

**PENGEMBANGAN MODEL CNN LSTM UNTUK PREDIKSI SERANGAN
GANODERMA PADA CITRA UAV PERKEBUNAN KELAPA SAWIT**

TUGAS AKHIR



**VINCENTIUS ADITYA PUTRALIM
NIM: 311910014**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2023**

LEMBAR PENGESAHAN
TUGAS AKHIR

**PENGEMBANGAN MODEL CNN LSTM UNTUK PREDIKSI SERANGAN
GANODERMA PADA CITRA UAV PERKEBUNAN KELAPA SAWIT**

Oleh:
VINCENTIUS ADITYA PUTRALIM
NIM. 311910014

dari:
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS dan TEKNOLOGI
UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer (S.Kom.)

Dosen Pembimbing I,

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

Dosen Pembimbing II,

Hendry Setiawan, ST., M.Kom.
NIP. 20100006

Dekan Fakultas Sains dan Teknologi,



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

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Pengembangan Model CNN LSTM Untuk Prediksi Serangan Ganoderma Pada Citra UAV Perkebunan Kelapa Sawit” 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, 20 Juli 2023



A handwritten signature in black ink, appearing to read "V. Aditya Putralim".

Vincentius Aditya Putralim
NIM. 311910014

PENGEMBANGAN MODEL CNN LSTM UNTUK PREDIKSI SERANGAN GANODERMA PADA CITAR UAV PERKEBUNAN KELAPA SAWIT

Vincentius Aditya Putralim, Kestrilia Rega Prilianti, Hendry Setiawan
Universitas Ma Chung

Abstrak

Serangan Ganoderma pada perkebunan kelapa sawit dapat menyebabkan kerugian yang signifikan. Oleh karena itu, pengembangan model prediktif yang efektif untuk mendeteksi serangan Ganoderma menjadi penting dalam upaya pengendalian dan manajemen penyakit ini. Dalam penelitian ini akan menggunakan model *Convolutional Neural Network Long Short-Term Memory* (CNN LSTM) untuk memprediksi serangan *ganoderma* pada citra *Unmanned Aerial Vehicle* (UAV) perkebunan kelapa sawit dengan melihat indeks vegetasi *Visible Atmospherically Resistant Index* (VARI). Pada eksperimen penelitian ini akan digunakan 6 tuning parameter yakni dengan menggunakan metode optimasi Adam dan SGD serta *learning rate* sebesar 0.001, 0.001, dan 0.0001. Hasil prediksi model dengan keenam tuning parameter tersebut akan diamati pada kelipatan epoch 500 sampai dengan epoch ke-3000. Model yang digunakan pada penelitian ini akan menggunakan layer ConvLSTM2D yang memiliki peran yang sama dengan CNN LSTM, layer tersebut akan melakukan ekstraksi fitur spasial yang ada pada input. Model yang telah dilatih akan dievaluasi dengan menggunakan *Mean Square Error* (MSE) sebagai metrik evaluasi. Hasil evaluasi menunjukan bahwa model yang dikembangkan dapat bekerja dengan baik dalam memprediksi serangan *ganoderma* pada citra UAV perkebunan kelapa sawit. Prediksi serangan yang dihasilkan model ditampilkan dalam bentuk *heatmap*. Model dengan menggunakan metode optimasi Adam dan *learning rate* 0.001 pada epoch ke-2500 merupakan model dengan nilai MSE terendah yang dihasilkan eksperimen dengan nilai MSE sebesar $2,56 \cdot 10^{-5}$.

Kata kunci: *Convolutional Neural Network, Ganoderma, Kelapa Sawit, Long Short-Term Memory, Visible Atmospherically Resistant Index*

DEVELOPMENT OF CNN LSTM MODEL FOR PREDICTION OF GANODERMA ATTACKS ON UAV IMAGES OF OIL PALM PLANTATIONS

Vincentius Aditya Putralim, Kestrilia Rega Prilianti, Hendry Setiawan
Universitas Ma Chung

Abstract

Ganoderma attacks in oil palm plantations can cause significant losses. Therefore, the development of an effective predictive model for detecting Ganoderma attacks is crucial for disease control and management efforts. This research utilizes a Convolutional Neural Network Long Short-Term Memory (CNN LSTM) model to predict Ganoderma attacks on Unmanned Aerial Vehicle (UAV) images of oil palm plantations, using the Visible Atmospherically Resistant Index (VARI) vegetation index. The experiment involves using 6 tuning parameters, employing the Adam and SGD optimization methods with learning rates of 0.001, 0.001, and 0.0001. The predictions of the model with these six tuning parameters are observed at multiples of 500 epochs up to the 3000th epoch. The model used in this research utilizes the ConvLSTM2D layer, which serves the same purpose as the CNN LSTM layer, extracting spatial features from the input. The trained model is evaluated using Mean Square Error (MSE) as the evaluation metric. The evaluation results show that the developed model performs well in predicting Ganoderma attacks on UAV images of oil palm plantations. The predicted attacks are displayed in the form of a heatmap. The model utilizing the Adam optimization method and a learning rate of 0.001 at the 2500th epoch achieves the lowest MSE value among the experiments, with an MSE of $2,56 \cdot 10^{-5}$.

Kata kunci: *Convolutional Neural Network, Ganoderma, Long Short Term-Memory, Oil Palm, Visible Atmospherically Resistant*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas anugerah-Nya tugas dan laporan Tugas Akhir ini dapat selesai dengan baik. Laporan ini ditulis untuk melaporkan hasil pencapaian dari proses penggerjaan Tugas Akhir yang telah dilaksanakan.

Pada kesempatan ini, penulis ingin berterima kasih kepada seluruh pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Ibu Dr. Kestrilia Rega Prilianti, M.Si., Selaku Dekan Fakultas Sains dan Teknologi sekaligus dosen pembimbing dalam proyek Tugas Akhir ini,
2. Bapak Hendry Setiawan, ST., M. Kom., Selaku kepala Program Studi Teknik Informatika Universitas Ma Chung sekaligus dosen pembimbing dan dosen pembimbing akademik dalam proyek Tugas Akhir ini,
3. Orang tua penulis, yang memberikan dukungan dan semangat sehingga proyek Tugas Akhir ini bisa selesai,
4. Dan kepada teman-teman yang sudah memberikan bantuan dan kontribusinya dalam penggerjaan proyek Tugas Akhir ini.

Laporan ini ditulis berdasarkan hasil dari Tugas Akhir dengan judul “Pengembangan Model CNN LSTM Untuk Prediksi Serangan *Ganoderma* Pada Citra UAV Perkebunan Kelapa Sawit”. Tugas Akhir ini adalah mata kuliah yang wajib diselesaikan oleh mahasiswa Teknik Informatika Universitas Ma Chung Malang sebagai salah satu prasyarat kelulusan.

Malang, 20 Juli 2023

Vincentius Aditya Putralim

311910014

DAFTAR ISI

LEMBAR PENGESAHAN TUGAS AKHIR	i
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
Abstrak	iii
Abstract	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	3
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	4
1.5 Tujuan Penelitian	4
1.6 Manfaat	4
1.7 Luaran Penelitian	5
1.8 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	6
2.1 Komoditas Kelapa Sawit	6
2.1.1 Penyakit Busuk Pangkal Batang (<i>ganoderma</i>)	7
2.2 QGIS	8
2.3 Visible Atmospherically Resistant Index (VARI)	8
2.4 Ruang Warna Red Green Blue (RGB)	9
2.5 Python	9
2.6 <i>Convolutional Neural Network</i> (CNN)	10
2.7 <i>Long Short-Term Memory</i> (LSTM)	14
2.8 CNN-LSTM	14
2.9 Keras	17
2.10 Scikit-Learn	17
2.11 Pandas	17
2.12 Numpy	18

2.13 Matplotlib	18
2.14 Geowombat	18
2.15 Google Colaboratory	19
2.16 Visual Studio Code (VS Code)	19
2.17 MSE (<i>Mean Squared Error</i>)	19
2.18 Penelitian Terdahulu	20
BAB III ANALISIS DAN PERANCANGAN SISTEM	22
3.1 Alur Penelitian	22
3.2 Identifikasi Masalah	23
3.3 Analisis Kebutuhan	24
3.3.1 Kebutuhan Peneliti	24
3.3.2 Kebutuhan Pengguna	25
3.4 Studi Pustaka	25
3.5 Pembuatan Dataset	28
3.6 Desain Model Arsitektur CNN LSTM	34
3.7 Pengujian Model Arsitektur CNN LSTM	36
BAB IV HASIL DAN PEMBAHASAN	38
4.1 Profil Dataset	38
4.2 <i>Preprocessing</i> Data	41
4.3 Exsperimen Model CNN LSTM	45
4.3.1 Model CNN LSTM <i>Optimizer</i> Adam Dengan <i>Learning Rate</i> 0.001	45
4.3.2 Model CNN LSTM <i>Optimizer</i> Adam Dengan <i>Learning Rate</i> 0.0001	53
4.3.3 Model CNN LSTM <i>Optimizer</i> Adam Dengan <i>Learning Rate</i> 0.00001	61
4.3.4 Model CNN LSTM <i>Optimizer</i> SGD Dengan <i>Learning Rate</i> 0.001	68
4.3.5 Model CNN LSTM <i>Optimizer</i> SGD Dengan <i>Learning Rate</i> 0.0001	76
4.3.6 Model CNN LSTM <i>Optimizer</i> SGD Dengan <i>Learning Rate</i> 0.00001	83

4.4 Ringkasan Hasil Eksperimen	90
4.4.1 Ringkasan Waktu Proses <i>Training</i> Model	90
4.4.2 Model CNN LSTM	91
BAB V KESIMPULAN DAN SARAN	93
5.1 Kesimpulan	93
5.2 Saran	94
DAFTAR PUSTAKA	95

DAFTAR GAMBAR

Gambar 2.1	Contoh Pohon Kelapa Sawit Terinfeksi <i>Ganoderm</i>	8
Gambar 2.2	Ruang Warna RGB	9
Gambar 2.3	Arsitektur <i>Convolutional Neural Network</i> (CNN)	11
Gambar 2.4	Fungsi Aktivasi ReLU	13
Gambar 2.5	Fungsi Aktivasi Sigmoid	13
Gambar 2.6	Gambar LSTM Cell	14
Gambar 2.7	Arsitektur Layer ConvLSTM2D	16
Gambar 3.1	Alur Penelitian	22
Gambar 3.2	Flow Chart Pembuatan Dataset	28
Gambar 3.3	Citra Pohon Kelapa Sawit Tumbang Yang Telah diberi Tanda	30
Gambar 3.4	Proses Penelusuran Pokok Pohon Kelapa Sawit Terinfeksi <i>ganoderma</i> , (A) Citra Titik Pohon Kelapa Sawit Tumbang Pada Bulan Januari 2022; (B) Citra Titik Pohon Kelapa Sawit Sebelum Tumbang Pada Bulan Desember 2021	30
Gambar 3.5	Perbandingan Koordinat Pohon Kelapa Sawit, (A) Titik Sebelum Buffering; (B) Titik Sesudah Buffering	31
Gambar 3.6	Contoh Hasil Grid Pada Citra Perkebunan Kelapa Sawit	33
Gambar 3.7	Desain Sistem Model <i>Convolutional Neural Network Long Short Term Memory</i> Untuk Deteksi Penyakit <i>Ganoderma</i>	34
Gambar 3.8	Plot Model CNN LSTM	35
Gambar 3.9	Hasil Output Nilai VARI Model	37
Gambar 3.10	Perbandingan Kondisi Titik Pohon Kelapa Sawit Sebenarnya Pada Citra UAV dan Hasil Representasi Pada Heatmap	37
Gambar 4.1	<i>Boxplot</i> Nilai VARI Pada Setiap Kondisi di Kedua Dataset	39
Gambar 4.2	<i>Barchart Mean</i> VARI Pada Setiap Kondisi Dikedua Dataset	40
Gambar 4.3	<i>Heatmap</i> Dari Array Pertama Yang Merupakan Representasi Kondisi Pada Bulan Agustus	43
Gambar 4.4	<i>Heatmap</i> Array y Pada Data Series Test	44
Gambar 4.5	Grafik <i>Loss</i> Model Adam Lr 0.001	46
Gambar 4.6	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.001 Pada Epoch 500	46

Gambar 4.7	Grafik <i>Loss</i> Model Adam Lr 0.001 Pada Epoch 500	47
Gambar 4.8	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.001 Pada Epoch 1000	48
Gambar 4.9	Grafik <i>Loss</i> Model Adam Lr 0.001 Pada Epoch 500-1000	48
Gambar 4.10	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.001 Pada Epoch 1500	49
Gambar 4.11	Grafik <i>Loss</i> Model Adam Lr 0.001 Pada Epoch 1000-1500	49
Gambar 4.12	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.001 Pada Epoch 2000	50
Gambar 4.13	Grafik <i>Loss</i> Model Adam Lr 0.001 Pada Epoch 1500-2000	50
Gambar 4.14	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.001 Pada Epoch 2500	51
Gambar 4.15	Grafik <i>Loss</i> Model Adam Lr 0.001 Pada Epoch 2000-2500	51
Gambar 4.16	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.001 Pada Epoch 3000	52
Gambar 4.17	Grafik <i>Loss</i> Model Adam Lr 0.001 Pada Epoch 2500-3000	52
Gambar 4.18	Grafik <i>Loss</i> Model Adam Lr 0.0001	54
Gambar 4.19	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.0001 Pada Epoch 500	54
Gambar 4.20	Grafik <i>Loss</i> Model Adam Lr 0.0001 Pada Epoch 500	55
Gambar 4.21	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.0001 Pada Epoch 1000	55
Gambar 4.22	Grafik <i>Loss</i> Model Adam Lr 0.0001 Pada Epoch 500-1000	56
Gambar 4.23	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.0001 Pada Epoch 1500	56
Gambar 4.24	Grafik <i>Loss</i> Model Adam Lr 0.0001 Pada Epoch 1000-1500	57
Gambar 4.25	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.0001 Pada Epoch 2000	57
Gambar 4.26	Grafik <i>Loss</i> Model Adam Lr 0.0001 Pada Epoch 1500-2000	58
Gambar 4.27	<i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.0001 Pada Epoch 2500	58
Gambar 4.28	Grafik <i>Loss</i> Model Adam Lr 0.0001 Pada Epoch 2000-2500	59

Gambar 4.29 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.0001 Pada Epoch 3000	59
Gambar 4.30 Grafik <i>Loss</i> Model Adam Lr 0.0001 Pada Epoch 2500-3000	60
Gambar 4.31 Grafik <i>Loss</i> Model Adam Lr 0.00001	61
Gambar 4.32 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.00001 Pada Epoch 500	62
Gambar 4.33 Grafik <i>Loss</i> Model Adam Lr 0.00001 Pada Epoch 500	62
Gambar 4.34 Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.00001 Pada Epoch 1000	63
Gambar 4.35 Grafik <i>Loss</i> Model Adam Lr 0.00001 Pada Epoch 500-1000	63
Gambar 4.36 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.00001 Pada Epoch 1500	64
Gambar 4.37 Grafik <i>Loss</i> Model Adam Lr 0.00001 Pada Epoch 1000-1500	64
Gambar 4.38 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.00001 Pada Epoch 2000	65
Gambar 4.39 Grafik <i>Loss</i> Model Adam Lr 0.00001 Pada Epoch 1500-2000	65
Gambar 4.40 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.00001 Pada Epoch 2500	66
Gambar 4.41 Grafik <i>Loss</i> Model Adam Lr 0.00001 Pada Epoch 2000-2500	66
Gambar 4.42 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model Adam Lr 0.00001 Pada Epoch 3000	67
Gambar 4.43 Grafik <i>Loss</i> Model Adam Lr 0.00001 Pada Epoch 2500-3000	67
Gambar 4.44 Grafik <i>Loss</i> Model SGD Lr 0.001	69
Gambar 4.45 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.001 Pada Epoch 500	69
Gambar 4.46 Grafik <i>Loss</i> Model SGD Lr 0.001 Pada Epoch 500	70
Gambar 4.47 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.001 Pada Epoch 1000	70
Gambar 4.48 Grafik <i>Loss</i> Model SGD Lr 0.001 Pada Epoch 500-1000	71
Gambar 4.49 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.001 Pada Epoch 1500	71
Gambar 4.50 Grafik <i>Loss</i> Model SGD Lr 0.001 Pada Epoch 1000-1500	72

Gambar 4.51 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.001 Pada Epoch 2000	72
Gambar 4.52 Grafik <i>Loss</i> Model SGD Lr 0.001 Pada Epoch 1500-2000	73
Gambar 4.53 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.001 Pada Epoch 2500	73
Gambar 4.54 Grafik <i>Loss</i> Model SGD Lr 0.001 Pada Epoch 2000-2500	74
Gambar 4.55 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.001 Pada Epoch 3000	74
Gambar 4.56 Grafik <i>Loss</i> Model SGD Lr 0.001 Pada Epoch 2500-3000	75
Gambar 4.57 Grafik <i>Loss</i> Model SGD Lr 0.0001	76
Gambar 4.58 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.0001 Pada Epoch 500	76
Gambar 4.59 Grafik <i>Loss</i> Model SGD Lr 0.0001 Pada Epoch 500	77
Gambar 4.60 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.0001 Pada Epoch 1000	77
Gambar 4.61 Grafik <i>Loss</i> Model SGD Lr 0.0001 Pada Epoch 500-1000	78
Gambar 4.62 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.0001 Pada Epoch 1500	78
Gambar 4.63 Grafik <i>Loss</i> Model SGD Lr 0.0001 Pada Epoch 1000-1500	79
Gambar 4.64 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.0001 Pada Epoch 2000	79
Gambar 4.65 Grafik <i>Loss</i> Model SGD Lr 0.0001 Pada Epoch 1500-2000	80
Gambar 4.66 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.0001 Pada Epoch 2500	80
Gambar 4.67 Grafik <i>Loss</i> Model SGD Lr 0.0001 Pada Epoch 2000-2500	81
Gambar 4.68 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.0001 Pada Epoch 3000	81
Gambar 4.69 Grafik <i>Loss</i> Model SGD Lr 0.0001 Pada Epoch 2500-3000	82
Gambar 4.70 Grafik <i>Loss</i> Model SGD Lr 0.00001	83
Gambar 4.71 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.00001 Pada Epoch 500	83
Gambar 4.72 Grafik <i>Loss</i> Model SGD Lr 0.00001 Pada Epoch 500	84

Gambar 4.73 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.00001 Pada Epoch 1000	84
Gambar 4.74 Grafik <i>Loss</i> Model SGD Lr 0.00001 Pada Epoch 500-1000	85
Gambar 4.75 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.00001 Pada Epoch 1500	85
Gambar 4.76 Grafik <i>Loss</i> Model SGD Lr 0.00001 Pada Epoch 1000-1500	86
Gambar 4.77 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.00001 Pada Epoch 2000	86
Gambar 4.78 Grafik <i>Loss</i> Model SGD Lr 0.00001 Pada Epoch 1500-2000	87
Gambar 4.79 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.00001 Pada Epoch 2500	87
Gambar 4.80 Grafik <i>Loss</i> Model SGD Lr 0.00001 Pada Epoch 2000-2500	88
Gambar 4.81 <i>Heatmap</i> Perbandingan Hasil Prediksi dan <i>Ground Truth</i> Model SGD Lr 0.00001 Pada Epoch 3000	88
Gambar 4.82 Grafik <i>Loss</i> Model SGD Lr 0.00001 Pada Epoch 2500-3000	89
Gambar 4.83 Grafik Batang Perbandingan Nilai MSE Hasil Eksperimen	92

DAFTAR TABEL

Tabel 3.1	Ringkasan Hasil Penelitian Terdahulu	26
Tabel 3.2	Rincian Dataset Citra UAV	29
Tabel 4.1	Rincian <i>Dataset Train</i> dan <i>Test</i>	38
Tabel 4.2	<i>Dataframe</i> Pada <i>Dataset Train</i>	38
Tabel 4.3	<i>Dataframe</i> Pada <i>Dataset Test</i>	39
Tabel 4.4	Perbandingan Nilai <i>Mean</i> , Maksimal dan Minimal VARI	40
Tabel 4.5	Dataframe Setelah Diberi Label <i>Date Time</i>	41
Tabel 4.6	Rincian Nilai Limitasi Koordinat Dataset	42
Tabel 4.7	Rincian Nilai MSE Pada Data <i>Test</i> Dimodel Adam Lr 0.001 Di setiap 500 Epoch	53
Tabel 4.8	Rincian Nilai MSE Pada Data <i>Test</i> Dimodel Adam Lr 0.0001 Di setiap 500 Epoch	60
Tabel 4.9	Rincian Nilai MSE Pada Data <i>Test</i> Dimodel Adam Lr 0.00001 Di setiap 500 Epoch	68
Tabel 4.10	Rincian Nilai MSE Pada Data <i>Test</i> Dimodel SGD Lr 0. 001 Di setiap 500 Epoch	75
Tabel 4.11	Rincian Nilai MSE Pada Data <i>Test</i> Dimodel SGD Lr 0. 0001 Di setiap 500 Epoch	82
Tabel 4.12	Rincian Nilai MSE Pada Data <i>Test</i> Dimodel SGD Lr 0. 00001 Di setiap 500 Epoch	89
Tabel 4.13	Rincian Waktu Proses <i>Training</i> Model Setiap 500 Epoch	90
Tabel 4.14	Perbandingan Selisih Waktu Pelatihan dan Nilai MSE Pada Model Adam Lr 0.001	91
Tabel 4.15	Rincian Nilai MSE Seluruh Model Pada Kelipatan 500 Epoch	92

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kelapa sawit merupakan komoditas perkebunan yang cukup berpengaruh di sektor perkebunan dalam menghasilkan pendapatan negara (Abdina, 2019). Sehingga untuk memaksimalkan tanaman kelapa sawit ini tidak heran jika banyak perusahaan yang berlomba-lomba dan serius untuk mengembangkan sistem yang dapat digunakan untuk memaksimalkan setiap tanaman kelapa sawit milik perusahaan. *Precision agriculture* merupakan salah satu contoh sistem yang dapat diterapkan untuk dapat memaksimalkan tanaman sawit mulai dari mempercepat pertumbuhan tanaman kelapa sawit, mempercepat panen, memperpanjang umur, mencegah penyakit, hingga memaksimalkan hasil produksi dari pohon sawit tersebut.

Untuk mempertahankan produktivitas tanaman kelapa sawit agar tetap berproduksi dengan baik, maka diperlukanlah pengendalian penyakit untuk mencegah menurunnya produksi kelapa sawit yang telah dihasilkan. Penyakit Ganoderma sendiri merupakan penyakit yang sering menjangkit dan umum diderita oleh tanaman kelapa sawit, ganoderma sendiri memiliki sifat parasitik dan saprophytic dan biasanya hidup ditanah, penyakit Ganoderma dapat menyebabkan pembusukan akar dan batang sehingga dapat menyebabkan kematian dan kegagalan panen terhadap tanaman kelapa sawit tersebut, dengan demikian perlu untuk menentukan tanaman kelapa sawit mana yang terindikasi terjangkit penyakit Ganoderma untuk dapat dilakukan pencegahan terhadap pohon tersebut ataupun pemusnahan tanaman kelapa sawit untuk menghindari penularan yang mungkin terjadi pada perkebunan (Suwartono, 2018). Gejala umum yang ditunjukkan tanaman kelapa sawit yang terjangkit Ganoderma antara lain, daun berwarna hijau pucat, daun muda yang terbentuk mengurang, daun tua menjadi layu, pelepas banyak yang patah dan menggelantung (Corteva, 2020).

Pada penelitian yang telah dilakukan Putralim (2023) didapatkan hasil bahwa nilai RGB dan VARI yang dimiliki tanaman kelapa sawit disetiap kondisi yakni 4 hingga 1 bulan tanaman kelapa sawit tersebut tumbang memiliki pola

menurun dan memiliki pola nilai RGB dan VARI yang berbeda antara tanaman kelapa sawit sehat dan yang terserang Ganoderma. RGB merupakan model warna yang menggabungkan cahaya merah, hijau, dan biru yang digunakan untuk mendapatkan rangkaian warna yang lebih luas, sedangkan VARI atau *Visible Atmospherically Resistant Index* yang merupakan indeks vegetasi yang digunakan untuk memperkirakan *vegetation fraction*.

Sehingga berlandaskan penelitian yang dilakukan Putralim (2023) tersebut bahwa pembaruan dari penelitian ini adalah dapat melakukan deteksi dini dengan memprediksi tanaman kelapa sawit yang terkena *ganoderma* beberapa bulan sebelum tanaman kelapa sawit tersebut tumbang dikarenakan penyakit *ganoderma*. Dengan kebutuhan untuk melakukan prediksi pada data series nilai RGB dan VARI yang dihasilkan pada citra UAV dari tanaman kelapa sawit, maka digunakanlah *Long Short-Term Memory* (LSTM) sebagai teknik yang spesifik berfungsi untuk deteksi data series. Beberapa hasil yang telah dikembangkan untuk melakukan prediksi menggunakan teknik LSTM pada data series yakni prediksi harga saham berdasarkan data historis / data series (Fischer & Krauss, 2018), dan juga LSTM digunakan untuk memprediksi arus lalu lintas di wilayah perkotaan (Wang & Xu, 2018).

Pada penelitian tugas akhir ini berfokus pada pengembangan model CNN LSTM yang merupakan model spatio temporal, konsep spatio temporal sendiri mengacu pada kombinasi dimensi spasial yang berkaitan dengan lokasi fisik dalam ruang dan temporal berkaitan dengan waktu atau urutan peristiwa, pada model CNN LSTM CNN akan bekerja pada bagian spatio sedangkan LSTM akan bekerja pada bagian temporal fungsi dari model ini akan digunakan sebagai alat untuk melakukan prediksi penyakit ganoderma pada tanaman kelapa sawit dengan input citra UAV. Arsitektur CNN LSTM sendiri sudah banyak digunakan dan terbukti mencapai kinerja yang baik dalam beberapa tugas yang diberikan seperti pengenalan tindakan manusia dalam sebuah video (Shahroudy et al., 2016), pengenalan emosi pada ucapan (Zhao et al., 2019), dan prediksi harga saham(Fischer & Krauss, 2018). Sehingga pada penelitian ini akan berfokus menggunakan model CNN LSTM yang akan digunakan untuk melakukan prediksi

tanaman kelapa sawit yang terserang Ganoderma atau tidak dengan menggunakan citra UAV pada perkebunan kelapa sawit.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang sudah dipaparkan, diperoleh identifikasi masalah berdasarkan dari hasil pada penelitian sebelumnya didapat hasil bahwa nilai RGB dan VARI yang diperlihatkan oleh tanaman kelapa sawit yang terkena ganoderma 1, 2, 3, 4 bulan sebelum tumbang dan dinyatakan ganoderma memiliki pola unik dan jika dibandingkan dengan nilai RGB dan VARI dari tanaman kelapa sawit yang sehat terdapat perbedaan nilai RGB dan VARI antara tanaman kelapa sawit sakit dan sehat, sehingga dampak yang timbul dari terlambatnya dideteksi tanaman kelapa sawit yang terserang *ganoderma* akan menimbulkan kerugian panen dan penularan penyakit pada tanaman kelapa sawit lain disekitar tanaman kelapa sawit yang terserang *ganoderma* tersebut.

1.3 Batasan Masalah

Batasan masalah yang ada dalam proyek ini adalah sebagai berikut.

- a. Citra Input dihasilkan dengan menggunakan UAV dari pihak partner yakni PT Perkebunan Nusantara (PTPN)
- b. Citra input yang digunakan merupakan citra UAV dengan format warna RGB.
- c. Citra input yang digunakan memiliki rentang waktu 1 bulan pada setiap citra dan pada lokasi yang sama sebagai dataset *training model*.
- d. Penggunaan *feature* pada metode CNN LSTM hanya berfokus pada nilai indeks vegetasi VARI dari tanaman kelapa sawit.
- e. Penggunaan dataset nilai indeks vegetasi VARI terbatas pada rentang hingga 4 bulan tanaman kelapa sawit sebelum tumbang dan dinyatakan Ganoderma
- f. Input citra UAV yang digunakan hanya terbatas sebanyak 64 x 64 pohon kelapa sawit yang ada pada citra UAV

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah yang disampaikan, rumusan masalah dari penelitian ini sebagai berikut.

Bagaimana membangun model spatio temporal untuk deteksi dini penyakit *ganoderma* pada tanaman kelapa sawit.

1.5 Tujuan Penelitian

Tujuan dari pengerjaan proyek penelitian tugas akhir ini sebagai berikut.

Mengembangkan dan mengimplementasikan model spatio temporal yakni model *deep learning* CNN LSTM yang dibuat untuk dapat melakukan prediksi nilai VARI tanaman kelapa sawit yang terserang *ganoderma* atau tidak dengan menggunakan input citra UAV perkebunan kelapa sawit.

1.6 Manfaat

Manfaat dari pengerjaan penelitian ini sebagai berikut.

- a. Bagi penulis, dapat menambah wawasan dan keterampilan baru dalam membuat model CNN LSTM yang diperlukan untuk memprediksi tanaman kelapa sawit yang terserang Ganoderma atau tidak pada citra UAV dengan melihat nilai indeks vegetasi VARI pada tanaman kelapa sawit tersebut.
- b. Bagi Universitas khususnya Program Studi Teknik Informatika, dapat membantu mempersiapkan lulusan yang siap kerja dan kompeten dengan memberikan bekal kepada mahasiswa berupa proses pembelajaran yang intens selama kegiatan penelitian Tugas Akhir.
- c. Bagi Masyarakat khususnya petani sawit dan golongannya, model CNN LSTM yang dihasilkan pada penelitian Tugas Akhir dapat manfaatkan dan dipergunakan untuk melakukan prediksi penyakit ganoderma yang akan terjadi pada tanaman kelapa sawit dengan melihat nilai VARI yang ada pada tanaman kelapa sawit melalui citra UAV sebagai input model.

1.7 Luaran Penelitian

Luaran hasil penelitian ini merupakan model CNN LSTM yang dipergunakan untuk melakukan prediksi serangan Ganoderma yang ada pada citra UAV. Selain itu, akan dilanjutkan dengan publikasi ilmiah ke jurnal.

1.8 Sistematika Penulisan

Sistematika dalam penulisan proposal Tugas Akhir ini akan dibagi menjadi lima bab seperti berikut.

Bab I Pendahuluan

Pada bab pendahuluan ini berisi latar belakang, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian, luaran tugas akhir dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab tinjauan pustaka ini berisi uraian sistematis terkait dengan literatur yang digunakan dalam proses penyusunan Tugas Akhir sehingga diperoleh landasan teori terkait pengolahan citra UAV, CNN, LSTM, dan penyakit Ganoderma pada tanaman kelapa sawit.

Bab III Metodologi Penelitian

Bab ini menjelaskan tahapan pengerjaan serta analisis perancangan awal sistem yang akan dibuat. Tahapan ini sendiri terdiri atas identifikasi masalah, studi pustaka, pengumpulan data, desain sistem dan pengujian.

Bab IV Hasil dan Pembahasan

Bab ini menjelaskan tahapan pengerjaan serta analisis perancangan awal sistem yang akan dibuat. Tahapan ini sendiri terdiri atas identifikasi masalah, studi pustaka, pengumpulan data, desain sistem dan pengujian.

Bab V Kesimpulan dan Saran

Bab ini berisi simpulan dari hasil penelitian Tugas Akhir yang telah dilakukan beserta saran yang mungkin dapat dilakukan untuk memperbaiki sistem aplikasi untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Komoditas Kelapa Sawit

Komoditas kelapa sawit merupakan salah satu komoditas dalam sektor perkebunan yang memiliki dampak cukup besar dalam membantu membangun perekonomian di Indonesia. Dengan dampaknya yang cukup besar dalam membantu membangun perekonomian di Indonesia, pihak perusahaan ataupun pelaku usaha perkebunan kelapa sawit perlu melakukan tindakan yang tepat dalam menjaga tingkat produktivitas tanaman kelapa sawit yang mereka kelola untuk dapat menghasilkan produktivitas yang maksimal dan efisien (Suwartono, 2018).

Dalam praktik dan pengelolaan dilapangan tanaman kelapa sawit memiliki berbagai macam penyakit yang dapat menyerang ataupun menginfeksi tanaman kelapa sawit. Salah satu contoh penyakit yang menjadi momok menakutkan para pelaku usaha perkebunan kelapa sawit di asia tenggara yakni penyakit *ganoderma* (Hushiaran et al., 2013).

Tanaman kelapa sawit yang telah terinfeksi penyakit *ganoderma* akan menjadi permasalahan yang serius bagi pengelola kebun kelapa sawit, dampak yang terjadi akan menjadi lebih signifikan jika penyakit *ganoderma* tersebut menginfeksi pada tanaman kelapa sawit didalam perkebunan tua yang menyebabkan penurunan yang signifikan pada tingkat produktivitas tanaman kelapa sawit tersebut dan dapat hingga menyebabkan kematian pada pohon kelapa sawit yang telah terinfeksi penyakit *ganoderma*.

Pada perkebunan kelapa sawit yang telah terinfeksi *ganoderma* terdapat 2 permasalahan umum yang pasti terjadi pada perkebunan kelapa sawit tersebut yakni permasalahan pertama yaitu mengurangnya populasi pohon kelapa sawit yang ada pada perkebunan yang terinfeksi *ganoderma* bahkan persentase tumbangnya pohon kelapa sawit pada perkebunan yang terinfeksi *ganoderma* dapat mencapai 50 sampai dengan 60% dari populasi pohon generasi 3 dan 4 yang ada pada perkebunan kelapa sawit tersebut, dengan kerugian yang ditaksir hingga Rp 210 miliar pertahunnya pada skala luas kebun sebesar 10.000 hektare. Sedangkan permasalahan kedua yang pasti terjadi pada tanaman kelapa sawit

yang terinfeksi penyakit *ganoderma* yakni penurunan berat buah tanaman kelapa sawit yang berefek pada menurunnya produksi yang dihasilkan dari setiap pohon tanaman kelapa sawit dan secara tidak langsung juga akan menurunkan pendapatan dimiliki oleh pemilik kebun kelapa sawit (Suwartono, 2018).

2.1.1 Penyakit Busuk Pangkal Batang (*ganoderma*)

Penyakit *ganoderma* atau biasa juga disebut dengan penyakit busuk pangkal batang. Merupakan penyakit tanaman kelapa sawit yang disebabkan oleh jamur *ganoderma applanatum*, *ganoderma psuedofferum*, dan *ganoderma lucidum*, jamur-jamur ini menyerang tanaman kelapa sawit melalui pangkal batang pada tanaman kelapa sawit. Efek samping yang muncul dan terlihat pada tanaman kelapa sawit yang terinfeksi penyakit *ganoderma* ini akan memiliki pangkal batang pohon yang lunak dan membusuk. Penyakit *ganoderma* sendiri dapat menular dengan kondisi apabila akar pohon tanaman kelapa sawit yang sehat bersentuhan dengan akar pohon tanaman kelapa sawit yang terinfeksi *ganoderma* ataupun juga bersentuhan dengan sisa-sisa pohon tanaman kelapa sawit yang terinfeksi *ganoderma* (Corteva, 2020).

Waktu yang dibutuhkan *ganoderma* untuk membunuh pohon tanaman kelapa sawit saat pertama kali *ganoderma* tersebut muncul di batang pohon kelapa sawit yakni 6 sampai dengan 24 bulan pada pohon kelapa sawit yang masih belum berbuah atau berproduksi, sedangkan pada pohon kelapa sawit yang telah berproduksi atau berbuah kematian pohon kelapa sawit akan terjadi pada kurun waktu 2 sampai dengan 3 tahun setelah munculnya gejala pertama *ganoderma* pada pohon kelapa sawit (Hendarjati, 2014).

Contoh gambar pohon kelapa sawit yang telah terjangkit *ganoderma* dapat dilihat pada gambar 2.1. Gejala umum yang dapat timbul pada pohon kelapa sawit yang terinfeksi *ganoderma* antara lain.

- Daun berwarna hijau pucat.
- Daun tua menjadi layu.
- Daun muda yang tumbuh mulai sedikit.
- Pelepas pohon kelapa sawit banyak yang patah dan menggantung.
- Pangkal batang berubah menjadi warna hitam.

