	0.0083528	0.2015938	0.3083330	0.0966824
	1.0.E-02	0.1065022	0.1931994	0.0691029	0.0181800	0.0504058	0.0076959	0.2667841	0.4279463	0.1462277
	1.0.E-03	0.1018437	0.1610080	0.0705368	0.0167210	0.0385086	0.0080934	0.2023208	0.3059188	0.0960097
150	5.0.E-04	0.1031854	0.1647636	0.0692033	0.0172547	0.0413527	0.0079339	0.1958483	0.3022754	0.0720380
	Mean	0.1033350	0.1714937	0.0699078	0.0172175	0.0423000	0.0079111	0.2270431	0.3540608	0.1136237
	Max	0.1078510	0.1954307	0.0722683	0.0185969	0.0516901	0.0083528	0.2667841	0.4279463	0.1481489
	Min	0.0992959	0.1566563	0.0679566	0.0158965	0.0354711	0.0075075	0.1958483	0.3022754	0.0720380

^aEpochs

^bLearning Rate

Sementara itu, dengan tiga *layer*, model menunjukkan kinerja yang sama efektifnya dalam mendeteksi anomali. Model ini secara konsisten menghasilkan nilai MAE yang tinggi (rata-rata 0.10323) dan MSE yang juga tinggi (rata-rata 0.0172), mengindikasikan bahwa model kesulitan merekonstruksi citra yang menyimpang dari pola normal (citra pohon sawit sehat). Namun, bila diteliti lebih

lanjut, performa model tiga *layer* sedikit lebih mirip dengan model dua layer. Hal ini terlihat dari rata-rata nilai MAE (0.10302) yang sedikit menurun bila dibandingkan dengan satu layer (0.10383) dan nilai rata-rata MSE (0.0172) yang hampir sama dengan dua layer. Ini menunjukkan bahwa meskipun performa deteksinya tetap baik, ada sedikit penurunan sensitivitas seiring dengan penambahan layer dan kompleksitas model. Terlepas dari konfigurasi yang diuji, model dengan 100 *epochs* dan *learning rate* 0.01 menunjukkan performa deteksi anomali yang paling optimal pada CVAE tiga *layer*.

Table 4.11 Hasil metrik CVAE tiga *layer*

E ^a	LR ^b	MAE			MSE			SSIM		
		MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
50	1.0.E-02	0.1051494	0.1893160	0.0691998	0.0176486	0.0483606	0.0077090	0.2662871	0.4303199	0.1473309
	1.0.E-03	0.0994217	0.1540661	0.0705501	0.0159593	0.0361837	0.0078722	0.2115616	0.2983007	0.1195843
	5.0.E-04	0.1018405	0.1740922	0.0715895	0.0170964	0.0463780	0.0084254	0.2106356	0.3392256	0.0848429
	1.0.E-02	0.1078605	0.1931268	0.0701838	0.0186086	0.0503117	0.0079458	0.2653313	0.4236479	0.1462884
100	1.0.E-03	0.1014233	0.1559846	0.0711551	0.0167573	0.0374126	0.0082201	0.2012249	0.2868941	0.1052548
	5.0.E-04	0.1021293	0.1678033	0.0684070	0.0171371	0.0435580	0.0074340	0.2010325	0.3003490	0.0984139
	1.0.E-02	0.1073019	0.1954038	0.0694053	0.0185099	0.0516163	0.0077782	0.2663463	0.4289261	0.1471891
	1.0.E-03	0.1007231	0.1553570	0.0727348	0.0164528	0.0371674	0.0083783	0.1955662	0.2828159	0.0883089
150	5.0.E-04	0.1013811	0.1586819	0.0722715	0.0166894	0.0385223	0.0083783	0.2010946	0.2900859	0.0828614
	Mean	0.1030257	0.1715369	0.0706108	0.0172066	0.0432790	0.0080157	0.2243422	0.3422850	0.1133416
	Max	0.1078605	0.1954038	0.0727348	0.0186086	0.0516163	0.0084254	0.2663463	0.4303199	0.1473309
	Min	0.0994217	0.1540661	0.0684070	0.0159593	0.0361837	0.0074340	0.1955662	0.2828159	0.0828614

^aEpochs

^bLearning Rate

Model CVAE secara konsisten menghasilkan nilai MAE dan MSE yang tinggi (> 0.107) serta SSIM yang rendah (± 0.265). Ini menegaskan bahwa arsitektur CVAE memiliki kemampuan bawaan yang kuat untuk membedakan anomali. Dari hasil pengujian ketiga *layer* tersebut, dapat disimpulkan bahwa model CVAE, terlepas dari jumlah *layer*-nya, sangat efektif dalam mendekripsi anomali. Berbeda dengan model CAE, arsitektur CVAE secara inheren mampu membedakan citra sehat dari citra sakit.

Secara keseluruhan, model CVAE adalah pilihan yang jauh lebih optimal untuk tujuan deteksi anomali dibandingkan model CAE. Perbedaan fundamentalnya terletak pada cara kedua model merespons peningkatan kompleksitas. Model CAE menunjukkan bahwa kemampuannya untuk mendekripsi anomali meningkat secara bertahap seiring dengan penambahan layer, mengindikasikan bahwa model memerlukan arsitektur yang lebih dalam untuk

membedakan anomali. Sebaliknya, model CVAE sudah mampu beroperasi secara optimal bahkan dengan hanya satu *layer*. Hasil pengujian menunjukkan bahwa penambahan *layer* pada CVAE tidak memberikan peningkatan melainkan sedikit menurunkan performa deteksi. Oleh karena itu, CVAE dengan tiga *layer* merupakan model yang paling optimal dalam mendekripsi anomali pada citra sakit.

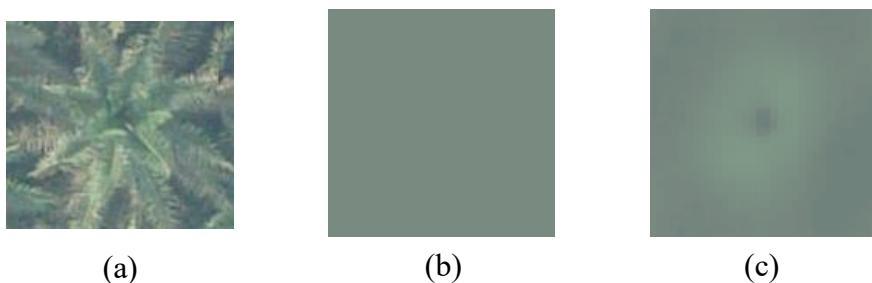


Gambar 4.13 Grafik perbandingan Nilai MAE, MSE, dan SSIM CVAE

4.4 Perbandingan Hasil Rekonstruksi Citra

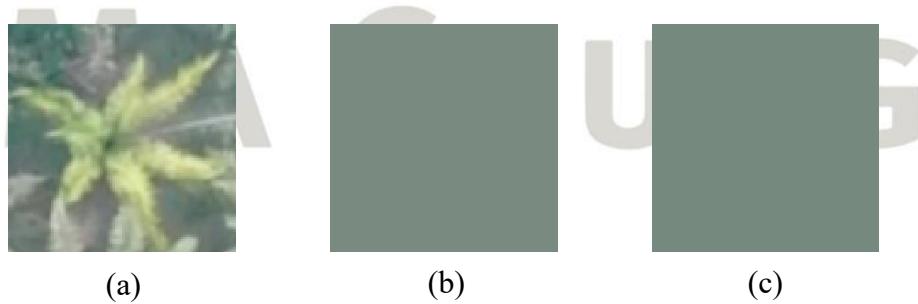
Pada citra asli yang menunjukkan kondisi pohon sawit sehat, model CAE menunjukkan kinerja rekonstruksi yang kurang efektif. Citra yang direkonstruksi terlihat sangat berbeda dari aslinya, dengan detail dan struktur yang hampir tidak terlihat. Hal ini mengindikasikan bahwa model CAE belum berhasil mempelajari dan mereplikasi pola visual dari pohon sehat secara akurat. Sebaliknya, model CVAE memperlihatkan hasil yang lebih baik. Meskipun citra rekonstruksinya tidak identik dengan aslinya, ia mampu mereplikasi sebagian kecil dari struktur citra asli.

