

**ANALISIS PERBANDINGAN ALGORITMA *TEMPORAL CONVOLUTIONAL NETWORK* (TCN) DAN *BIDIRECTIONAL LSTM* DALAM PREDIKSI *MULTI-LABEL* SERANGAN HAMA TEBU**

**(Skripsi)**

**Oleh**

**MUKTI PRABOWO  
NPM 2217051055**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

**ANALISIS PERBANDINGAN ALGORITMA *TEMPORAL CONVOLUTIONAL NETWORK* (TCN) DAN *BIDIRECTIONAL LSTM* DALAM PREDIKSI MULTI-LABEL SERANGAN HAMA TEBU**

Oleh

**MUKTI PRABOWO**

**Skripsi**

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

## ABSTRAK

### ANALISIS PERBANDINGAN ALGORITMA *TEMPORAL CONVOLUTIONAL NETWORK* (TCN) DAN *BIDIRECTIONAL LSTM* DALAM PREDIKSI *MULTI-LABEL* SERANGAN HAMA TEBU

Oleh

MUKTI PRABOWO

Industri tebu pada Perusahaan Gula A menghadapi ancaman serius dari serangan hama penggerek batang (*Chilo sacchariphagus* Bojer), penggerek pucuk (*Scirpophaga excerptalis* Walker), dan kutu perisai (*Aulacaspis tegalensis* Zehntner) yang seringkali terjadi secara simultan. Keterlambatan deteksi serangan dapat menyebabkan kerusakan yang tidak dapat dipulihkan, sehingga dibutuhkan sistem prediksi yang akurat. Penelitian ini membandingkan dua arsitektur *deep learning*, yaitu *Temporal Convolutional Network* (TCN) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM), dalam prediksi multi-label tingkat keparahan serangan hama tebu. *Dataset* yang digunakan merupakan data pengamatan hama tebu Perusahaan Gula A tahun 2022-2023. Tahapan penelitian meliputi *preprocessing*, *feature engineering*, *hyperparameter tuning* dengan *random search*, pemodelan independen per jenis hama, serta evaluasi menggunakan metrik Akurasi, *F1-Macro*, dan ROC-AUC. Hasil penelitian menunjukkan bahwa TCN secara konsisten mengungguli Bi-LSTM pada hampir seluruh skenario pengujian dengan nilai *F1-Macro* dan ROC-AUC yang lebih tinggi. TCN juga terbukti jauh lebih efisien dari segi waktu komputasi. Berdasarkan hasil tersebut, TCN direkomendasikan sebagai model utama untuk sistem prediksi hama tebu, sementara Bi-LSTM dapat difungsikan sebagai model alternatif yang kompetitif.

Kata kunci: *Temporal Convolutional Network*, *Bidirectional LSTM*, Prediksi Hama Tebu, *Klasifikasi Multi-Label*.

## **ABSTRACT**

### **COMPARATIVE ANALYSIS OF TEMPORAL CONVOLUTIONAL NETWORK (TCN) AND BIDIRECTIONAL LSTM ALGORITHMS IN MULTI-LABEL PREDICTION OF SUGARCANE PEST ATTACKS**

**By**

**MUKTI PRABOWO**

*The sugarcane industry at Sugar Company A faces serious threats from attacks by stem borers (*Chilo sacchariphagus* Bojer), top borers (*Scirpophaga excerptalis* Walker), and scale insects (*Aulacaspis tegalensis* Zehntner), which often occur simultaneously. Delayed detection of attacks can cause irreparable damage, requiring an accurate prediction system. This study compares two deep learning architectures, namely Temporal Convolutional Network (TCN) and Bidirectional Long Short-Term Memory (Bi-LSTM), in predicting the multi-label severity of sugarcane pest attacks. The dataset used is data on sugarcane pest observations at Sugar Company A in 2022-2023. The research stages included preprocessing, feature engineering, hyperparameter tuning with random search, independent modeling per pest type, and evaluation using Accuracy, F1-Macro, and ROC-AUC metrics. The results showed that TCN consistently outperformed Bi-LSTM in almost all testing scenarios with higher F1-Macro and ROC-AUC values. TCN also proved to be much more efficient in terms of computation time. Based on these results, TCN is recommended as the primary model for sugarcane pest prediction systems, while Bi-LSTM can be used as a competitive alternative model.*

*Keywords: Temporal Convolutional Network, Bidirectional LSTM, Sugarcane Pest Prediction, Multi-Label Classification.*

Judul Skripsi : **ANALISIS PERBANDINGAN ALGORITMA  
TEMPORAL CONVOLUTIONAL NETWORK (TCN)  
DAN BIDIRECTIONAL LSTM DALAM PREDIKSI  
MULTI-LABEL SERANGAN HAMA TEBU**