- Terdapat getah ataupun lendir pada bagian pohon kelapa sawit yang terinfeksi
- Batang pohon kelapa sawit membusuk dan berwarna coklat muda



Gambar 2.1 Contoh Pohon Kelapa Sawit Terinfeksi *Ganoderma*

(Sumber: www.gdm.id/ganoderma/)

2.2 QGIS

Quantum geographic information system (QGIS) merupakan tools *geographic information system* yang rilis pada tahun 2008 dan dapat diakses secara open-source, gratis, dan *cross-platform*, QGIS sendiri juga mendukung bahasa pemrograman python dan C++ sebagai plugin yang dapat digunakan didalam QGIS. Secara teknis QGIS berintegrasi dengan *geospatial data abstraction library* (GDAL), yang mana *library* tersebut membantu QGIS untuk dapat membaca, memproses dan menvisualisasikan *geographic image* yang memiliki ukuran besar (Moyroud & Portet, 2018).

2.3 Visible Atmospherically Resistant Index (VARI)

Visible Atmospherically Resistant Index (VARI) merupakan indeks vegetasi yang dipergunakan untuk dapat memperkirakan *vegetation fraction* secara kuantitatif dengan menggunakan rentang spektrum yang terlihat atau biasa disebut dengan rentang spektrum *red, green, blue* (RGB) dengan mengurangi perbedaan efek atmosfer dan iluminasi yang ada pada citra digital (Gitelson et al., 2002).

Rumus perhitungan yang digunakan untuk mendapatkan nilai VARI sebagai berikut.

$$VARI = \frac{(Green-Red)}{(Green+Red-Blue)} \quad (2-1)$$

Keterangan:

Red: nilai pixel dari band merah

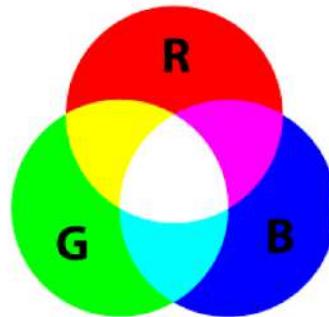
Green: nilai pixel dari band hijau

Blue: nilai pixel dari band biru

2.4 Ruang Warna Red Green Blue (RGB)

Ruang warna *red*, *green*, *blue* (RGB) merupakan model warna yang menggabungkan cahaya merah, hijau dan biru untuk mereproduksi rangkaian warna baru yang lebih luas. Nama model ini sendiri berasal dari inisial setiap warna primer aditif yakni merah, hijau, dan biru yang ditujukan untuk menghasilkan representasi dan tampilan citra yang lebih baik dan menarik (Trivusi, 2022).

Representasi ruang warna RGB dapat dilihat pada gambar 2.3 untuk melihat representasi warna yang dihasilkan dari gabungan tiga warna primer aditif yang digunakan pada model ruang warna RGB.



Gambar 2.2 Ruang Warna RGB

2.5 Python

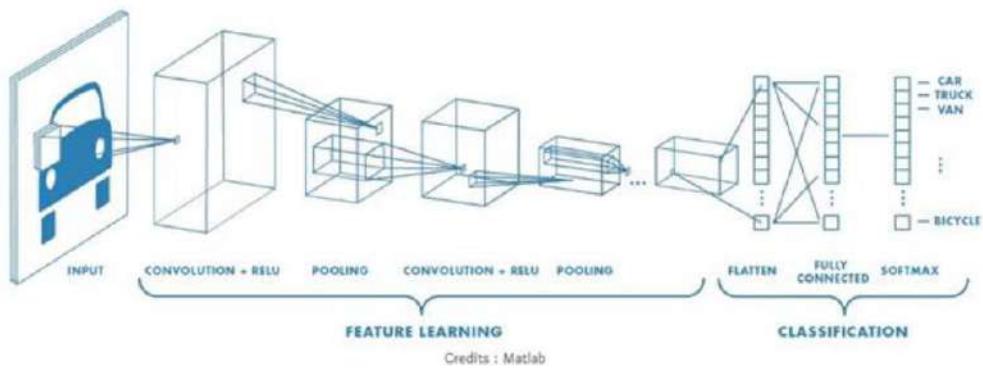
Python merupakan bahasa pemrograman yang diperuntukan untuk banyak hal atau dapat juga disebut bahasa pemrograman multifungsi, didevelop oleh Guido Van Rossum pada sekitar tahun 1980 sebagai pengganti bahasa ABC. Diperkenalkan secara terbuka pada tahun 1991 hingga saat ini python sudah

dikembangkan sebanyak tiga tahap yakni pada tahun 1990, 1995, dan awal tahun 2000 dengan versi terbaru yang digunakan saat ini yaitu Python versi 3.

Adapun kelebihan dan kekurangan yang ada pada bahasa pemograman Python yakni, Python dianggap mudah untuk dipelajari oleh para pemula dikarenakan kode yang ada didalamnya mudah untuk dibaca dan dengan sedikit baris koding sudah dapat menjalankan banyak fungsi yang tingkat kompleksitasnya cukup tinggi, hal ini dapat terjadi karena bahasa pemograman Python sendiri memiliki banyak library yang dapat digunakan untuk berbagai permasalahan sesuai dengan fungsinya masing-masing, dengan dasar metode *Object Oriented Programming* (OOP) bahasa pemograman Python ini dapat memiliki library-library tersebut. Sedangkan kekurangan yang dirasakan ketika menggunakan bahasa pemograman Python antara lain dibutuhkan waktu yang cukup lama untuk menjalankan program yang telah dibuat, dan untuk pengembangan program pada platform berbasis android dan IOS masih kurang mendukung (KOMINFO, 2019).

2.6 *Convolutional Neural Network (CNN)*

Convolutional neural network (CNN) merupakan algoritma deep learning yang yang algoritmanya memiliki layer-layer yang berfungsi untuk memproses dan mengekstraksi fitur dari data yang masuk kedalam alogritma CNN. Algoritma CNN ini biasa digunakan untuk objek deteksi pada gambar dan meproses gambar input tersebut (Setiawan, 2021). Cara kerja dari algoritma CNN dengan yakni dengan cara memecah input gambar/citra menjadi citra yang ukurannya lebih kecil dari citra asli dan tumpeng tindih, setelah dilakukan pemecahan algoritma CNN akan memasukan hasil citra yang telah dipecah tersebut ke *small neural network* dan disimpan kedalam sebuah array yang baru, lalu CNN akan melanjutkan algortimanya dengan melakukan *downsampling* yang bertujuan untuk mengambil nilai pixel terbesar yang ada disetiap *pooling kernel*, terakhir algoritma CNN akan membuat prediksi dari hasil yang telah ada (Lina, 2019). Arsitektur *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 2.4.



Gambar 2.3 Arsitektur *Convolutional Neural Network* (CNN)

Backpropagation dalam *deep learning* secara umum merupakan salah satu metode pelatihan, merupakan metode pelatihan terawasi atau biasa disebut *supervised learning* dengan menggunakan jaringan *multi-layer* yang memiliki ciri untuk meminimalkan *error* yang dihasilkan jaringan pada *output* (Prasetyo, 2019).

Ada banyak metode optimasi yang ada dan dapat digunakan dalam *deep learning*, beberapa metode optimasi yang ada pada *deep learning* antara lain sebagai berikut.

- ***Gradient Descent***

Merupakan salah satu metode optimasi paling populer dalam *deep learning* yang memiliki tujuan untuk mencari nilai parameter (koefisien) yang dapat memminimumkan fungsi biaya (cost) dengan mengikuti turunan parsial (gradien) dari fungsinya (Brownlee, 2019).

- ***Stochastic Gradient Descent (SGD)***

Stochastic Gradient Descent (SGD) merupakan salah satu variasi metode optimasi hasil turunan metode optimasi *Gradient Descent*, dimana pada setiap iterasinya gradien dihitung secara stokastik. Stokastik memiliki maksud yakni gradien dihitung dengan mengambil subset dari data secara acak dengan tujuan untuk dapat mempercepat konvergensi dan menghindari terjebak pada minimum lokal (Srinivasan, 2019).

- ***Adaptive Moment Estimation (Adam)***

Merupakan salah satu metode optimasi yang populer digunakan pada *deep learning*. Metode optimasi Adam merupakan hasil pengembangan dari

algoritma SGD yang mengkombinasikan gradien pertama dan gradien kedua untuk mengatur learning rate yang ada pada setiap iterasi. Gradien stokastik yang ada pada metode optimasi Adam dihitung dengan menggunakan *moving average* yang dihasilkan dari gradien pertama dan kedua. *Moving average* ini digunakan untuk memprediksi gradien aktual yang ada pada setiap iterasi. Selain *moving average* metode optimasi Adam ini juga menggunakan *learning rate* yang diatur secara adaptif pada setiap parameter yang ada pada model (Kingma & Ba, 2015).

- ***Root Mean Square Propagation (RMSprop)***

RMSprop sering dan populer digunakan untuk mengatasi masalah learning rate yang tidak stabil seperti pada metode optimasi SGD. *Moving average* yang dihitung pada metode RMSprop dihitung dari gradien kuadrat untuk mengatur *learning rate* yang ada pada setiap iterasi. Moving average pada metode optimasi RMSprop digunakan untuk memperbarui nilai learning rate yang ada pada setiap parameter model. Bertujuan untuk mempercepat konvergensi dan juga menghindari terjebak pada minimum lokal (Hinton & Tieleman, 2012).

- ***Adagrad***

Sama halnya dengan metode optimasi RMSprop metode Adagrad juga dapat digunakan untuk mengatasi masalah learning rate yang tidak stabil pada metode optimasi SGD. Metode Adagrad memperbarui learning rate pada setiap parameter secara adaptif, sehingga pengoptimalan dapat lebih akurat pada setiap iterasi yang ada (Duchi et al., 2011).

- ***Nadam***

Nadam merupakan algoritma optimasi yang menggabungkan dua buah algoritma berbeda menjadi satu. Algoritma tersebut yakni algoritma RMSprop dengan *momentum Nesterov*. Pada metode optimasi Nadam, moving average dari gradien pertama dan kedua bertujuan untuk memprediksi gradien aktual pada setiap iterasi. Sedangkan penggunaan teknik Nesterov Accelerated Gradient pada metode Nadam bertujuan untuk mempercepat konvergensi pada metode ini (Dozat, 2016).

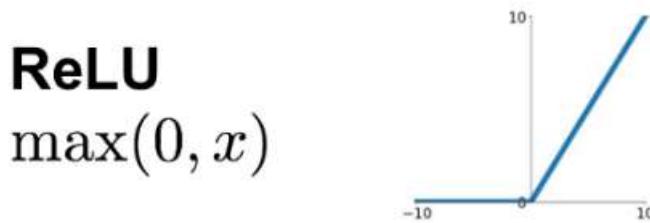
Selain metode optimasi ada juga berbagai fungsi aktivasi yang dimiliki pada *deep learning*. Fungsi aktivasi pada *deep learning* bertujuan untuk melakukan mengaktifkan atau menonaktifkan *neuron* yang ada pada jaringan saraf (Julpan et al., 2015). Beberapa contoh fungsi aktivasi pada *deep learning* sebagai berikut.

- **ReLU**

Merupakan salah satu fungsi aktivasi yang sering digunakan, fungsi aktivasi ReLU membuat pembatas pada bilangan nol, yang memiliki arti apabila $x \leq 0$ maka $x = 0$ dan bila $x > 0$ maka $x = x$.

Kelebihan dari fungsi aktivasi ReLU antara lain mudah dihitung dan efisien dalam proses pelatihan, dan dapat mengatasi masalah *vanishing gradient* pada jaringan saraf tiruan dalam (Goodfellow et al., 2016).

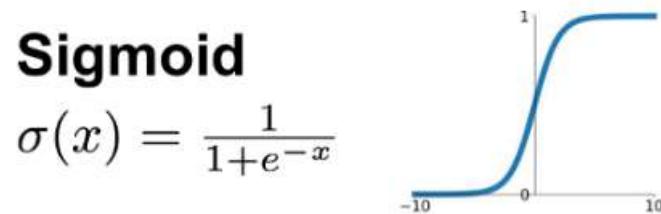
Gambar fungsi aktivasi ReLU dapat dilihat pada pada gambar 2.4.



Gambar 2.4 Fungsi Aktivasi ReLU

- **Sigmoid**

Sama dengan fungsi aktivasi ReLU, fungsi aktivasi Sigmoid juga merupakan salah satu fungsi aktivasi yang populer digunakan. Dengan karakteristik kontinyu dan tidak menurun secara monoton. *Output* yang dihasilkan bersifat *nonlinear* (Julpan et al., 2015). Gambar fungsi aktivasi Sigmoid dapat dilihat pada gambar 2.5.

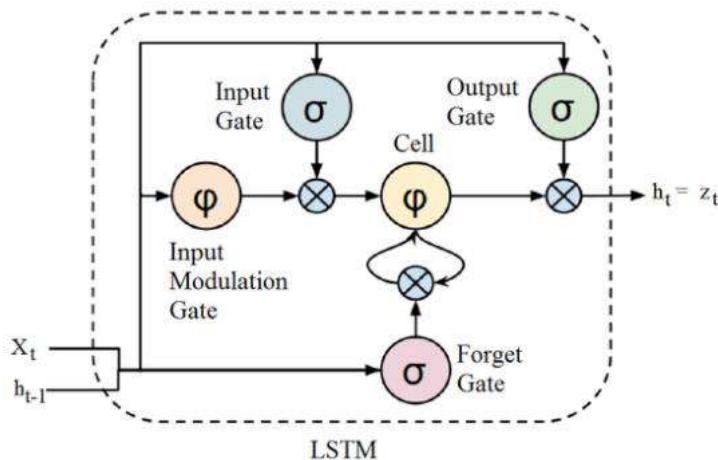


Gambar 2.5 Fungsi Aktivasi Sigmoid

2.7 Long Short-Term Memory (LSTM)

Long short-term memory network (LSTM) merupakan algoritma deep learning dengan tipe *Recurrent Neural Network* (RNN) yang dapat dan sering digunakan untuk mempelajari data *time series*. Menjadi salah satu algoritma terbaik yang ada pada algoritma *deep learning* untuk mempelajari informasi jangka panjang dengan performa yang sangat baik (Setiawan, 2021). LSTM ini dikembangkan berdasarkan adanya keterbatasan pada algoritma RNN dalam melakukan *vanishing gradient* pada proses data *sequential* panjang (Girsang, 2019).

LSTM menggunakan struktur sel yang terdiri dari beberapa gerbang (*gate*) yang berfungsi sebagai pengatur aliran informasi kedalam atau keluar sel. Terdapat tiga gate yang ada pada LSTM antara lain *forget gate* berfungsi sebagai kontrol banyaknya informasi yang akan dilupakan dari sel sebelumnya, *input gate* berfungsi sebagai kontrol banyaknya informasi yang akan disimpan pada sel, dan *output gate* berfungsi sebagai kontrol banyaknya informasi yang akan dikeluarkan dari sel. Gambar 2.5 merupakan contoh gambar *cell* LSTM.



Gambar 2.6 Gambar LSTM Cell

2.8 CNN-LSTM

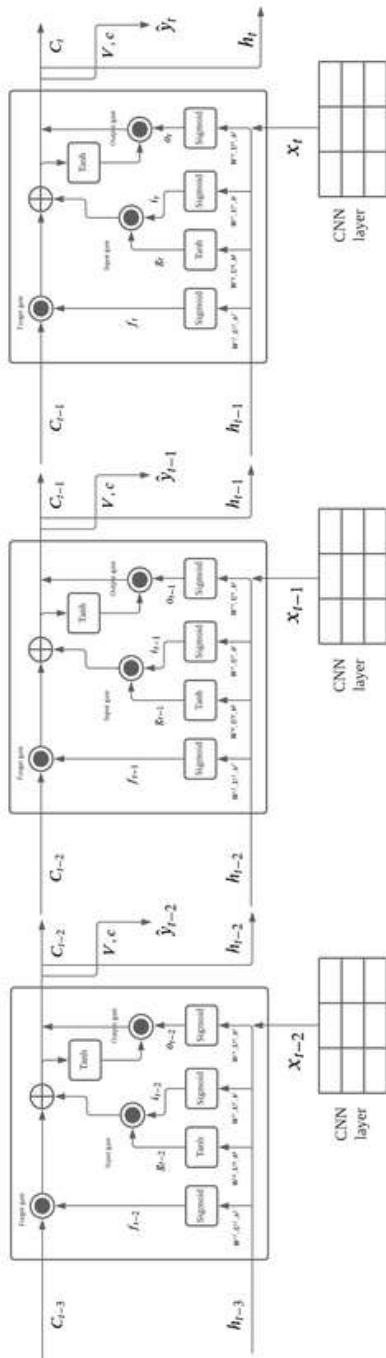
CNN LSTM merupakan *neural networks* yang biasa digunakan dalam menganalisis data spatio temporal, data spatio temporal merupakan data yang

memiliki ruang dan waktu. Cara kerja dan konsep dari CNN LSTM bergantung pada penggabungan kedua arsitektur tersebut untuk mengekstrak fitur secara spasial dan temporal. CNN LSTM efektif digunakan untuk menganalisis data spatio temporal seperti video ataupun gambar. CNN menerapkan filter yang digunakan untuk menangkap pola dan fitur yang ada pada data spasial, sedangkan pada LSTM digunakan untuk data temporal. Cara kerja dari CNN LSTM dimulai dengan menggunakan *layer* konvolusi untuk mengekstrak fitur-fitur spasial dari input. Hasil dari ekstrak fitur akan dilanjutkan ke *layer* LSTM yang akan memproses data sekuensial dan juga mempertahankan informasi kontekstual penting yang diperlukan untuk memahami urutan data yang ada pada data input. Secara umum CNN LSTM digunakan untuk memproses data sekuensial berdimensi tinggi, seperti gambar, video, dan teks dalam konteks NLP (*Natural Language Processing*).

Pada penelitian tugas akhir ini digunakan layer ConvLSTM2D yang memiliki peran yang sama dengan CNN LSTM beberapa fungsi dari layer tersebut antara lain, pertama layer ConvLSTM2D memadukan konvolusi spasial dan memori jangka panjang pada setiap selnya, dan dapat mengekstrak fitur-fitur spasial yang penting dari input. Kedua layer ConvLSTM2D dapat digunakan untuk mengatasi vanishing gradient, hal ini disebabkan karena layer ConvLSTM2D menggunakan *gating mechanism* pada setiap selnya untuk mengatur aliran informasi yang ada dalam jaringan. Ketiga penggunaan layer ConvLSTM2D dapat terbukti meningkatkan akurasi prediksi yang dihasilkan. Arsitektur layer ConvLSTM2D dapat dilihat pada gambar 2.6.

Cara kerja layer ConvLSTM2D yakni pertama input data akan masuk ke dalam layer lalu data tersebut akan melalui beberapa layer konvolusi yang memiliki tujuan untuk mengekstrak fitur-fitur yang penting pada data *input*, selanjutnya pada setiap sel, layer ConvLSTM2D melakukan 2 perhitungan yakni perhitungan memori jangka panjang (LSTM) dan *output*, pada memori jangka panjang ini akan menyimpan informasi tentang input yang telah diproses sebelumnya dan setiap sel juga menerima *input* dari sel sebelumnya dan *output* dari sel saat ini. Perhitungan *output* pada setiap sel, layer ConvLSTM2D

menggunakan informasi yang telah dihasilkan memori jangka panjang dan input dari sel sebelumnya. *Output* dapat digunakan sebagai representasi dari input yang telah diproses. Terakhir *output* seluruh sel akan dilanjutkan ke layer selanjutnya yang ada pada arsitektur jaringan saraf tiruan.



Gambar 2.7 Arsitektur Layer ConvLSTM2D

2.9 Keras

Keras merupakan *library* berbasis *open-source* yang ditulis dengan bahasa pemrograman Python yang dirancang dan berfungsi untuk menyederhanakan model dari kerangka Deep Learning. Library keras ini juga dapat dijalankan diatas *framework* kecerdasan buatan *Tensorflow*, *Theano*, dan masih banyak lagi. Keras saat ini dianggap sebagai *library machine learning* terbaik di bahasa pemograman Python, dalam menyusun model, memproses dataset, menvisualisasikan grafik dan hal lain keras juga telah menyediakan utilitas terbaik dalam memproses pekerjaan-pekerjaan tersebut (Fachrizal, 2019).

2.10 Scikit-Learn

Library Scikit-Learn merupakan library yang sangat erat kaitannya dengan NumPy dan SciPy dirilis pada tahun 2007, dianggap sebagai *library* yang bekerja dengan performa baik dalam data kompleks. *Library* ini berisi algoritma-algoritma yang digunakan untuk menerapkan pembelajaran mesin standar dan bertugas untuk *data mining* seperti melakukan regresi, klasifikasi, pengelompokan data, pengurangan dimensi, dan melakukan pemilihan model (Fachrizal, 2019). Algoritma yang ada pada *Scikit-Learn* antara lain regresi, klasifikasi, *decision tree*, *neural networks*, *svm*, dan *naïve bayes*. Fungsi lain yang diberikan khususnya oleh *Library Scikit-Learn* yakni metode ensemble, deteksi outlier, manipulasi fitur, pemilihan dan validasi model (Gifari, 2020).

2.11 Pandas

Pandas merupakan library yang disediakan dalam ruang lingkup bahasa pemograman Python, ditulis oleh Wes McKinney yang merperan sebagai penulis utama *library* pandas ini. Merupakan sebuah *library* yang menyediakan rutinitas intuitif dan terintegrasi dalam melakukan manipulasi dan analisis data (McKinney, 2011).

Pandas sendiri dapat digunakan dalam banyak ruang lingkup antara lain analisis web, ilmu sosial, keuangan, dan ekonomi melalui pengolahan data yang diolah dengan menggunakan *library* pandas (DataCamp, 2001).

2.12 Numpy

Merupakan *fundamental package* yang biasa diperuntukan untuk *scientific computing* yang telah disediakan oleh bahasa pemrograman Python. *Library* Numpy menyediakan berbagai macam objek turunan seperti *masked array*, *multidimensional array* dan matriks, dan menyediakan juga operasi cepat yang dapat dilakukan didalam array seperti operasi logika, matematika, manipulasi bentuk, *sorting*, *selecting*, dan sejenisnya (NumPy, 2008).

2.13 Matplotlib

Library yang disediakan dalam bahasa pemrograman Python yang diperuntukan untuk melakukan tugas-tugas seperti melakukan *ploting* 2D, *power spectra*, histogram, *bar chart*, *scatterplots*, *error charts*, dan sejenisnya, hanya dengan menulis beberapa baris koding. *Library* matplotlib juga memberikan fitur untuk melakukan *styling* sedemikian rupa seperti mengganti *line style*, *axes properties*, *front properties*, dan sejenisnya melalui antarmuka objek orientasi (Hunter et al., 2010).

2.14 Geowombat

Geowombat merupakan *library* hasil dari gabungan beberapa *library* yang ada didalam bahasa pemrograman Python antara lain *library* pandas, geopandas, rasterio, xarray, dan dask. Fitur utama yang dimiliki *library* geowombat yakni terbiasa diperuntukan untuk menangani banyak file secara langsung sekaligus memproses *raster data* geospasial dan data *time series* berskala besar. Memiliki keunggulan dapat dengan mudah memproses data landsat, sentinel, *plantoscope*, RGB dan sejenisnya. Fungsi lain yang disediakan dalam *library* geowombat ini adalah melakukan *extract* data, point / polygon raster *sampling*, dan melakukan analisis data *time series* (GeoWombat, 2020).

2.15 Google Colaboratory

Google colaboratory merupakan layanan yang dikembangkan oleh *google* sebagai layanan berbasis *cloud* yang mereplikasi *Jupyter Notebook* didalam *cloud*. Beberapa keuntungan yang diberikan oleh *google colaboratory* kepada pengguna dalam mengembangkan proyek dengan basis bahasa pemograman Python antara lain *library* yang telah terinstall didalam *cloud google colaboratory* sehingga pengguna tidak perlu melakukan *installing library* umum yang sering digunakan, *file* dapat langsung disimpan didalam *cloud*, dapat melakukan kolaborasi dalam mengerjakan proyek secara bersamaan bersama dengan rekan kerja, dan juga penggunaan CPU dan TPU dilakukan di *cloud* sehingga tidak menggunakan CPU maupun GPU lokal milik pengguna yang mana ini berguna untuk meringankan kerja pc pengguna (Wibowo, 2021).

2.16 Visual Studio Code (VS Code)

Visual Studio Code merupakan editor teks yang memiliki kemampuan untuk digunakan dalam berbagai bahasa pemograman antara lain bahasa pemograman Python, Java, JavaScript, C++, C, dan masih banyak lagi. VS code dapat diinstal pada berbagai sistem operasi dan merupakan teks editor yang populer digunakan oleh pengembang. Kelebihan yang VS code berikan antara lain terdapat extension yang dapat diinstall dan digunakan untuk berbagai kebutuhan dengan didukung bahasa pemograman yang luas dan juga teks editor ini dapat digunakan gratis.

2.17 MSE (*Mean Squared Error*)

Mean Squared error sebuah tolak ukur untuk mengevaluasi rata-rata kesalahan kuadrat antara nilai aktual dan nilai prediksi dalam sebuah model. MSE biasa digunakan untuk mengevaluasi estimasi besarnya nilai kesalahan pada nilai prediksi.

$$\mathbf{MSE} = \sum \frac{(Y' - Y)^2}{n} \quad (2-2)$$

Keterangan:

Y': Nilai Prediksi

Y : Nilai Sebenarnya

n : Jumlah Data

2.18 Penelitian Terdahulu

Dasar dari dilakukannya proyek penelitian ini didasari dari hasil penelitian yang telah dilakukan (Wicaksono, 2022), bahwa indeks vegetasi VARI dapat digunakan sebagai input untuk melakukan deteksi penyakit *ganoderma* pada citra pohon kelapa sawit, dengan menggunakan metode ANN hasil penelitian tersebut mendapatkan akurasi tertinggi sebesar 82.67 dan *loss* 0.4283 dengan menggunakan *input median* VARI, sendangkan pada *input mean* VARI didapatkan akurasi sebesar 74% dan *loss* 0.4872.

Penggunaan indeks vegetasi VARI juga didukung pada penelitian yang telah dilakukan oleh Putralim (2023), pada penelitian tersebut dilakukan *profiling* terhadap indeks vegetasi VARI pada pohon kelapa sawit dengan kondisi 1 sampai dengan 4 bulan sebelum pohon kelapa sawit tersebut tumbang dan dinyatakan *ganoderma* dengan indeks vegetasi VARI pada pohon kelapa sawit sehat. Didapatkan hasil bahwa *mean* maupun *median* dari indeks vegetasi VARI memiliki rentang nilai yang berbeda-beda pada setiap kondisinya, dimana pola unik dapat terlihat pada hasil *profiling* tersebut yakni penurunan nilai indeks vegetasi VARI pada setiap kondisinya kecuali pada kondisi pohon kelapa sawit 1 bulan sebelum tumbang yang memiliki sedikit peningkatan nilai.

Pada penelitian yang dilakukan oleh Wicaksono (2022) dan Putralim (2023) belum dikembangkan model CNN LSTM yang dapat digunakan untuk memprediksi penyakit *ganoderma* yang akan muncul pada bulan-bulan setelahnya. Sehingga pada proyek penelitian tugas akhir ini akan berfokus pada pengimplementasian dan pengembangan model CNN LSTM yang digunakan sebagai model prediksi serangan *ganoderma* pada citra pohon kelapa sawit. Sehingga pelaku industri kelapa sawit dapat melakukan pencegahan dini terhadap pohon kelapa sawit yang telah terindikasi terserang penyakit *ganoderma* dan dapat meminimalisir kerugian yang mungkin terjadi.

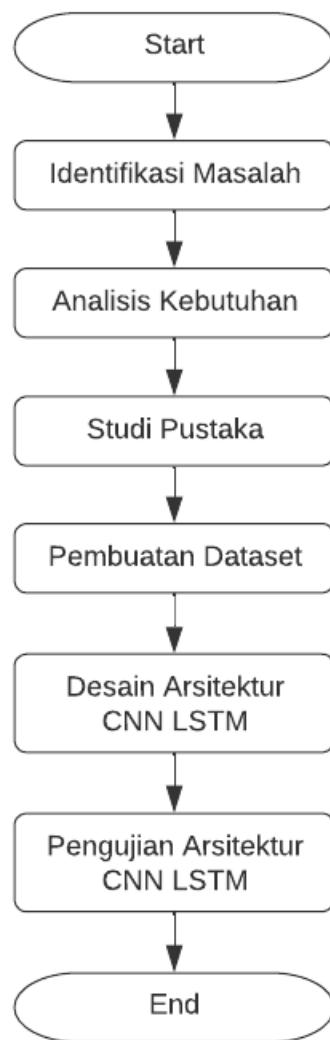
Penggunaan model CNN LSTM yang dilakukan pada penelitian tugas akhir ini mengacu pada penelitian yang dilakukan oleh Hopping (2020), dimana model CNN LSTM tersebut digunakan untuk melakukan prediksi risiko kecelakaan di kota New York, model yang dibuat pada penelitian tersebut menghasilkan hasil yang baik, dimana model tersebut mendapatkan nilai koefisien korelasi Michael sebesar 0.84. Terdapat juga pernyataan bahwa model CNN LSTM yang dilakukan oleh Sun et al (2019) memberikan hasil bahwa model CNN LSTM yang digunakan untuk melakukan prediksi hasil panen kedelai memiliki kinerja yang lebih unggul dibandingkan model CNN ataupun model LSTM murni. Pengujian kinerja tersebut dilakukan pada 2 kondisi yakni pada akhir masa panen dan pada masa panen kedelai yang mana hasil dari pengujian tersebut model CNN LSTM memiliki kinerja yang lebih unggul dibanding model CNN ataupun model LSTM murni.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Alur Penelitian

Dalam proyek tugas akhir ini memiliki tujuan untuk dapat melakukan deteksi penyakit *ganoderma* pada pohon kelapa sawit pada citra UAV dengan mengembangkan dan menggunakan arsitektur CNN LSTM. Alur penelitian untuk proyek tugas akhir ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur Penelitian

Penelitian ini diawali dengan melakukan identifikasi masalah yang muncul dan memiliki tujuan yang jelas akan permasalahan yang dibahas. Dilanjutkan dengan melakukan studi pustaka yang berguna untuk mendapatkan ilmu dan sudut pandang lain dari penelitian yang telah ada yang berkaitan dengan proyek tugas akhir ini sehingga dapat dikembangkan menjadi lebih baik. Setelah dilakukan studi pustaka selanjutnya dilakukan pembuatan dataset dari citra UAV perkebunan kelapa sawit yang nantinya akan menjadi input didalam model CNN LSTM yang digunakan pada penelitian ini. Alur akhir pada penelitian ini yakni melakukan desain model CNN LSTM beserta melakukan pengujian terhadap model CNN LSTM yang telah dibuat dan digunakan pada proyek tugas akhir ini sehingga dapat diuji apakah model CNN LSTM pada proyek penelitian tugas akhir ini telah memenuhi kriteria untuk dapat mendeteksi penyakit *ganoderma* pada citra UAV atau belum dapat memenuhi kriteria untuk dapat mendeteksi penyakit *ganoderma* tersebut. Setiap tahap pada alur penelitian ini akan dipaparkan lebih lanjut pada setiap sub bab berikut.

3.2 Identifikasi Masalah

Masalah yang timbul dalam perkebunan kelapa sawit yakni melakukan monitoring kondisi setiap pohon kelapa sawit yang mana setiap perkebunan pasti memiliki luas wilayah yang sangat luas sehingga waktu yang diperlukan dalam melakukan monitoring juga tidak sedikit. Sehingga perlu adanya pengembangan sistem cepat yang dapat dengan baik melakukan prediksi kemungkinan terjadinya infeksi penyakit khususnya penyakit *ganoderma* yang mungkin akan menginfeksi pohon kelapa sawit yang ada pada kebun kelapa sawit hanya dengan menggunakan citra UAV untuk dapat mengefisiensikan waktu yang diperlukan pihak pengelola kebun kelapa sawit dengan cara memprediksi keadaan serangan penyakit *ganoderma* pada citra bulan selanjutnya dengan memantau keadaan pada bulan-bulan sebelumnya.

3.3 Analisis Kebutuhan

Dalam melakukan proyek penelitian tugas akhir dalam mengembangkan sistem cepat prediksi penyakit *ganoderma* pada tanaman kelapa sawit, dibutuhkan analisis kebutuhan agar penelitian dapat dilaksanakan dengan lancar dan dapat berjalan dengan baik. Cakupan dari analisis yang diperlukan untuk melaksanakan penelitian tugas akhir ini antara lain kebutuhan perangkat keras dan perangkat lunak dari sisi peneliti dan pengguna.

3.3.1 Kebutuhan Peneliti

Berikut merupakan beberapa kebutuhan perangkat keras dan perangkat lunak yang digunakan peneliti dalam melakukan proyek penelitian tugas akhir ini.

1. Perangkat Keras
 - a. Laptop
 - i. Prosesor: AMD Ryzen 5-3550H 2.1GHz
 - ii. Ram: 8 GB
 - iii. SSD: 240 GB
 - iv. HDD: 1 TB
 - v. GPU: AMD Radeon RX 560X
 - vi. Sistem Operasi: Windows 11 Home 64 bit
2. Perangkat Lunak
 - a. Google Colaboratory
 - b. Python 3
 - c. Qgis
 - d. Pandas
 - e. Numpy
 - f. Matplotlib
 - g. Scikit-Learn
 - h. Keras
 - i. Tensorflow

3.3.2 Kebutuhan Pengguna

Pengembangan model CNN LSTM ini berguna khususnya pada pelaku industri kebun kelapa sawit, yang mana pada penelitian ini bekerja sama dengan pihak PTPN dapat terbantu perihal pemantauan dan pengambilan tindakan cepat untuk pohon kelapa sawit yang terindikasi terserang penyakit *ganoderma* sehingga pihak manajemen dapat meminimalisir kerugian dengan berbagai cara yang paling tepat sesuai dengan prosedur yang dimiliki setiap perusahaan. Pengguna sendiri membutuhkan *Google Colaboratory* dan Qgis untuk dapat menggunakan model yang diimplementasikan dan dikembangkan pada proyek penelitian ini. Qgis pada sisi pengguna digunakan untuk melakukan *cropping* citra UAV perkebunan kelapa sawit sesuai dengan batasan masalah pada model yang dibuat ini dengan menggunakan 64 x 64 pohon kelapa sawit pada citra input. Output yang diberikan dari model ini berupa citra dalam *heatmap* yang menggambarkan nilai VARI pada setiap *grid* yang merupakan representasi setiap pohon kelapa sawit yang ada pada citra UAV.

3.4 Studi Pustaka

Tahap ini dilakukan studi pustaka terhadap penelitian terdahulu yang masih berkaitan dengan penelitian tugas akhir ini dan akan dijadikan panduan dan acuan dalam menyelesaikan masalah yang ada pada penelitian tugas akhir ini. Penelitian terdahulu yang masih berkaitan yang dijadikan referensi merupakan penelitian yang berkaitan dengan prediksi penyakit *ganoderma* pada pohon kelapa sawit khususnya pada citra pohon kelapa sawit serta penelitian yang berkaitan dengan nilai RGB dan indeks vegetasi VARI pada pohon kelapa sawit. Sehingga tujuan dari studi pustaka ini untuk dapat menambah wawasan mengenai deteksi penyakit *ganoderma* pada pohon kelapa sawit khususnya pada citra pohon kelapa sawit. Ringkasan penelitian terdahulu dapat dilihat pada tabel 3.1.

Tabel 3.1 Ringkasan Hasil Penelitian Terdahulu

No	Topik	Pengetahuan	Temuan
1	Indeks nilai VARI sebagai tolak ukur mengetahui kondisi pohon kelapa sawit	Indeks VARI dapat digunakan untuk menprediksi kondisi pohon kelapa sawit	Terdapat korelasi antara nilai VARI dengan kondisi pohon kelapa sawit, yang mana jika nilai VARI mendekati 1 maka pohon kelapa sawit dikategorikan sebagai pohon sehat dan sebaliknya (Nur Anisa et al., 2020).
2	Penggunaan citra UAV sebagai alat pengamatan untuk pohon kelapa sawit	Terdapat pembahasan nilai RGB dan indeks vegetasi dapat diperuntukan untuk mengamati pertumbuhan pohon kelapa sawit	RGB dan indeks vegetasi yang diperoleh dari citra UAV dijadikan tolak ukur tingkat kesehatan pohon kelapa sawit (Tugi et al., 2015).
3	Deteksi penyakit <i>ganoderma</i> dengan <i>artificial intelligence</i>	Beberapa metode AI dapat diperuntukan guna mendeteksi penyakit <i>ganoderma</i> beserta keparahannya	Metode ANN, CNN, GAN merupakan metode yang masih dapat berpotensi untuk dikembangkan dalam upaya mendeteksi penyakit <i>ganoderma</i> , sedangkan metode terbaik yang disarankan pada penelitian ini yakni metode SVM (Tee et al., 2021).
4	Metode deteksi penyakit <i>ganoderma</i>	Berdasarkan jenis data inputnya ada berbagai metode yang dapat digunakan untuk mendeteksi penyakit <i>ganoderma</i>	Citra UAV (multispectral data) merupakan salah satu input data yang dapat digunakan untuk mendeteksi penyakit <i>ganoderma</i> pada

Tabel 3.1 Tabel Lajutan

No	Topik	Pengetahuan	Temuan
			pohon kelapa sawit beserta tingkat terinfeksinya dengan menggunakan beberapa teknik yang dibahas pada penelitian ini antara lain teknik tomografi dan sensor radar, <i>spatial mapping</i> , <i>Intteligent Electronic Nose</i> (e-nose), Terrestrial Laser Scanning (Baharim et al., 2021)
5	Model CNN LSTM untuk prediksi risiko kecelakaan di Kota New York	Model CNN LSTM dapat digunakan untuk melakukan prediksi nilai maupun keadaan pada masa depan	Model CNN LSTM yang digunakan berdasarkan arsitektur Keras ConvLSTM2D dapat bekerja dengan baik untuk melakukan prediksi risiko kecelakaan di kota New York dengan koefisien korelasi Michael sebesar 0.84
6	Model CNN LSTM untuk prediksi hasil panen kedelai	Bahwa model CNN LSTM yang diusulkan dapat digunakan untuk melakukan prediksi hasil panen kedelai	Model CNN LSTM yang diusulkan memiliki kinerja yang lebih unggul dibandingkan dengan model CNN atau LSTM murni, baik pada saat melakukan prediksi panen di akhir musim maupun pada saat musim kedelai (Sun et al., 2019).