Secara keseluruhan, dalam merekonstruksi citra sehat, model CVAE terbukti lebih unggul dibandingkan dengan model CAE (Gambar 4.14).



Gambar 4. 14 Citra Sehat (a), Citra Rekonstruksi CAE (b)
dan Citra Rekonstruksi CVAE (c)

Ketika dihadapkan pada citra asli yang menunjukkan pohon sawit sakit (anomali), kedua model (baik CAE maupun CVAE) menunjukkan hasil yang serupa: detail dan struktur rekonstruksi citra tersebut hampir tidak terlihat (Gambar 4.15). Kinerja ini sebenarnya adalah indikator keberhasilan dalam konteks deteksi anomali. Kondisi ini terjadi karena kedua model mengalami kesulitan dalam merekonstruksi citra yang menyimpang dari pola normal yang telah mereka pelajari. Ketidakmampuan untuk mereplikasi anomali tersebut menghasilkan kesalahan rekonstruksi yang tinggi, yang menjadi sinyal utama untuk mengidentifikasi adanya penyakit.



Gambar 4. 15 Citra Sehat (a), Citra Rekonstruksi CAE (b)
dan Citra Rekonstruksi CVAE (c)

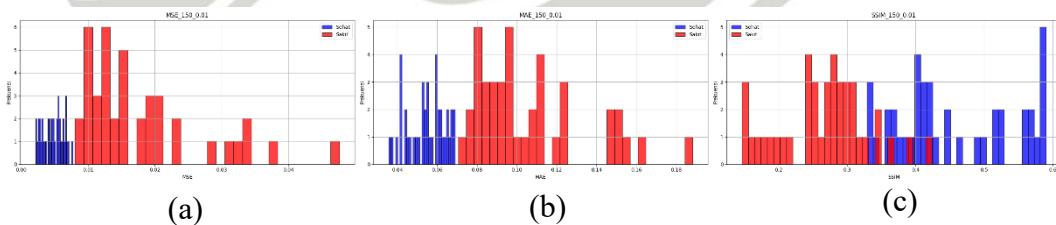
4.5 Perbandingan Kinerja CAE dan CVAE

Setelah nilai metrik MAE, MSE dan SSIM diperoleh, hasil distribusi data dari model terbaik kedua artitektur akan divisualisasikan menggunakan histogram

untuk melihat distribusi data dari ketiga metrik tersebut. Dengan membandingkan distribusi data pohon sawit sehat dengan pohon sawit sakit, analisis visual akan dilakukan untuk mengidentifikasi perbedaan signifikan yang menunjukkan keberhasilan model dalam mendeteksi anomali. Tingkat perbedaan ini juga akan diukur secara kuantitatif menggunakan metrik JSD dan BD.

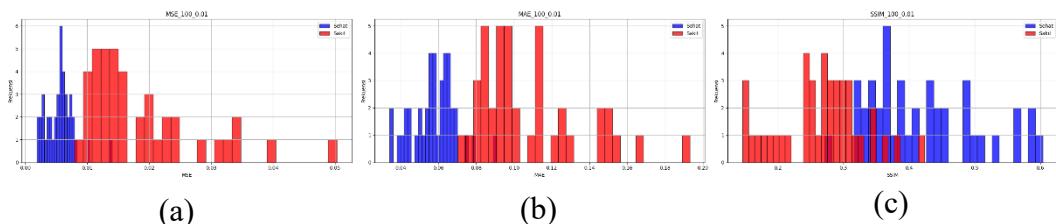
4.5.1 Satu bulan sebelum tumbang

Berdasarkan pengujian pada 41 citra yang diambil satu bulan sebelum pohon tumbang, model dievaluasi kemampuannya dalam mendeteksi anomali. Model yang efektif akan menunjukkan kesulitan dalam merekonstruksi citra sakit, yang tercermin dari nilai MAE dan MSE yang tinggi dan SSIM yang rendah. Berikut adalah grafik distribusi data metrik MSE, MAE, dan SSIM pada CAE yang menunjukkan sensitivitas yang baik karena model berhasil membedakan antara citra sehat dengan citra sakit (Gambar 4.16).



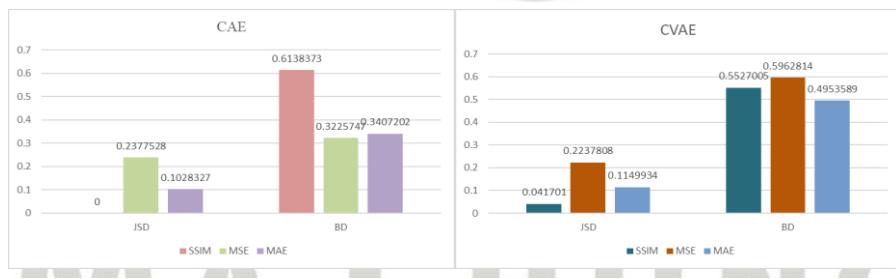
Gambar 4.16 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE satu bulan sebelum tumbang

Sementara itu, model CVAE menunjukkan karakteristik data yang sedikit berbeda dalam mendeteksi anomali pada citra sakit (Gambar 4.17).



Gambar 4.17 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE satu bulan sebelum tumbang

Berdasarkan distribusi data dari kedua arsitektur jarak antara citra sehat dan citra sakit pada CAE lebih terlihat bila dibandingkan dengan CVAE. Distribusi data pada CVAE terkhususnya MAE (Gambar 4.15 tengah) masih terlihat ada tumpang tindih antara distribusi data citra sehat dan citra sakit. Hal ini menunjukkan bahwa arsitektur CAE lebih unggul dari pada model CVAE. Kemudian pada analisis metrik JSD dan BD, terlihat perbedaan kinerja yang signifikan antara arsitektur CAE dan CVAE. Model CAE menunjukkan kemampuan terbaiknya dalam memisahkan distribusi metrik MSE ($JSD=0.238$, $BD=0.323$) dan MAE ($JSD=0.103$, $BD=0.341$), sedangkan metrik SSIM tidak menunjukkan perbedaan distribusi sama sekali ($JSD=0$, $BD=0.614$), hal ini mengindikasikan bahwa CAE tidak mampu membedakan citra sehat dan sakit berdasarkan kemiripan struktural. Di sisi lain, CVAE menunjukkan performa yang lebih konsisten dan seimbang di seluruh metrik. CVAE berhasil memisahkan distribusi metrik SSIM dengan nilai JSD 0.042 dan BD 0.553, dan memiliki performa yang serupa untuk metrik MAE ($JSD=0.115$, $BD=0.495$). Meskipun nilai JSD dan BD untuk MSE pada CVAE lebih rendah dibandingkan CAE ($JSD=0.224$, $BD=0.596$).



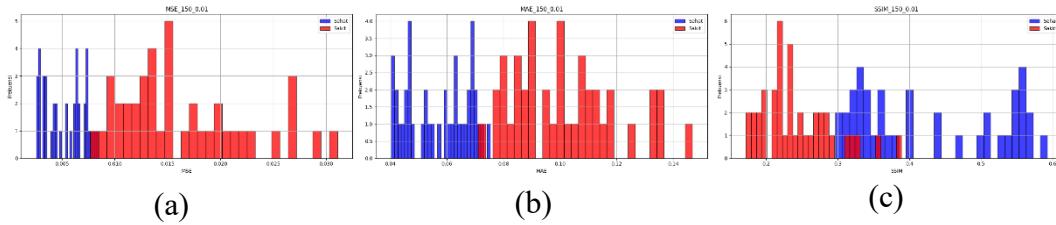
Gambar 4. 18 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE

Secara keseluruhan pada periode ini, CVAE terbukti lebih unggul karena kemampuannya dalam memisahkan distribusi data pada metrik yang paling relevan untuk deteksi anomali pada citra, yaitu SSIM.