Nama Mahasiswa : **Mukti Prabowo**

NPM : 2217051055

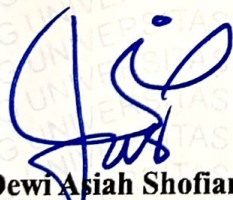
Program Studi : S1 Ilmu Komputer

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing



**Dewi Asiah Shofiana, S.Komp., M.Kom.**  
NIP. 199509292020122030

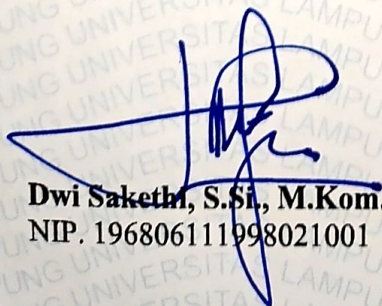


**Ridho Sholehurrohman, M.Mat.**  
NIK. 232111970128101

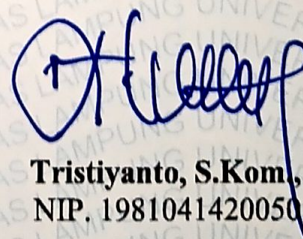
2. Mengetahui

Ketua Jurusan Ilmu Komputer

Ketua Program Studi Ilmu Komputer



**Dwi Sakethi, S.Si., M.Kom.**  
NIP. 196806111998021001



**Tristiyanto, S.Kom., M.I.S., Ph.D.**  
NIP. 198104142005011001

## MENGESAHKAN

### 1. Tim Penguji

Ketua Penguji : Dewi Asiah Shofiana, S.Komp., M.Kom.

Sekretaris Penguji : Ridho Sholehurrohman, M.Mat.

Penguji Utama : Tristiyanto, S.Kom., M.I.S., Ph.D.

### 2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



*Heri Satria*  
Dr. Eng. Heri Satria, S.Si., M.Si.  
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 9 Maret 2026

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Mukti Prabowo

NPM : 2217051055

Dengan ini menyatakan bahwa skripsi saya yang berjudul "**Analisis Perbandingan Algoritma *Temporal Convolutional Network* (TCN) dan *Bidirectional LSTM* dalam Prediksi *Multi-Label* Serangan Hama Tebu**" merupakan karya saya sendiri, bukan karya orang lain. Semua tulisan yang tertulis dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa karya tulis ilmiah saya terbukti hasil menjiplak karya orang lain, maka saya siap menerima sanksi berupa pencabutan gelar yang saya peroleh.

Bandar Lampung, 9 April 2026



**Mukti Prabowo**  
NPM. 2217051055

## RIWAYAT HIDUP



Penulis lahir di Mataram Udik Lampung Tengah pada tanggal 16 Maret 2004 sebagai anak ketiga dari pasangan Bapak Parsudi dan Ibu Suwanti. Penulis menempuh pendidikan formal Sekolah Dasar (SD) di SDS 01 Gula Putih Mataram dan lulus pada tahun 2016. Kemudian, pendidikan Sekolah Menengah Pertama (SMP) SMPS Sugar Group hingga Tahun 2019, lalu melanjutkan pendidikan Sekolah Menengah Atas (SMA) di SMAS Sugar Group yang diselesaikan pada Tahun 2022. Pada tahun 2022 penulis terdaftar menjadi mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Selama menjalani masa perkuliahan penulis melakukan berbagai kegiatan akademik maupun nonakademik sebagai berikut.

1. Menjadi Anggota Bidang Media Informasi Himpunan Mahasiswa Jurusan Ilmu Komputer FMIPA Unila pada Periode 2023/2024.
2. Menjadi Ketua Bidang Media Informasi Himpunan Mahasiswa Jurusan Ilmu Komputer FMIPA Unila pada Periode 2024/2025.
3. Menjadi pembicara pada kegiatan Pelatihan Desain dan Video oleh Media Informasi Himpunan Mahasiswa Jurusan Ilmu Komputer FMIPA Unila.
4. Menjadi Kepala Departemen 2 *Digital Creative* pada acara CSS 2024.
5. Menjadi Sekretaris Koordinator Divisi Desain dan Multimedia di Pekan Raya Jurusan x Himakom Tournament 2023.
6. Menjadi Asisten Dosen mata kuliah Matematika D3 di Jurusan Ilmu Komputer pada Tahun 2023.

7. Menjadi Asisten Dosen mata kuliah Rekayasa Perangkat Lunak S1 di Jurusan Ilmu Komputer pada Tahun 2024.
8. Mengikuti Program Studi Independen yang diselenggarakan oleh Dicoding Indonesia pada jalur pembelajaran (*learning path*) *Front-End Web and Back-End Developer*.
9. Mengikuti Kuliah Kerja Nyata (KKN) di Kelurahan Surabaya, Kecamatan Kedaton, Kota Bandar Lampung.
10. Melaksanakan Kerja Praktik (KP) di Badan Pusat Statistik dengan fokus pada pengembangan fitur *booking* ruangan rapat pada aplikasi web PARABOLA.