3.5 Pembuatan Dataset

Pada proyek penelitian tugas akhir ini *flowchart* pembuatan dataset dapat dilihat pada gambar 3.2 untuk lebih jelasnya pada setiap langkah akan dibahas pada bab 3.5 ini.



Gambar 3.2 Flow Chart Pembuatan Dataset

Pembuatan dataset akan didapat dari 6 file citra UAV perkebunan kelapa sawit yang dimiliki oleh pihak PTPN sebagai pihak *partner* dari Kelompok Riset Precision Agriculture. Keenam citra UAV tersebut diambil dari jarak foto 300M pada rentang pengambilan citra perbulan, 6 citra tersebut merupakan citra pada bulan Agustus 2021 sampai dengan bulan Januari 2022. Rincian dataset citra UAV yang digunakan pada proyek penelitian ini dapat dilihat pada tabel 3.2.

Citra yang digunakan pada penelitian ini akan menggunakan keenam citra yang ada, dengan acuan citra bulan januari 2022 sebagai penanda pohon kelapa sawit tersebut yang ada pada citra merupakan pohon kelapa sawit yang terinfeksi penyakit *ganoderma* atau pohon kelapa sawit sehat. Sedangkan citra bulan Agustus, September, Oktober, November, dan Desember 2022 digunakan nantinya sebagai dataset time series yang akan digunakan pada model CNN

LSTM yang akan dibuat pada proyek penelitian tugas akhir ini. Pada penelitian ini akan dilakukan terlebih dahulu penandaan koordinat pohon kelapa sawit yang telah tumbang pada citra UAV bulan Januari 2022 sebagai indikator bahwa pohon kelapa sawit tersebut terinfeksi *ganoderma* atau pohon kelapa sawit tersebut sehat. Hasil koordinat yang telah dibuat akan disimpan kedalam sebuah *shape file* yang nantinya akan digunakan sebagai acuan melakukan pencarian pada citra UAV bulan-bulan sebelumnya untuk mendapatkan *extract* nilai RGB, VARI, *longitude*, *latitude*, dan koordinat x, y dari pohon kelapa sawit tersebut.

Tabel 3.2 Rincian Dataset Citra UAV

No	Nama File	Ekstensi	Bulan	Tahun	Ukuran
1	Ortho_GUB_4_300M	TIF	Agustus	2021	9.06 GB
2	Gub_afd_IV_09_2021_300M_Ortho	TIF	September	2021	9.45 GB
3	GUB_AFD_IV_300M_10_2021_Ortho	TIF	Oktober	2021	9.34 GB
4	Gub_Afd_IV_11_2021_300m_Ortho	TIF	November	2021	9.65 GB
5	GUB_AFD_IV_DES_300M_Ortho	TIF	Desember	2021	38.2 GB
6	GUB_Januari22_ORTHO	TIF	Januari	2022	38.9 GB

Pada proyek penelitian tugas akhir ini akan berfokus pada pengambilan nilai RGB dan VARI pada pohon kelapa sawit yang ada pada citra UAV bulan Agustus sampai dengan bulan Desember 2021 yang telah dipotong pada bagian tertentu untuk meringankan beban model CNN LSTM, menurut keterangan dari pihak PTPN seluruh pohon kelapa sawit yang telah tumbang pada citra UAV bulan Januari 2022 merupakan pohon kelapa sawit yang tumbang karena infeksi dari penyakit *ganoderma*, berdasarkan keterangan tersebut maka bulan Januari 2022 dijadikan sebagai acuan untuk melakukan pencarian nilai RGB dan VARI yang muncul beberapa bulan sebelum pohon tersebut tumbang pada citra UAV antara bulan Agustus sampai dengan Desember 2021.

Gambar 3.3 merupakan contoh citra UAV pada bulan Januari 2022 yang telah diberi penanda pada pohon kelapa sawit yang telah tumbang dan dinyatakan *ganoderma* menurut pihak PTPN, penandaan ini sendiri dilakukan pada aplikasi QGIS sebagai tools bantuan yang digunakan untuk melakukan penandaan lokasi pohon kelapa sawit yang setelah dilakukan seluruh penandaan akan disimpan

kedalam sebuah file dengan ekstensi TIF. File yang telah dibuat tersebut akan menjadi tahapan awal dataset yang akan dibuat pada proyek penelitian ini.



Gambar 3.3 Citra Pohon Kelapa Sawit Tumbang Yang Telah diberi Tanda

Untuk mendapatkan nilai RGB dan VARI pohon kelapa sawit pada kondisi 1 sampai dengan 4 bulan sebelum tumbang dilakukan penelusuran pada setiap citra UAV dengan menggunakan koordinat titik yang telah disimpan pada *shape file* yang telah dibuat dilakukan *over laying* untuk menentukan pohon mana yang akan ditandai, proses penandaan dan *over laying* dapat dilihat pada gambar 3.4.



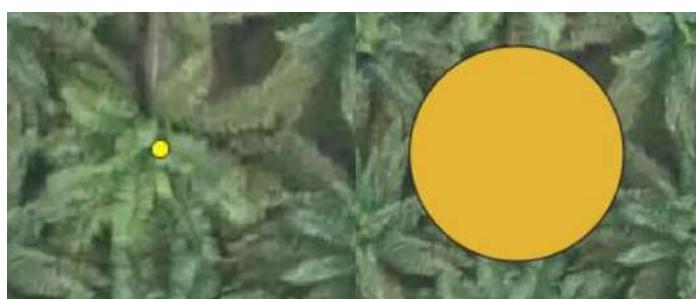
(A)

(B)

Gambar 3.4 Proses Penelusuran Pokok Pohon Kelapa Sawit Terinfeksi *ganoderma*, (A) Citra Titik Pohon Kelapa Sawit Tumbang Pada Bulan Januari 2022; (B) Citra Titik Pohon Kelapa Sawit Sebelum Tumbang Pada Bulan Desember 2021

Dapat diperhatikan bahwa ada perbedaan koordinat yang dihasilkan pada kedua titik yang ada pada kedua citra, hal ini mungkin terjadi ketika proses pengambilan citra dengan drone. Proses penandaan ini akan berlangsung berulang sampai dengan kondisi pohon kelapa sawit 4 bulan sebelum tumbang. Seluruh koordinat akan disimpan didalam *shape file* yang berbeda-beda sesuai dengan kondisi pohon kelapa sawit tersebut dan diberi label pembeda yakni pada nama filenya disetiap kondisinya. dataset yang dibuat ini terbatas pada kondisi pohon kelapa sawit 4 bulan sebelum tumbang, dikarenakan keterbatasan file yang tersedia pada saat proyek penelitian ini berlangsung.

Setelah didapat seluruh titik koordinat, maka dilakukan *buffering* pada setiap titik koordinat yang telah ditemukan. Fungsi dari proses melakukan *buffering* ini bertujuan untuk melakukan *limitasi* pada saat proses *ekstraksi* nilai RGB yang akan dilakukan setelah proses *buffering* ini. Agar nilai RGB yang ter-*ekstrak* hanya nilai RGB yang merepresentasikan pohon kelapa sawit itu sendiri. Ukuran *buffer* yang digunakan pada penelitian ini sebesar 2.5 m, ukuran tersebut merupakan hasil yang didapat dari melakukan *trial* dan *error* untuk mendapatkan ukuran yang tepat dengan kanopi pohon kelapa sawit yang ada pada citra UAV miliki PTPN ukuran tersebut juga didukung dari hasil penelitian yang menjelaskan bahwa lebar kanopi pohon kelapa sawit yang berumur dewasa memiliki lebar antara 1.8 m sampai dengan 2.2 m (Yudistina et al., 2017). Hasil buffering ini juga disimpan kedalam shape file dengan membedakan pada setiap kondisi pohon kelapa sawit. Perbandingan sebelum dan sesudah dilakukan proses *buffering* dapat dilihat pada gambar 3.5.



(A)

(B)

Gambar 3.5 Perbandingan Koordinat Pohon Kelapa Sawit, (A) Titik Sebelum *Buffering*; (B) Titik Sesudah *Buffering*

Setelah seluruh koordinat pohon kelapa sawit dengan bentuk *buffer* telah didapatkan, dilakukan *extract* nilai RGB yang ada pada citra pohon kelapa sawit dalam ruang lingkup pengambilan nilai RGB didalam *buffer* yang telah dibuat pada proses sebelumnya. Pengambilan nilai RGB ini dilakukan dengan bantuan *library* geowombat dengan fungsi *geowombat.extract* yang memiliki dua parameter antara lain shape file yang berbentuk polygon sebagai limitasi pengambilan nilai RGB yang akan dilakukan dan citra UAV dengan extensi TIF. Didalam fungsi *extract* yang disediakan pada *library* geowombat ini juga disediakan beberapa sensor yang dapat digunakan sesuai dengan jenis citra UAV yang dimiliki peneliti. Pada penelitian ini akan digunakan sensor ‘rgbn’ yang menurut peneliti paling sesuai dengan citra UAV miliki PTPN yang hanya memiliki tiga band yang ada pada citra sedangkan pada sensor ‘n’ nantinya akan dibuang karena tidak memberikan informasi apapun. Hasil yang diberikan pada *extract* nilai RGB yang dilakukan geowombat ini akan memberikan output berupa *data frame* yang berisi informasi mengenai id, kondisi pohon, koordinat, nilai band *red*, *green*, *blue*, dan nir.

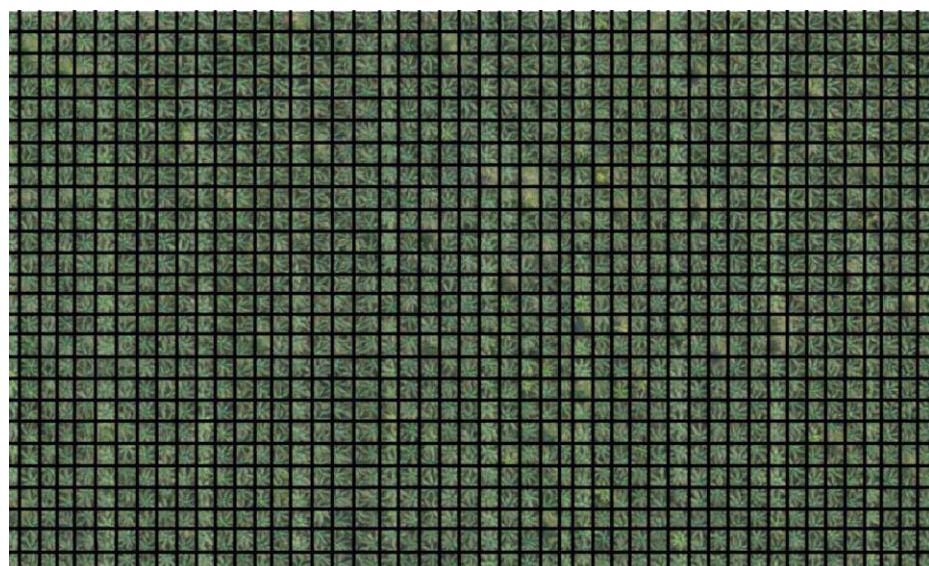
Proses ekstrak nilai ini akan dilakukan secara manual dalam maksud mengganti parameter yang diperlukan pada saat proses ekstraksi dan berulang sampai dengan mendapatkan seluruh nilai RGB yang ada pada setiap kondisi pohon kelapa sawit satu sampai dengan empat bulan sebelum tumbang beserta pohon kelapa sawit sehat. Hasil *data frame* setiap kondisi tersebut akan disimpan kedalam file csv yang kedepannya akan digunakan sebagai dasar untuk mendapatkan nilai VARI yang ada pada setiap pixel.

Langkah selanjutnya untuk mendapatkan dataset yang nantinya akan digunakan sebagai input pada model CNN LSTM pada penelitian ini yakni menghitung nilai indeks vegetasi VARI yang diperoleh dari nilai RGB yang telah diekstrak pada proses sebelumnya. Rumus perhitungan nilai VARI dapat dilihat pada bab 2.3. proses perhitungan akan dibantu dengan menggunakan *library* pandas sebagai dasar dilakukannya perhitungan nilai VARI, hasil perhitungan akan disimpan terlebih dahulu didalam sebuah array yang nantinya array tersebut akan digabungkan kedalam *data frame* yang sudah memiliki informasi nilai RGB pada proses sebelumnya. Untuk hasil perhitungan nilai VARI juga akan disimpan

kedalam file csv untuk nantinya dijadikan sebagai input pada model CNN LSTM prediksi *ganoderma* pada citra pohon kelapa sawit.

Setelah didapatkan nilai VARI pada setiap pixel yang ada di dalam *buffer* selanjutnya dilakukan *summary* untuk mendapatkan nilai *mean* VARI yang ada pada setiap pohon, yang nantinya menjadi nilai VARI yang merepresentasikan setiap pohon kelapa sawit yang ada pada citra. Setelah dilakukan *summary data* selanjutnya akan dilakukan pembersihan pada *dataset* dengan membuang beberapa kolom yang tidak digunakan, dan menyisakan kolom antar lain kondisi pohon kelapa sawit, koordinat, dan nilai *mean* VARI. Setelah dibersihkan akan kembali disimpan kedalam sebuah file CSV.

Pada proses selanjutnya akan memisahkan setiap pohon kelapa sawit yang ada pada citra menjadi *grid* yang pada konsepnya nanti *grid* tersebut akan merepresentasikan satu pohon kelapa sawit pada setiap *grid*. Pada tahap ini akan citra akan dibagi menjadi 64×64 *grid* dengan menghitung maksimal dan minimal dari koordinat yang ada pada *dateset* lalu jarak pada setiap *grid* akan dihitung dengan menggunakan koordinat maksimal x dikurang dengan koordinat minimal x sehingga didapat jarak *grid* x, pada jarak *grid* y juga dilakukan perhitungan yang sama dengan jarak *grid* y, sempurnanya setiap *grid* nantinya hanya akan memiliki sebuah pohon didalamnya . Contoh gambar dapat dilihat pada gambar 3.6.

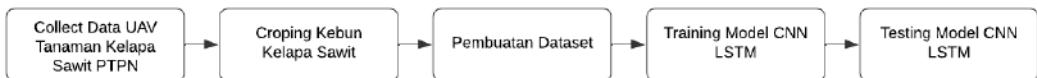


Gambar 3.6 Contoh Hasil *Grid* Pada Citra Perkebunan Kelapa Sawit

Setelah menentukan *grid* proses selanjutnya akan diubah menjadi *dataset time series* yang diperlukan untuk menyesuaikan dengan input yang diperlukan pada model arsitektur CNN LSTM yang digunakan pada penelitian ini dengan mengubah *dataset* menjadi *dataset time series* dengan menggunakan fungsi *time series generator* yang dimiliki *library* keras. Proses tersebut akan dilakukan pada data *train* dan *test*.

3.6 Desain Model Arsitektur CNN LSTM

Tahap ini merupakan paparan desain sistem yang nantinya digunakan pada penelitian ini. Gambar 3.7 merupakan desain sistem untuk metode CNN LSTM yang akan digunakan untuk medeteksi penyakit *ganoderma* pada citra pohon kelapa sawit.



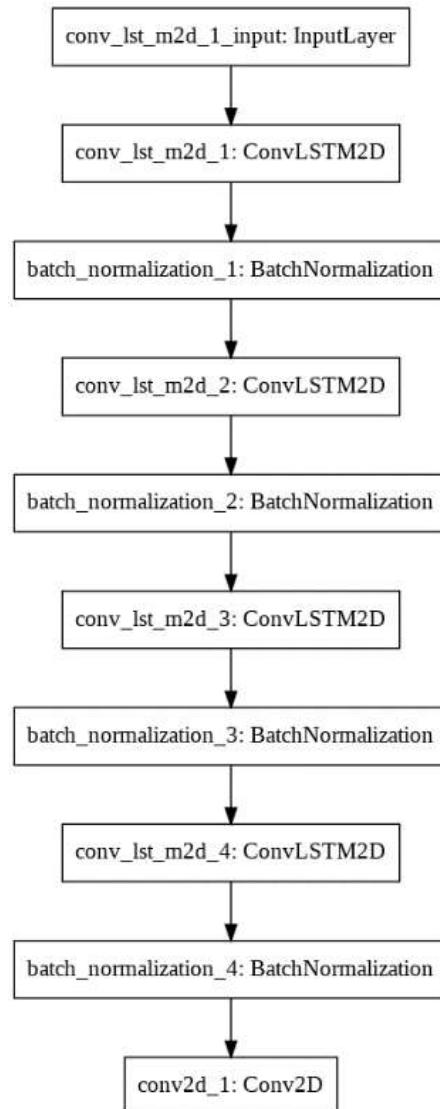
Gambar 3.7 Desain Sistem Model *Convolutional Neural Network Long Short Term Memory* Untuk Deteksi Penyakit *Ganoderma*

Model CNN LSTM yang digunakan pada proyek penelitian ini terdiri dari beberapa *layer* yang mana 1 *layer* merupakan *input layer*, 4 *layer* ConvLSTM2D, 4 *layer* Normalization dan 1 *layer* Conv2D. Plot model dapat dilihat pada gambar 3.8.

Input layer memiliki input size sebesar 64x64 dimana input ini merupakan hasil dari pembuatan *grid* yang telah dilakukan pada proses pembuatan dataset yang prosesnya dapat dilihat pada bab 3.5, ukuran kernel 5x5, menggunakan jenis padding *same*, dengan fungsi aktivasi *relu*, pada *layer* ConvLSTM2D menggunakan ukuran kernel 5x5, dengan jenis padding *same*, dan fungsi aktivasi *relu*, terakhir pada *layer* Conv2D kernel yang digunakan berukuran 5x5, menggunakan padding *same*, dan dengan fungsi aktivasi *sigmoid*.

Training model yang dilakukan pada proyek penelitian ini akan dilakukan 2 eksperimen dengan menggunakan 2 metode optimasi yang berbeda, eksperimen

pertama menggunakan metode optimasi Adam sedangkan pada eksperimen kedua menggunakan metode optimasi SGD, kedua eksperimen tersebut akan menggunakan 3 parameter *learning rate* (Lr) yang memiliki ukuran yang berbeda yakni 0.001, 000.1, dan 0.00001. Eksperimen ini menggunakan 6 variasi epoch yang digunakan mulai dari 500, 1000, 1500, 2000, 2500, dan 3000 epoch. Seluruh eksperimen ini menggunakan fungsi *loss mean squared error*, dan dengan *metrics mean squared error*. Output yang dihasilkan oleh model ini berupa heatmap berukuran 64x64 grid yang pada setiap gridnya merupakan representasi setiap pohon kelapa sawit pada citra UAV, pada setiap grid tersebut akan memiliki nilai mean VARI hasil dari prediksi yang telah dilakukan model CNN LSTM yang digunakan pada penelitian ini.



Gambar 3.8 Plot Model CNN LSTM

Fungsi *layer* ConvLSTM2D yang pertama merupakan *layer* yang bertugas sebagai penerima *input* dengan ukuran 64x64. *Layer* ini memiliki 64 filter dan menggunakan kernel yang berukuran 5x5 untuk melakukan konvolusi spasial pada *input*, *layer* ini akan menghasilkan urutan *output*, yang nantinya akan digunakan pada *layer* selanjutnya. Seluruh *layer* BatchNormalization digunakan untuk mempercepat pelatihan model dan membantu dalam normalisasi aktivasi *output* dari *layer* sebelumnya. *Layer* ini membantu menjaga stabilitas dan mempercepat konvergensi dalam proses pelatihan. Fungsi *layer* ConvLSTM2D pada *layer* ke tiga, lima dan tujuh memiliki fungsi yang sama dengan *layer* ConvLSTM2D yang pertama dengan perbedaan bahwa tidak ada *input shape* yang ditentukan secara eksplisit karena itu *input layer* ini akan mengikuti *output* yang yang dihasilkan *layer* sebelumnya. Terakhir *layer* Conv2D merupakan *layer* konvolusi 2D biasa yang menghasilkan *output* dalam bentuk citra biner. *Output* dari *layer* ini diaktivasi dengan menggunakan sigmoid yang bertujuan untuk menghasilkan probabilitas yang berkisar antara 0 dan 1 untuk setiap *grid*.

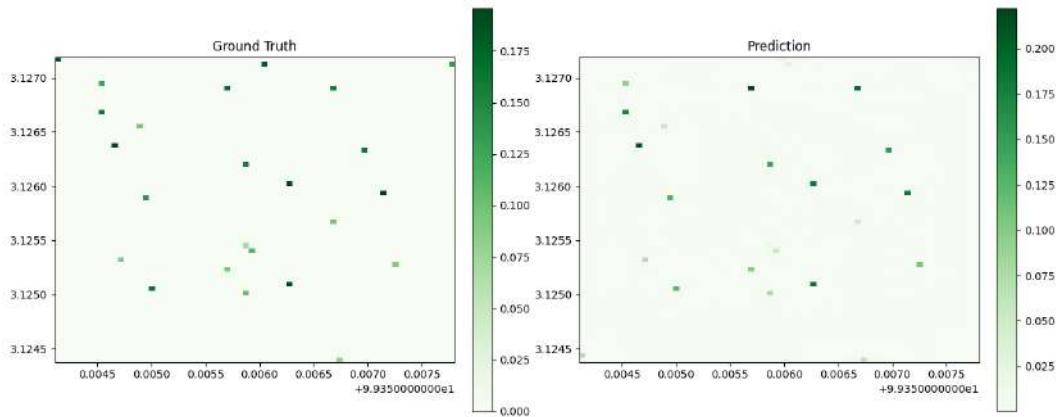
3.7 Pengujian Model Arsitektur CNN LSTM

Pengujian model arsitektur CNN LSTM yang dilakukan pada proyek penelitian ini akan menggunakan perbandingan nilai *mean VARI* yang ada pada *grid* dari *ground truth* dengan nilai *mean VARI* yang telah diprediksi dari model CNN LSTM yang digunakan pada proyek penelitian ini. *Ground truth* yang dimaksud tersebut merupakan keadaan aktual yang terjadi pada citra pohon kelapa sawit, apakah pohon tersebut termasuk pohon dengan kondisi sehat atau pohon dengan kondisi terserang *ganoderma* yang dijadikan sebagai acuan perbandingan hasil prediksi yang telah dibuat berdasarkan model yang digunakan pada proyek penelitian tugas akhir ini.

Pengujian kinerja model CNN LSTM yang dibuat pada proyek penelitian ini akan menggunakan perhitungan MSE (Mean Squared Error) untuk mendapatkan akurasi nilai VARI dan nilai *loss* yang dihasilkan oleh model yang telah dibuat, rumus perhitungan nilai MSE dapat dilihat pada sub bab 2.16. Sehingga nantinya dapat disimpulkan bahwa model tersebut dapat bekerja dengan baik atau kurang dapat bekerja dengan baik dengan input dan batasan yang ada

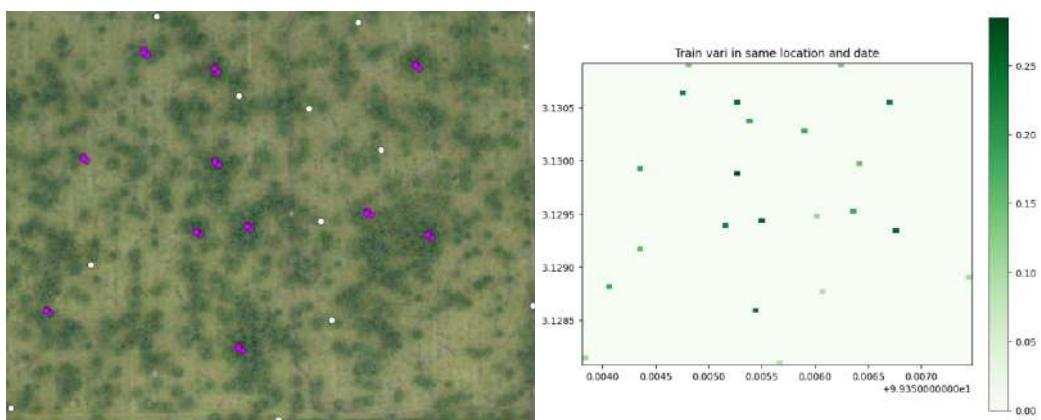
pada proyek penelitian tugas akhir ini dengan melihat nilai MSE yang dihasilkan model.

MSE akan dibandingkan pada setiap model yang dibuat dengan parameter yang berbeda-beda dan akan dievaluasi model dengan metode optimasi, parameter *learning rate* dan jumlah epoch mana yang memiliki nilai MSE terbaik sehingga dapat ditentukan model terbaik dari beberapa model yang digunakan pada penelitian ini.



Gambar 3.9 Hasil Output Nilai VARI Model

Gambar 3.9 merupakan hasil output dari model yang telah digunakan, yang nantinya dari hasil tersebut akan dihitung nilai MSE yang dihasilkan dari prediksi dan dibandingkan dengan *ground truth*. Titik-titik yang ada pada setiap *grid* didalam *heatmap ground truth* merupakan representasi pohon kelapa sawit yang ada pada kebun kelapa sawit. Perbandingan keadaan titik pohon kelapa sawit yang direpresentasikan pada *heatmap* dan pada citra perkebunan kelapa sawit dapat dilihat pada gambar 3.10.



Gambar 3.10 Perbandingan Kondisi Titik Pohon Kelapa Sawit Sebenarnya Pada Citra UAV dan Hasil Representasi Pada *Heatmap*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Profil Dataset

Pada penelitian yang dilakukan pada tugas akhir ini digunakan 2 dataset yang dibagi menjadi *dataset training* dan *dataset testing*, kedua *dataset* ini diambil pada blok kelapa sawit yang berbeda namun keduanya memiliki informasi nilai *mean VARI* yang ada pada setiap titik pohon kelapa sawit yang digunakan pada penelitian tugas akhir ini. Dataset ini disimpan kedalam sebuah *file* dengan format csv, sehingga memudahkan peneliti untuk melakukan eksperimen yang akan dilakukan selanjutnya. Rincian jumlah titik pohon kelapa sawit yang digunakan dan ukuran dataset dapat dilihat pada tabel 4.1.

Tabel 4.1 Rincian *Dataset Train* dan *Test*

Dataset	Titik Pohon Sakit	Titik Pohon Sehat	Ukuran File
Train	11	11	6KB
Test	12	12	7KB

Setiap titik pohon kelapa sawit yang ada pada dataset akan diberi label yang menandakan bahwa pohon tersebut terinfeksi ganoderma atau pohon tersebut merupakan pohon dengan kondisi sehat. Label yang diberikan pada dataset ini antara lain 0 untuk melabeli pohon kelapa sawit dengan kondisi sehat, sedangkan 1 untuk melabeli pohon kelapa sawit yang terinfeksi ganoderma.

Contoh *dataframe* yang dihasilkan dari *dataset train* dan *test* dapat dilihat pada tabel 4.2 dan tabel 4.3.

Tabel 4.2 *Dataframe* Pada *Dataset Train*

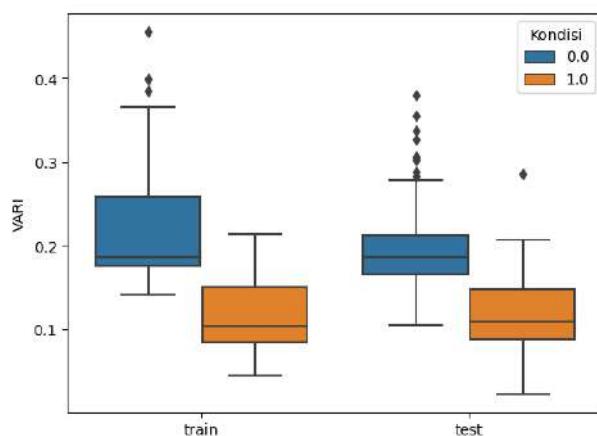
No	Kondisi	x	y	VARI
0	0	99.3555	3.12942	0.26165
1	0	99.35527	3.12988	0.28468
2	0	99.35634	3.12952	0.18044
3	0	99.35434	3.1299	0.18583
4	0	99.35544	3.12857	0.24438
...
85	1	99.3557	3.12808	0.05559
86	1	99.35381	3.12816	0.07204
87	1	99.35748	3.12889	0.05305

Tabel 4.3 Dataframe Pada Dataset Test

No	Kondisi	x	y	VARI
0	0	99.35571	3.12692	0.212
1	0	99.35628	3.12602	0.19801
2	0	99.35587	3.12618	0.15063
3	0	99.35468	3.12637	0.23707
4	0	99.35606	3.12713	0.19385
...
91	1	99.35595	3.12539	0.11624
92	1	99.35567	3.12523	0.09454
93	1	99.35588	3.12502	0.10286
94	1	99.3567	3.1257	0.09554
95	1	99.35727	3.12529	0.09049

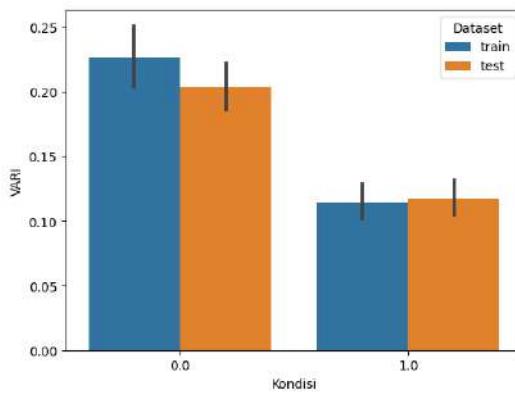
Kolom kondisi merupakan label yang berguna pembeda kondisi dari pohon kelapa sawit yang ada pada dataset, kolom x dan y merupakan koordinat titik pohon kelapa sawit tersebut pada citra UAV, dimana nilai x dan y merupakan koordinat pohon kelapa sawit, kolom VARI merupakan nilai mean VARI dari pohon kelapa sawit tersebut, dan kolom datetime merupakan kolom untuk memberi informasi pada bulan berapa nilai VARI tersebut diambil. Pada penelitian ini nilai mean VARI diambil pada rentang waktu bulan Agustus, September, Oktober, dan November yang nantinya akan diubah menjadi data *time series* dan digunakan pada model CNN LSTM yang dibuat pada penelitian ini.

Perbandingan nilai mean VARI dari pohon kelapa sawit sehat dan pohon kelapa sawit yang terinfeksi ganoderma dapat dilihat pada gambar 4.1.



Gambar 4.1 Boxplot Nilai VARI Pada Setiap Kondisi di Kedua Dataset

Pada gambar 4.1 merupakan boxplot nilai VARI yang ada pada kedua dataset, dapat dilihat bahwa median nilai VARI dari kedua dataset memiliki nilai median yang mirip. Pada data pohon kelapa sawit sehat nilai median kurang lebih 0.19 sedangkan pada pohon kelapa sawit sakit nilai median berada diantara 0.11. selain dilihat dari median yang dihasilkan dari boxplot, dapat juga dilihat pada grafik batang yang ada pada gambar 4.2 yang memberikan informasi mean VARI pada setiap kondisinya di kedua dataset.



Gambar 4.2 *Barchart Mean VARI Pada Setiap Kondisi Dikedua Dataset*

Dilihat dari hasil *barchart* maupun *boxplot* yang telah dihasilkan didapat kesimpulan bahwa nilai VARI yang dimiliki pohon kelapa sawit sehat dan sakit (terinfeksi ganoderma) memiliki perbedaan nilai rata-rata VARI dimana nilai VARI pada pohon kelapa sawit sehat memiliki nilai lebih tinggi jika dibandingkan dengan nilai VARI pada pohon kelapa sawit sakit. Rincian nilai *mean* VARI pada setiap kondisi dapat dilihat pada tabel 4.4.

Tabel 4.4 Perbandingan Nilai *Mean*, Maksimal dan Minimal VARI

Nilai	VARI	
	Train	Test
Mean Sehat	0.226289	0.203218
Mean Sakit	0.114751	0.117159
Maks Sehat	0.456065	0.379466
Maks Sakit	0.213194	0.285542
Min Sehat	0.140664	0.104575
Min Sakit	0.043435	0.022047

4.2 Preprocessing Data

Preprocessing dilakukan untuk menyesuaikan dataset yang telah dibuat agar sesuai dengan input yang diminta pada model CNN LSTM yang digunakan pada penelitian ini.

Tahap awal dilakukan pengurutan data antara data kelapa sawit sakit dan kelapa sawit sehat, urutan data yang digunakan yakni sesuai urutan waktu kondisi kelapa sawit mulai dari bulan Agustus sampai dengan bulan November, pengurutan ini dilakukan pada kedua dataset *train* maupun *test*.

Selanjutnya pada tahap kedua dilakukan pemberian label terhadap setiap titik pohon kelapa sawit yang ada pada dataframe, pada penelitian ini digunakan *date time* sebagai label pembeda yang menentukan nilai VARI pada titik pohon kelapa sawit tersebut diambil berapa bulan sebelum pohon kelapa sawit tersebut tumbang dan dinyatakan ganoderma. Contoh *dataframe* yang dihasilkan dapat dilihat pada tabel 4.5.

Tabel 4.5 Dataframe Setelah Diberi Label *Date Time*

No	Kondisi	x	y	VARI	datetime
0	0	99.3555	3.12942	0.26165	8/1/2021
1	0	99.35527	3.12988	0.28468	8/1/2021
2	0	99.35634	3.12952	0.18044	8/1/2021
3	0	99.35434	3.1299	0.18583	8/1/2021
4	0	99.35544	3.12857	0.24438	8/1/2021
...
83	1	99.35436	3.12917	0.10799	11/1/2021
84	1	99.35607	3.12878	0.04976	11/1/2021
85	1	99.3557	3.12808	0.05559	11/1/2021
86	1	99.35381	3.12816	0.07204	11/1/2021
87	1	99.35748	3.12889	0.05305	11/1/2021

Pemberian label ini dilakukan pada kedua dataset yakni dataset *train* dan *test*, yang menjadikan dataset ini sebagai *clean dataset* yang kedepannya akan digunakan dan diolah pada penelitian ini.

Setelah didapat *clean dataset* akan dihitung limitasi dari data koordinat x dan y untuk digunakan sebagai syarat melakukan pembuatan *histogram 2D* yang akan dibuat pada tahapan selanjutnya, pada penlitian ini akan dihitung x minimal,

x maksimal, x *range*, y minimal, y maksimal, dan y *range*. Perhitungan limitasi nilai tersebut akan dilakukan pada kedua dataset, dengan rincian dapat dilihat pada tabel 4.6.