4.5.2 Dua bulan sebelum tumbang

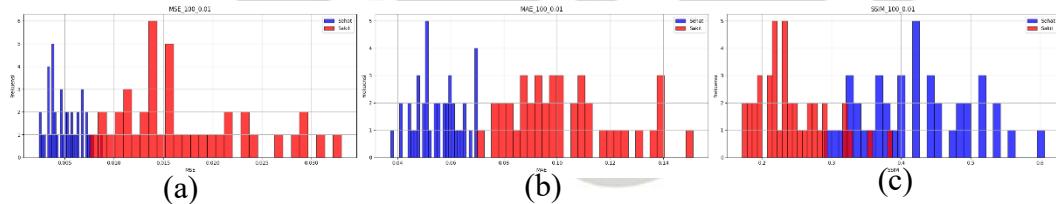
Pada periode dua bulan sebelum tumbang, pola infeksi Ganoderma masih dapat terlihat dengan jelas. Model CAE menunjukkan sedikit penurunan kinerja sementara model CVAE menunjukkan kinerja yang konsisten. Jarak distribusi

metrik MAE, MSE, dan SSIM antara citra sehat dan sakit pada CAE masih berjarak, menandakan model masih mampu dalam mengidentifikasi anomali (Gambar 4.19).



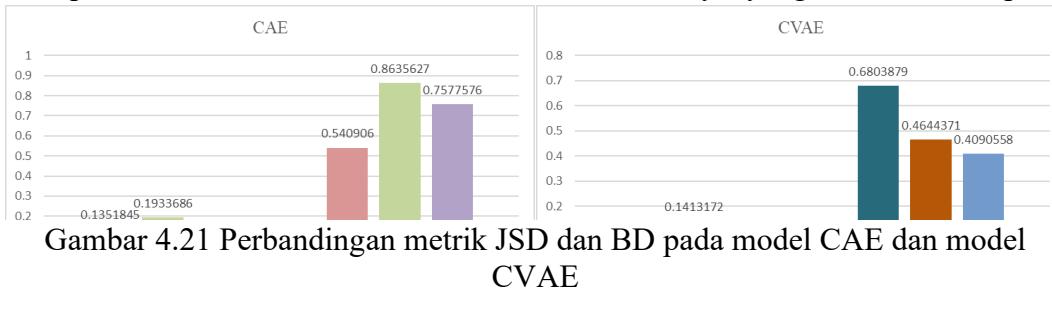
Gambar 4.19 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE dua bulan sebelum tumbang

Namun jarak yang terlihat mengecil antara citra sehat dan sakit bila dibandingkan pada periode satu bulan sebelum tumbang. Sementara itu, CVAE konsisten dalam distribusi data antara citra sehat dan citra sakit, menegaskan kemampuannya dalam mendeteksi anomali secara dini (Gambar 4.20).



Gambar 4.20 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE dua bulan sebelum tumbang

Berdasarkan analisis JSD dan BD pada kedua model, terlihat bahwa CAE menunjukkan performa yang lebih unggul dalam memisahkan distribusi metrik. Dengan nilai JSD dan BD yang lebih tinggi untuk MSE (JSD=0.193, BD=0.864) dan MAE (JSD=0.084, BD=0.758), model CAE lebih efektif dalam menciptakan perbedaan yang signifikan antara data sehat dan anomali. Meskipun CVAE juga mampu membedakan anomali, nilai JSD dan BD-nya yang lebih rendah pada

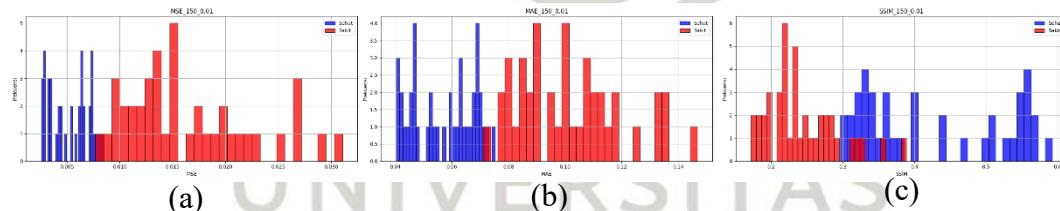


metrik MAE dan MSE ($JSD=0.049$, $BD=0.409$) menunjukkan pemisahan distribusi yang tidak setajam CAE. Menariknya, pada metrik SSIM, CVAE menunjukkan nilai JSD nol, mengindikasikan bahwa distribusi SSIM untuk citra sehat dan sakit hampir tidak dapat dibedakan, yang berlawanan dengan performa CAE ($JSD=0.135$, $BD=0.541$).

Dengan demikian pada periode ini, CAE terbukti lebih andal dan efektif dalam mendeteksi anomali karena kemampuannya yang lebih baik dalam memisahkan distribusi metrik evaluasi.

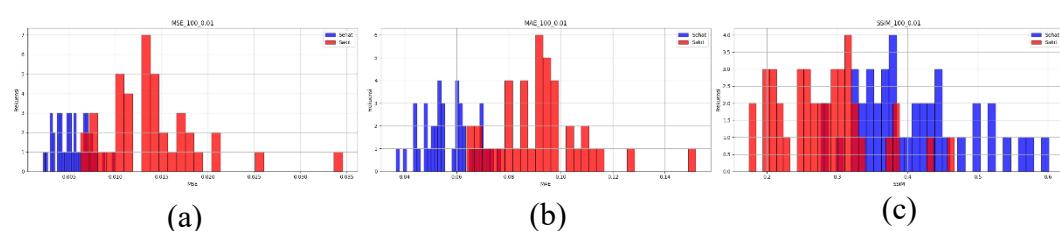
4.5.3 Tiga bulan sebelum tumbang

Pada periode ini, kemampuan deteksi model CAE mulai menunjukkan sedikit penurunan. Meskipun nilai MAE dan MSE masih relatif tinggi dan SSIM rendah. Pada CAE jarak antara distribusi metrik citra sehat dan sakit pada histogram sedikit menyempit dan tumpeng tindih bila dibandingkan bulan sebelumnya. (Gambar 4.22)



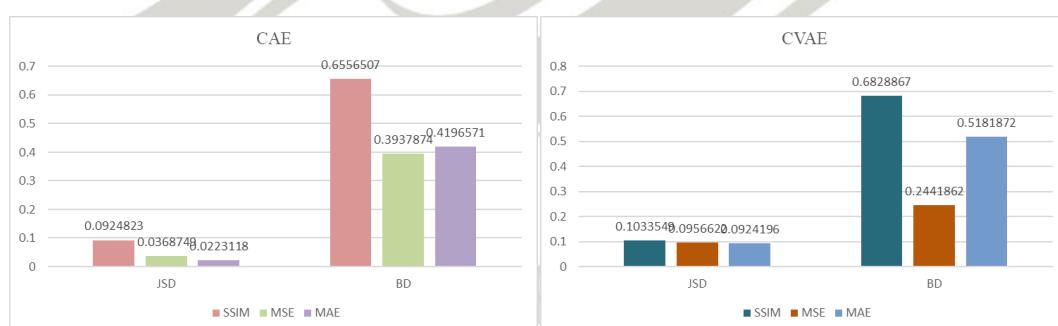
Gambar 4.22 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE tiga bulan sebelum tumbang

Sebaliknya model CVAE menunjukkan penurunan kinerja deteksi, jarak antara distribusi metrik citra sehat dan sakit pada histogram sebagian mengalami tumpang tindih bila dibandingkan bulan sebelumnya (Gambar 4.23).