## **MOTTO**

“Hidup Bukan untuk Saling Mendahului.”

- Hindia

“*Whatever happens, happens.*”

- Spike Spiegel, Cowboy Bebop

## **PERSEMBAHAN**

Puji syukur kehadirat Allah Subhanahu Wa Ta'ala atas segala limpahan rahmat dan karunia-Nya, sehingga skripsi ini berhasil diselesaikan dengan baik. Shalawat serta salam selalu tercurahkan kepada Nabi Muhamad Shallahu 'Alaihi Wasallam sebagai suri teladan bagi seluruh umat manusia.

### **Skripsi ini penulis persembahkan kepada:**

Kedua orang tua tercinta, atas semua doa, dukungan, kasih, serta pengorbanan yang tidak pernah berhenti. Terima kasih atas setiap usaha, bimbingan, dan kepercayaan yang menjadi landasan dalam setiap langkah penulis.

Selain itu, almamater tercinta Jurusan Ilmu Komputer FMIPA Universitas Lampung sebagai tempat belajar, bertumbuh dan memperoleh pengetahuan yang menjadi bekal untuk masa depan.

## SANWANCANA

Puji syukur kehadiran Allah Subhanahu Wa Ta'ala, yang telah memberikan nikmat kesehatan, rahmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis Perbandingan Algoritma *Temporal Convolutional Network* (TCN) Dan *Bidirectional LSTM* Dalam Prediksi *Multi-Label* Serangan Hama Tebu” dengan lancar. Terima kasih penulis ucapkan kepada semua pihak yang telah memberikan dukungan, bimbingan dan arahan dalam penyusunan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala rahmat, pertolongan, dan kemudahan yang diberikan, sehingga setiap proses dapat berjalan dengan baik dan lancar.
2. Kedua orang tua tercinta, Bapak Parsudi dan Ibu Suwanti yang selalu memberikan kasih sayang, doa dan dukungan, sehingga penulis dapat menyelesaikan perkuliahan dan penyusunan skripsi ini. Terima kasih selalu ada di saat penulis membutuhkan kepercayaan dan dukungan, serta senantiasa percaya pada setiap langkah penulis lakukan.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Program Studi S1 Ilmu Komputer FMIPA Universitas Lampung dan Pembahas yang telah memberikan masukan serta saran yang bermanfaat untuk perbaikan skripsi ini.
7. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom. selaku Pembimbing Akademik

Pembimbing Utama yang telah memberikan arahan, ide, kritik, serta saran dalam menyelesaikan penelitian ini.

8. Bapak Ridho Sholehurrohman, M. Mat. selaku Pembimbing Kedua yang selalu dapat meluangkan waktunya untuk membimbing, memberikan arahan, dan bantuan ketika mengalami kesulitan dalam menyelesaikan penelitian ini.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan banyak ilmu dan motivasi selama perkuliahan.
10. Seluruh Staf dan Karyawan Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah membantu perihal administrasi, sarana, dan prasarana.
11. Himpunan Mahasiswa Jurusan Ilmu Komputer Periode 2023 & 2024, atas pengalaman dan kebersamaan selama berorganisasi dan masa perkuliahan.

Penulis menyadari bahwa masih terdapat kekurangan dalam penyusunan skripsi ini. Namun, penulis berharap penelitian ini dapat memberikan manfaat dan menjadi referensi bagi para pembaca. Semoga penelitian ini dapat menjadi langkah awal untuk penelitian yang lebih baik di masa mendatang.

Bandar Lampung, 9 April 2026

Penulis,



Mukti Prabowo

2217051055

## DAFTAR ISI

<b>DAFTAR ISI</b> .....	<b>xiv</b>
<b>DAFTAR TABEL</b> .....	<b>xvii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xviii</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xx</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	4
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian .....	5
1.5. Manfaat Penelitian .....	6
<b>II. TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1. Penelitian Terdahulu .....	7
2.2. Landasan Teori.....	10
2.2.1. <i>Machine Learning</i> .....	10
2.2.2. <i>Deep Learning</i> .....	12
2.2.3. <i>Recurrent Neural Network</i> .....	13
2.2.4. <i>Long Short-Term Memory</i> .....	14
2.2.5. <i>Bidirectional Long Short-Term Memory</i> .....	16
2.2.6. <i>Convolutional Neural Network (CNN)</i> .....	17
2.2.7. <i>Temporal Convolutional Network (TCN)</i> .....	20
2.2.8. Python .....	22
2.2.8.1. Pandas .....	22
2.2.8.2. NumPy .....	23
2.2.8.3. Matplotlib.....	23