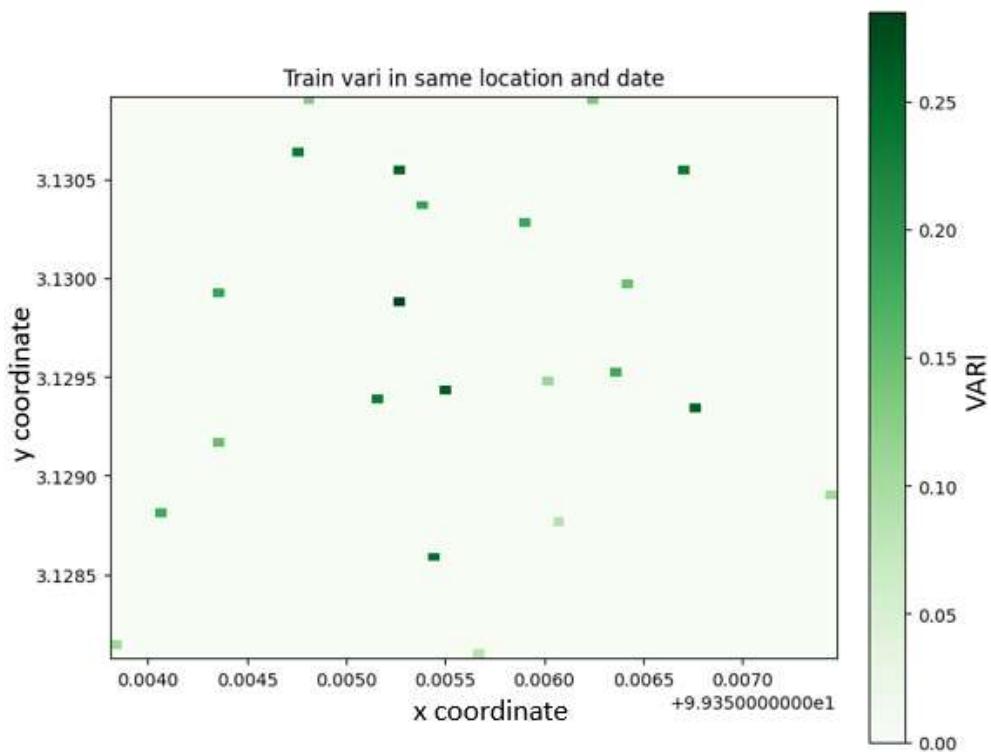
Tabel 4.6 Rincian Nilai Limitasi Koordinat Dataset

dataset	x min	x max	x range	y min	y max	y range
train	99.3538	99.3575	$5,64 \cdot 10^{-5}$	3.12808	3.13092	$4,37 \cdot 10^{-5}$
test	99.3541	99.3578	$5,68 \cdot 10^{-5}$	3.12438	3.12719	$4,33 \cdot 10^{-5}$

Fungsi nilai x minimal, x maksimal, y minimal, dan y maksimal sebagai batasan dalam membuat histogram 2D yang akan digunakan pada penelitian ini, nilai minimal dan maximal tersebut didapat dari menentukan nilai terendah dan tertinggi yang ada pada column x dan y yang merupakan nilai koordinat titik pohon kelapa sawit. Sedangkan fungsi nilai x *range* dan y *range* digunakan sebagai rentang nilai pada setiap grid yang akan dibuat pada histogram 2D, nilai range tersebut didapat dari menghitung nilai maximal dikurangi dengan nilai minimal dan dibagi 65 bagian sehingga akan didapat rentang nilai x dan y untuk seluruh grid yang ada pada histogram 2D.

Setelah diperoleh *clean dataset* dan nilai limitasi yang membatasi x dan y yang diperlukan pada tahapan pembuatan histogram 2D selanjutnya data yang telah didapat ini akan diolah dan diubah menjadi format numpy array histogram 2D yang memiliki informasi nilai VARI dititik-titik yang telah ditentukan pada setiap bulannya, hasil dari langkah ini akan menghasilkan 4 buah array sesuai dengan bulan diambilnya nilai VARI pada setiap titik pohon kelapa sawit dan disimpan kedalam sebuah array yang memiliki keempat informasi nilai VARI pada setiap grid yang telah dibuat pada proses pembuatan histogram 2D dengan format numpy, setelah array tersebut dihasilkan maka diubah menjadi array dalam format numpy sehingga bentuk dari array tersebut akan menjadi array 3 dimensi yang bejumlah 4, 64, 64.

Hasil dari pembuatan histogram 2D dapat dilihat pada gambar 4.3 yang merupakan contoh histogram 2D yang ada pada urutan array pertama atau bisa disebut sebagai histogram 2D yang merepresentasikan kondisi pada bulan Agustus.

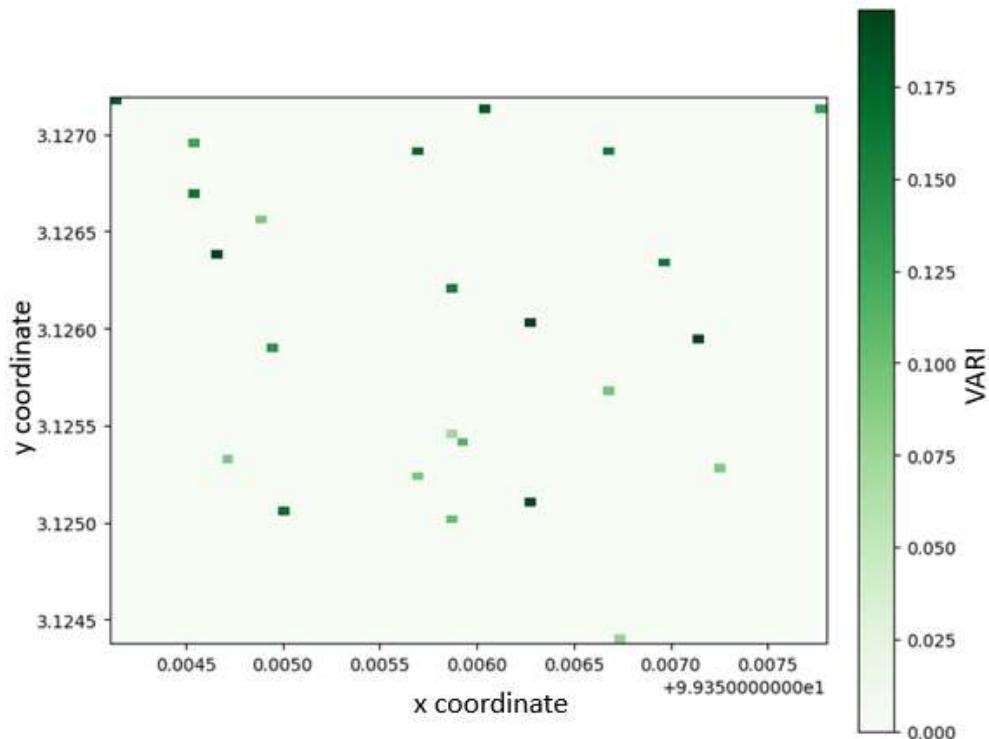


Gambar 4.3 *Heatmap* Dari Array Pertama Yang Merupakan Representasi Kondisi Pada Bulan Agustus

Warna yang dihasilkan pada *heatmap* tersebut merupakan representasi nilai VARI yang ada pada setiap *grid*, dimana semakin pekat warna berarti nilai VARI yang ada pada *grid* tersebut semakin tinggi, dalam kasus ini semakin tinggi nilai VARI menandakan pohon tersebut semakin sehat dan sebaliknya semakin pudar warna yang ada pada *grid* semakin rendah nilai VARI yang ada pada *grid* tersebut yang menandakan pohon tersebut semakin sakit.

Proses pembuatan array ini dilakukan terhadap kedua dataset *train* dan *test*, array ini yang menjadi dasar untuk dijadikan *input* pada model yang digunakan pada penelitian ini. Setelah dihasilkan array 3 dimensi yang memiliki informasi nilai VARI pada setiap gridnya, langkah selanjutnya array 3 dimensi tersebut diubah menjadi array 4 dimensi yang berjumlah 4, 64, 64, 1 dengan menggunakan fungsi numpy *reshape*, perubahan bentuk ini diperlukan mengingat layer ConvLSTM2D yang digunakan pada penelitian membutuhkan input array 4 dimensi. Pengubahan bentuk array ini dilakukan terhadap kedua dataset *train* dan *test*. Setelah didapat array *train* dan *test* yang memiliki bentuk 4 dimensi maka bentuk array tersebut akan diubah menjadi bentuk *time series* dengan

menggunakan fungsi dari *library* Keras yakni TimeSeriesGenerator, parameter yang digunakan pada proses ini akan menggunakan *length* = 3, dengan besar *batch size* sebesar 1. Fungsi dari *length* = 3 merupakan jumlah waktu yang akan digunakan untuk memprediksi langkah selanjutnya dalam deret waktu, sedangkan *batch size* = 1 merupakan jumlah sampel yang dihasilkan pada setiap iterasi dimana model akan memperbarui bobot setiap model memasukan 1 sampel data. Hasil dari *data series train* dan *test* akan dipisah menjadi X dan y, dimana X merupakan urutan waktu yang digunakan sebagai fitur dalam pelatihan model sedangkan y merupakan target atau nilai yang ingin diprediksi. Contoh hasil yang diberikan setelah dilakukan pemisahan antara array yang digunakan sebagai data *train* dan *test* akan menghasilkan grid yang sama seperti sebelum dilakukan perubahan *dataset* menjadi *data series* hasil dari histogram 2D yang digunakan berdasar *data series* dapat dilihat pada gambar 4.4 yang merupakan contoh histogram 2D dari array y *test* yang merupakan data target pada *data series test*.



Gambar 4.4 *Heatmap* Array y Pada Data Series *Test*

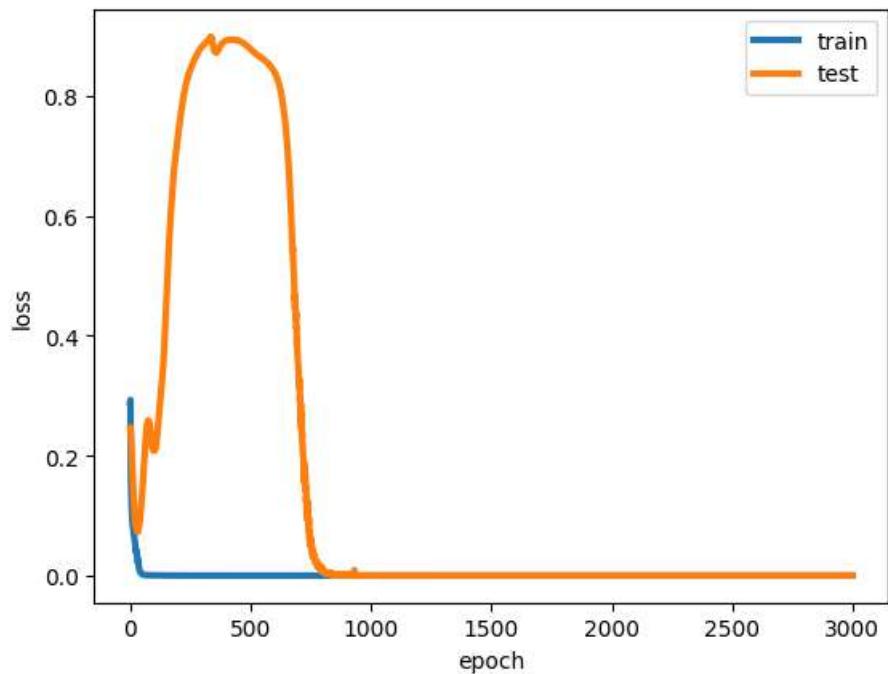
Hasil dari array X dan y pada data *train* dan *test* yang telah dibuat ini yang kedepannya akan digunakan sebagai data yang diperuntukan sebagai data latih maupun data *test* pada eksperimen yang dilakukan pada eksperimen yang ada pada penelitian tugas akhir ini.

4.3 Exsperimen Model CNN LSTM

Pada penelitian ini akan digunakan 1 model dengan 6 *tuning* yang berbeda-beda dengan menggunakan metode optimasi adam dan SGD, dengan parameter *learning rate* 0.001, 0.0001, dan 0.00001 pada setiap optimasi. Eksperimen dengan seluruh model yang digunakan pada penelitian ini akan dilakukan sebanyak 3000 epoch dengan membaginya pada setiap kelipatan 500 epoch untuk melihat bagaimana kondisi perbandingan hasil prediksi dan *ground truth* beserta nilai MSE yang dihasilkan pada setiap 500 epoch akan dipaparkan pada setiap model. Warna yang dihasilkan pada setiap *grid* akan merepresentasikan nilai VARI yang ada pada *grid* tersebut, semakin hijau warna pada *grid* maka nilai VARI yang dimiliki *grid* tersebut semakin tinggi (sehat) dan sebaliknya. Eksperimen model akan dilakukan pada *dataset test* yang memiliki *ground truth* titik pohon sebanyak 24 titik. Hasil dari eksperimen keenam model ini akan dijelaskan lebih rinci pada setiap sub bab 4.3.

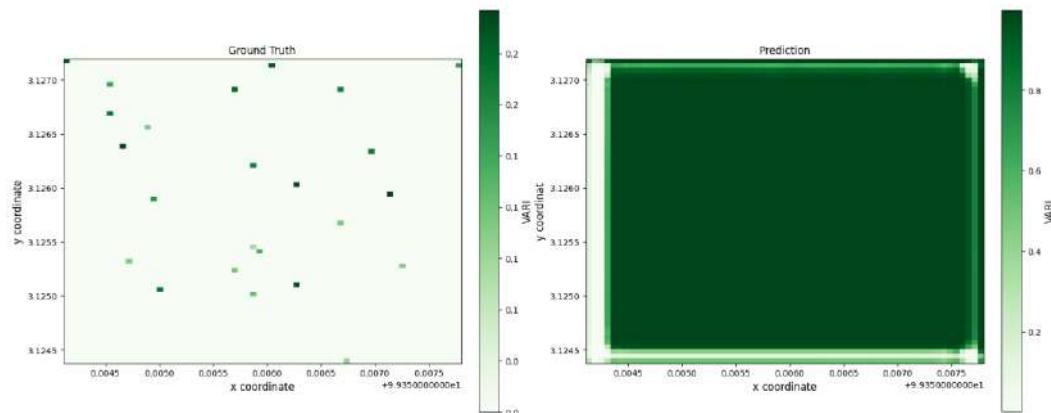
4.3.1 Model CNN LSTM Optimizer Adam Dengan Learning Rate 0.001

Pada model CNN LSTM ini digunakan metode optimasi Adam dengan parameter *learning rate* sebesar 0.001. Gambar 4.5 merupakan grafik *loss* yang dihasilkan model sampai dengan epoch ke-3000, dapat dilihat bahwa dari grafik *loss* yang dihasilkan model tidak terindikasi adanya *overfit*, grafik *loss* yang dihasilkan model ini akan dilihat kondisinya lebih rinci pada setiap 500 epoch yang akan dibahas pada sub bab ini.



Gambar 4.5 Grafik *Loss* Model Adam Lr 0.001

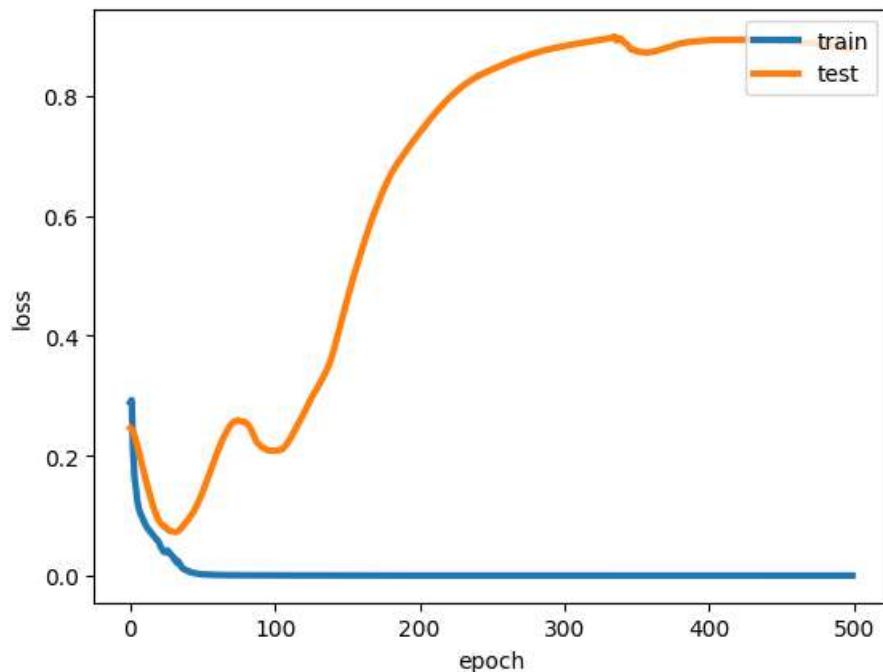
Gambar 4.6 merupakan perbandingan antara hasil prediksi dan *ground truth* pada epoch 500 dengan menggunakan *dataset test*.



Gambar 4.6 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.001 Pada Epoch 500

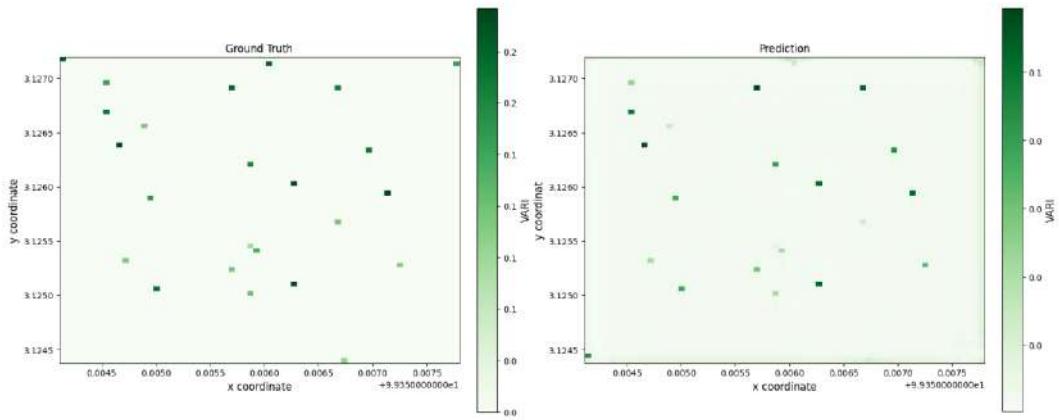
Dapat dilihat bahwa hasil prediksi yang dihasilkan pada model CNN LSTM dengan menggunakan metode optimasi Adam pada parameter *learning rate* 0.001 di epoch 500 memberikan hasil yang tidak merepresentasikan keadaan sebenarnya dimana sebagian besar *grid* diprediksi memiliki warna hijau pekat

yang berarti memiliki nilai VARI tinggi sedangkan pada *ground truth* sebagian besar *grid* memiliki warna putih atau tidak memiliki nilai VARI, sehingga perlu untuk menganalisis bagaimana keadaan loss yang diberikan pada model tersebut, gambar 4.7 merupakan grafik loss dari model ini pada epoch 500.

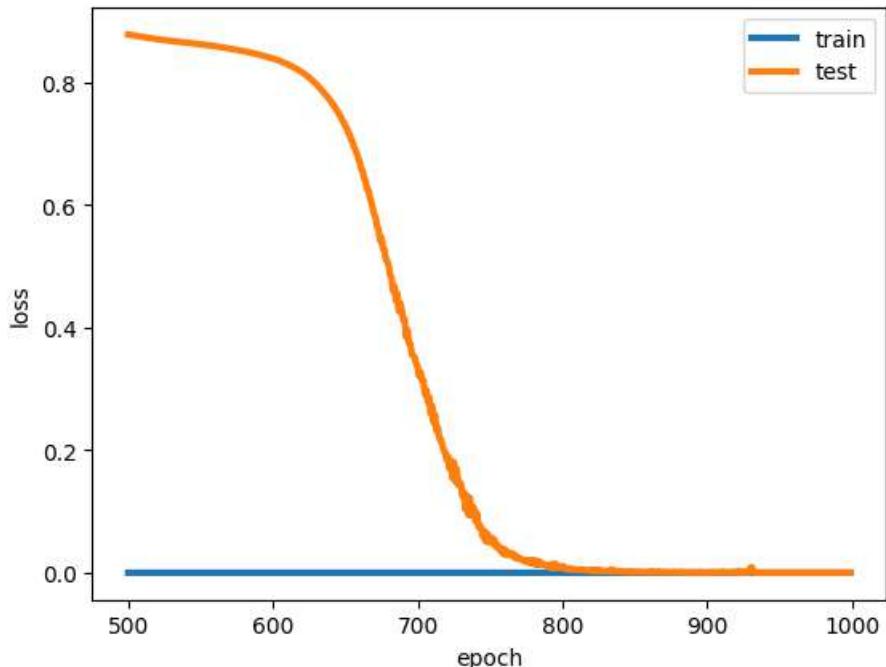


Gambar 4.7 Grafik *Loss* Model Adam Lr 0.001 Pada Epoch 500

Dapat dilihat bahwa grafik loss pada data test seiring dengan berjalannya jumlah epoch memiliki pola yang terus meningkat hingga mendekati 1, dimana jika loss mendekati 1 maka model tersebut belum bekerja dengan baik, sedangkan loss pada data train memiliki pola menurun dengan besaran nilai MSE yang dihasilkan model dengan menggunakan optimasi Adam dan parameter *learning rate* sebesar 0.001 ini pada epoch 500 sebesar 0.879. Gambar 4.8 dan 4.9 akan menyajikan perbandingan *ground truth* dengan hasil prediksi model pada epoch 1000 beserta grafik *loss*-nya.



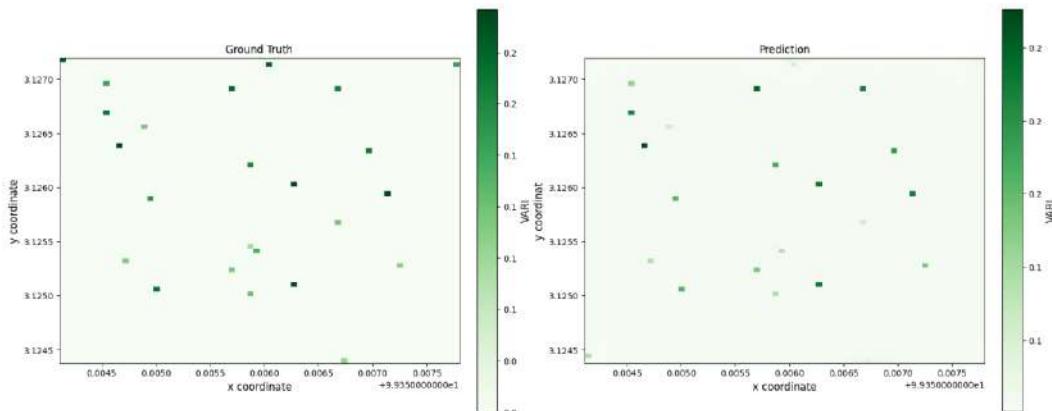
Gambar 4.8 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.001 Pada Epoch 1000



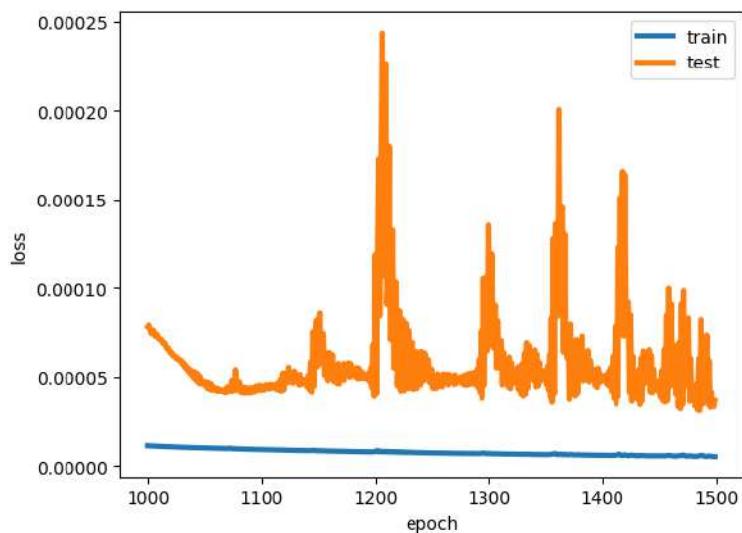
Gambar 4.9 Grafik *Loss* Model Adam Lr 0.001 Pada Epoch 500-1000

Dapat dilihat bahwa hasil yang ditunjukkan bahwa hasil prediksi pada gambar 4.8 sudah memiliki pola yang mirip jika dibandingkan dengan *ground truth*, dibandingkan juga dengan pergerakan nilai *loss* yang dihasilkan memiliki kecenderungan nilai yang menurun, merupakan pertanda baik bahwa model tersebut dapat dilanjutkan pembelajarannya, namun dilihat dari rentang nilai yang diperlihatkan pada hasil prediksi ini memiliki nilai tertinggi sebesar 0.059. Nilai MSE yang dihasilkan pada model ini di epoch 1000 yakni sebesar $7,84 \cdot 10^{-5}$.

Dilanjutkan pada gambar 4.10 dan 4.11 merupakan hasil dari model pada epoch sebesar 1500.



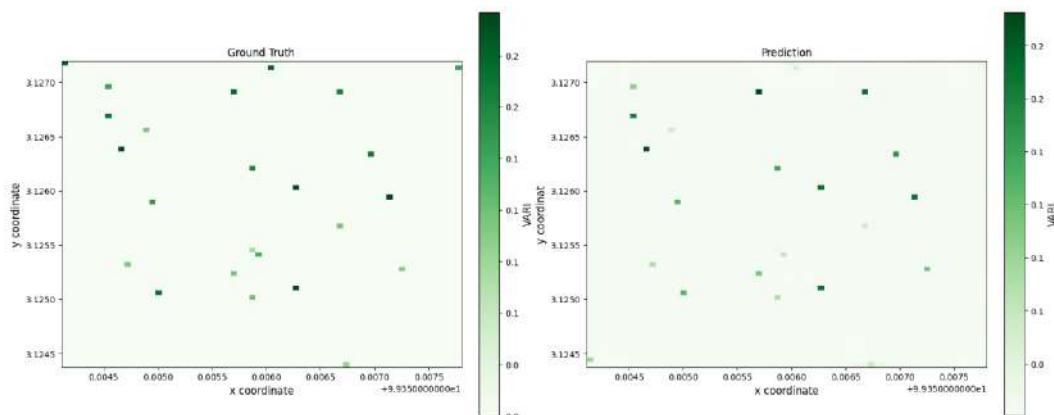
Gambar 4.10 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.001 Pada Epoch 1500



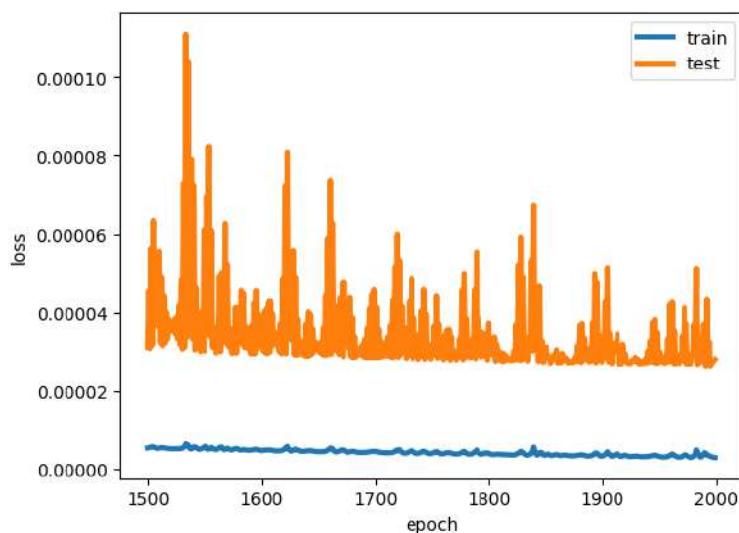
Gambar 4.11 Grafik *Loss* Model Adam Lr 0.001 Pada Epoch 1000-1500

Hasil yang diberikan model ini dalam melakukan prediksi pada epoch sebesar 1500 terbilang cukup baik dapat dilihat pada gambar 4.10 pola yang dimiliki hasil prediksi dan *ground truth* memiliki pola yang sama, yang berarti model sudah dapat belajar dengan baik, namun jika dilihat pada grafik *loss* yang dimiliki model ini pada rentang epoch 1000 sampai dengan 1500, terdapat tanda-tanda terjadinya *overfit* yakni terjadinya fluktuasi nilai *loss* yang dihasilkan model ini. Namun nilai MSE yang dihasilkan dari model ini pada epoch 1500 mengalami

penurunan nilai dibandingkan pada epoch 1000 yakni sebesar $3,72 \cdot 10^{-5}$. Gambar 4.12 dan 4.13 merupakan hasil yang diberikan model pada epoch 2000.



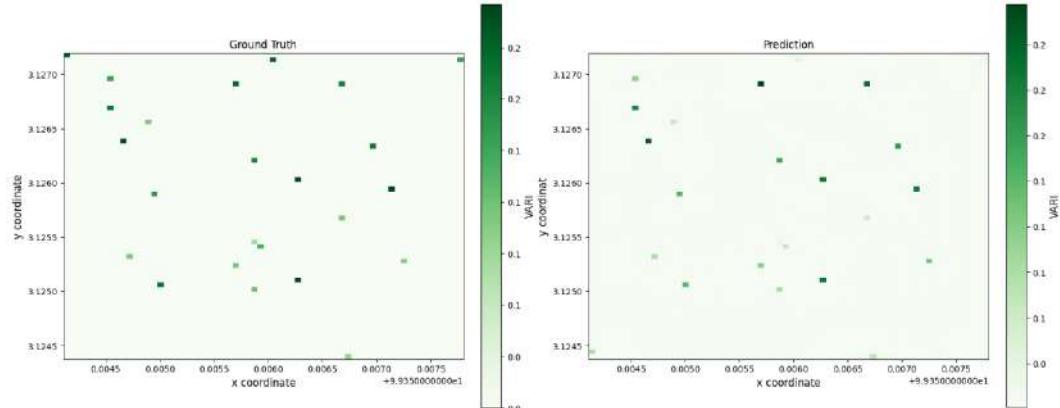
Gambar 4.12 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.001 Pada Epoch 2000



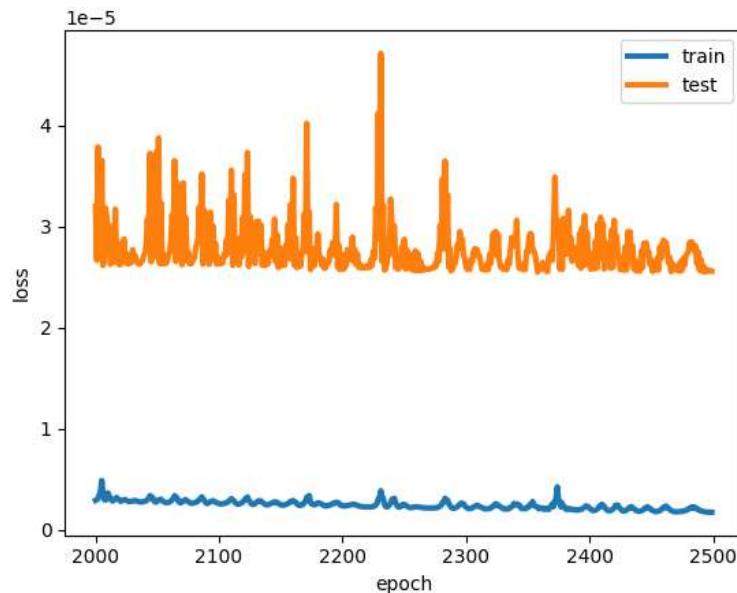
Gambar 4.13 Grafik *Loss* Model Adam Lr 0.001 Pada Epoch 1500-2000

Dilihat dari rentang nilai yang dihasilkan antara hasil prediksi dan *ground truth* model ini memiliki prediksi yang cukup baik, dengan nilai maksimal yang diberikan pada hasil prediksi yakni 0.190605 sedangkan nilai maksimal pada *ground truth* sebesar 0.195932 model pada epoch ini cukup mendekati nilai asli tertinggi dengan selisih 0.005, nilai MSE yang dihasilkan pada model diepoch ini sebesar $2,78 \cdot 10^{-5}$ memiliki penurunan nilai dibandingkan nilai MSE pada epoch 1500. Namun hasil yang diberikan oleh grafik *loss* pada model dengan epoch 2000

ini memiliki tanda-tanda *overfit* yang lebih jelas dengan fluktuasi nilai loss pada setiap iterasinya. Gambar 4.14 dan 4.15 akan menyajikan hasil dari model ini pada jumlah epoch 2500.



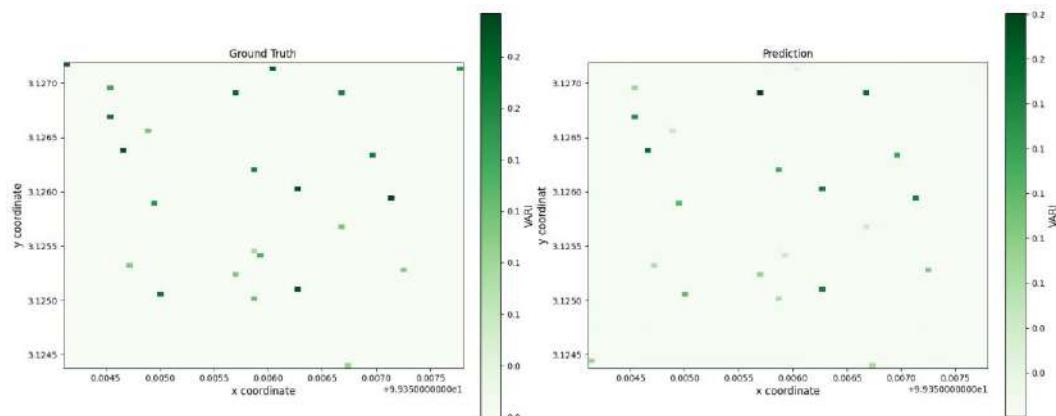
Gambar 4.14 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.001 Pada Epoch 2500



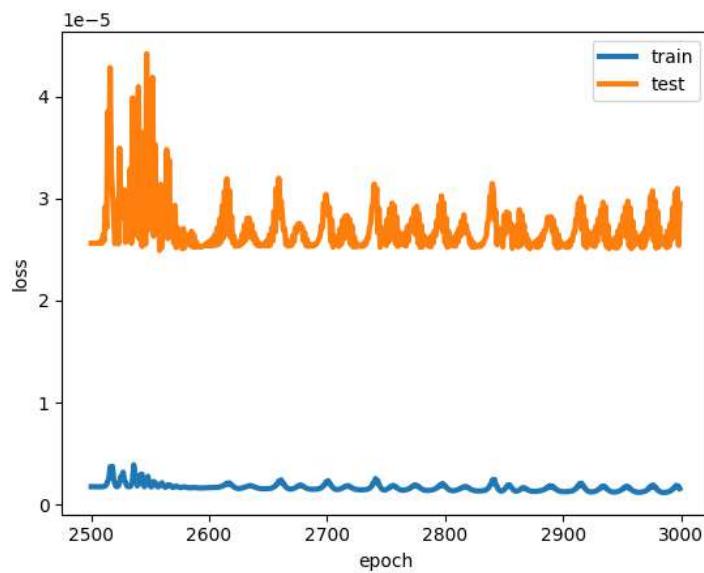
Gambar 4.15 Grafik *Loss* Model Adam Lr 0.001 Pada Epoch 2000-2500

Hasil yang diberikan model dengan epoch 2500 ini memiliki nilai MSE sebesar $2,56 \cdot 10^{-5}$, jika dibandingkan dengan epoch sebelumnya nilai MSE yang dihasilkan pada model ini memiliki penurunan nilai sebesar $0,22 \cdot 10^{-5}$, yang berarti model dengan epoch 2500 ini dapat dikategorikan lebih baik dibanding dengan

model dengan epoch yang lebih kecil, namun jika dilihat pada grafik *loss* yang dihasilkan model ini, terdapat fluktuasi nilai pada setiap iterasinya yang memiliki tanda-tanda bahwa pada model ini terjadi *overfit*. Gambar 4.16 dan 4.17 merupakan hasil prediksi model dengan epoch sebesar 3000, ini merupakan epoch terakhir yang batasi pada penelitian ini untuk dilihat bagaimana kondisi nilai MSE dan hasil prediksi yang diberikan model tersebut pada perbandingan epoch yang digunakan.



Gambar 4.16 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.001 Pada Epoch 3000



Gambar 4.17 Grafik *Loss* Model Adam Lr 0.001 Pada Epoch 2500-3000

Hasil prediksi model dengan optimasi Adam dan *learning rate* 0.001 yang dijalankan pada epoch 3000 memiliki nilai MSE sebesar $2,95 \cdot 10^{-5}$, terdapat

peningkatan nilai MSE yang dihasilkan pada epoch 3000 ini, hal ini mungkin terjadi karena fluktuasi nilai *loss* yang dapat dilihat pada gambar 4.17 yang merupakan grafik loss dari model dengan epoch 3000 ini.

Rincian perbandingan nilai MSE pada setiap epoch yang digunakan dan yang diamati dapat dilihat pada tabel 4.7.