Gambar 4.23 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE tiga bulan sebelum tumbang

Hal ini mengindikasikan bahwa kedua model masih dapat mendeteksi anomali, tetapi pada model CVAE, tingkat kepastian sedikit berkurang karena distribusi data citra sehat dan citra sakit mengalami tumpang tindih yang signifikan. Sementara itu, berdasarkan hasil analisis metrik JSD dan BD, terlihat perbedaan performa yang signifikan antara arsitektur CAE dan CVAE. Model CAE menunjukkan kemampuan terbaiknya dalam memisahkan distribusi metrik SSIM ($JSD=0.092$, $BD=0.656$) dan MSE ($JSD=0.037$, $BD=0.394$), sedangkan performanya sedikit menurun pada metrik MAE ($JSD=0.022$, $BD=0.420$). Di sisi lain, CVAE menunjukkan performa yang lebih konsisten di seluruh metrik, dengan nilai JSD dan BD yang lebih tinggi untuk SSIM ($JSD=0.103$, $BD=0.683$) dan MAE ($JSD=0.092$, $BD=0.518$) jika dibandingkan dengan CAE. Namun, CVAE menunjukkan performa yang lebih lemah pada metrik MSE ($JSD=0.096$, $BD=0.244$).



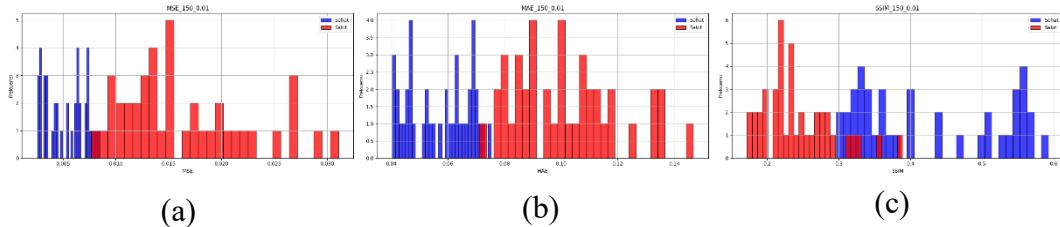
Gambar 4. 24 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE

Secara keseluruhan pada periode ini, CVAE terbukti lebih unggul karena kemampuannya yang seimbang dalam memisahkan distribusi metrik yang relevan untuk deteksi anomali pada citra, terutama pada metrik SSIM.

4.5.4 Empat bulan sebelum tumbang

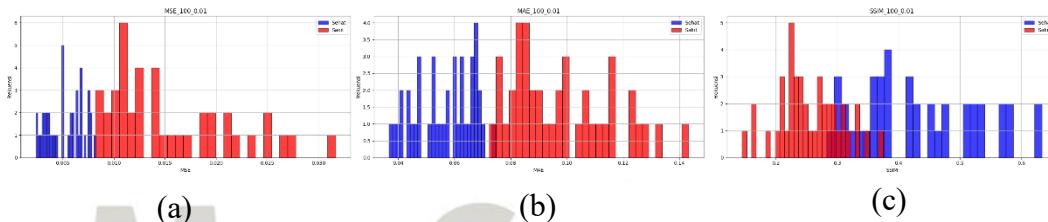
Masuk ke periode keempat, secara teori gejala penyakit pada tanaman mungkin masih minim dan belum muncul secara kasat mata. Namun hasil kedua model tetap menunjukkan perbedaan antara citra sehat dan sakit bahkan lebih baik

dibandingkan pada periode ketiga. Pada CAE, distribusi metrik MAE dan MSE antara kedua kelompok citra (sehat dan sakit) tetap konsisten memberikan kinerja yang mampu memisahkan kedua kelompok.



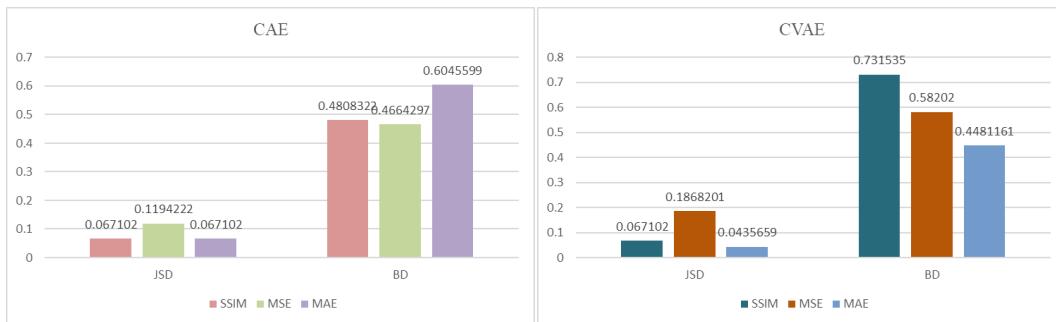
Gambar 4.25 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE empat bulan sebelum tumbang

Akan tetapi pada CVAE, distribusi yang sebelumnya tumpang tindih, kini kembali meningkat, memisahkan kedua kelompok data. Ini menunjukkan bahwa kedua model masih efektif dalam mendeteksi anomali empat bulan sebelum pohon tumbang.



Gambar 4.26 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE empat bulan sebelum tumbang

Berdasarkan hasil analisis metrik JSD dan BD, terlihat perbedaan performa antara arsitektur CAE dan CVAE. Model CAE menunjukkan kemampuan yang lebih baik dalam memisahkan distribusi metrik MSE ($JSD=0.119$, $BD=0.466$) dan MAE ($JSD=0.067$, $BD=0.605$). Di sisi lain, CVAE menunjukkan performa yang lebih unggul pada metrik MSE ($JSD=0.187$, $BD=0.582$), mengindikasikan kemampuan yang lebih baik dalam membedakan anomali. Namun, CVAE menunjukkan performa yang lebih lemah pada metrik MAE ($JSD=0.044$, $BD=0.448$) dan SSIM ($JSD=0.067$, $BD=0.732$) jika dibandingkan dengan CAE.

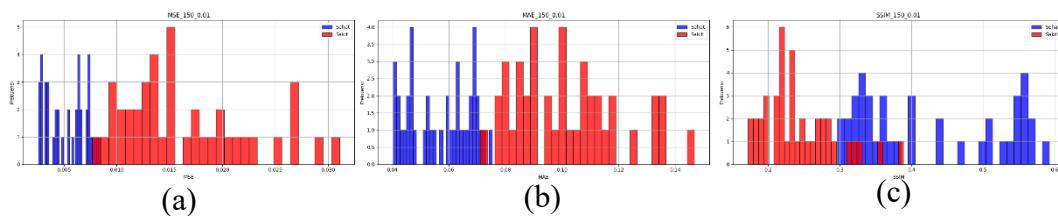


Gambar 4. 27 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE

Secara keseluruhan, CVAE terbukti lebih unggul karena kemampuannya yang seimbang dalam memisahkan distribusi metrik yang relevan untuk deteksi anomali pada citra, terutama pada metrik MSE.