2.2.8.4.	Scikit-Learn.....	24
2.2.9.	Hama Tebu .....	24
2.2.9.1.	Penggerek Pucuk ( <i>Scirpophaga excerptalis</i> Walker) .....	25
2.2.9.2.	Penggerek Batang ( <i>Chilo sacchariphagus</i> Bojer).....	26
2.2.9.3.	Kutu Perisai ( <i>Aulacaspis tegalensis</i> Zehntner) .....	27
2.2.10.	<i>Confusion Matrix</i> .....	28
2.2.11.	ROC AUC .....	29
2.2.12.	<i>Hyperparameter Tuning</i> .....	30
2.2.13.	<i>Multi-Label Classification</i> .....	32
<b>III.</b>	<b>METODE PENELITIAN .....</b>	<b>33</b>
3.1.	Tempat dan Waktu Penelitian .....	33
3.1.1.	Tempat Penelitian.....	33
3.1.2.	Waktu Penelitian .....	33
3.2.	Tahapan Penelitian .....	34
3.2.1.	Pengumpulan <i>Dataset</i> .....	35
3.2.2.	<i>Exploratory Data Analysis</i> .....	37
3.2.3.	<i>Preprocessing</i> .....	38
3.2.4.	<i>Split Data</i> Berdasarkan Blok.....	40
3.2.5.	<i>Preprocessing</i> Lanjutan .....	41
3.2.6.	<i>Hyperparameter Tuning</i> .....	44
3.2.7.	<i>Modelling</i> .....	46
3.2.8.	Evaluasi Model.....	49
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>51</b>
4.1.	Eksplorasi dan <i>Preprocessing Data</i> .....	51
4.1.1.	Data Awal.....	51
4.1.2.	Identifikasi Duplikasi Data dan <i>Missing Values</i> .....	53
4.1.2.1.	Identifikasi Duplikasi Data .....	53
4.1.2.2.	Identifikasi <i>Missing Values</i> .....	55
4.1.3.	<i>Data Cleaning</i> .....	58
4.1.3.1.	<i>Rename Columns</i> .....	58
4.1.3.2.	<i>Drop</i> Kolom yang Tidak Diperlukan .....	59
4.1.3.3.	<i>Drop</i> Nilai Spesifik pada Kolom Varietas .....	59

4.1.3.4.	Penanganan Duplikasi Data .....	60
4.1.3.5.	Penanganan <i>Missing Values</i> .....	61
4.1.4.	<i>Labelling Data</i> .....	62
4.1.5.	Analisis Distribusi Fitur .....	63
4.1.5.1.	Analisis Distribusi Serangan dan Label Hama.....	63
4.1.5.2.	Distribusi Fitur Dosis .....	64
4.1.5.3.	Distribusi Kolom Kategorikal .....	65
4.1.6.	Analisis Korelasi Antarfitur .....	66
4.2.	<i>Data Splitting</i> Berdasarkan Blok .....	67
4.3.	<i>Feature Engineering</i> .....	69
4.3.1.	Pembuatan Fitur Temporal Dasar .....	69
4.3.2.	Pembuatan <i>Lag Features</i> .....	70
4.3.3.	Pembuatan <i>Rolling Statistics</i> .....	71
4.3.4.	Pembuatan <i>Cummulative Features</i> untuk Dosis .....	73
4.3.5.	Pembuatan Fitur Curah Hujan Kategorikal.....	74
4.4.	<i>Feature Selection</i> .....	75
4.5.	<i>Modelling</i> .....	76
4.6.	Hasil <i>Evaluasi</i> .....	77
4.6.1.	Hasil Evaluasi Bi-LSTM.....	78
4.6.2.	Hasil Evaluasi TCN.....	81
4.7.	Analisis dan Perbandingan Performa Model.....	84
4.7.1.	Perbandingan Performa Antar Algoritma .....	85
4.7.2.	Perbandingan Performa Antar Jenis Hama .....	86
4.7.3.	Perbandingan Waktu Komputasi dan Efisiensi Antar Model .....	88
4.7.4.	Perbandingan Arsitektur dan Konfigurasi Model .....	89
<b>V.</b>	<b>SIMPULAN DAN SARAN.....</b>	<b>94</b>
5.1.	Simpulan .....	94
5.2.	Saran.....	95
	<b>DAFTAR PUSTAKA .....</b>	<b>97</b>
	<b>LAMPIRAN.....</b>	<b>108</b>

## DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu .....	7
2. <i>Dataset Features</i> .....	35
3. Aturan Label Serangan Penggerek Batang dan Penggerek Pucuk.....	40
4. Aturan Label Serangan Kutu Perisai.....	40
5. <i>Hyperparameter Search Space</i> untuk Model Bi-LSTM.....	44
6. <i>Hyperparameter Search Space</i> untuk Model TCN.....	45
7. Konfigurasi <i>Hyperparameter Tuning</i> .....	45
8. Parameter Model TCN dan Bi-LSTM .....	47
9. Karakteristik <i>Dataset</i> Serangan Hama.....	52
10. Karakteristik <i>Dataset</i> Curah Hujan.....	53
11. <i>Rename Columns</i> .....	59
12. Pembagian Data Latih, Data Validasi, dan Data Uji.....	68
13. Fitur Temporal yang Dibuat.....	70
14. <i>Lag Features</i> yang Dibuat .....	71
15. <i>Rolling Statistics</i> yang Dibuat.....	72
16. <i>Cumulative Features</i> yang Dibuat .....	73
17. Fitur Curah Hujan Kategorikal .....	74
18. Tabel Kategori Fitur.....	75
19. Tabel Fitur Spesifik per Target Hama .....	76
20. Performa Model pada Masing-Masing Label .....	85
21. Performa Model pada Masing-Masing Kelas .....	86
22. Perbandingan Waktu Komputasi dan Efisiensi Antar Model .....	88
23. Perbandingan Arsitektur dan Konfigurasi Model .....	89

## DAFTAR GAMBAR

Gambar	Halaman
1. <i>Supervised, Unsupervised, dan Reinforcement Learning</i> .....	11
2. Perbedaan Proses <i>Machine Learning</i> dan <i>Deep Learning</i> .....	13
3. Arsitektur Standar RNN .....	14
4. Struktur <i>Long Short-Term Memory</i> .....	15
5. (a) Konfigurasi LSTM, (b) Konfigurasi Bi-LSTM .....	16
6. Arsitektur CNN .....	18
7. <i>Convolutional Layers</i> .....	19
8. <i>Pooling Layers</i> .....	19
9. <i>Fully Connected Layers</i> .....	20
10. <i>TCN Layers</i> .....	20
11. Arsitektur Utama TCN .....	21
12. Penggerek Pucuk .....	25
13. Penggerek Batang Tebu .....	26
14. Serangan Kutu Perisai .....	27
15. Tabel <i>Confusion Matrix</i> .....	28
16. Perbedaan <i>Grid Search</i> dan <i>Random Search</i> .....	31
17. <i>Flowchart</i> Metode Penelitian .....	34
18. Hasil Identifikasi Duplikasi Berdasarkan (a) Tanggal Pengamatan (b) Nama Blok dan Umur .....	54
19. Contoh Blok dengan Duplikat (a) Tanggal Pengamatan Sama (b) Tanggal Pengamatan Berbeda .....	54
20. <i>Missing Values</i> pada 'Kawasan' (a) 10 Baris Pertama (b) 10 Baris Terakhir. 55	
21. Blok dengan Kawasan Terdapat NaN .....	55
22. Identifikasi Pengamatan Blok .....	56

23. Contoh Blok dengan Data Pengamatan Lengkap .....	57
24. Contoh Blok dengan Pengamatan (a) Sampai 9 Bulan (b) Sampai 10 Bulan	57
25. Contoh Blok dengan Data Pengamatan Tidak Lengkap .....	58
26. Varietas Tebu dengan Nilai 'lain-lain' dan 'Lain-lain' .....	60
27. Contoh Blok yang Sudah Tidak Ada Data Duplikat.....	60
28. Blok (a) Sebelum Imputasi (b) Setelah Imputasi .....	61
29. Blok dengan Pengamatan (a) Sebelum Imputasi (b) Setelah Imputasi.....	62
30. Distribusi Serangan Hama .....	63
31. Distribusi Label Hama .....	64
32. Distribusi Dosis.....	65
33. Distribusi Fitur Kategorikal .....	66
34. Heatmap Korelasi antara Serangan Hama, Dosis, dan Umur .....	66
35. Grafik Distribusi Target.....	68
36. Konfigurasi Parameter Optimal untuk Bi-LSTM .....	77
37. Konfigurasi Parameter Optimal untuk TCN .....	77
38. Hasil <i>Confusion Matrix</i> Model <i>Bidirectional LSTM</i> pada <i>Stem Borer</i> .....	78
39. Hasil <i>Confusion Matrix</i> Model <i>Bidirectional LSTM</i> pada <i>Top Borer</i> .....	79
40. Hasil <i>Confusion Matrix</i> model <i>Bidirectional LSTM</i> pada Kutu Perisai.....	79
41. Hasil <i>Confusion Matrix</i> model TCN pada <i>Stem Borer</i> .....	82
42. Hasil <i>Confusion Matrix</i> model TCN pada <i>Top Borer</i> .....	82
43. Hasil <i>Confusion Matrix</i> model TCN pada Kutu Perisai .....	83
44. Kurva ROC TCN per Kelas (Semua Target) .....	84