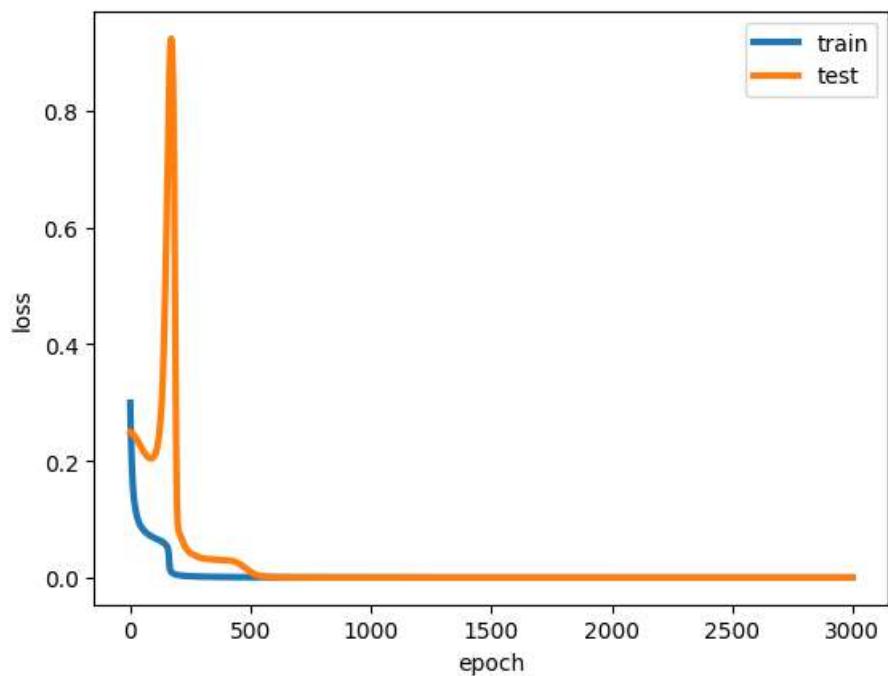
Tabel 4.7 Rincian Nilai MSE Pada Data *Test* Dimodel Adam Lr 0.001 Di setiap 500 Epoch

MSE	Epoch					
	500	1000	1500	2000	2500	3000
	0.879	$7,84 \cdot 10^{-5}$	$3,72 \cdot 10^{-5}$	$2,78 \cdot 10^{-5}$	$2,56 \cdot 10^{-5}$	$2,95 \cdot 10^{-5}$

Jika dibandingkan nilai MSE yang diamati pada setiap keliapan 500 epoch, nilai MSE yang dihasilkan memiliki pola menurun sampai pada epoch 2500 sedangkan pada epoch 3000 terdapat peningkatan nilai MSE dibandingkan epoch sebelumnya, hal ini mungkin terjadi karena fluktuasi nilai *loss* yang dapat diperhatikan pada grafik *loss* yang ada pada gambar 4.17. Pola nilai MSE yang dihasilkan pada perbandingan nilai MSE disetiap epoch tersebut menandakan bahwa jumlah epoch cukup berpengaruh terhadap akurasi nilai MSE yang dihasilkan model CNN LSTM dengan optimasi Adam dan *learning rate* 0.001.

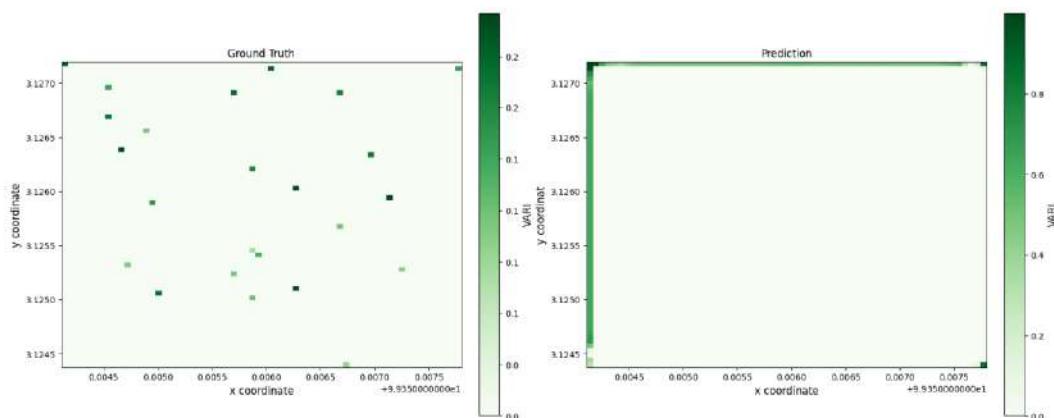
4.3.2 Model CNN LSTM *Optimizer* Adam Dengan *Learning Rate* 0.0001

Model ini menggunakan optimasi Adam dengan parameter *learning rate* sebesar 0.0001, gambar 4.18 merupakan grafik *loss* yang dihasilkan model sampai dengan epoch ke-3000 dapat dilihat bahwa tidak ada tanda terjadinya *overfit* pada model ini, grafik *loss* tersebut akan dilihat lebih rinci pada epoch kelipatan 500 yang akan dibahas pada sub bab ini.



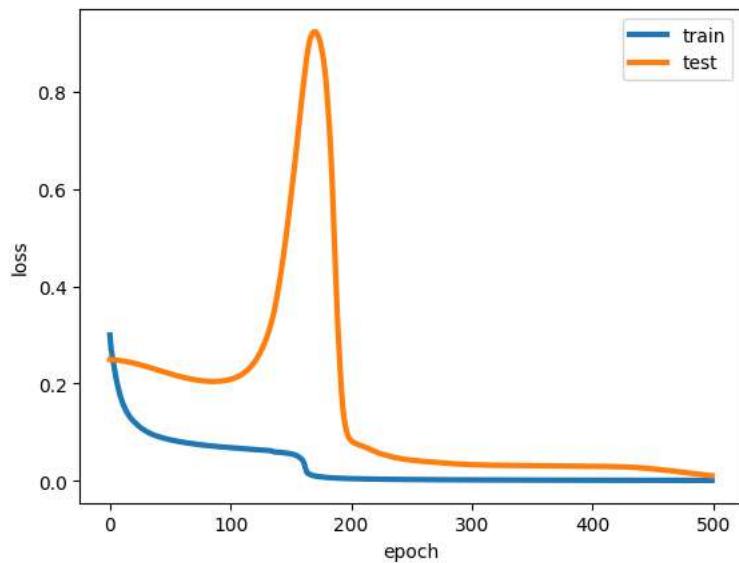
Gambar 4.18 Grafik *Loss* Model Adam Lr 0.0001

Gambar 4.19 merupakan perbandingan hasil prediksi antara *ground truth* dan hasil prediksi dengan menggunakan *dataset test* dengan jumlah epoch 500.



Gambar 4.19 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.0001 Pada Epoch 500

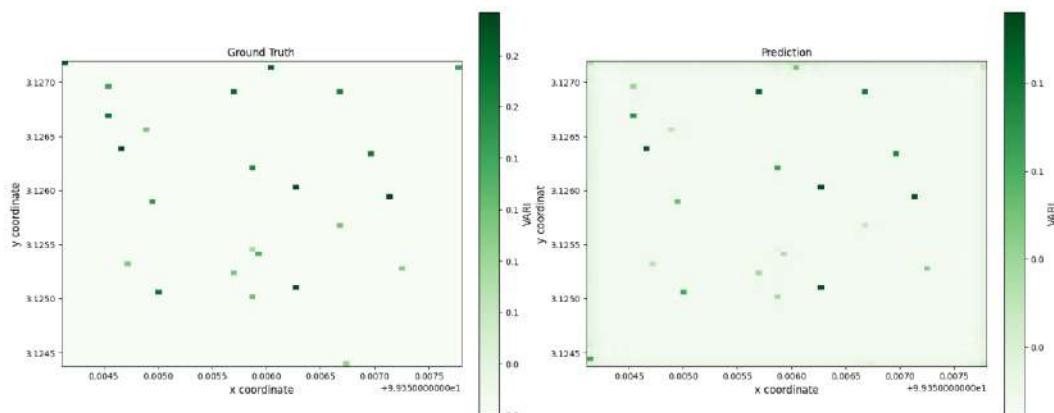
gambar 4.20 merupakan grafik *loss* yang dihasilkan pada model ini dengan epoch sebesar 500.



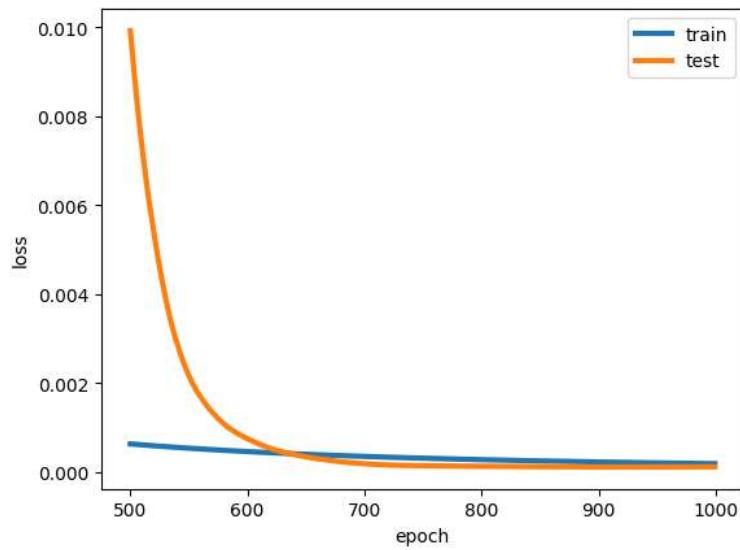
Gambar 4.20 Grafik *Loss* Model Adam Lr 0.0001 Pada Epoch 500

Dilihat dari perbandingan hasil prediksi dan *ground truth*, model pada epoch 500 ini hasil prediksi masih belum menghasilkan pola unik yang dimiliki *ground truth*, namun jika dilihat dari grafik *loss* yang dihasilkan model masih dalam tahap belajar karena nilai *loss* yang dihasilkan masih terus menurun pada epoch 500 ini. Nilai MSE yang diberikan pada epoch 500 ini sebesar 0.0102.

Gambar 4.21 dan 4.22 merupakan perbandingan hasil prediksi dan *ground truth* dari pada penggunaan epoch sebanyak 1000 beserta grafik *loss* nya.

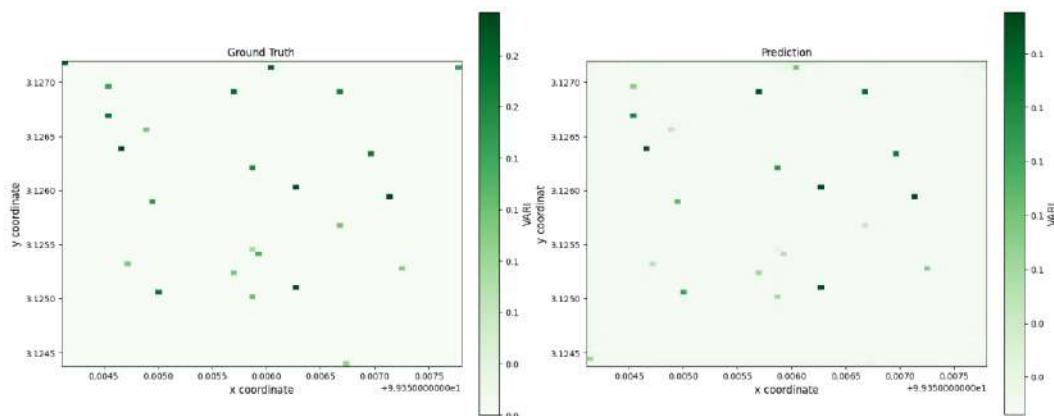


Gambar 4.21 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.0001 Pada Epoch 1000

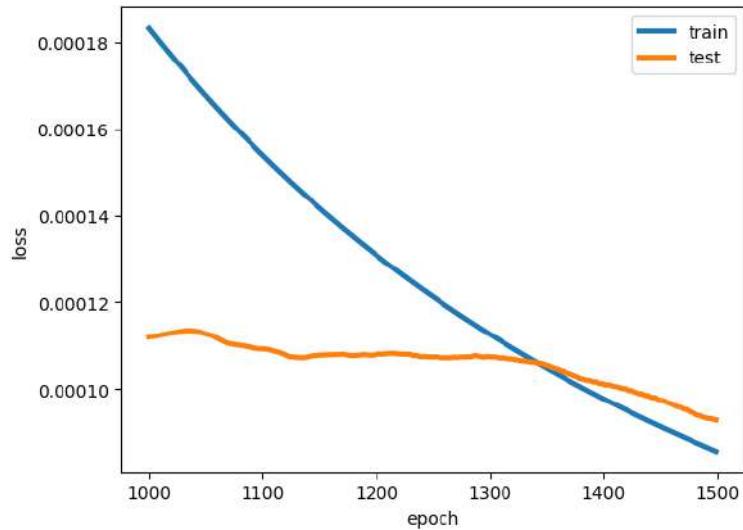


Gambar 4.22 Grafik *Loss* Model Adam Lr 0.0001 Pada Epoch 500-1000

Hasil yang diperlihatkan pada gambar 4.21 yakni hasil dari prediksi sudah mampu untuk memberikan pola unik yang dimiliki *ground truth*, dengan melihat kondisi grafik *loss* yang diberikan juga mengalami penurunan hingga epoch 1000 merupakan tanda yang baik bahwa model melakukan pembelajaran dengan baik. Nilai MSE yang dihasilkan oleh model ini pada epoch 1000 mengalami penurunan nilai dibandingkan pada epoch 500 yakni nilai MSE yang dihasilkan sebesar 0.00011. Dilanjutkan pada gambar 4.23 dan gambar 4.24 merupakan hasil dari epoch sebesar 1500.

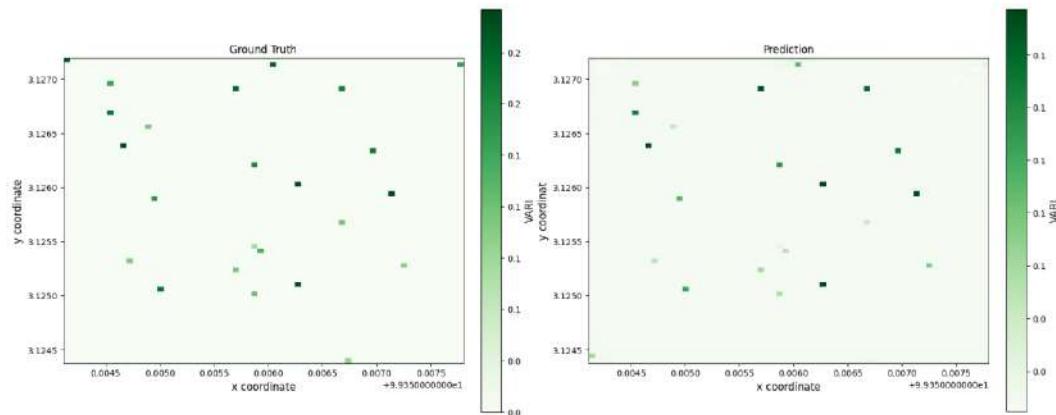


Gambar 4.23 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.0001 Pada Epoch 1500

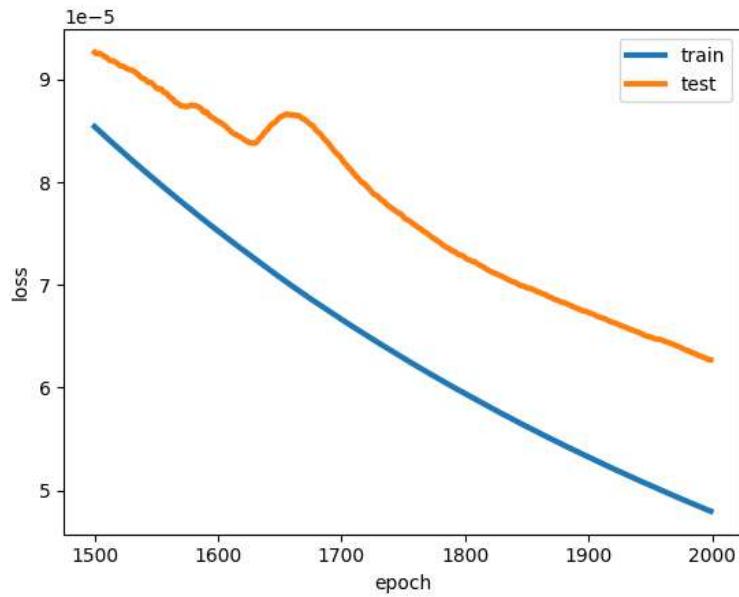


Gambar 4.24 Grafik *Loss* Model Adam Lr 0.0001 Pada Epoch 1000-1500

Pada epoch 1500 ini model yang dihasilkan juga sudah menghasilkan pola khusus yang dimiliki *ground truth*, dan jika dilihat pada grafik *loss* nilai yang dihasilkan selalu mengalami penurunan. Nilai MSE yang dihasilkan yakni sebesar $9,28 \cdot 10^{-5}$, nilai tersebut lebih kecil dibandingkan nilai MSE yang dihasilkan pada epoch 1000 sehingga hal tersebut merupakan indikasi baik bahwa model melakukan pembelajaran dengan baik. Gambar 4.25 dan 2.26 merupakan hasil yang diperoleh pada epoch 2000.

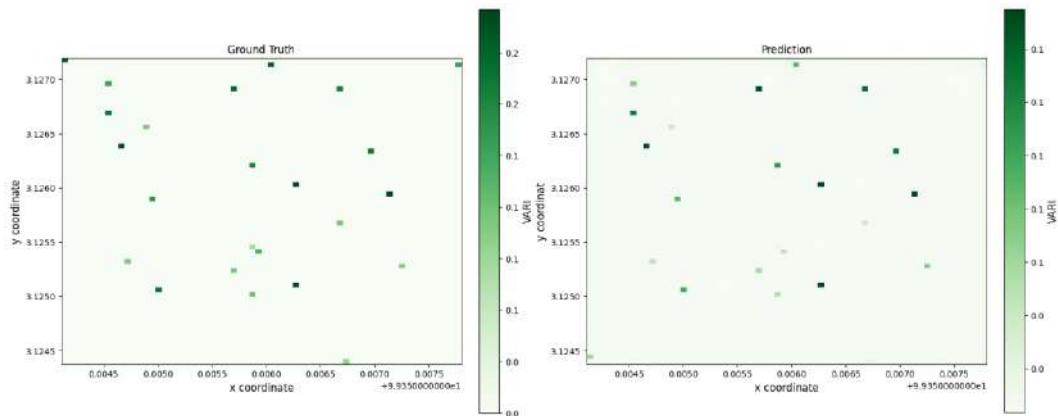


Gambar 4.25 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.0001 Pada Epoch 2000

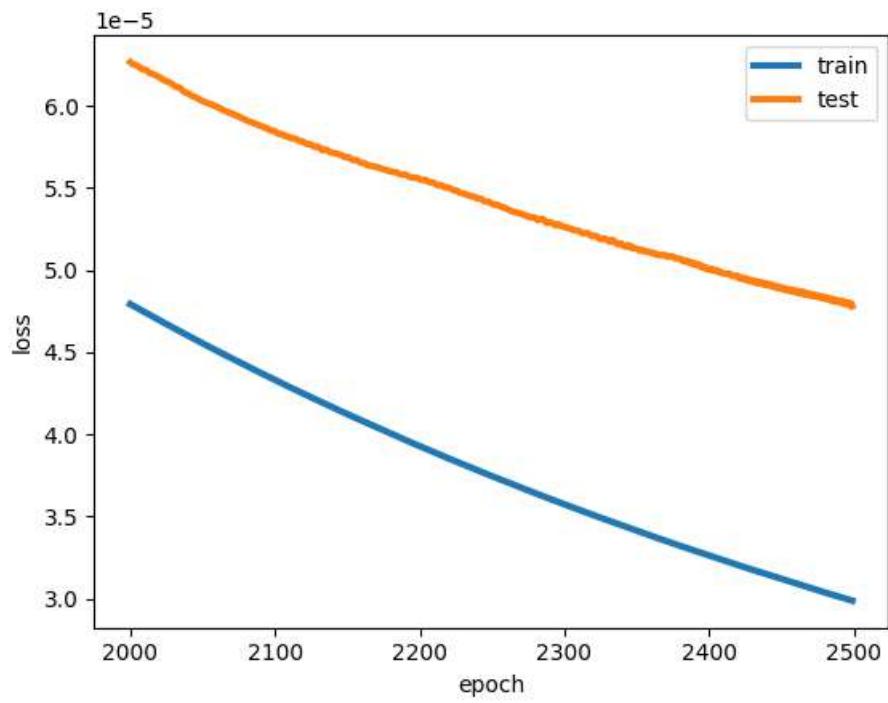


Gambar 4.26 Grafik *Loss* Model Adam Lr 0.0001 Pada Epoch 1500-2000

Pada epoch 2000 ini dilihat dari grafik *loss* yang dihasilkan terlihat pola nilai yang menurun, dengan nilai MSE yang dihasilkan yakni sebesar $6,27 \cdot 10^{-5}$ merupakan nilai yang lebih kecil dibandingkan dengan nilai MSE yang dihasilkan dari jumlah epoch kurang dari 2000, sehingga penurunan nilai MSE ini merupakan indikasi baik bahwa model bekerja dengan baik. Gambar 4.27 dan 4.28 merupakan hasil dari model dengan jumlah epoch 2500.

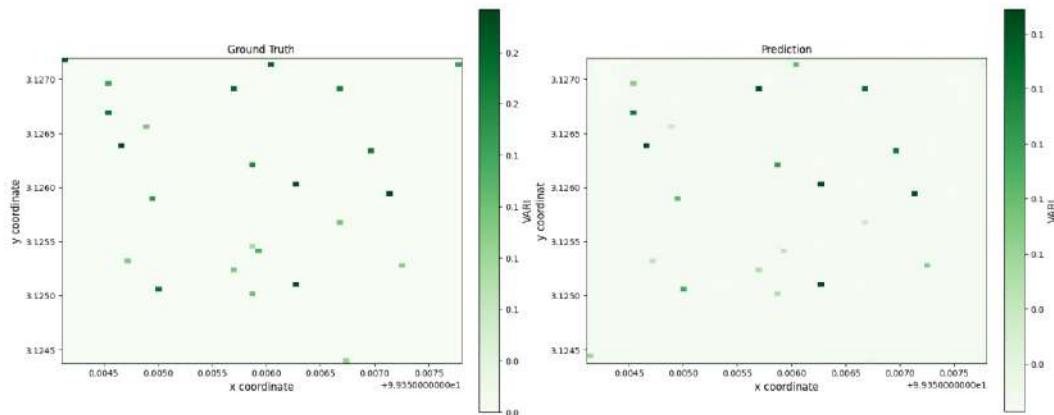


Gambar 4.27 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.0001 Pada Epoch 2500

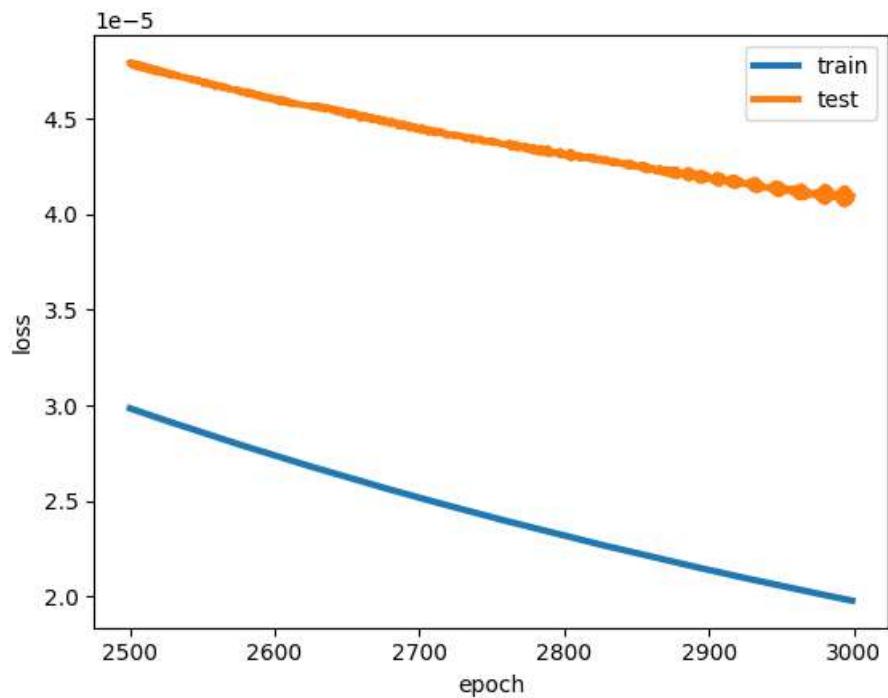


Gambar 4.28 Grafik *Loss* Model Adam Lr 0.0001 Pada Epoch 2000-2500

Hasil yang diberikan pada model ini menunjukkan hasil yang baik, dengan indikator nilai MSE yang menurun sampai dengan epoch 2500 diperoleh nilai MSE sebesar $4,78 \cdot 10^{-5}$. Gambar 4.29 dan gambar 4.30 merupakan hasil prediksi dan grafik *loss* pada epoch ke-3000.



Gambar 4.29 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.0001 Pada Epoch 3000



Gambar 4.30 Grafik *Loss* Model Adam Lr 0.0001 Pada Epoch 2500-3000

Hasil yang ditunjukan pada epoch ke-3000 menunjukan hasil yang baik dengan nilai MSE sebesar $4,09 \cdot 10^{-5}$ yang merupakan nilai MSE terkecil dibandingkan dengan epoch sebelumnya. Rincian perbandingan nilai MSE pada setiap 500 iterasi epoch dapat dilihat pada tabel 4.8.

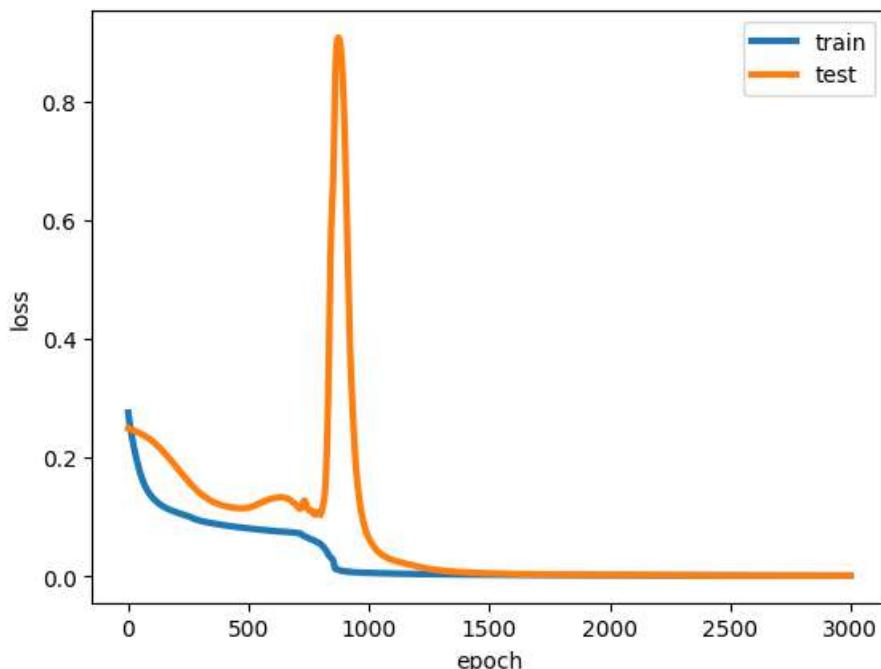
Tabel 4.8 Rincian Nilai MSE Pada Data *Test* Dimodel Adam Lr 0.0001 Di setiap 500 Epoch

MSE	Epoch					
	500	1000	1500	2000	2500	3000
	0.01022	0.00011	$9,28 \cdot 10^{-5}$	$6,27 \cdot 10^{-5}$	$4,78 \cdot 10^{-5}$	$4,09 \cdot 10^{-5}$

Dapat disimpulkan bahwa jumlah epoch yang digunakan pada model CNN LSTM dengan optimasi Adam dan *learning rate* 0.0001 memiliki keterkaitan dimana jumlah epoch yang semakin tinggi akan menghasilkan nilai MSE yang lebih rendah dibanding hasil yang diberikan pada jumlah epoch yang lebih rendah.

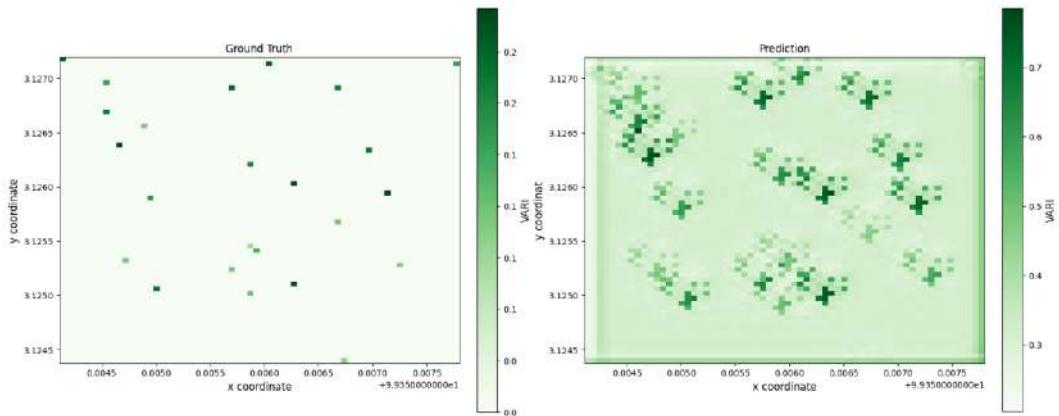
4.3.3 Model CNN LSTM Optimizer Adam Dengan Learning Rate 0.00001

Pada eksperimen ini model menggunakan optimasi Adam dengan *learning rate* sebesar 0.00001, hasil yang akan dipaparkan pada sub-sub bab 4.3.3 ini akan membahas hasil prediksi dari *dataset test*. Gambar 4.31 merupakan grafik *loss* yang dihasilkan model ini sampai dengan epoch ke-3000 dapat dilihat dari grafik tersebut tidak ada indikasi terjadinya *overfit* yang dihasilkan dari model ini. Sampai dengan epoch ke-1000 grafik sudah tidak terlihat terjadi penurunan nilai, sehingga pada sub bab ini akan dilihat lebih rinci pada kelipatan 500 epoch bagaimana kondisi nilai *loss* yang dihasilkan model ini.

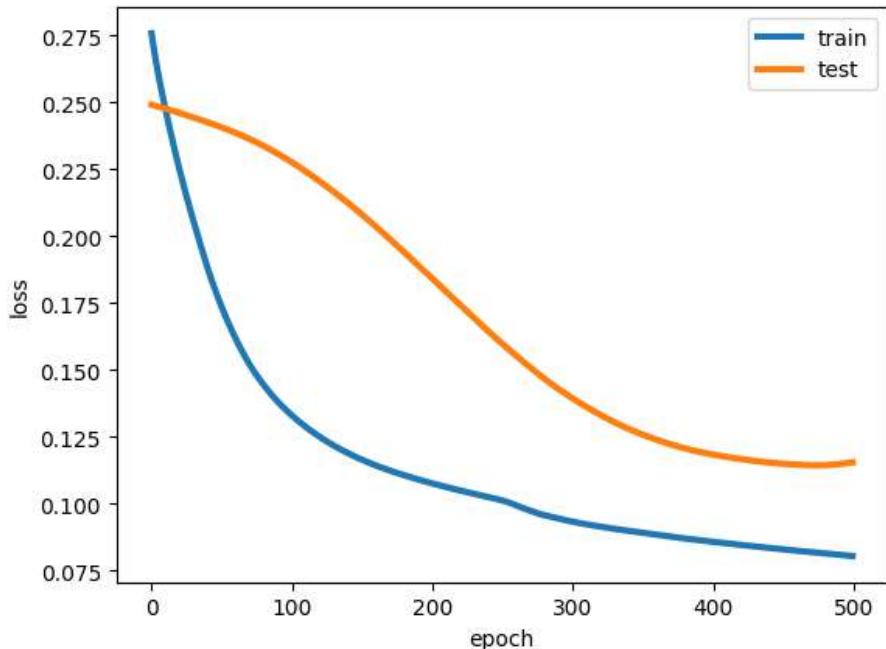


Gambar 4.31 Grafik *Loss* Model Adam Lr 0.00001

Gambar 4.32 merupakan perbandingan hasil prediksi dan *ground truth* pada epoch 500, sedangkan pada gambar 4.33 merupakan grafik *loss* yang dihasilkan model pada epoch sebanyak 500.



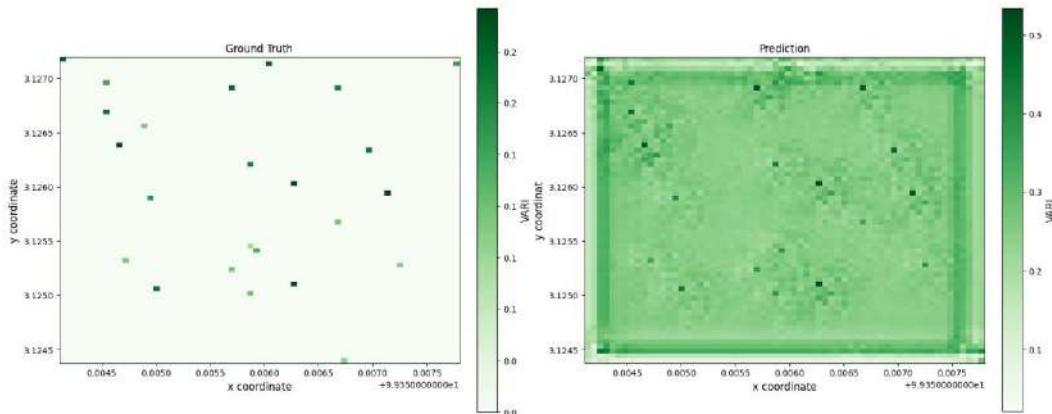
Gambar 4.32 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.00001 Pada Epoch 500



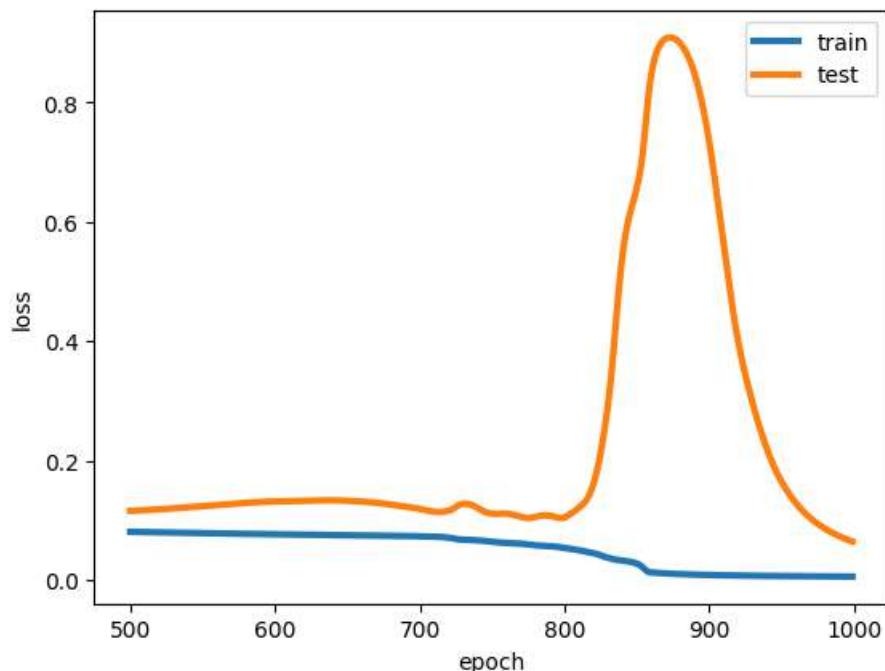
Gambar 4.33 Grafik *Loss* Model Adam Lr 0.00001 Pada Epoch 500

Hasil prediksi yang dihasilkan oleh model ini pada gambar 4.32 sudah menghasilkan pola unik yang dimiliki pada *ground truth* dan jika diperhatikan pada grafik *loss* nya juga memiliki pola menurun yang mengindikasikan bahwa model melakukan pembelajaran dengan baik. Nilai MSE yang dihasilkan model pada epoch 500 ini sebesar 0.11539, nilai MSE yang dihasilkan pada model ini cenderung lebih tinggi jika dibandingkan 2 model sebelumnya yang menggunakan

learning rate yang lebih rendah. Gambar 4.34 dan gambar 4.35 merupakan hasil dari epoch sebesar 1000.