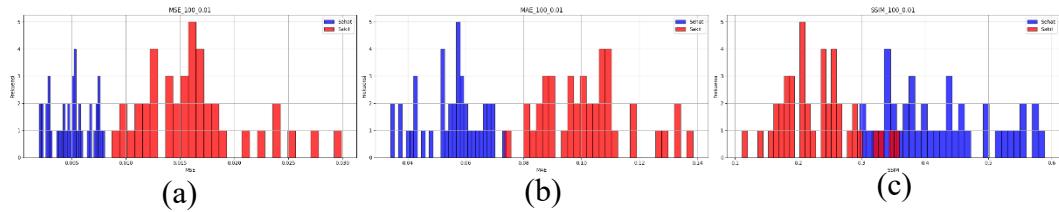
4.5.5 Lima bulan sebelum tumbang

Masuk ke periode terakhir, secara teori gejala penyakit pada tanaman seharusnya sulit untuk dideteksi. Namun hasil kedua model tetap menunjukkan perbedaan antara citra sehat dan sakit. CAE tetap konsisten dalam memisahkan distribusi metrik MAE dan MSE antara kedua kelompok citra (sehat dan sakit).



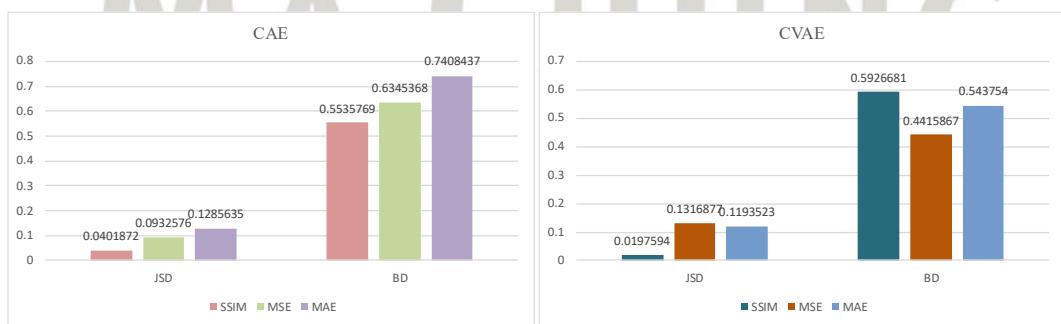
Gambar 4.28 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CAE lima bulan sebelum tumbang

Menariknya, CVAE mengalami peningkatan dengan jarak antara distribusi metrik MAE dan MSE antara kedua kelompok citra (sehat dan sakit) semakin melebar. Ini menunjukkan bahwa kedua model masih efektif dalam mendeteksi anomali empat bulan sebelum pohon tumbang.



Gambar 4.29 Hasil distribusi data metrik (a) MSE, (b) MAE, dan (c) SSIM pada model CVAE lima bulan sebelum tumbang

Pada analisis JSD dan BD, model CAE menunjukkan kemampuan yang lebih unggul dalam memisahkan distribusi metrik, menjadikannya lebih efektif untuk deteksi anomali. Model CAE memiliki nilai JSD dan BD tertinggi pada metrik MAE ($JSD=0.129$, $BD=0.741$) dan MSE ($JSD=0.093$, $BD=0.635$). Ini menunjukkan bahwa CAE berhasil menciptakan perbedaan yang signifikan antara distribusi citra sehat dan sakit. Di sisi lain, CVAE menunjukkan performa yang lebih seimbang, tetapi nilai JSD dan BD-nya cenderung lebih rendah dibandingkan CAE pada metrik MAE dan MSE ($JSD=0.119$, $BD=0.544$ dan $JSD=0.132$, $BD=0.442$), yang mengindikasikan pemisahan distribusi yang tidak setajam CAE. Meskipun CVAE menunjukkan nilai BD yang sedikit lebih tinggi pada metrik SSIM, nilai JSD-nya sangat rendah ($JSD=0.020$), menandakan perbedaan distribusi yang minimal. Secara keseluruhan, CAE terbukti lebih andal karena mampu menghasilkan pemisahan distribusi yang lebih kuat dan jelas.

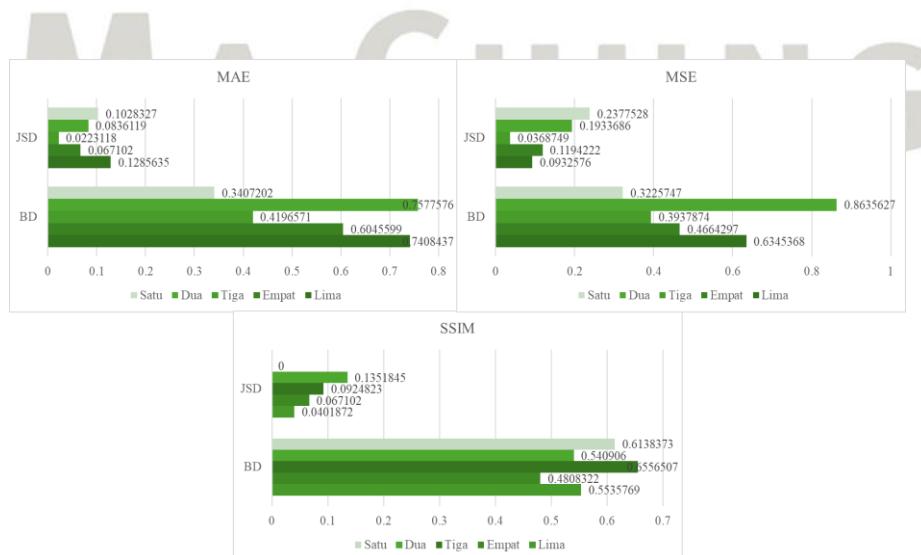


Gambar 4. 30 Perbandingan metrik JSD dan BD pada model CAE dan model CVAE

Dari hasil pengujian sepanjang periode (satu hingga lima bulan) sebelum tumbang model CVAE terbukti sebagai pilihan paling optimal dan andal untuk deteksi anomali *Ganoderma*. Meskipun CAE dapat bekerja dengan baik pada 1 bulan dan 5 bulan sebelum tumbang, arsitektur CVAE yang berbasis probabilistik memberikan keunggulan inheren yang memungkinkannya untuk secara konsisten mendeteksi anomali sejak tahap paling dini, bahkan ketika gejala visual belum terlihat. Dengan demikian, CVAE adalah solusi yang lebih efisien dan efektif untuk deteksi dini dan pencegahan penyebaran penyakit, yang sangat krusial untuk menjaga produktivitas perkebunan.

4.6 Ringkasan Perbandingan Hasil

Berdasarkan metrik MAE dan MSE, model CAE menunjukkan fluktiasi yang signifikan, dengan nilai BD (*Bhattacharyya Distance*) tertinggi di bulan kedua dan kelima. Hal ini mengindikasikan adanya variasi besar dalam akurasi prediksi. Sebaliknya, nilai JSD (*Jensen-Shannon Distance*) pada model ini cenderung lebih stabil, meskipun ada sedikit lonjakan di bulan kelima. Pada metrik SSIM, kualitas prediksi model CAE bervariasi, mencapai puncaknya di bulan ketiga (0.6557) dan nilai terendah di bulan keempat (0.4808). Ini menunjukkan inkonsistensi dalam kualitas visual hasil prediksi. Secara keseluruhan, model CAE menunjukkan kinerja yang tidak stabil dalam hal akurasi dan kualitas visual.



Gambar 4. 31 Grafik progresif nilai JSD dan BD metrik MAE, MSE, dan SSIM pada CAE untuk diskriminasi sawit sehat dan sakit.

Sebaliknya, model CVAE menunjukkan performa yang lebih konsisten pada metrik BD. Nilai BD pada metrik MAE dan MSE relatif lebih rendah dan stabil dibandingkan model CAE, tanpa lonjakan ekstrem. Ini mengindikasikan bahwa CVAE memiliki akurasi prediksi yang lebih konsisten dari waktu ke waktu. Untuk metrik SSIM, CVAE menunjukkan peningkatan kualitas visual secara progresif dari bulan ke bulan, mencapai nilai tertinggi di bulan keempat (0.7315), sebelum sedikit menurun di bulan kelima. Nilai JSD pada model CVAE juga menunjukkan stabilitas yang lebih baik.