## DAFTAR LAMPIRAN

Lampiran	Halaman
1. Jadwal Penelitian .....	109
2. Kode Program Integrasi Data Curah Hujan .....	109
3. Kode Program <i>Splitting</i> Data berdasarkan blok.....	109
4. Distribusi Target Pembagian Data Skema 60:20:20 .....	110
5. Distribusi Target Pembagian Data Skema 80:10:10 .....	110
6. Distribusi Target Pembagian Data Skema 70:15:15 .....	111
7. Kode Program <i>Feature Engineering</i> .....	111
8. Kode Program Pembuatan Fitur Temporal Dasar.....	112
9. <i>Confusion matrix</i> proporsi 60:20:20 .....	112
10. <i>Confusion matrix</i> proporsi 70:15:15 .....	113
11. Kurva ROC Semua Model per Kelas <i>Split</i> 70:15:15 .....	114
12. Kurva ROC Semua Model per Kelas <i>Split</i> 60:20:20 .....	115
13. Grafik <i>Loss</i> dan Akurasi Bi-LSTM Pembagian Data 60:20:20 .....	116
14. Grafik <i>Loss</i> dan Akurasi TCN Pembagian Data 60:20:20 .....	116
15. Grafik <i>Loss</i> dan Akurasi Bi-LSTM Pembagian Data 70:15:15 .....	117
16. Grafik <i>Loss</i> dan Akurasi TCN Pembagian Data 70:15:15 .....	117
17. Grafik <i>Loss</i> dan Akurasi Bi-LSTM Pembagian Data 80:10:10 .....	118
18. Grafik <i>Loss</i> dan Akurasi TCN Pembagian Data 80:10:10 .....	119

## I. PENDAHULUAN

### 1.1. Latar Belakang

Indonesia dikenal sebagai negara agraris dengan kondisi tanah yang subur, sehingga mayoritas penduduknya menggantungkan hidup sebagai petani. Kesuburan tanah tersebut menjadikan Indonesia memiliki potensi besar dalam pengembangan sektor pertanian, yang sekaligus memegang peranan penting dalam pembangunan ekonomi dan perekonomian nasional secara keseluruhan (Syahnaz et al., 2022). Meski demikian, sektor pertanian masih dihadapkan pada beragam tantangan yang cukup kompleks, antara lain keterbatasan lahan produktif, minimnya pemanfaatan teknologi modern, perubahan iklim, serta praktik budidaya yang kurang berkelanjutan (Zahra & Aulia, 2025).

Di antara berbagai tantangan tersebut, ancaman serius datang dari organisme pengganggu tanaman (OPT) atau hama (Bonnu, 2025). Serangan hama secara langsung merusak tanaman budidaya sehingga mengakibatkan penurunan signifikan pada kuantitas maupun kualitas hasil panen, yang pada akhirnya menimbulkan kerugian ekonomi besar. Upaya pengendalian dengan penggunaan pestisida secara berlebihan sering kali tidak efektif, bahkan justru membuat hama dan penyakit menjadi semakin resisten terhadap racun tersebut (Wedastra et al., 2020). Kondisi ini mendorong perlunya inovasi menuju sistem pengelolaan hama yang lebih cerdas, proaktif, dan berkelanjutan.

Tantangan tersebut menjadi semakin relevan ketika difokuskan pada subsektor perkebunan, khususnya pada komoditas tebu yang merupakan bahan baku utama untuk industri gula yang menjadi komoditas strategis nasional (Jeni et al., 2025). Tebu (*Saccharum officinarum*) tidak hanya menjadi bahan utama industri gula,

tetapi juga termasuk salah satu komoditas pokok yang banyak dikonsumsi masyarakat Indonesia, dengan kebutuhan yang terus meningkat seiring bertambahnya jumlah penduduk, baik untuk konsumsi rumah tangga maupun sektor industri (Heryanto et al., 2024). Namun demikian, industri per-tebu-an nasional masih menghadapi persoalan serius, salah satunya produksi gula yang belum mampu memenuhi kebutuhan konsumsi dalam negeri (Sinuraya et al., 2024). Hal diperparah dengan adanya serangan hama yang berdampak pada penurunan hasil panen (Adrian et al., 2019a).

Studi kasus dalam penelitian ini, Perusahaan Gula A, adalah salah satu entitas agribisnis yang operasionalnya sangat bergantung pada pasokan tebu yang stabil dan berkualitas. Kelangsungan produksi perusahaan ini secara rutin menghadapi ancaman dari berbagai jenis hama endemik, seperti penggerek batang (*Chilo sacchariphagus* Bojer), penggerek pucuk (*Scirpophaga excerptalis* Walker), dan kutu perisai (*Aulacaspis tegalensis* Zehntner) yang tidak hanya menyerang secara individual tetapi seringkali terjadi dalam bentuk serangan *multi-label* yang simultan dan saling berinteraksi. Keterlambatan dalam deteksi dan penanganan serangan hama dapat menyebabkan kerusakan yang tidak dapat dipulihkan, sehingga diperlukan sistem prediksi yang akurat dan responsif untuk mengantisipasi serangan sebelum mencapai tingkat yang merugikan.