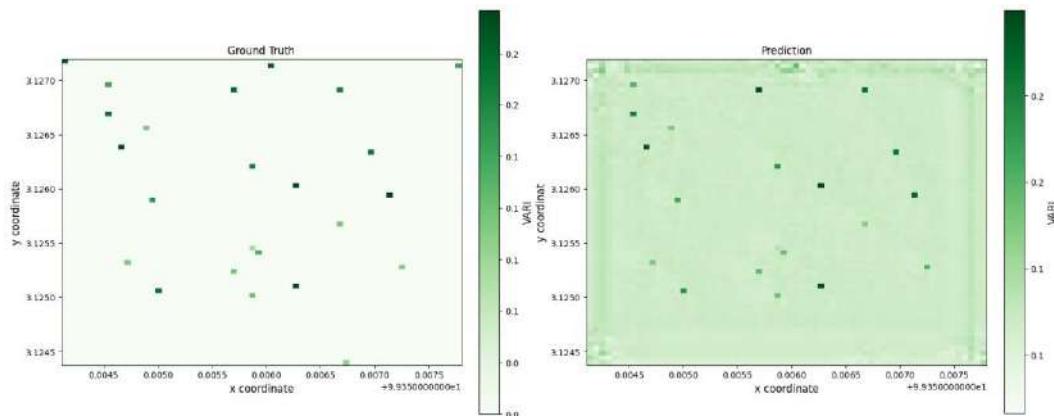
Gambar 4.34 Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.00001 Pada Epoch 1000



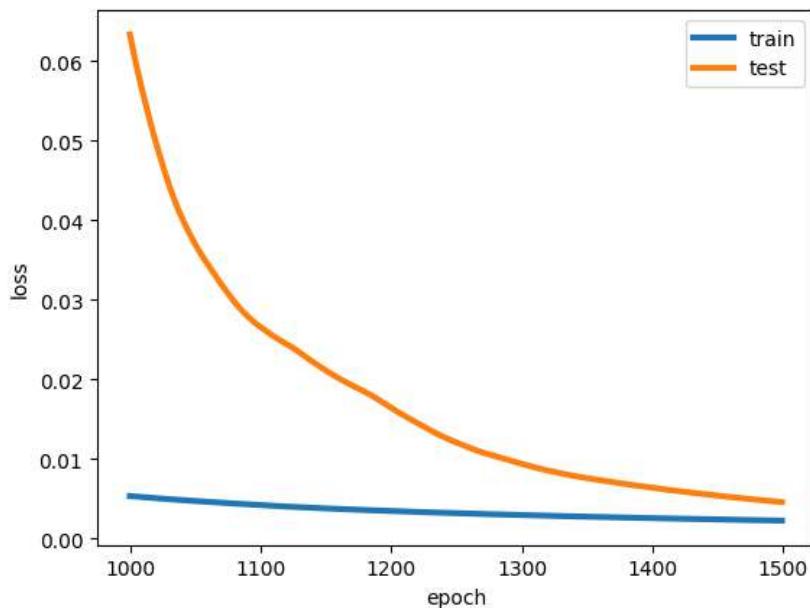
Gambar 4.35 Grafik *Loss* Model Adam Lr 0.00001 Pada Epoch 500-1000

Grafik *loss* yang dihasilkan model pada epoch 1000 mengalami pola unik yakni terdapat peningkatan nilai *loss* yang signifikan pada rentang epoch 800 sampai 900 namun pada epoch 900 sampai dengan 1000 nilai *loss* yang dihasilkan kembali mengalami penurunan nilai. Dengan nilai MSE sebesar 0.06417 model

dapat memprediksi dengan cukup baik dengan indikasi bahwa titik yang menandakan pohon sudah dapat diprediksi oleh model namun pada bagian yang tidak terdapat pohon masih diprediksi terdapat pohon kelapa sawit. Gambar 4.36 dan gambar 4.37 merupakan hasil prediksi beserta grafik *loss* pada epoch 1500.



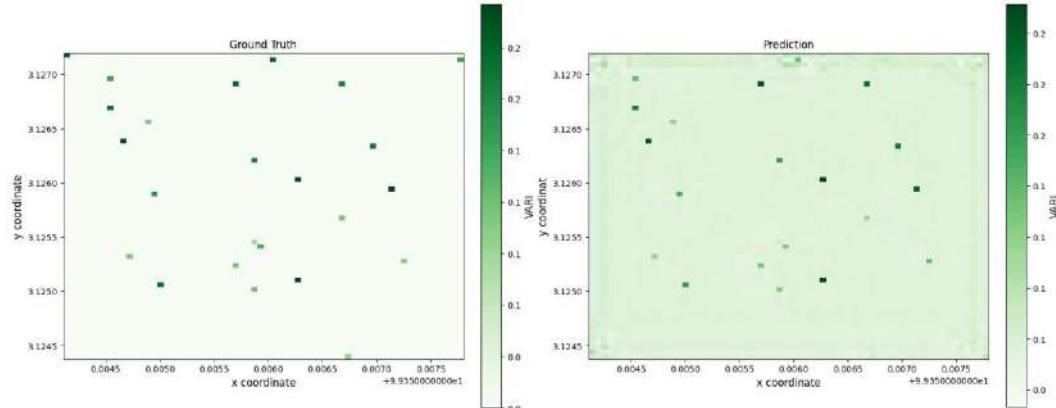
Gambar 4.36 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.00001 Pada Epoch 1500



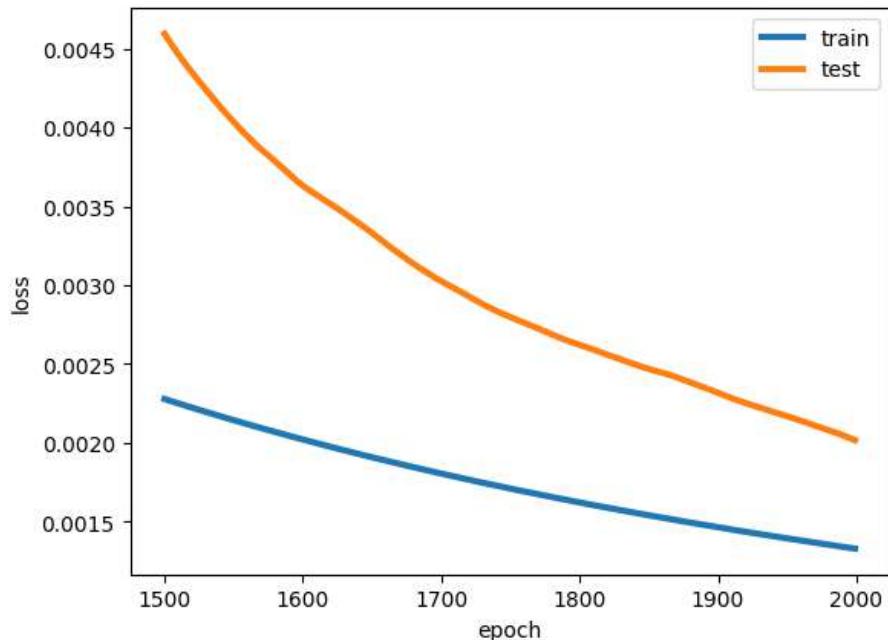
Gambar 4.37 Grafik *Loss* Model Adam Lr 0.00001 Pada Epoch 1000-1500

Hasil yang diperoleh dari model ini pada epoch 1500 yakni *loss* yang dihasilkan mengalami penurunan pada setiap iterasinya, penurunan tersebut dapat dilihat pada gambar 4.37. Dengan nilai MSE sebesar 0.00461, model sudah dapat

memprediksi dengan baik namun pada bagian yang bukan merupakan pohon kelapa sawit masih diprediksi sebagai pohon kelapa sawit. Dilanjutkan pada epoch 2000 hasil yang diperoleh dapat dilihat pada gambar 4.38 dan gambar 4.39.



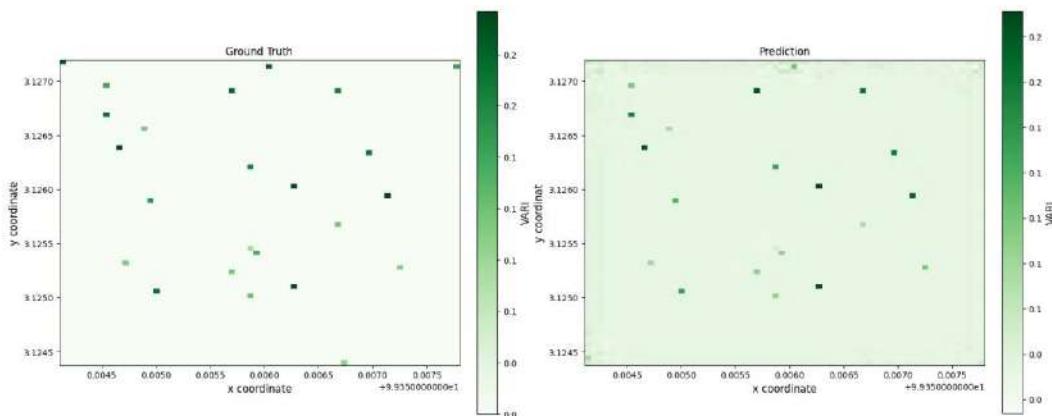
Gambar 4.38 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.00001 Pada Epoch 2000



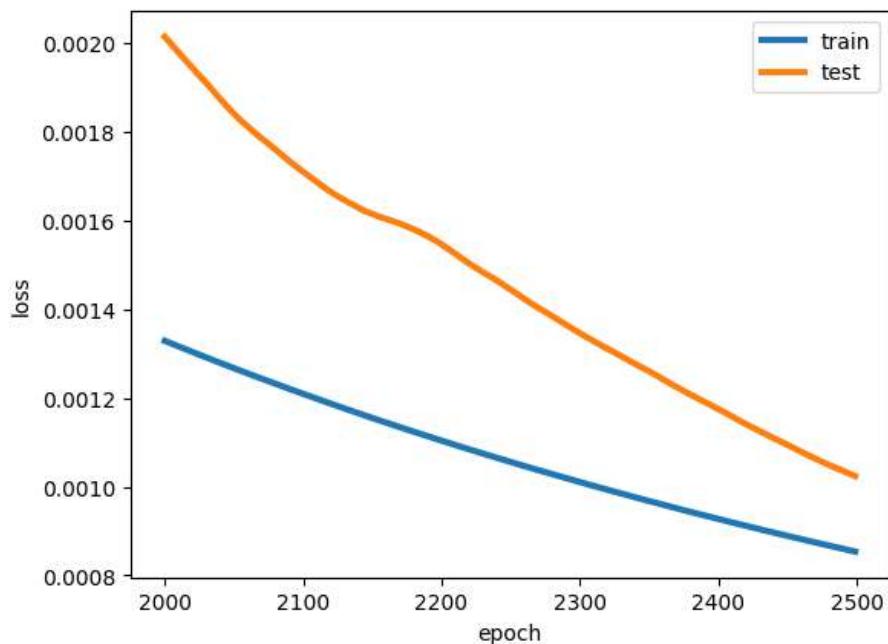
Gambar 4.39 Grafik *Loss* Model Adam Lr 0.00001 Pada Epoch 1500-2000

Hasil yang ditunjukan pada gambar 4.38 yakni pada bagian yang bukan merupakan pohon kelapa sawit nilai yang diperoleh sudah menurun dibandingkan dengan hasil yang diperoleh pada epoch sebelumnya, dan *noise* atau titik yang

seharusnya tidak memiliki nilai namun diprediksi memiliki nilai yang ada pada bagian sekitar titik pohon kelapa sawit sudah mulai menghilang. Nilai MSE yang dihasilkan pada model ini juga sudah menurun dibandingkan dengan nilai MSE pada epoch sebelumnya yakni nilai MSE yang diperoleh pada epoch 2000 sebesar 0.00202. Gambar 4.40 dan 4.41 merupakan hasil yang diperoleh model pada epoch sebesar 2500.

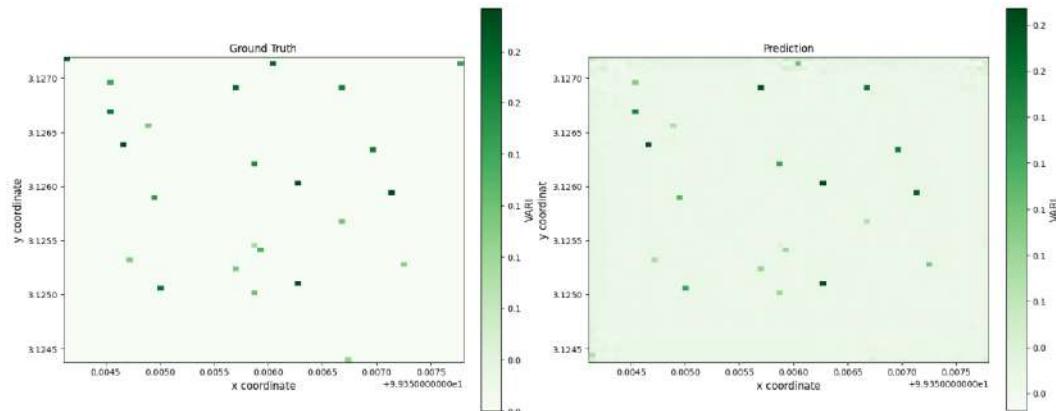


Gambar 4.40 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.00001 Pada Epoch 2500

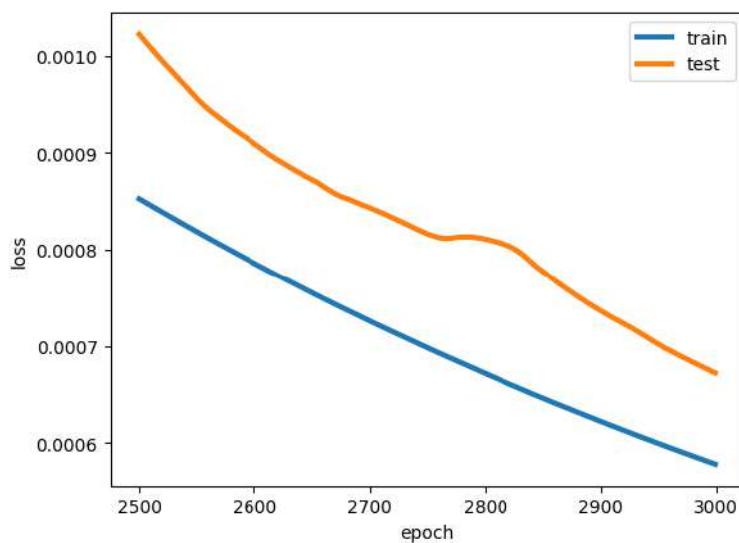


Gambar 4.41 Grafik *Loss* Model Adam Lr 0.00001 Pada Epoch 2000-2500

Hasil yang diperlihatkan pada gambar 4.38 yang merupakan grafik *loss* menandakan bahwa model bekerja dengan baik dalam proses pembelajarannya dengan indikasi bahwa sampai dengan epoch yang ke-2500 nilai *loss* yang dihasilkan masih menurun. Dengan nilai MSE yang dihasilkan yakni sebesar 0.00102 model sudah lebih baik dalam membedakan titik yang terdapat pohon kelapa sawit dan yang tidak dengan indikasi warna yang lebih terang pada *grid* yang tidak terdapat pohon kelapa sawit. Gambar 4.42 dan gambar 4.43 merupakan hasil prediksi dari model pada epoch 3000 yang merupakan batas penggunaan epoch yang digunakan pada penelitian ini.



Gambar 4.42 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model Adam Lr 0.00001 Pada Epoch 3000



Gambar 4.43 Grafik *Loss* Model Adam Lr 0.00001 Pada Epoch 2500-3000

Hasil yang diperlihatkan grafik pada gambar 4.43 mengindikasikan bahwa nilai *loss* yang dihasilkan dalam proses pembelajaran model bekerja dengan baik dengan indikasi bahwa nilai *loss* yang terus menurun pada setiap iterasinya. Nilai MSE yang diperoleh pada epoch ke-3000 ini sebesar 0.00067, nilai MSE yang dihasilkan pada epoch ke-3000 ini merupakan nilai MSE terkecil yang dihasilkan pada model Adam dengan *learning rate* 0.00001. Dilihat dari hasil perbandingan yang diperoleh pada gambar 4.39 juga merupakan hasil yang terbaik dibandingkan dengan epoch-epoch sebelumnya. Rincian perbandingan nilai MSE yang diperoleh pada setiap 500 epoch sampai dengan epoch ke-3000 dapat dilihat pada tabel 4.9.

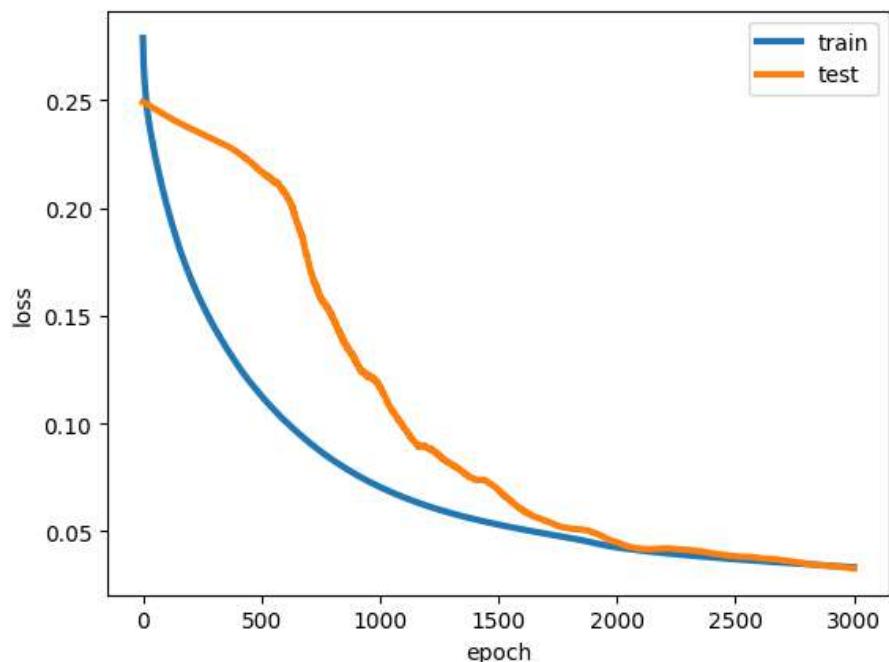
Tabel 4.9 Rincian Nilai MSE Pada Data *Test* Dimodel Adam Lr 0.00001 Di setiap 500 Epoch

	Epoch					
	500	1000	1500	2000	2500	3000
MSE	0.11539	0.06417	0.00461	0.00202	0.00102	0.00067

Dapat diperhatikan bahwa nilai epoch yang semakin tinggi akan menghasilkan nilai MSE yang semakin rendah, dimana nilai MSE yang semakin rendah mengindikasikan bahwa model akan menghasilkan hasil prediksi yang lebih akurat.

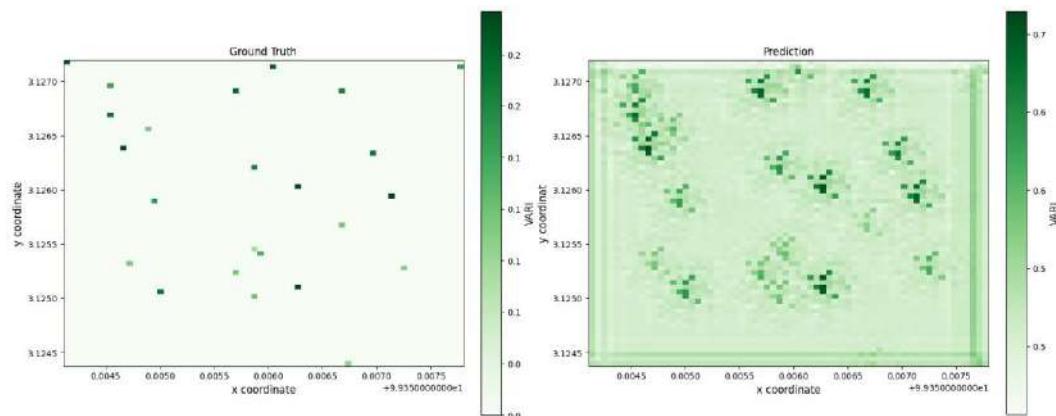
4.3.4 Model CNN LSTM *Optimizer SGD* Dengan *Learning Rate* 0.001

Pada model ini akan menggunakan tuning parameter dengan menggunakan metode optimasi SGD pada *learning rate* 0.001, gambar 4.44 merupakan grafik *loss* yang dihasilkan model ini, dapat dilihat dari grafik *loss* yang dihasilkan model ini tidak terdapat fluktuansi nilai *loss* sampai dengan epoch ke-3000 dengan demikian dapat ditarik kemungkinan bahwa model ini tidak memiliki indikasi adanya *overfit*. Penurunan nilai *loss* yang dihasilkan model juga terus menurun baik pada data *train* maupun data *test* sampai dengan epoch ke-3000.

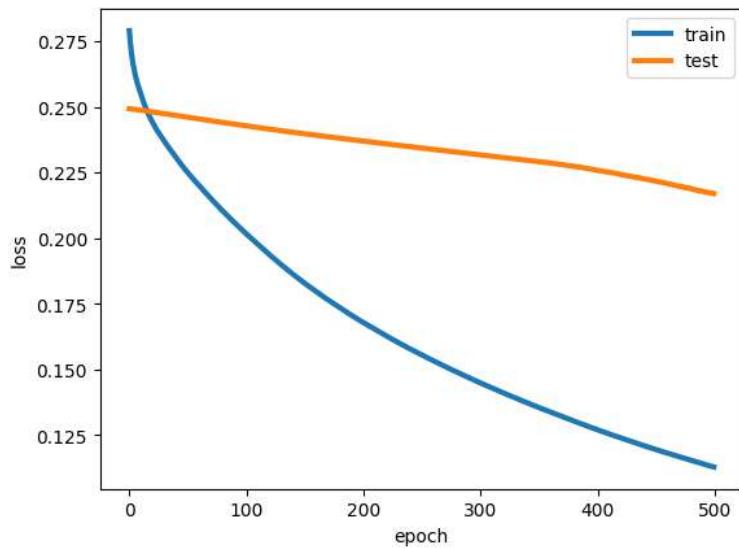


Gambar 4.44 Grafik *Loss* Model SGD Lr 0.001

Pada gambar 4.45 dan gambar 4.46 merupakan hasil prediksi model dan grafik nilai *loss* yang diberikan model pada epoch ke-0 sampai dengan ke-500.

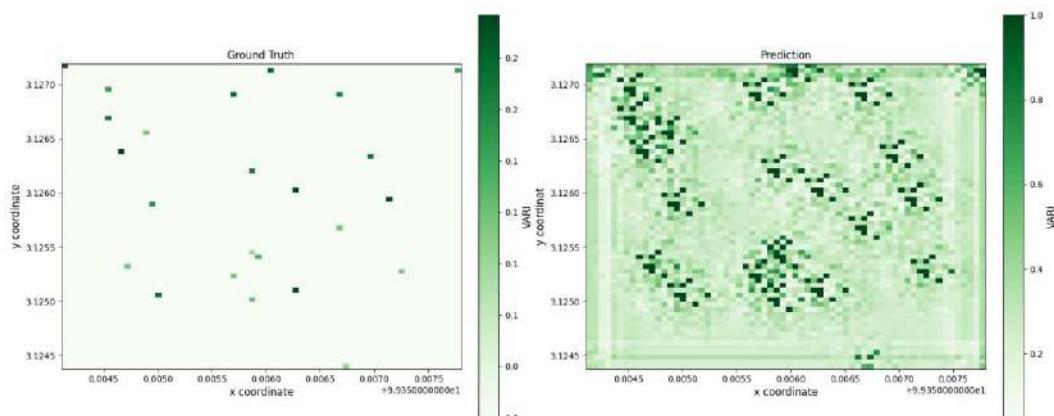


Gambar 4.45 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.001 Pada Epoch 500

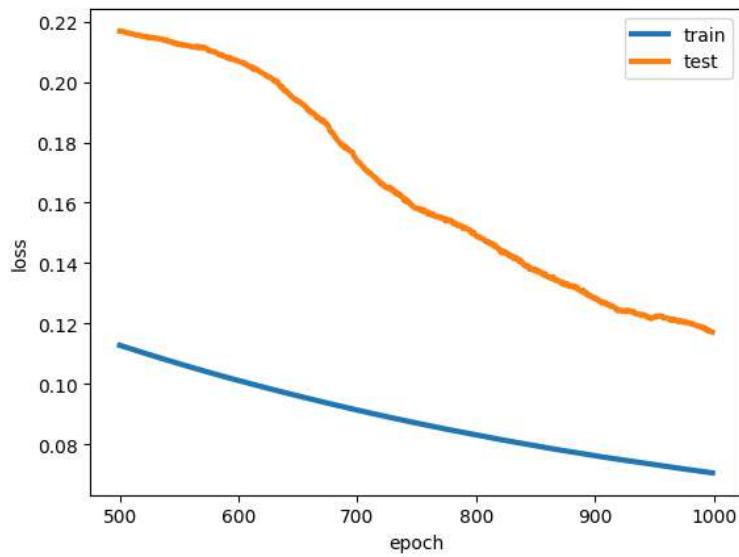


Gambar 4.46 Grafik *Loss* Model SGD Lr 0.001 Pada Epoch 500

Dilihat dari grafik *loss* yang dihasilkan sampai dengan epoch ke-500, nilai *loss* yang diperoleh mengalami penuruan baik pada data *train* dan data *test*. Hasil prediksi yang diperlihatkan pada gambar 4.45 juga sudah menghasilkan pola yang ada pada *ground truth* namun masih terdapat *grid* yang seharusnya tidak ada pohon kelapa sawit atau bernilai 0 namun diprediksi memiliki nilai pada model. Nilai MSE yang diperoleh model pada epoch ke-500 ini sebesar 0.21697. Gambar 4.47 dan gambar 4.48 merupakan hasil dari model pada epoch ke-1000.

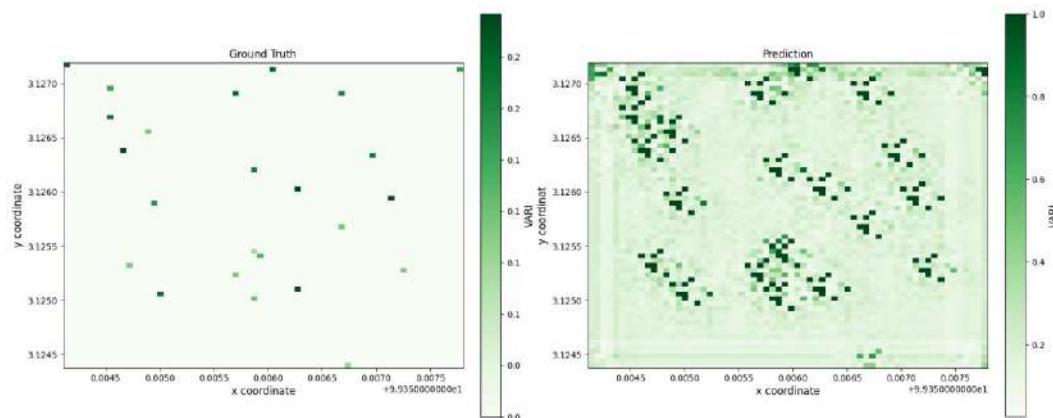


Gambar 4.47 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.001 Pada Epoch 1000

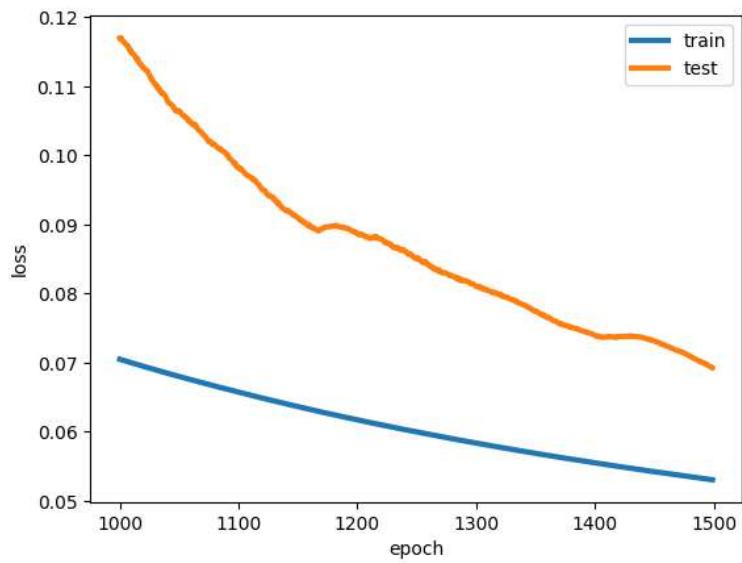


Gambar 4.48 Grafik *Loss* Model SGD Lr 0.001 Pada Epoch 500-1000

Dari Hasil yang diperoleh model mengalami penurunan nilai *loss* sampai dengan epoch ke-1000 baik pada data *train* maupun pada data *test*. Dengan nilai MSE sebesar 0.11717 model masih belum dapat menghilangkan *noise* atau titik yang seharusnya tidak memiliki nilai namun diprediksi memiliki nilai yang ada pada hasil prediksi. Gambar 4.49 dan gambar 4.50 merupakan hasil yang diberikan model pada epoch ke-1500.

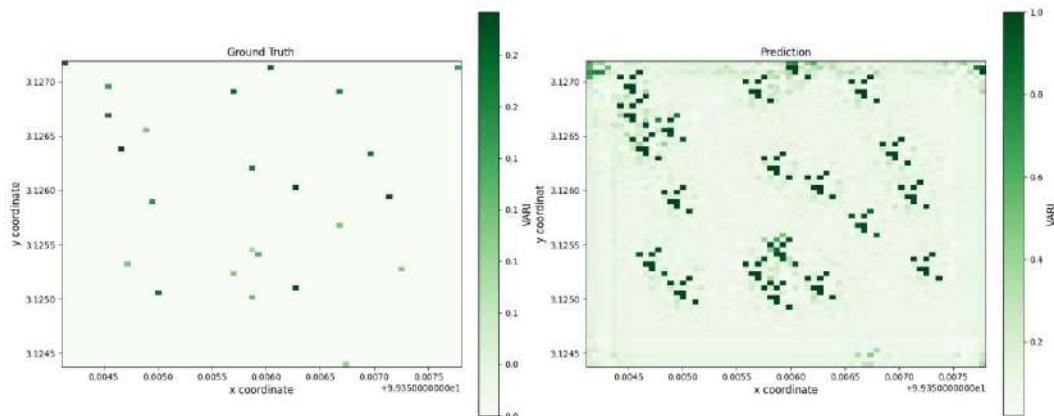


Gambar 4.49 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.001 Pada Epoch 1500

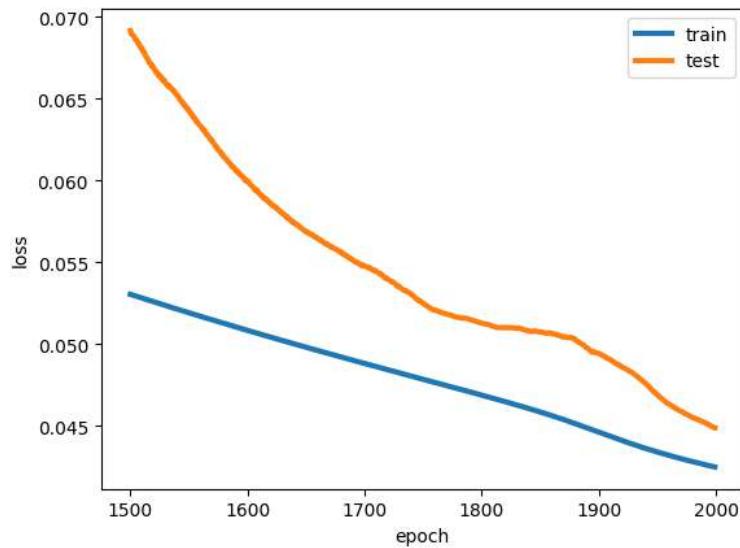


Gambar 4.50 Grafik *Loss* Model SGD Lr 0.001 Pada Epoch 1000-1500

Hasil sampai dengan epoch ke-1500 menunjukan indikasi yang baik dengan nilai *loss* yang terus menurun pada setiap iterasinya baik pada data *test* maupun data *train*. Diperoleh nilai MSE sebesar 0.06928 dengan nilai MSE tersebut jika dilihat pada gambar 4.45 model sudah dapat mengurangi *grid* yang seharusnya tidak memiliki nilai namun diprediksi memiliki nilai, namun masih terdapat *grid* yang seharusnya bernilai 0 tetapi diprediksi memiliki nilai oleh model. Gambar 4.51 dan gambar 4.52 merupakan hasil yang diberikan model pada epoch ke-2000.

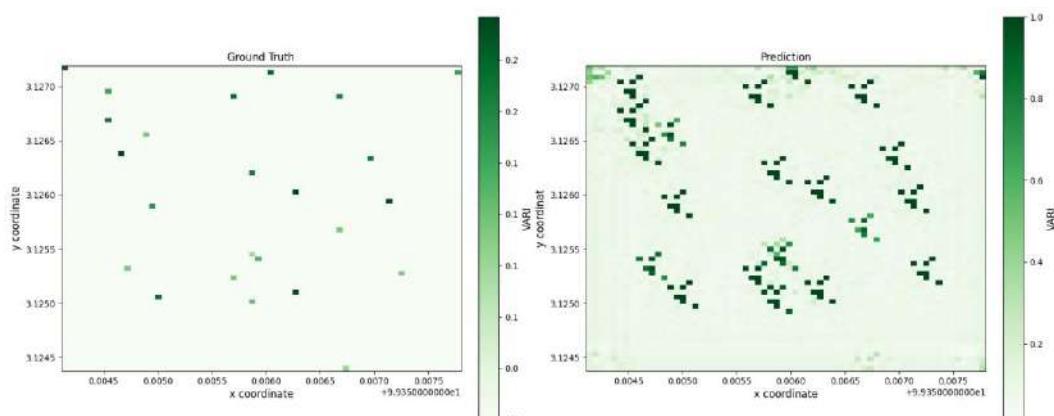


Gambar 4.51 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.001 Pada Epoch 2000

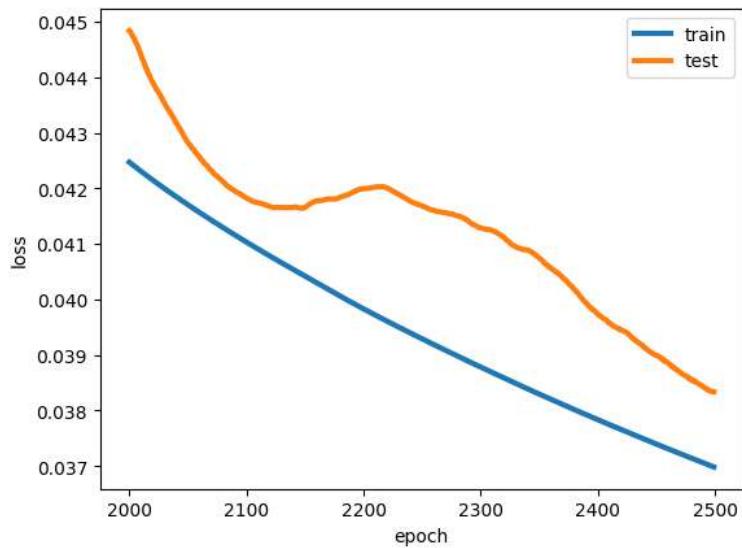


Gambar 4.52 Grafik *Loss* Model SGD Lr 0.001 Pada Epoch 1500-2000

MSE yang dihasilkan model pada epoch ke-2000 ini sebesar 0.04488, nilai MSE yang dihasilkan pada epoch ke-2000 ini memiliki penurunan nilai dibandingkan dengan epoch sebelumnya. Nilai MSE yang menurun ini merupakan indikasi yang baik bahwa model melakukan pembelajaran dengan baik dan didukung juga dengan nilai *loss* yang menurun pada setiap iterasinya sampai dengan epoch ke-2000, penurunan nilai *loss* terjadi pada data *test* maupun data *train*. Namun hasil yang terlihat pada gambar 4.51 model masih belum dapat mengenali *grid* yang terdapat pohon dan tidak. Gambar 4.53 dan gambar 4.54 merupakan hasil yang diberikan model pada epoch ke-2500.