Gambar 4. 32 Grafik progresif nilai JSD dan BD metrik MAE, MSE, dan SSIM pada CVAE untuk diskriminasi sawit sehat dan sakit.

Berdasarkan analisis progresif, model CVAE secara konsisten mengungguli model CAE dalam hal stabilitas dan akurasi prediksi. Fluktuasi nilai BD dan JSD yang lebih rendah pada CVAE menunjukkan bahwa model ini lebih andal dan menghasilkan prediksi yang konsisten dari bulan ke bulan. Sementara itu, model CAE menunjukkan kinerja yang tidak stabil, dengan variasi besar dalam akurasi dan kualitas hasil. Oleh karena itu, model CVAE adalah pilihan yang lebih optimal untuk tugas prediksi ini karena memiliki performa yang lebih konsisten dan dapat diandalkan seiring berjalannya waktu.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini berhasil membuktikan bahwa deteksi dini *Ganoderma* pada pohon sawit melalui perubahan warna daun dapat dilakukan menggunakan model *deep learning* CAE dan CVAE. Kedua model mampu mengenali anomali dengan tingkat akurasi yang memuaskan, terutama pada tahap infeksi lanjut.

Berdasarkan analisis *timeseries* selama lima bulan, model CAE menunjukkan efektivitas tertinggi dalam mendeteksi anomali hingga dua bulan sebelum pohon tumbang. Namun, kemampuan deteksinya menurun signifikan pada rentang waktu tiga hingga lima bulan sebelum pohon tumbang. Sebaliknya, model CVAE terbukti lebih unggul dan konsisten. Dengan arsitektur yang lebih sederhana, CVAE secara efektif mampu mendeteksi anomali pada tahap paling awal serangan infeksi *Ganoderma*, bahkan hingga lima bulan sebelum pohon tumbang, ketika gejala visual belum terlihat.

Dengan demikian, tujuan penelitian untuk mengembangkan model deteksi dini dan menganalisis batas kemampuannya telah tercapai. Dalam kurun waktu lima bulan, Model CVAE adalah pilihan yang lebih optimal karena kemampuannya yang luar biasa dalam mendeteksi anomali.

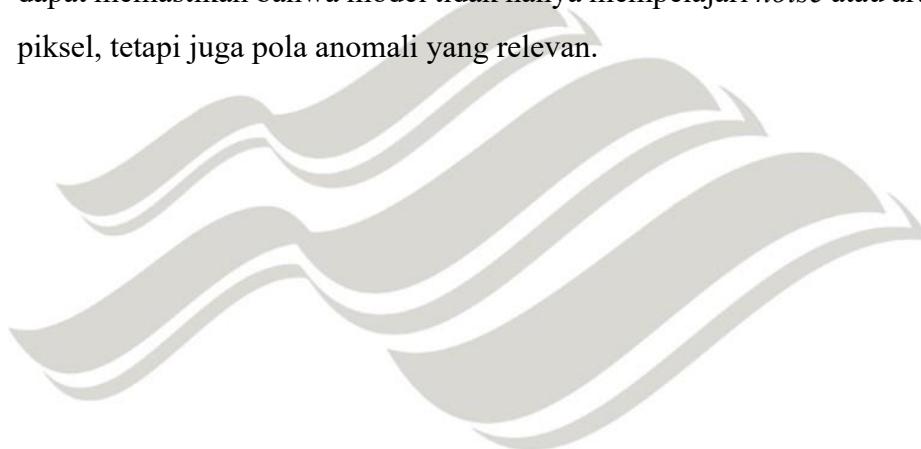
5.2. Saran

Untuk penelitian selanjutnya, ada beberapa aspek kunci yang perlu diperbaiki dari penelitian ini:

- 1. Periode Data *Timeseries* yang Lebih Panjang:** Deteksi dini *Ganoderma* sangat menantang karena gejala awal sulit dikenali. Periode data *timeseries* lima bulan yang digunakan mungkin sudah termasuk fase akhir infeksi. Oleh karena itu, disarankan untuk memperpanjang periode pengumpulan data menjadi **satu hingga lima tahun** untuk mendapatkan gambaran yang lebih akurat tentang perkembangan penyakit dari fase awal hingga akhir.
- 2. Memanfaatkan Pola Spasial Anomali:** Penelitian selanjutnya dapat memanfaatkan fakta bahwa pohon yang terinfeksi *Ganoderma* cenderung

muncul secara acak, sementara pohon yang kekurangan air atau unsur hara biasanya menunjukkan pola kerusakan yang seragam. Memasukkan analisis spasial ini ke dalam model dapat meningkatkan akurasi dan membedakan anomali penyakit dari kondisi stres lainnya.

3. **Peningkatan Kualitas Citra:** Kualitas data sangat memengaruhi performa model. Penelitian di masa depan harus fokus pada peningkatan kualitas citra dengan melakukan pra-pemrosesan yang lebih baik, termasuk normalisasi ukuran (*resizing*) dan pembersihan *noise*. Kualitas citra yang lebih baik dapat memastikan bahwa model tidak hanya mempelajari *noise* atau artefak piksel, tetapi juga pola anomali yang relevan.



UNIVERSITAS
MA CHUNG

DAFTAR PUSTAKA

- Aaryadev., 2024. *Convolutional Autoencoders Anomaly Detection*, [Online] Available at: <<https://github.com/aaryadevchandra/convolutional-autoencoders-anomaly-detection>>
- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C. et al., 2016, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*. doi:<https://doi.org/10.48550/arXiv.1603.04467>
- Ahmadi, P., Mansor, S., Farjad, B., Ghaderpour, E., 2020, Unmanned Aerial Vehicle (UAV)-Based Remote Sensing for Early-Stage Detection of Ganoderma. *Remote Sensing*, **14**(5), 1239. doi:<https://doi.org/10.3390/rs14051239>
- Angel, Y., Turner, D., Parkes, S., Malbeteau, Y., Lucieer, A., McCabe, M. F., 2020, Automated Georectification and Mosaicking of UAV-Based Hyperspectral Imagery from Push-Broom Sensors. *Remote Sensing*, **12**(1). doi:<https://doi.org/10.3390/rs12010034>
- Anuar, I. M., Arof, H. b., Mohd Nor, N. b., Abu Seman, I. b., Masri, M. M., Ibrahim, S. M. et al., 2021, Remote Sensing for Detection of Ganoderma Disease and Bagworm Infestation in Oil Palm. *Advances in Agricultural and Food Research Journal*, **2**(1). doi:<https://doi.org/10.36877/aafjr.a0000189>
- Araújo, S. O., Peres, R. S., Ramalho, J. C., Lidon, F., Barata, J., 2023, Machine Learning Applications in Agriculture: Current Trends, Challenges, and Future Perspectives. *Agronomy*, **13**(12). doi:<https://doi.org/10.3390/agronomy13122976>
- Aszkowski, P., Ptak, B., Kraft, M., Pieczyński, D., Drapikowski, P., 2023, Deepness: Deep neural remote sensing plugin for QGIS. *SoftwareX*, **23**. doi:<https://doi.org/10.1016/j.softx.2023.101495>
- Aziz, D., Rafiq, S., Saini, P., Ahad, I., Gonal, B., Rehman, S. A., 2025, Remote sensing and artificial intelligence: revolutionizing pest management in agriculture. *Frontiers in Sustainable Food Systems*, **9**, 1-16. doi:<https://doi.org/10.3389/fsufs.2025.1551460>
- Azmi, A. N., Bejo, S. K., Jahari, M., Muharam, F. M., Yule, I., & Husin, N. A., 2020, Early Detection of Ganoderma boninense in Oil Palm Seedlings Using Support Vector Machines. *Remote Sensing*, **12**(23), 3920. doi:<https://doi.org/10.3390/rs12233920>
- Badan Pengelola Dana Perkebunan Kelapa Sawit (BPDPKS), 2021, *eNose-G Solusi Deteksi Dini Ganoderma*. [online] Tersedia di: <www.bpdp.or.id:
<https://www.bpdp.or.id/enose-g-solusi-deteksi-dini-ganoderma>>
- Baharim, M. S., Adnan, N. A., Mohd, F. A., Seman, I. A., Anuar, M. I., Latif, Z. A. et al., 2023, Optimization of machine learning classifier using multispectral