Dalam konteks globalisasi dan revolusi industri 4.0, peran teknologi dalam sektor pertanian menjadi sangat krusial. Teknologi pertanian modern berpotensi untuk mendefinisikan cara kita memproduksi dan mengelola hasil pertanian, meningkatkan efisiensi dan produktivitas serta memastikan keberlanjutan sektor ini di masa depan (Mendrofa et al., 2024). Salah satu teknologi yang banyak diterapkan dalam bidang pertanian adalah *machine learning*. Metode ini merupakan bagian dari kecerdasan buatan yang memungkinkan komputer mempelajari data serta menghasilkan prediksi atau keputusan berdasarkan pola yang terkandung di dalamnya (Anggarda et al., 2023). Pendekatan berbasis data dan algoritma pembelajaran mesin juga memungkinkan analisis pola temporal yang kompleks dari data historis serangan hama, kondisi lingkungan, dan faktor-faktor lain yang mempengaruhi dinamika populasi hama.

Untuk menjawab tantangan tersebut, penelitian ini akan mengusulkan dan membandingkan dua arsitektur *deep learning* modern yang dirancang khusus untuk data sekuensial atau deret waktu: *Temporal Convolutional Network* (TCN) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM). Bi-LSTM merupakan pengembangan dari LSTM yang telah terbukti andal dalam menangkap dependensi data dari dua arah (masa lalu dan masa depan), sehingga mampu memahami konteks sekuensial secara lebih utuh (Praveenkumar et al., 2025). Di sisi lain, TCN adalah arsitektur yang lebih baru yang menggunakan jaringan konvolusional untuk memproses data deret waktu, menawarkan keunggulan dalam pemrosesan paralel dan kemampuannya menangkap dependensi jangka panjang secara efisien (Shaikh et al., 2023). Kedua algoritma ini memiliki potensi besar namun belum banyak dibandingkan secara langsung untuk kasus peramalan hama yang kompleks.

Sejumlah penelitian sebelumnya telah menunjukkan potensi besar algoritma *deep learning*, khususnya *Bidirectional Long Short-Term Memory* (Bi-LSTM) dan *Temporal Convolutional Network* (TCN), dalam memprediksi data deret waktu (*time series*). Dalam pengembangan model prediksi hama dan penyakit tanaman, Bi-LSTM terbukti secara konsisten mengungguli metode konvensional. Chen et al. (2020) berhasil menerapkan Bi-LSTM untuk memprediksi kejadian hama dan penyakit kapas dengan kinerja yang sangat stabil, mencapai nilai *Area Under Curve* (AUC) sebesar 0.9543 dan akurasi 87.73%, mengungguli arsitektur RNN standar. Keunggulan ini diperkuat oleh Wahyono et al. (2020) yang membuktikan bahwa Bi-LSTM menghasilkan *Mean Squared Error* (MSE) terendah sebesar 0.135 dalam peramalan serangan hama berbasis data cuaca, lebih baik dibandingkan *Vanilla LSTM*. Lebih lanjut, Yadav et al. (2024) menegaskan dominasi Bi-LSTM dalam memprediksi populasi hama penggerek batang padi (*Yellow Stem Borer*), di mana model ini mencatatkan RMSE sebesar 315.11, jauh lebih rendah dibandingkan algoritma *machine learning* tradisional seperti *Random Forest* (336.13) dan *Support Vector Regression* (415.89).

Sementara itu, *Temporal Convolutional Network* (TCN) juga menunjukkan performa yang luar biasa dalam analisis deret waktu pertanian, khususnya untuk data jangka panjang. Sangat relevan dengan komoditas tebu, Silva & Gomedede

(2025) melaporkan bahwa TCN mampu mencapai akurasi generalisasi 96.6% dalam pemantauan tanaman berbasis satelit, secara signifikan mengungguli efisiensi model LSTM dan GRU. Kemampuan TCN menangkap fitur temporal non-linear juga dibuktikan oleh Liu & Song (2025) dalam prediksi hasil panen, di mana TCN menghasilkan MSE 4.59, lebih rendah dibandingkan SVM yang memiliki error 6.42. Selain itu, Ngo et al. (2021) mencatat bahwa TCN memberikan stabilitas yang lebih baik daripada LSTM untuk prediksi jangka panjang pada koloni lebah, dengan akurasi peringatan dini mencapai 90.0%. Rangkaian penelitian ini menegaskan bahwa baik Bi-LSTM maupun TCN memiliki kemampuan yang tinggi dalam menangkap pola temporal kompleks, sehingga layak diimplementasikan untuk prediksi *multi-label* serangan hama pada tanaman tebu.