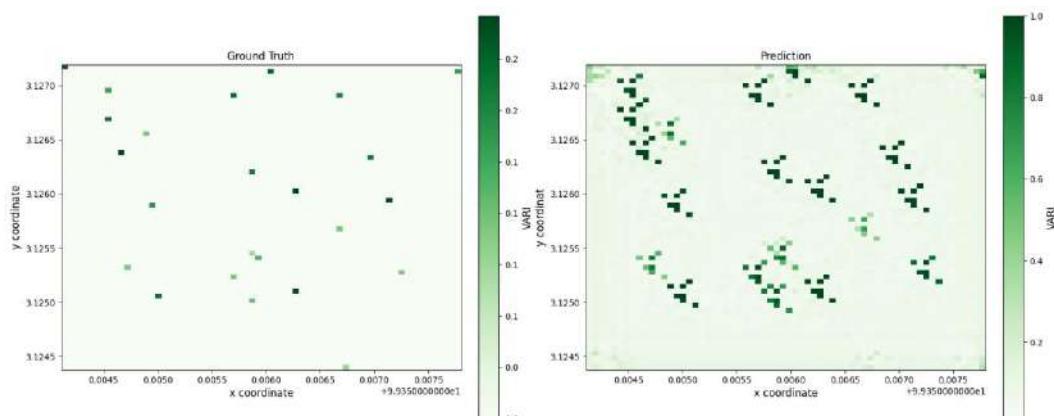


Gambar 4.53 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.001 Pada Epoch 2500

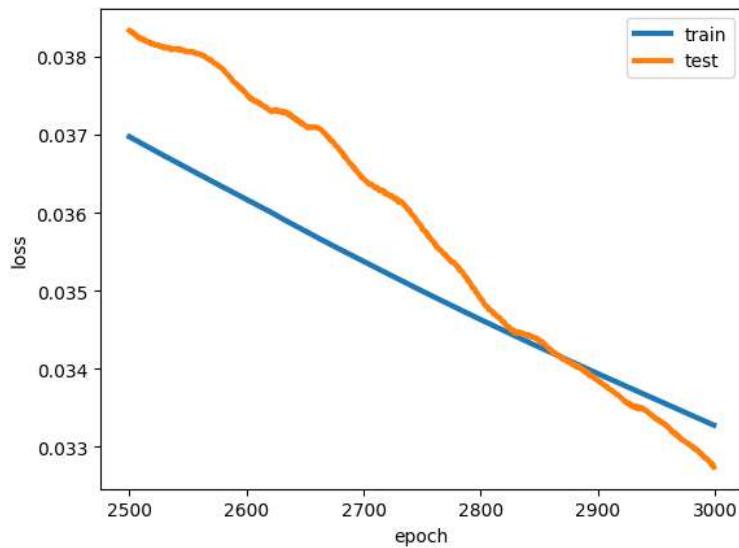


Gambar 4.54 Grafik *Loss* Model SGD Lr 0.001 Pada Epoch 2000-2500

Dari hasil yang diberikan model pada epoch ke-2500, nilai MSE yang dihasilkan yakni sebesar 0.03834 dimana nilai MSE yang dihasilkan tersebut mengalami penurunan dibanding nilai MSE yang diperoleh pada epoch sebelumnya. Grafik *loss* yang diperlihatkan juga mengalami peningkatan dengan nilai *loss* yang dihasilkan memiliki pola menurun pada setiap iterasinya kecuali pada epoch 2100 sampai dengan 2200 terdapat peningkatan nilai *loss*. Gambar 4.55 dan gambar 4.56 merupakan hasil yang diberikan model pada epoch 3000, yang mana epoch 3000 merupakan batasan jumlah epoch yang digunakan dan dianalisis pada penelitian ini.



Gambar 4.55 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.001 Pada Epoch 3000



Gambar 4.56 Grafik Loss Model SGD Lr 0.001 Pada Epoch 2500-3000

Hasil yang diberikan model pada epoch ke-3000 yakni pada nilai *loss* data test memiliki nilai yang lebih rendah dibandingkan nilai *loss* yang dimiliki data *train*. Nilai MSE yang diperoleh dari model ini pada epoch ke-3000 yakni sebesar 0.03274, nilai tersebut merupakan nilai terkecil yang diperoleh model dengan optimasi SGD dan *learning rate* 0.001. Namun sampai dengan epoch ke-3000 hasil prediksi yang diberikan model ini masih belum dapat melakukan prediksi dengan baik, dimana masih dilakukannya prediksi nilai VARI pada *grid* yang sebenarnya tidak memiliki nilai VARI. Rincian perbandingan nilai MSE pada model dengan parameter SGD dan *learning rate* 0.001 dapat dilihat pada tabel 4.10.

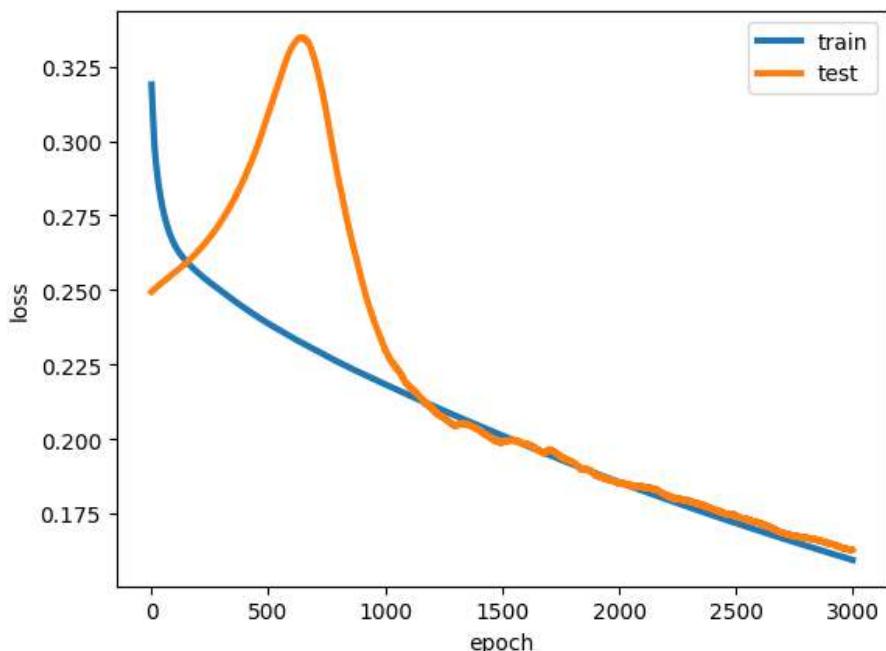
Tabel 4.10 Rincian Nilai MSE Pada Data *Test* Dimodel SGD Lr 0.001 Di setiap 500 Epoch

	Epoch					
	500	1000	1500	2000	2500	3000
MSE	0.21697	0.11717	0.06928	0.04488	0.03834	0.03274

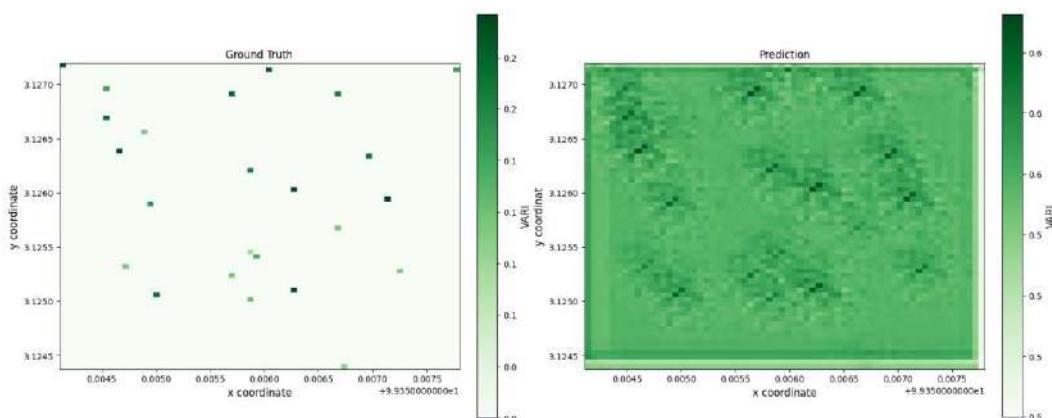
Dapat disimpulkan bahwa jumlah epoch berpengaruh terhadap penurunan nilai MSE yang merupakan nilai akurasi yang digunakan pada model dengan optimasi SGD dan *learning rate* 0.001, dimana semakin tinggi jumlah epoch yang digunakan nilai MSE akan semakin kecil atau menurun.

4.3.5 Model CNN LSTM Optimizer SGD Dengan Learning Rate 0.0001

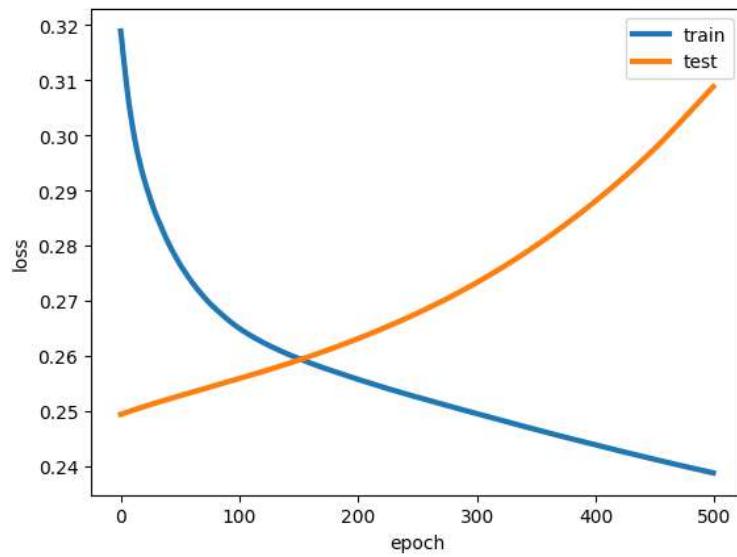
Pada Eksperimen ini model akan menggunakan optimasi SGD dengan *learning rate* 0.0001. Gambar 4.57 merupakan grafik *loss* yang dihasilkan model ini sampai dengan epoch ke-3000 dan akan dijelaskan lebih rinci di sub bab ini dengan kondisi pengamatan setiap kelipatan 500 epoch. Sedangkan Gambar 4.58 dan gambar 4.59 menunjukkan hasil pada saat epoch 500.



Gambar 4.57 Grafik *Loss* Model SGD Lr 0.0001

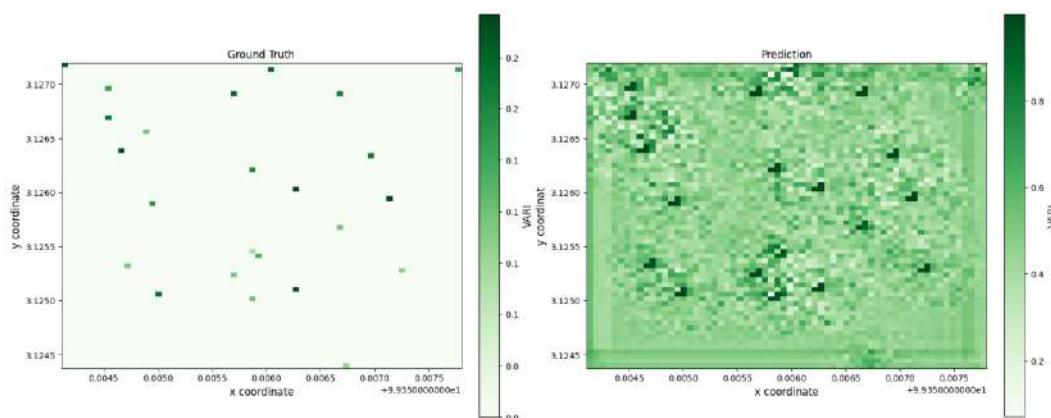


Gambar 4.58 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.0001 Pada Epoch 500

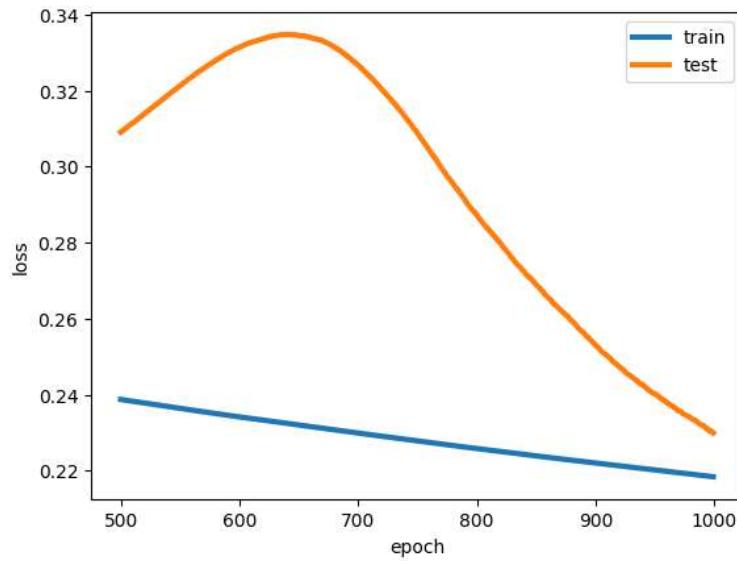


Gambar 4.59 Grafik *Loss* Model SGD Lr 0.0001 Pada Epoch 500

Dilihat dari grafik *loss* yang ada pada gambar 4.59 terjadi peningkatan nilai *loss* pada data *test* sedangkan pada data *train* nilai *loss* menurun pada setiap iterasinya. Nilai MSE yang diperoleh pada epoch ke-500 yakni sebesar 0.30884, dengan nilai MSE tersebut hasil prediksi sudah dapat memberikan pola yang ada pada *ground truth* namun pada *grid* yang tidak memiliki nilai atau sama dengan 0 model memprediksi *grid* tersebut sama dengan *grid* yang memiliki nilai. Gambar 4.60 dan gambar 4.61 merupakan hasil yang diberikan model pada epoch ke-1000.

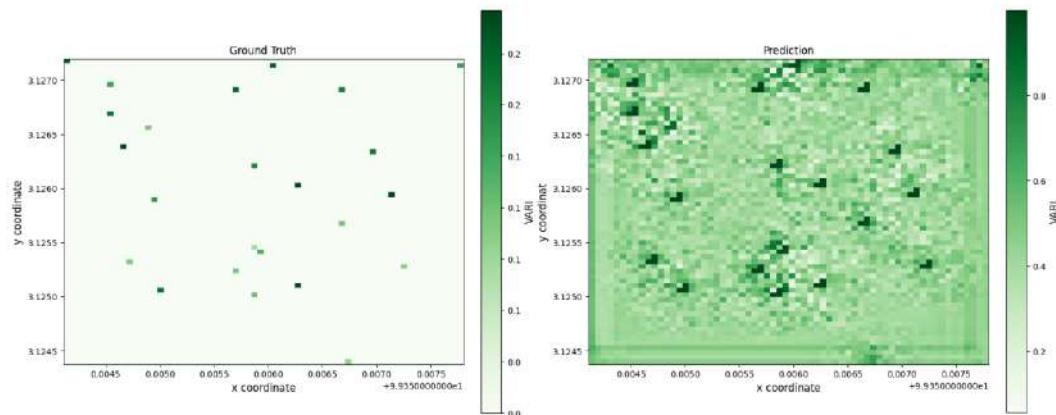


Gambar 4.60 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.0001 Pada Epoch 1000

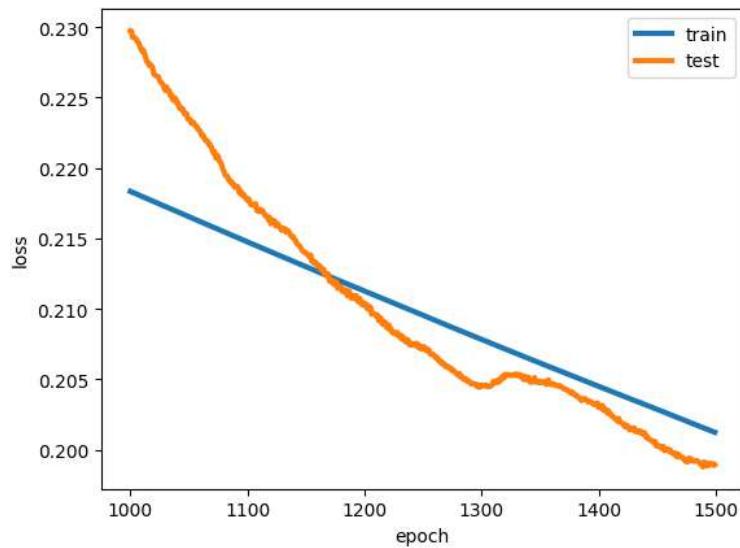


Gambar 4.61 Grafik *Loss* Model SGD Lr 0.0001 Pada Epoch 500-1000

Terjadi penurunan nilai *loss* yang terjadi pada data *test* diePOCH ke-200 sampai dengan epoch ke-1000, sedangkan pada data *train* nilai *loss* masih mengalami penurunan. Nilai MSE yang diberikan model pada epoch ke-1000 yakni sebesar 0.22991, dengan nilai MSE tersebut model masih belum dapat membedakan *grid* yang memang terdapat nilai VARI atau yang tidak terdapat nilai VARI. Gambar 4.62 dan gambar 4.63 menunjukkan hasil dari model pada epoch ke-1500.

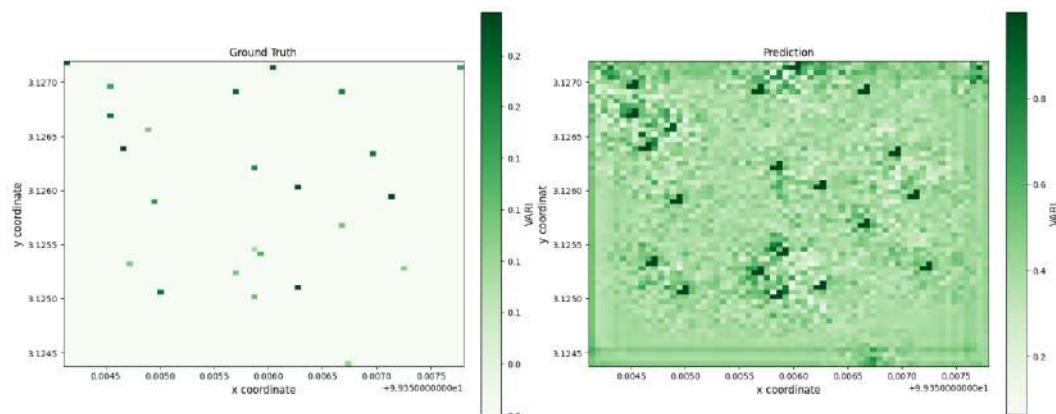


Gambar 4.62 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.0001 Pada Epoch 1500

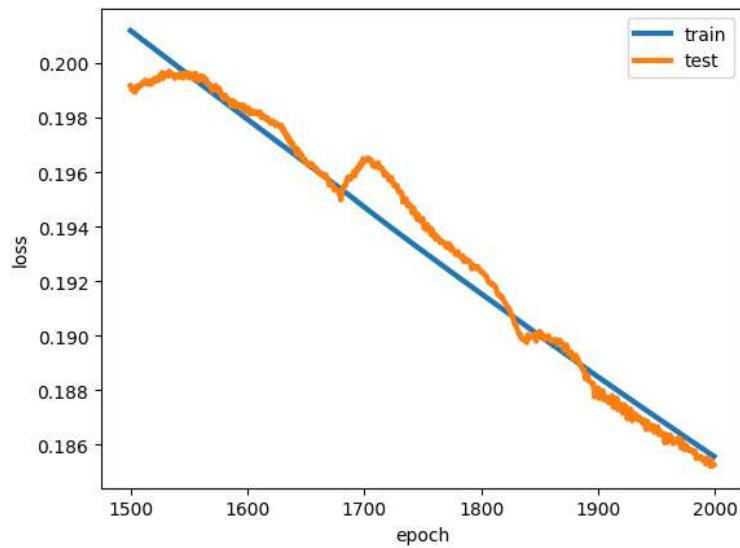


Gambar 4.63 Grafik *Loss* Model SGD Lr 0.0001 Pada Epoch 1000-1500

Dari grafik *loss* yang dihasilkan pada epoch ke-1000 sampai dengan 1500 terjadi penurunan nilai *loss* yang dihasilkan pada setiap iterasinya, nilai *loss* yang dihasilkan pada data *test* bahkan memiliki nilai yang lebih rendah dibandingkan dengan nilai *loss* yang dihasilkan pada data *train*. Nilai MSE yang diperoleh pada epoch ke-1500 ini sebesar 0.19893, dengan nilai MSE tersebut model masih memprediksi nilai pada *grid* yang dimana pada keadaan sebenarnya *grid* tersebut tidak memiliki nilai VARI. Gambar 4.64 dan gambar 4.65 merupakan lanjutan hasil yang diberikan model pada epoch ke-2000.

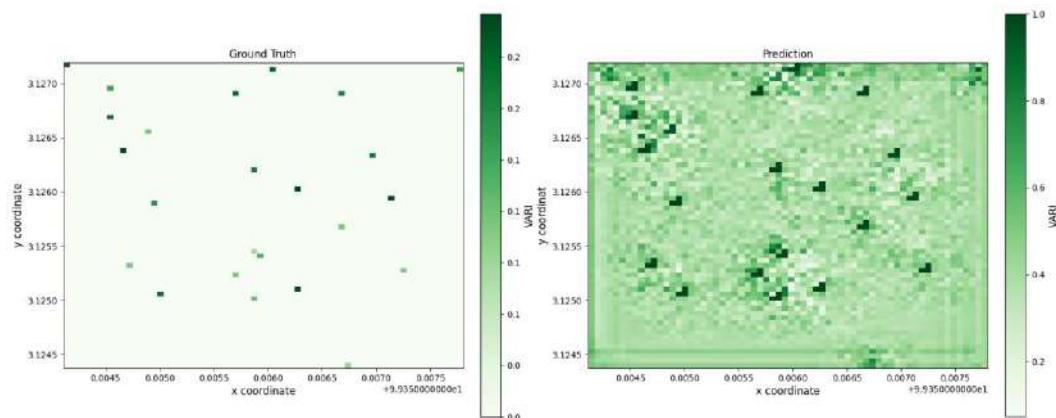


Gambar 4.64 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.0001 Pada Epoch 2000

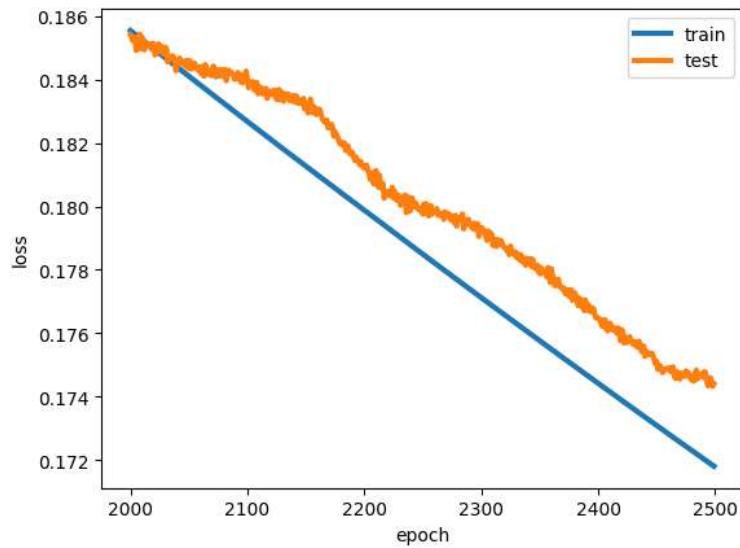


Gambar 4.65 Grafik *Loss* Model SGD Lr 0.0001 Pada Epoch 1500-2000

Pada data *train* nilai *loss* yang dihasilkan masih mengalami penurunan pada setiap iterasinya sedangkan pada data *test* nilai *loss* sempat mengalami peningkatan namun masih secara keseluruhan masih nilai *loss* yang diberikan pada data *test* memiliki pola menurun. Nilai MSE yang dihasilkan pada epoch ke-2000 yakni 0.18526, dengan nilai MSE tersebut model masih memberikan prediksi pada *grid* yang seharusnya tidak memiliki nilai VARI. Gambar 4.66 dan gambar 4.67 merupakan hasil yang diberikan model pada epoch ke-2500.

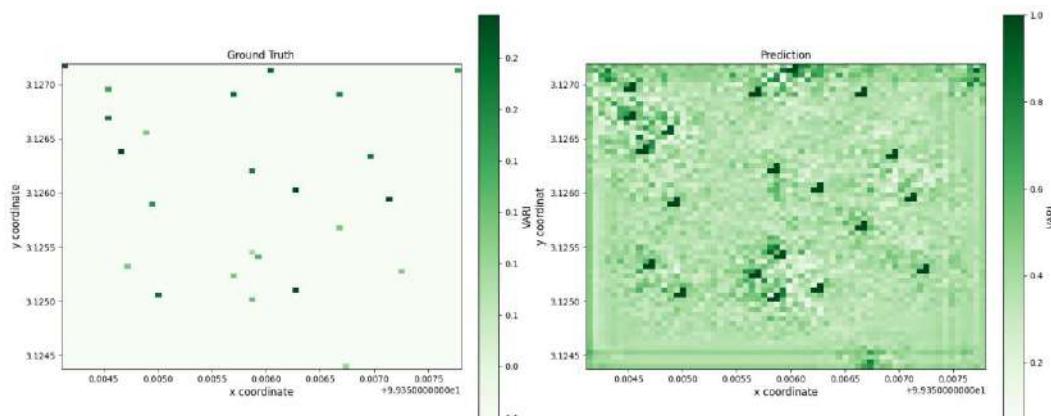


Gambar 4.66 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.0001 Pada Epoch 2500

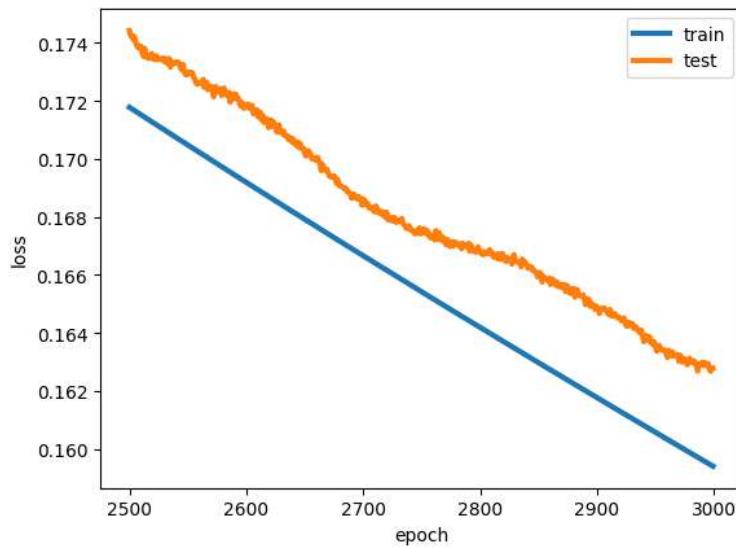


Gambar 4.67 Grafik *Loss* Model SGD Lr 0.0001 Pada Epoch 2000-2500

Sampai dengan epoch ke-2500 nilai *loss* yang diberikan pada model dengan optimasi SGD dan *learning rate* 0.0001 ini memiliki penurunan nilai *loss* pada setiap iterasinya. Nilai MSE yang dihasilkan model pada epoch ke-2500 yakni sebesar 0.17441, nilai MSE yang diberikan model pada epoch ke-2500 ini mengalami penurunan nilai dibanding pada epoch-epoch sebelumnya, namun dengan nilai MSE yang lebih rendah model masih memprediksi nilai pada *grid* yang pada kondisi sebenarnya tidak memiliki nilai VARI. Gambar 4.68 dan gambar 4.69 merupakan hasil yang diberikan model pada epoch ke-3000.



Gambar 4.68 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.0001 Pada Epoch 3000



Gambar 4.69 Grafik *Loss* Model SGD Lr 0.0001 Pada Epoch 2500-3000

Sampai dengan epoch ke-3000 nilai *loss* yang dihasilkan pada model ini masih mengalami penurunan nilai baik pada data *test* maupun pada data *train*, penurunan nilai *loss* ini merupakan indikasi yang baik bahwa model melakukan pembelajaran dengan baik. Nilai MSE yang diberikan model pada epoch ke-3000 yakni sebesar 0.16279, Nilai MSE yang diberikan pada epoch ke-3000 tersebut merupakan nilai terkecil yang dihasilkan model jika dibandingkan dengan penggunaan epoch yang lebih rendah. Rincian perbandingan nilai MSE yang dihasilkan pada setiap 500 epoch dapat dilihat pada tabel 4.11.

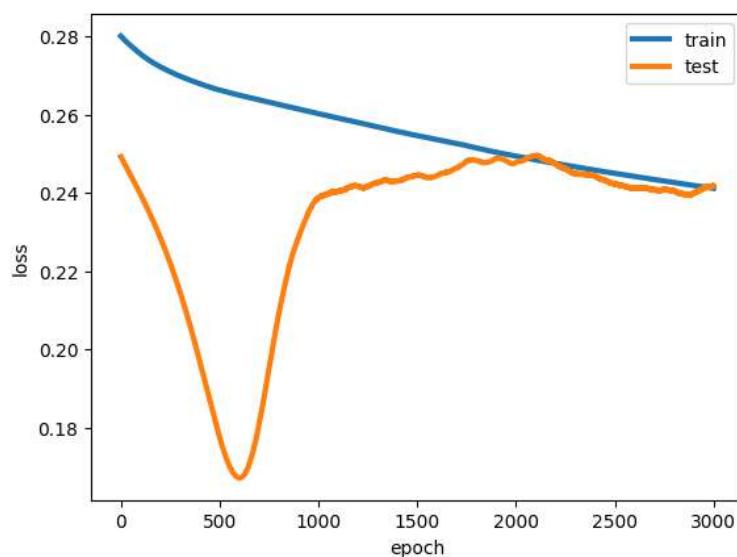
Tabel 4.11 Rincian Nilai MSE Pada Data *Test* Dimodel SGD Lr 0.0001 Di setiap 500 Epoch

	Epoch					
	500	1000	1500	2000	2500	3000
MSE	0.30884	0.22991	0.19893	0.18526	0.17441	0.16279

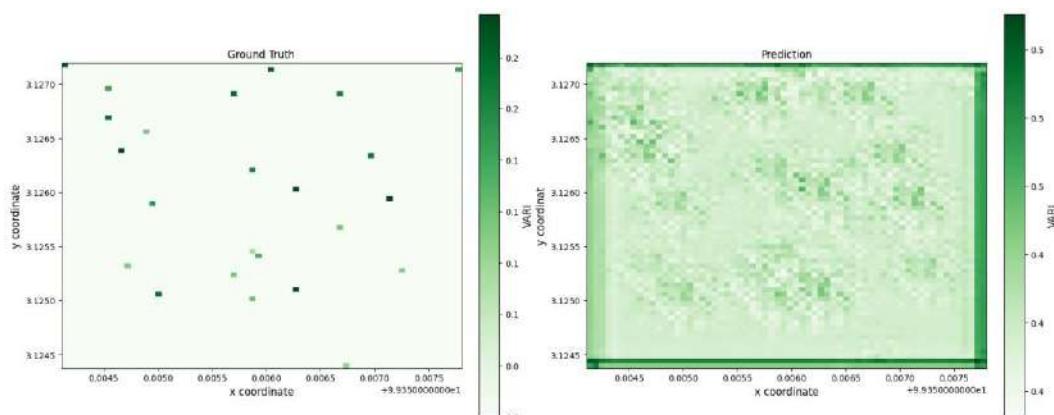
Dari hasil yang diberikan pada tabel 4.11, dapat disimpulkan bahwa semakin banyak epoch yang digunakan akan mempengaruhi nilai MSE menjadi semakin kecil, hal ini merupakan indikasi yang baik karena MSE merupakan metrik akurasi yang digunakan sebagai indikator bahwa model bekerja dengan baik atau tidak pada eksperimen model ini.

4.3.6 Model CNN LSTM Optimizer SGD Dengan Learning Rate 0.00001

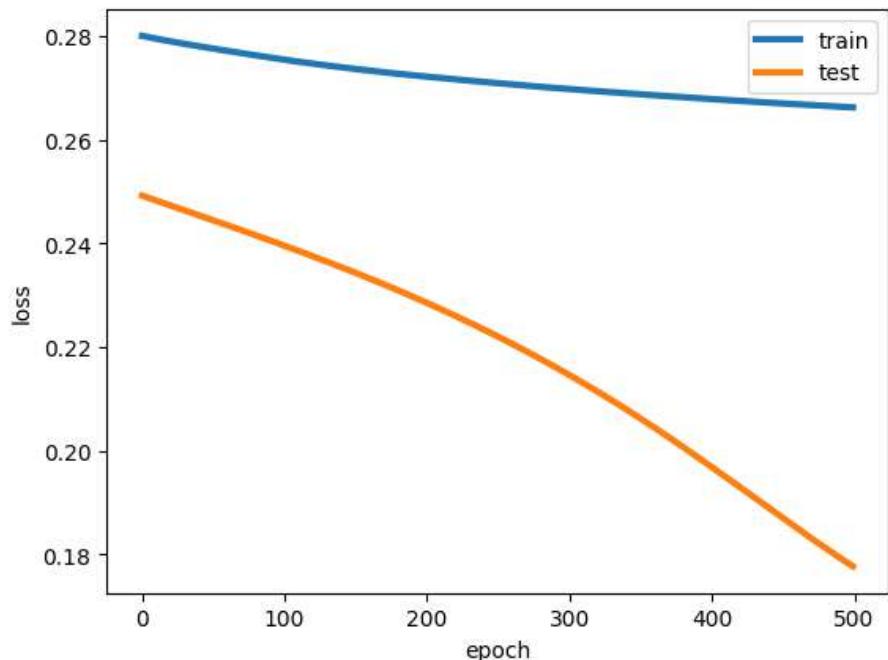
Model ini menggunakan optimasi SGD dengan parameter *learning rate* sebesar 0.00001, hasil yang dibahas pada sub-sub bab ini dengan melihat hasil prediksi dan grafik *loss* model pada setiap kelipatan 500 epoch sampai dengan epoch ke-3000. Gambar 4.70 merupakan grafik *loss* yang dihasilkan model ini, terdapat keunikan dimana nilai *loss* yang dihasilkan data *test* lebih rendah dibandingkan nilai *loss* yang ada pada data *train*. Gambar 4.71 dan gambar 4.72 merupakan hasil dari model pada epoch ke-500.



Gambar 4. 70 Grafik *Loss* Model SGD Lr 0.00001

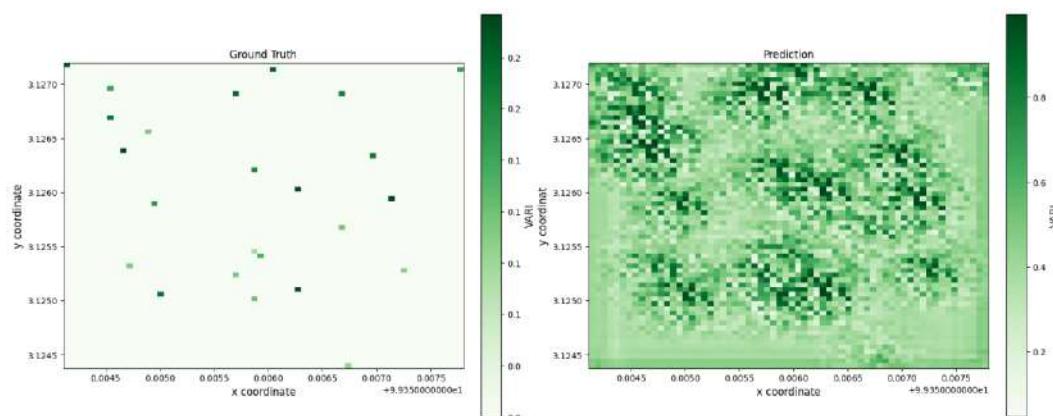


Gambar 4.71 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.00001 Pada Epoch 500

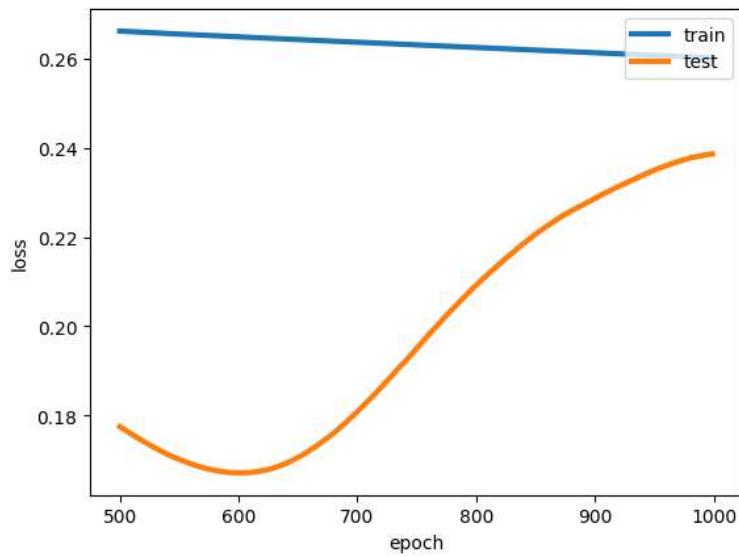


Gambar 4.72 Grafik *Loss* Model SGD Lr 0.00001 Pada Epoch 500

Hasil yang diperoleh pada model ini jika dilihat pada gambar 4.72 nilai *loss* yang dihasilkan pada data *test* lebih rendah dibandingkan pada data *train* dan jika dilihat pada hasil prediksi pada gambar 4.71 model hanya dapat memberikan pola samar yang dimiliki pada *ground truth*. Nilai MSE yang dihasilkan model pada epoch ke-500 ini sebesar 0.17769. Gambar 4.73 dan gambar 4.74 merupakan hasil dari model pada epoch ke-1000.