- data in assessment of Ganoderma basal stem rot (BSR) disease in oil palm plantation. *International Journal of Information Technology*, 4259–4273.
- Bedi, P., & Gole, P, 2021, Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artificial Intelligence in Agriculture*, 5, 90-101. doi:<https://doi.org/10.1016/j.aiia.2021.05.002>
- Bergmann, D., & Stryker, C., 2023, *Apa itu autoencoder?* [online] Tersedia di: <<https://www.ibm.com/id-id/think/topics/autoencoder>> [diakses pada April 2025]
- Bharudin , I., Ab Wahab, A. F., Abd Samad , M. A., Xin Yie, N., Zairun, M. A., Abu Bakar, F. D., Abdul Murad , A. M., 2022, Review Update on the Life Cycle, Plant–Microbe Interaction, Genomics, Detection and Control Strategies of the Oil Palm Pathogen Ganoderma boninense. *Biology*, 11(2), 251. doi:<https://doi.org/10.3390/biology11020251>
- Chawla, A., 2024, *What is Bhattacharyya Distance?* [online] Available at: <<https://blog.dailydoseofds.com/p/what-is-bhattacharyya-distance>>
- Daranagama, S., and Takeuchi, W., 2025, Smartphone application for detecting and visualizing basal stem rot disease stages in oil palm. *Frontiers in Remote Sensing*, 6, 1-12. doi:<https://doi.org/10.3389/frsen.2025.1553844>
- Deasy, J., Simidjievski, N., and Liò, P., 2020, Constraining Variational Inference with Geometric Jensen-Shannon Divergence. *Advances in Neural Information Processing Systems*, 33, 10647-10658. doi:<https://doi.org/10.48550/arXiv.2006.10599>
- Dozat, T., 2016, Incorporating Nesterov Momentum into Adam. *International Conference on Learning Representations*. [online] Available at: <<https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ>>
- E+E Leader., 2025, *AI and Remote Sensing Combine for Early, Accurate Crop Mapping.* [online] Available at: <<https://c3newsmag.com/ai-and-remote-sensing-combine-for-early-accurate-crop-mapping>>
- Faisal, H. M., Aqib, M., Mahmood , K., Safran , M., Alfarhood, S., Ashraf , I., 2025, A customized convolutional neural network-based approach for weeds identification in cotton crops. *Frontiers in Plant Science*, 15. doi: <https://doi.org/10.3389/fpls.2024.1435301>
- Fuentes-Peñaillillo, F., Gutter, K., Vega, R., Silva, G. C., 2024, Transformative Technologies in Digital Agriculture Leveraging Internet of Things, Remote Sensing, and Artificial Intelligence for Smart Crop Management. *Journal of Sensor and Actuator Networks*, 13(4), 39. doi:<https://doi.org/10.3390/jsan13040039>
- Ginting, E. N., dan Wiratmoko, D. (2021). Potensi dan Tantangan Penerapan Precision Farming dalam Upaya Membangun Perkebunan Kelapa Sawit yang Berkelanjutan. *Warta PPKS*, 55-56.

- Google Colab., 2025, *Google Colaboratory*. [online] Available at: <<https://colab.google/>>
- Isinkaye, F. O., Olusanya, M. O., and Akinyelu, A. A., 2025, A multi-class hybrid variational autoencoder and vision transformer model for enhanced plant disease identification. *Intelligent Systems with Applications*, **26**. doi:<https://doi.org/10.1016/j.iswa.2025.200490>
- Jana, D., Patil, J., Herkal, S., Nagarajaiah, S., Osorio, L. D., 2022, CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction. *Mechanical Systems and Signal Processing*, **169**, 108723,. doi:<https://doi.org/10.1016/j.ymssp.2021.108723>.
- Junarto, R., Djurdjani, Permadi, F. B., Ferdiansyah, D., Admaja, P. K., Sholikin, A. R., Rahmansani, R., 2020, Pemanfaatan Teknologi Unmanned Aerial Vehicle (Uav) Untuk Pemetaan Kadaster. *BHUMI: Jurnal Agraria dan Pertanahan*, **6**(1), 105-118. doi:<https://doi.org/10.31292/jb.v6i1.428>
- Kanna, K., Ramalingam, K., P, P., R, J., and C., P. P., 2024, YOLO deep learning algorithm for object detection in agriculture: a review. *Journal of Agricultural Engineering*, **55**(4). doi:<https://doi.org/10.4081/jae.2024.1641>
- Kementerian Pertanian Republik Indonesia., 2025, *Statistik Perkebunan 2023-2025*. Sekretariat Direktorat Jenderal Perkebunan, Direktorat Jenderal Perkebunan, Kementerian Pertanian.
- KesKes, M., 2025, Review of The Current State of Deep Learning Applications in Agriculture. *Preprints*, **1**. doi:<https://doi.org/10.20944/preprints202504.1290.v1>
- Khoirunnisa, A., Wesnawa, I. A., Citra, I. A., Dwipayana, M., 2024, Deteksi Kerapatan Vegetasi Dan Suhu Permukaan Lahan Kabupaten Banyuwangi Menggunakan Citra Landsat 8-Oli (Studi Kasus Tahun 2023). *Innovative: Journal Of Social Science Research*, **4**(5), 1446–1460. doi:<https://doi.org/10.31004/innovative.v4i5.14882>
- Kingma, D. P., and Ba, J. L., 2014, Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv*. doi:<https://doi.org/10.48550/arXiv.1412.6980>
- Kosasih, D., Nasihin, I., dan Zulkarnain, E. R., 2020, Deteksi Kerapatan Vegetasi dan Suhu Permukaan Tanah Menggunakan Citra Landsat 8 (Studi Kasus: Stasiun Penelitian Pasir Batang Taman Nasional Gunung Ciremai). *Prosiding Seminar Nasional Konservasi untuk Kesejahteraan Masyarakat I*, **1**(1). [online] Tersedia di: <<https://journal.uniku.ac.id/index.php/prosiding-fahutan/article/view/3703>>
- Kurihara, J., Koo, V.-C., Guey, C. W., Lee, Y. P., Abidin, H., 2020, Early Detection of Basal Stem Rot Disease in Oil Palm Tree Using Unmanned Aerial Vehicle-Based Hyperspectral Imaging. *Remote Sensing*, **14**(3), 799. doi:<https://doi.org/10.3390/rs14030799>

- Kwang, C. S., Razak, S. F., Yogarayan, S., Zahisham, M. Z., Tam, T. H., Noor, M. A., 2025, Ganoderma Disease in Oil Palm Trees Using Hyperspectral Imaging and Machine Learning. *Journal of Human, Earth, and Future*, **6**(1). doi:<https://doi.org/10.28991/HEF-2025-06-01-05>
- Lee, C. C., Koo, V. C., Lim, T. S., Lee, Y. P., Abidin, H., 2022, A multi-layer perceptron-based approach for early detection of BSR disease in oil palm trees using hyperspectral images. *Heliyon*, **8**(4). doi: <https://doi.org/10.1016/j.heliyon.2022.e09252>
- Liaghat, S., Mansor , S., Ehsani, R., Shafri, H. Z., Meon, S., Sankaran, S., 2014, Mid-infrared spectroscopy for early detection of basal stem rot disease in oil palm. *Computers and Electronics in Agriculture*, **101**, 48-54. doi:<https://doi.org/10.1016/j.compag.2013.12.012>
- Muhammad, R., Zhang, W., Abbas, Z., Guo, F., Gwiazdzinski , L., 2022, Spatiotemporal Change Analysis and Prediction of Future Land Use and Land Cover Changes Using QGIS MOLUSCE Plugin and Remote Sensing Big Data: A Case Study of Linyi, China. *Land*, **11**(3), 419. doi:<https://doi.org/10.3390/land11030419>
- Neloy, A. A., and Turgeon , M., 2024, A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs. *Machine Learning with Applications*, **17**, 100572. doi:<https://doi.org/10.1016/j.mlwa.2024.100572>
- Ningrum., 2023, *Yuk Pahami Ciri-ciri Pohon Kelapa Sawit Yang Sehat*. [online] Tersedia di: <<https://www.sawitsetara.co/yuk-pahami-ciri-ciri-pohon-kelapa-sawit-yang-sehat/>>
- Ong , J. A., 2021, *Plant disease identification using autoencoder*. Master Thesis, Universiti Teknologi Malaysia, Faculty of Engineering - School of Electrical Engineering. [online] Available at: <<http://dms.library.utm.my:8080/vital/access/manager/Repository/vital:149764>>
- Pamungkas, A., 2023, *Ruang Warna dalam Pengolahan Citra*. [online] Tersedia di: <<https://pemrogramanmatlab.com/2023/09/30/ruang-warna-dalam-pengolahan-citra/>>
- Primakara, 2023, *Pengertian Bahasa Pemrograman Python & Kegunaannya*. [online] Tersedia di: <<https://primakara.ac.id/>> <<https://primakara.ac.id/blog/info-teknologi/python-adalah/>>. [diakses pada Maret 2025]
- PSI Perkebunan, 2023, *Mengatasi Tantangan Pengelolaan Perkebunan Kelapa Sawit Indonesia*. [online] Available at: <<https://perkebunan.bsim.pertanian.go.id/berita/mengatasi-tantangan-pengelolaan-perkebunan-kelapa-sawit-indonesia>>.
- Python Software Foundation, 2025, *General Python FAQ*. [online] Available at: <<https://docs.python.org/3/faq/general.html#what-is-python>>. [Accessed March 2025]

- Rahmana, B. A., Hayata, H., dan Hartawan, R., 2024, Intensitas Serangan Jamur Ganoderma sp Pada Perkebunan Kelapa Sawit Lahan Mineral dan Lahan Pasang Surut. *Jurnal Media Pertanian*, **9**(2), 148-154. doi:<http://dx.doi.org/10.33087/jagro.v9i2.255>
- Revou, 2025, *Apa itu Google Colab?* [online] Tersedia di: <<https://www.revou.co/kosakata/google-colab>>. [diakses pada April 2025]
- Roboflow, 2025, *Agriculture.* [online] Available at: <<https://universe.roboflow.com/browse/agriculture>>.
- Roboflow, 2025, *Computer Vision for Agriculture with Roboflow.* [online] Tersedia di: <<https://roboflow.com/industries/agriculture>>. [Accessed April 2025]
- S, P., 2023, *Unleashing the Power of Autoencoders: Applications and Use Cases.* [online] Available at: <<https://www.analyticsvidhya.com/blog/2023/05/unleashing-the-power-of-autoencoders-applications-and-use-cases/>>. [Accessed April 2025]
- Sagarseth06., n.d., *Implement Convolutional Autoencoder in PyTorch with CUDA.* [online] Available at: <<https://www.geeksforgeeks.org/implement-convolutional-autoencoder-in-pytorch-with-cuda/>> [Accessed April 2025]
- Santoso, H., 2020, Pengamatan dan Pemetaan Penyakit Busuk Pangkal Batang di Perkebunan Kelapa Sawit Menggunakan Unmanned Aerial Vehicle (UAV) dan Kamera Multispektral. *Jurnal Fitopatologi Indonesia*, **16**(2), 69-80. doi:<https://doi.org/10.14692/jfi.16.2.69-80>
- Selek, A., 2020, *plant-disease-classification-pytorch.* [online] Available at: <<https://github.com/abdullahselek/plant-disease-classification-pytorch>>
- Shi, Q., Pan, T., Lu, D., Li, H., Cha, Z., 2025, BPUM: A Bayesian Probabilistic Updating Model Applied to Early Crop Identification. *Journal of Remote Sensing*, **5**(0438). doi:<https://doi.org/10.34133/remotesensing.0438>
- Sipayung, T., 2024, Kerugian Ekonomi Serangan Ganoderma Sawit Dan Ancaman Masa Depan Industri Sawit Nasional. *Diseminasi Policy Brief*, (23). [online] Tersedia di: <<https://palmoilina.asia/jurnal-kelapa-sawit/serangan-ganoderma-sawit-rugi/>>
- Soetopo, D., Manohara, D., Wulandari, S., Djufry, F., Syafaruddin, S., 2022, Ganoderma Diseases on Oil Palm, Factors and Vectors Dispersal And Its Control Strategy Development / Penyakit Ganoderma pada Tanaman Kelapa Sawit, penyebaran dan Vektornya serta Strategi Pengendaliannya. *Perspektif: Review Penelitian Tanaman Industri*, **21**(1), 1-17. doi:<http://dx.doi.org/10.21082/psp.v21n1.2022.1-17>
- Suharyanto, Eris, D. D., Prakoso, H. T., Saragih, A. H., Darmono, T. W., 2016, Perangkat serologi untuk deteksi dini infeksi Ganoderma sp. pada kelapa sawit. *Menara Perkebunan*, **80**(1). doi:<https://doi.org/10.22302/iribb.jur.mp.v80i1.42>

- Tensorflow, 2024, *Convolutional Variational Autoencoder*. [online] Available at: <https://www.tensorflow.org/tutorials/generative/cvae#network_architecture>
- Tensorflow, 2024, *Intro to Autoencoders*. [online] Available at: <https://www.tensorflow.org/tutorials/generative/autoencoder#build_the_model>
- Ultralytics, 2023, November Frequently Asked Questions (FAQ)). *Frequently Asked Questions (FAQ)*. [online] Available at: <<https://docs.ultralytics.com/help/FAQ/>>
- Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., 2004, Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, **13**(4), 600-612. doi:10.1109/TIP.2003.819861
- Wicaksono, W., Prilianti, K. R., Setiawan, H., Mimboro, P., 2022, Metode Deteksi Cepat Serangan Ganoderma pada Perkebunan Kelapa Sawit dengan Penginderaan Jauh. *Journal of Embedded Systems Security and Intelligent Systems*, **3**(2), 135-142. doi:<http://dx.doi.org/10.26858/jessi.v3i2.38092>
- Wikipedia contributors, 2025, *Python (programming language)*. (Wikipedia, The Free Encyclopedia.) [online] Available at: <[https://en.wikipedia.org/w/index.php?title=Python_\(programming_language\)&oldid=1284741561](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=1284741561)> [Accessed March 2025]
- Xiang, P., Ali, S., Zhang , J., Jung, S. K., Zhou, H., 2024, Pixel-associated autoencoder for hyperspectral anomaly detection. *International Journal of Applied Earth Observation and Geoinformation*, **129**. doi:<https://doi.org/10.1016/j.jag.2024.103816>
- Xin, Y., Sun, L., and Hansen, M. C., 2022, Oil palm reconciliation in Indonesia: Balancing rising demand and environmental conservation towards 2050. *Journal of Cleaner Production*, **3800**(2), 1-30. doi:<https://doi.org/10.1016/j.jclepro.2022.135087>.