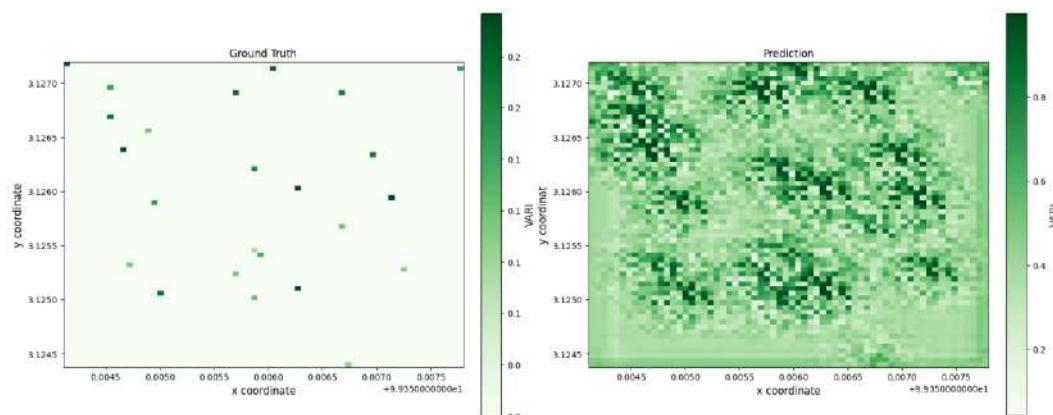


Gambar 4.73 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.00001 Pada Epoch 1000

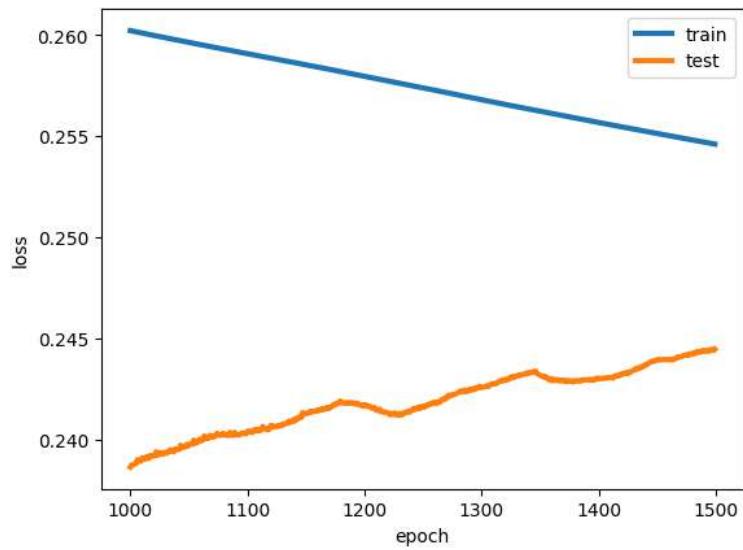


Gambar 4.74 Grafik *Loss* Model SGD Lr 0.00001 Pada Epoch 500-1000

Hasil yang diperoleh pada epoch ke-1000 ini dapat dilihat bahwa model memberikan prediksi pola unik namun masih belum dapat merepresentasikan prediksi yang mirip dibandingkan dengan pola yang dimiliki *ground truth*, dimana *grid* yang seharusnya tidak memiliki nilai masih banyak diprediksi memiliki nilai. Dilihat dari grafik *loss* yang dihasilkan pada model diePOCH ke-1000 terdapat peningkatan nilai *loss* pada data *test*. Nilai MSE yang diperoleh pada epoch ke-1000 ini juga meningkat dibanding nilai MSE yang diperoleh pada epoch ke-500, nilai MSE yang dihasilkan yakni sebesar 0.23865. Gambar 4.75 dan gambar 4.76 merupakan hasil dari model pada epoch ke-1500.

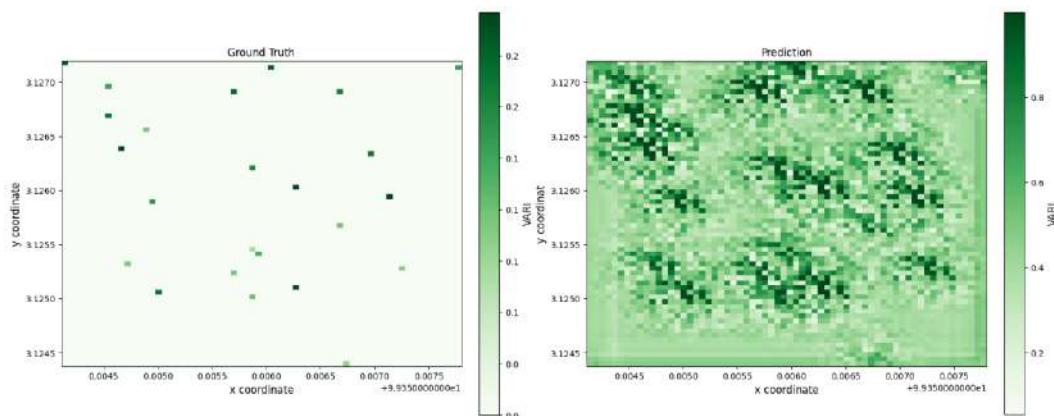


Gambar 4.75 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.00001 Pada Epoch 1500

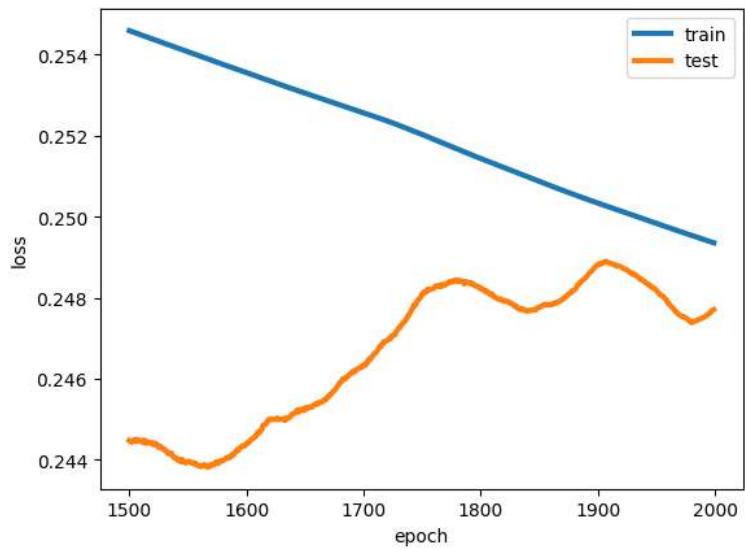


Gambar 4.76 Grafik *Loss* Model SGD Lr 0.00001 Pada Epoch 1000-1500

Nilai *loss* yang dihasilkan pada model terjadi peningkatan nilai *loss* pada data *test* sedangkan pada data *train* nilai *loss* yang dihasilkan masih mengalami penurunan nilai *loss* pada setiap iterasinya. Nilai MSE yang diberikan model pada epoch ke-1500 yakni sebesar 0.24447, nilai MSE yang dihasilkan pada epoch ke-1500 ini mengalami peningkatan nilai dibanding nilai MSE yang dihasilkan pada epoch sebelumnya, peningkatan nilai MSE ini merupakan salah satu indikasi bahwa model tidak belajar dengan baik. Gambar 4.77 dan gambar 4.78 merupakan hasil yang diberikan model pada epoch ke-2000.

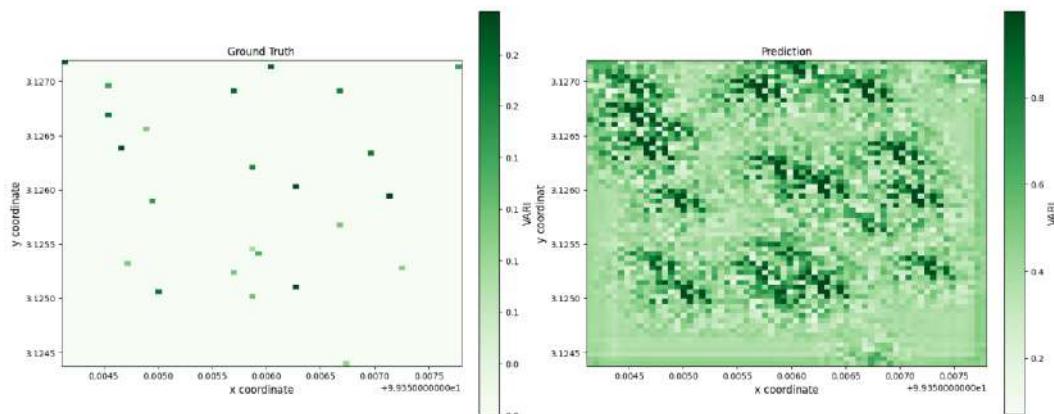


Gambar 4.77 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.00001 Pada Epoch 2000

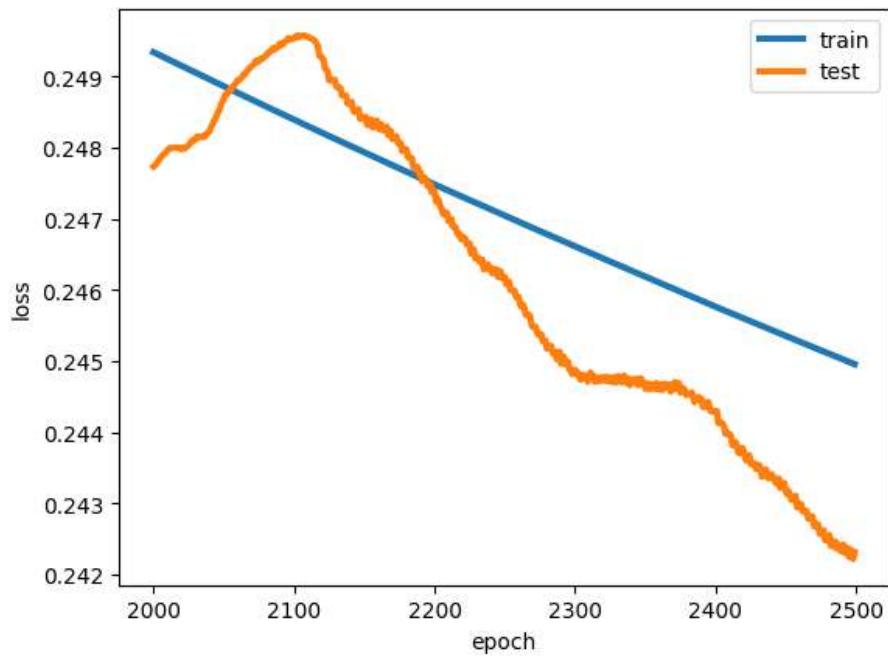


Gambar 4.78 Grafik *Loss* Model SGD Lr 0.00001 Pada Epoch 1500-2000

Dari Grafik *loss* yang diperlihatkan pada epoch ke-2000 ini, terjadi fluktuasi nilai *loss* yang ada pada data *test* dengan kecenderungan peningkatan nilai *loss* pada setiap iterasinya, sedangkan pada data *train* nilai *loss* yang dihasilkan pada setiap iterasinya mengalami penurunan nilai. Nilai MSE yang dihasilkan pada epoch ke-2000 yakni sebesar 0.24771, nilai MSE tersebut mengalami peningkatan nilai jika dibandingkan pada epoch sebelumnya. Peningkatan nilai MSE yang terjadi pada epoch ke-2000 ini merupakan salah satu indikasi bahwa model tidak belajar dengan baik. Gambar 4.79 dan gambar 4.80 merupakan hasil yang diberikan model pada epoch ke-2500.

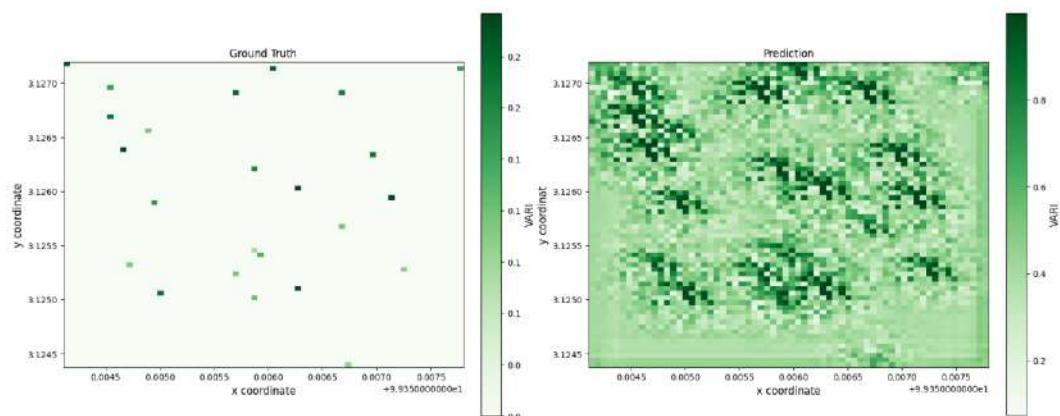


Gambar 4.79 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.00001 Pada Epoch 2500

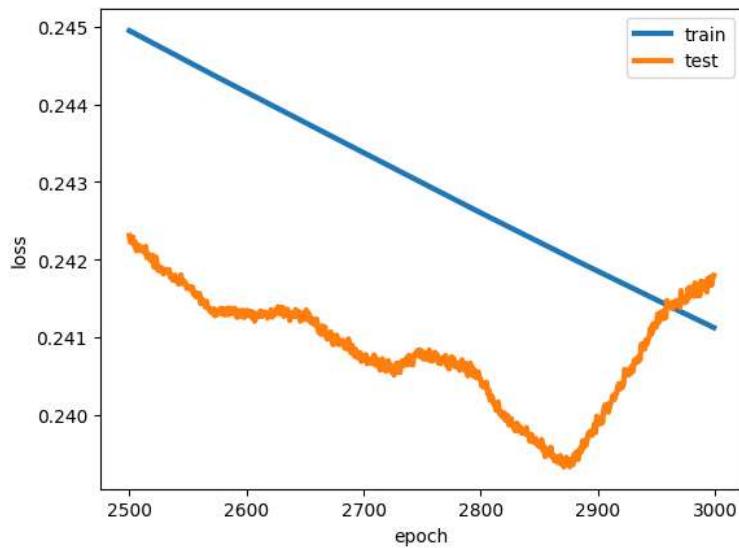


Gambar 4.80 Grafik *Loss* Model SGD Lr 0.00001 Pada Epoch 2000-2500

Hasil yang diperlihatkan grafik *loss* pada gambar 4.60 terjadi penurunan nilai *loss* yang ada pada data *test* pada epoch 2600, sedangkan pada data *train* nilai *loss* yang dihasilkan masih mengalami penurunan pada setiap iterasinya. Nilai MSE yang diberikan model pada epoch 2500 yakni sebesar 0.24231, nilai tersebut menurun dibandingkan pada epoch sebelumnya namun masih tetap lebih tinggi dibandingkan nilai MSE yang diberikan model pada epoch 500. Gambar 4.81 dan gambar 4.82 merupakan hasil yang diberikan model pada epoch ke-3000.



Gambar 4.81 *Heatmap* Perbandingan Hasil Prediksi dan *Ground Truth* Model SGD Lr 0.00001 Pada Epoch 3000



Gambar 4.82 Grafik *Loss* Model SGD Lr 0.00001 Pada Epoch 2500-3000

Sampai dengan epoch 3000 nilai *loss* yang dihasilkan pada data *train* terus mengalami penurunan berbanding terbalik dengan yang terjadi pada data *test*, nilai *loss* yang dihasilkan mengalami peningkatan pada epoch 3000. Nilai MSE yang dihasilkan model pada epoch ke-3000 yakni sebesar 0.24179, terdapat sedikit penurunan nilai MSE yang dihasilkan pada epoch ke-3000 jika dibandingkan pada epoch ke-2500 namun nilai MSE pada epoch ke-3000 masih lebih tinggi dibandingkan nilai MSE yang dihasilkan model pada epoch ke-500. Rincian perbandingan nilai MSE pada setiap 500 epoch dapat dilihat pada tabel 4.12.

Tabel 4.12 Rincian Nilai MSE Pada Data *Test* Dimodel SGD Lr 0.00001 Di setiap 500 Epoch

	Epoch					
	500	1000	1500	2000	2500	3000
MSE	0.17769	0.23865	0.24447	0.24771	0.24231	0.24179

Dari hasil nilai MSE yang diberikan pada setiap 500 epoch didapatkan kesimpulan bahwa pada model dengan optimasi SGD dan *learning rate* 0.00001 jumlah epoch tidak memiliki pengaruh terhadap penurunan nilai MSE yang dihasilkan oleh model. Dapat dilihat pada epoch ke-3000 nilai MSE yang dihasilkan memiliki nilai yang lebih tinggi dibandingkan nilai MSE yang ada pada epoch 500.

4.4 Ringkasan Hasil Eksperimen

Pada sub-bab ini akan dipaparkan waktu *train* yang diperlukan pada setiap model dan juga akan diperlihatkan dan disimpulkan model CNN LSTM terbaik yang digunakan pada penelitian ini berdasarkan nilai MSE terbaik yang diperoleh pada setiap kondisi. Untuk lebih jelasnya, dapat dilihat pada sub-bab 4.4.1 dan 4.4.2.

4.4.1 Ringkasan Waktu Proses *Training Model*

Dari hasil yang diberikan pada tabel perbandingan waktu *training* yang dilakukan setiap model waktu yang diperlukan untuk melakukan *training* setiap 500 epoch tidak berbeda jauh, dengan rata-rata jumlah waktu penggerjaan pada model dengan optimasi Adam sebesar 41 menit dan pada model dengan optimasi SGD sebesar 42 menit, terdapat perbedaan rata-rata 1 menit antara model dengan optimasi Adam dan SGD hal ini mungkin terjadi menginat proses pelatihan disetiap model menggunakan CPU sebagai *resource*. Rincian waktu yang dibutuhkan setiap model dalam melakukan pelatihan pada setiap 500 epoch dapat dilihat pada tabel 4.13.

Tabel 4.13 Rincian Waktu Proses *Training Model* Setiap 500 Epoch

Model		Epoch					
		500	1000	1500	2000	2500	3000
Adam		40m9s	82m39s	126m12s	167m36s	203m43s	244m05s
LR001		41m33s	80m12s	120m35s	161m29s	202m01s	242m20s
LR0001		40m22s	82m7s	144m21s	184m34s	223m23s	264m07s
Model		Epoch					
SGD		500	1000	1500	2000	2500	3000
LR001		39m21s	77m17s	116m41s	155m37s	210m33s	261m47s
LR0001		47m5s	91m26s	134m33s	179m9s	223m7s	267m57s
LR00001		41m32s	81m54s	123m55s	161m17s	201m4s	241m15s

Tabel 4.14 Perbandingan Selisih Waktu Pelatihan dan Nilai MSE Pada Model Adam Lr 0.001

Epoch	Waktu	Selisih Waktu	MSE	Selisih MSE
500	40,15m	0	0.87887	0
1000	82,65m	42,50m	$7,84 \cdot 10^{-5}$	0.87879
1500	126,20m	43,55m	$3,72 \cdot 10^{-5}$	$4,12 \cdot 10^{-5}$
2000	167,60m	41,40m	$2,78 \cdot 10^{-5}$	$0,94 \cdot 10^{-5}$
2500	203,72m	36,12m	$2,56 \cdot 10^{-5}$	$0,22 \cdot 10^{-5}$
3000	244,08m	40,37m	$2,95 \cdot 10^{-5}$	$-0,39 \cdot 10^{-5}$

Dari hasil yang diberikan pada tabel 4.14 yang merupakan perbandingan selisih waktu pelatihan dan selisih penurunan nilai MSE yang dihasilkan pada model Adam Lr 0.001 yang merupakan model terbaik yang dinyatakan pada penelitian tugas akhir ini. Didapat hasil bahwa dengan waktu 203,72 menit sudah mendapatkan nilai MSE terendah dibandingkan dengan waktu pelatihan model selama 244,08 menit untuk mencapai epoch ke-3000 namun nilai MSE yang dihasilkan pada epoch tersebut menjadi lebih besar dibandingkan nilai MSE pada epoch ke-2500 hanya dengan waktu pelatihan selama 203,72 menit.

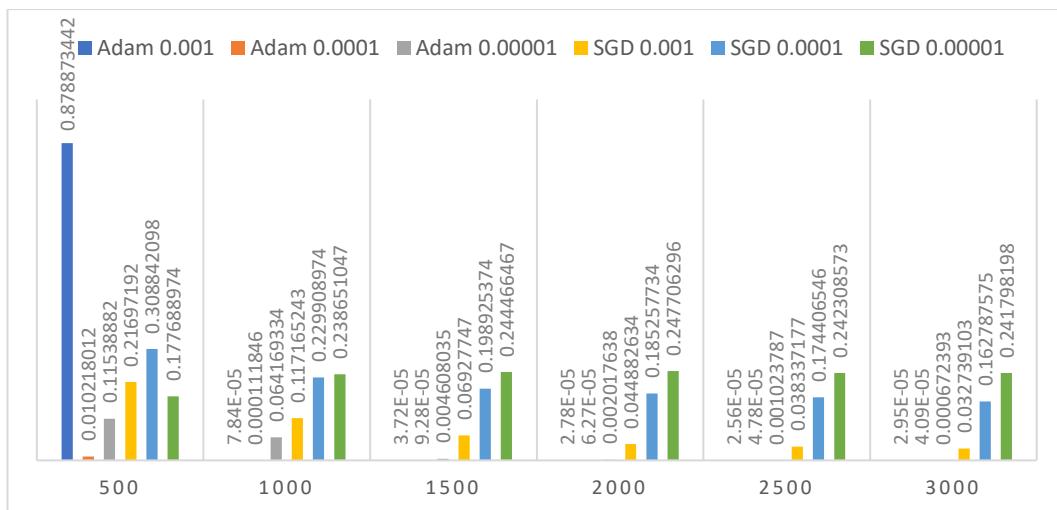
4.4.2 Model CNN LSTM

Model CNN LSTM terbaik yang didapat berdasarkan hasil penelitian yang telah dilakukan ini adalah model CNN LSTM dengan menggunakan optimasi Adam dan *learning rate* sebesar 0.001 pada epoch ke-2500, model tersebut disimpulkan sebagai model terbaik karena nilai MSE yang dihasilkan pada model tersebut memiliki nilai terendah dibandingkan seluruh model yang dilakukan pada penelitian ini. Nilai MSE yang diperoleh dari model terbaik ini yakni sebesar $2,56 \cdot 10^{-5}$. Rincian perbandingan nilai MSE yang diperoleh seluruh model dapat dilihat pada tabel 4.14 dan pada gambar 4.83 merupakan visualisasi grafik batang yang dihasilkan sesuai dengan tabel 4.14. Dari gambar 4.83 juga dapat diambil kesimpulan bahwa model dengan menggunakan metode optimasi Adam menghasilkan nilai MSE yang lebih rendah dibandingkan dengan model dengan menggunakan metode optimasi SGD.

Tabel 4.15 Rincian Nilai MSE Seluruh Model Pada Kelipatan 500 Epoch

Model		Epoch					
Adam		500	1000	1500	2000	2500	3000
0.001	0.87887	$7.84 \cdot 10^{-5}$	$3.72 \cdot 10^{-5}$	$2.78 \cdot 10^{-5}$	$2.56 \cdot 10^{-5}$	$2.95 \cdot 10^{-5}$	
0.0001	0.01022	0.00011	$9.28 \cdot 10^{-5}$	$6.27 \cdot 10^{-5}$	$4.78 \cdot 10^{-5}$	$4.09 \cdot 10^{-5}$	
0.00001	0.11539	0.06417	0.00461	0.00202	0.00102	0.00067	

Model		Epoch					
SGD		500	1000	1500	2000	2500	3000
0.001	0.21697	0.11717	0.06928	0.04488	0.03834	0.03274	
0.0001	0.30884	0.22991	0.19893	0.18526	0.17441	0.16279	
0.00001	0.17769	0.23865	0.24447	0.24771	0.24231	0.2418	



Gambar 4.83 Grafik Batang Perbandingan Nilai MSE Hasil Eksperimen

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Hasil yang dapat disimpulkan dari eksperimen yang dilakukan pada model CNN LSTM dalam memprediksi nilai VARI pada titik pohon kelapa sawit berjalan dengan baik dengan melihat warna yang dihasilkan pada setiap *grid*, semakin pekat warna yang dihasilkan pada *grid* menandakan bahwa pohon tersebut sehat, dan sebaliknya jika warna yang dihasilkan *grid* semakin memudar maka pohon tersebut kemungkinan besar sakit atau terserang *ganoderma*.

Hasil prediksi model prediksi model berjalan dengan baik dimana terdapat 6 tuning yang digunakan yakni model dengan menggunakan metode optimasi Adam dan SGD pada 3 parameter *learning rate* di 0.001, 0.0001, dan 0.00001 dan dianalisis disetiap 500 epoch sampai dengan 3000 epoch didapat kesimpulan bahwa model dengan menggunakan metode optimasi Adam pada *learning rate* 0.001 di epoch 2500 menghasilkan nilai MSE sebesar $2,56 \cdot 10^{-5}$, nilai MSE yang dihasilkan model tersebut pada epoch 2500 merupakan nilai MSE terkecil yang dihasilkan jika dibandingkan seluruh model yang lakukan pada penelitian ini.

Dari hasil eksperimen yang dilakukan didapat hasil bahwa metode optimasi Adam memiliki kinerja lebih baik dibandingkan dengan model yang menggunakan metode optimasi SGD. Parameter *learning rate* juga memiliki pengaruh dalam menghasilkan nilai MSE yang lebih baik dimana pada eksperimen ini disimpulkan bahwa *learning rate* dengan ukuran 0.001 bekerja dengan lebih baik dalam menurunkan nilai MSE yang diberikan mode. Selain kedua hasil tersebut didapatkan juga kesimpulan bahwa jumlah epoch memiliki pengaruh yang cukup besar dalam menurunkan nilai MSE yang dihasilkan model secara keseluruhan namun dalam kasus model pada metode optimasi Adam dan *learning rate* 0.001 didapat hasil terbaik yakni pada epoch ke-2500 dan pada model dengan metode optimasi SGD dan *learning rate* 0.00001 didapat hasil terbaik yakni pada epoch ke-500.

5.2 Saran

Berdasarkan hasil eksperimen yang dihasilkan, saran yang dapat peneliti berikan antara lain:

1. menambah urutan waktu pada *dataset*, agar model dapat bekerja dengan lebih baik untuk dapat memprediksi nilai VARI pada bulan selanjutnya, selain menambah urutan waktu pada *dataset* diperlukan juga penambah titik pohon kelapa sawit baik pada kondisi sakit maupun kondisi sehat sehingga model dapat memprediksi setiap *grid* dengan lebih akurat.
2. Dapat memperbaiki model sehingga *grid* yang digunakan untuk merepresentasikan keadaan nyata yang ada pada kebun kelapa sawit agar dapat melebihi kurang lebih 64x64 pohon kelapa sawit.
3. Menggunakan metode lain dalam merepresentasikan keadaan nyata kebun kelapa sawit selain menggunakan *grid-grid*.

DAFTAR PUSTAKA

- Abdina, M. F. (2019). Analisis Dampak Perkebunan Kelapa Sawit Terhadap Sosial Dan Ekonomi Masyarakat di Kabupaten Asahan. *Journal of Education, Humaniora and Social Sciences (JEHSS)*, 2(2). <https://doi.org/10.34007/jehss.v2i2.109>
- Baharim, M. S. A., Adnan, N. A., Mohd, F. A., Othman, A. N., Abdul Rahim, H., Azis, M. H. A., Seman, I. A., Izzuddin, M. A., Shahabuddin, N. A., & Nordiana, A. A. (2021). A review of ganoderma Basal Stem Rot (BSR) disease detection and mapping using geospatial and advanced techniques. In *Geocarto International*. <https://doi.org/10.1080/10106049.2021.1993350>
- Brownlee, J. (2019). *Gradient Descent For Machine Learning*. <https://machinelearningmastery.com/gradient-descent-for-machine-learning/>
- Corteva. (2020). *Penyakit Pada Tanaman Kelapa Sawit dan Cara Mencegahnya*. <https://www.corteva.id/berita/Penyakit-Pada-Tanaman-Kelapa-Sawit-dan-Cara-Mencegahnya.html>
- DataCamp. (2001). 【Pandas】 Python for Data Science. *Cheatsheet*, 71(10).
- Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. *ICLR Workshop*, 1.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12.
- Fachrizal, R. (2019). *Mengenal Beberapa Perpustakaan Machine Learning Berbasis Python*. <https://infokomputer.grid.id/read/121744301/mengenal-beberapa-perpustakaan-machine-learning-berbasis-python?page=all>
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2). <https://doi.org/10.1016/j.ejor.2017.11.054>
- GeoWombat. (2020). *GeoWombat: Utilities for geospatial data*. GeoWombat. <https://geowombat.readthedocs.io/en/latest/index.html>

- Gifari, J. (2020). *Belajar Machine Learning Dengan Library Python : Scikit-Learn*. <http://www.dqlab.id/belajar-machine-learning-dengan-library-python-scikit-learn>
- Girsang, S. A. (2019). *Long Short Term Memory (LSTM)*.
- Gitelson, A. A., Stark, R., Grits, U., Rundquist, D., Kaufman, Y., & Derry, D. (2002). Vegetation and soil lines in visible spectral space: A concept and technique for remote estimation of vegetation fraction. *International Journal of Remote Sensing*, 23(13). <https://doi.org/10.1080/01431160110107806>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning An MIT Press Book. In *Nature* (Vol. 29, Issue 7553).
- Hendarjati, H. (2014). *Ganoderma : “Momok Menakutkan” Planter Kelapa Sawit*. <https://sawitindonesia.com/ganoderma-momok-menakutkan-planter-kelapa-sawit/>
- Hinton, G., & Tieleman, T. (2012). RMSPROP: Divide the Gradient by a Running Average of its Recent Magnitude. *Coursera: Neural Networks for Machine Learning*, 4(2).
- Hopping, G. (2020, February 1). *Modeling the Risk of Traffic Accidents in New York City*. <Https://Towardsdatascience.Com/>.
- Hunter, J., Dale, D., & Droettboom, M. (2010). *matplotlib: python plotting — Matplotlib v1.0.0 documentation*. Matplotlib Website.
- Hushiaran, R., Yusof, N. A., & Dutse, S. W. (2013). Detection and control of Ganoderma boninense: Strategies and perspectives. In *SpringerPlus* (Vol. 2, Issue 1). <https://doi.org/10.1186/2193-1801-2-555>
- Julpan, Nababan, E. B., & Zarlis, M. (2015). Analisis Fungsi Aktivasi Sigmoid Biner Dan Sigmoid Bipolar Dalam Algoritma Backpropagation Pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*, 02(1).
- Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
- KOMINFO. (2019). *BAHASA PEMROGRAMAN PYTHON : PENGERTIAN, SEJARAH, KELEBIHAN DAN KEKURANGANNYA*.

- https://www.baktikominfo.id/id/informasi/pengetahuan/bahasa_pemrograman_python_pengertian_sejarah_kelebihan_dan_kekurangannya-954
- Lina, Q. (2019). *Apa itu Convolutional Neural Network?* <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>
- McKinney, W. (2011). pandas: a Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing*.
- Moyroud, N., & Portet, F. (2018). Introduction to QGIS. In *QGIS and Generic Tools* (Vol. 1, pp. 1–17). John Wiley & Sons, Inc. <https://doi.org/10.1002/9781119457091.ch1>
- NumPy. (2008). *What is NumPy?* NumPy. <https://numpy.org/doc/stable/user/whatisnumpy.html>
- Nur Anisa, M., Rokhmatuloh, & Hernina, R. (2020). UAV application to estimate oil palm trees health using Visible Atmospherically Resistant Index (VARI) (Case study of Cikabayan Research Farm, Bogor City). *E3S Web of Conferences*, 211. <https://doi.org/10.1051/e3sconf/202021105001>
- Prasetyo, H. (2019). *Backpropagation Neural Network (Jaringan Saraf Tiruan)*. <https://hendroprasetyo.com/backpropagation-neural-network-jaringan-saraf-tiruan/#.ZCUqm8pBxEa>
- Putralim, V. A. (2023). PROFILING REFLEKTANSI NILAI RGB DAN VARI POHON SAWIT YANG TERKENA GANODERMA DARI WAKTU KE WAKTU PADA CITRA UAV. *Machung.Ac.Id.*
- Setiawan, R. (2021). *Mengenal Deep Learning Lebih Jelas*. <https://www.dicoding.com/blog/mengenal-deep-learning/>
- Shahroudy, A., Liu, J., Ng, T. T., & Wang, G. (2016). NTU RGB+D: A large scale dataset for 3D human activity analysis. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*. <https://doi.org/10.1109/CVPR.2016.115>
- Srinivasan, V. A. (2019). *Stochastic Gradient Descent — Clearly Explained !!* <https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31>

- Sun, J., Di, L., Sun, Z., Shen, Y., & Lai, Z. (2019). County-level soybean yield prediction using deep CNN-LSTM model. *Sensors (Switzerland)*, 19(20). <https://doi.org/10.3390/s19204363>
- Suwartono, A. (2018, April 5). *Dampak Mengenakan Kerusakan dan Akibat Ganoderma*. <https://sawitindonesia.com/dampak-mengerikan-kerusakan-dan-akibat-ganoderma/>
- Tee, C. A. T., Teoh, Y. X., Yee, P. L., Tan, B. C., & Lai, K. W. (2021). Discovering the Ganoderma Boninense Detection Methods Using Machine Learning: A Review of Manual, Laboratory, and Remote Approaches. *IEEE Access*, 9. <https://doi.org/10.1109/ACCESS.2021.3098307>
- Trivusi. (2022). *Mengenal Model Ruang Warna pada Pengolahan Citra*. Trivusi. <https://www.trivusi.web.id/2022/11/model-ruang-warna.html>
- Tugi, A., Rasib, A. W., Suri, M. A., Zainon, O., Mohd Yusoff, A. R., Abdul Rahman, M. Z., Sari, N. A., & Darwin, N. (2015). Oil palm tree growth monitoring for smallholders by using unmanned aerial vehicle. *Jurnal Teknologi*, 77(26). <https://doi.org/10.11113/jt.v77.6855>
- Wang, X. X., & Xu, L. H. (2018). Short-term Traffic Flow Prediction Based on Deep Learning. *Jiaotong Yunshu Xitong Gongcheng Yu Xinxi/Journal of Transportation Systems Engineering and Information Technology*, 18(1). <https://doi.org/10.16097/j.cnki.1009-6744.2018.01.012>
- Wibowo, J. T. P. (2021). *Apa Itu Google Colab?* <https://wartaekonomi.co.id/read377409/apa-itu-google-colab>
- Wicaksono, W. (2022). PENGEMBANGAN MODEL DEEP LEARNING UNTUK PREDIKSI SERANGAN GANODERMA BERDASARKAN CITRA UNMANNED AERIAL VEHICLE (UAV) PADA PERKEBUNAN KELAPA SAWIT. *Machung.Ac.Id.*
- Yudistina, V., Santoso, M., & Aini, N. (2017). HUBUNGAN ANTARA DIAMETER BATANG DENGAN UMUR TANAMAN TERHADAP PERTUMBUHAN DAN HASIL TANAMAN KELAPA SAWIT. *BUANA SAINS*, 17(1). <https://doi.org/10.33366/bs.v17i1.577>

Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control*, 47. <https://doi.org/10.1016/j.bspc.2018.08.035>