Berdasarkan uraian tersebut, meskipun TCN dan Bi-LSTM masing-masing menunjukkan potensi besar untuk data deret waktu, evaluasi perbandingan kinerja keduanya secara langsung untuk kasus prediksi serangan hama tebu yang bersifat *multi-label* menjadi langkah yang relevan dan penting untuk dieksplorasi. Oleh karena itu, penelitian ini akan melakukan "Analisis Perbandingan Algoritma *Temporal Convolutional Network* (TCN) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM) dalam Prediksi *Multi-Label* Serangan Hama Tebu" dengan studi kasus di Perusahaan Gula A. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi model peramalan terbaik yang dapat diimplementasikan oleh perusahaan sebagai sistem pendukung keputusan untuk manajemen hama yang lebih efektif dan berbasis data.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah dalam penelitian ini adalah bagaimana membangun sebuah model prediksi yang akurat untuk mengestimasi tingkat keparahan serangan hama *multi-label* pada tanaman tebu, berdasarkan data historis, cuaca, dan kondisi agronomis yang ada, melalui analisis serta perbandingan komprehensif antara kinerja algoritma *Temporal Convolutional Network* (TCN) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM).

### 1.3. Batasan Masalah

Untuk memfokuskan penelitian dan menghindari pembahasan yang terlalu luas, maka penelitian ini dibatasi pada:

1. Penelitian ini menggunakan data yang disediakan oleh Perusahaan Gula A dalam kurun waktu 2022-2023. Kualitas dan kelengkapan data bergantung sepenuhnya pada sumber tersebut.
2. Fokus penelitian terbatas pada implementasi dan perbandingan dua algoritma, yaitu *Temporal Convolutional Network (TCN)* dan *Bidirectional Long Short-Term Memory (Bi-LSTM)*.
3. Jenis hama yang menjadi objek prediksi terbatas pada data yang tersedia dari perusahaan, yaitu penggerek batang (*Chilo sacchariphagus* Bojer), penggerek pucuk (*Scirpophaga excerptalis* Walker), dan kutu perisai (*Aulacaspis tegalensis* Zehntner).
4. *Output* dari penelitian ini adalah sebuah analisis perbandingan kinerja dan rekomendasi model terbaik, serta prediksi tingkat keparahan hama.
5. Penelitian ini berfokus pada aspek teknis pembangunan model prediksi dan tidak akan membahas secara mendalam aspek agronomis, biologis hama, atau efektivitas metode pengendalian hama di lapangan.

### 1.4. Tujuan Penelitian

Adapun, tujuan penelitian ini adalah sebagai berikut:

1. Mengimplementasikan arsitektur TCN dan Bi-LSTM untuk membangun model yang mampu melakukan prediksi serangan hama tebu secara *multi-label*.
2. Menganalisis dan membandingkan performa kedua algoritma menggunakan *confusion matrix* dan *Receiver Operating Characteristic (ROC) curve*.
3. Menentukan model prediksi yang paling unggul dan memberikan rekomendasi model terbaik bagi Perusahaan Gula A sebagai dasar pengembangan sistem pendukung keputusan untuk manajemen hama.

### 1.5. Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini meliputi:

1. Memberikan kontribusi ilmiah mengenai perbandingan kinerja TCN dan Bi-LSTM pada kasus data deret waktu di sektor pertanian, khususnya untuk masalah prediksi *multi-label* yang masih jarang dieksplorasi.
2. Meningkatkan efektivitas pengambilan keputusan dalam manajemen hama, dari reaktif menjadi lebih proaktif dan efisien.
3. Menyediakan model prediksi paling optimal sebagai landasan pengembangan sistem pendukung keputusan yang efektif untuk mengelola hama di Perusahaan Gula A.
4. Hasil prediksi tingkat serangan hama yang dapat dimanfaatkan untuk perencanaan alokasi sumber daya pengendalian hama yang lebih tepat sasaran, penjadwalan aplikasi pestisida atau agen biokontrol secara preventif, serta prioritas pemantauan pada blok-blok dengan serangan hama tinggi sebelum kerusakan pada perkebunan tebu meluas.
5. Mendorong adopsi teknologi berbasis *deep learning* di sektor pertanian Indonesia, sekaligus menunjukkan potensi teknologi modern untuk meningkatkan produktivitas dan keberlanjutan dalam menghadapi tantangan hama di perkebunan tebu.

## II. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian ini dilandasi oleh berbagai studi literatur terdahulu yang mengeksplorasi efektivitas metode *deep learning* dalam menangani data sekuensial dan deret waktu (*time series*). Secara khusus, kajian ini menyoroti kinerja kedua model, yaitu *Bidirectional Long Short-Term Memory* (Bi-LSTM) dan *Temporal Convolutional Network* (TCN). Rangkuman penelitian terdahulu yang relevan disajikan pada Tabel 1.

Tabel 1. Penelitian Terdahulu

No.	Penulis	Judul	Metode	Hasil
1.	(Chen et al., 2020)	<i>Occurrence Prediction of Cotton Pests and Diseases by Bidirectional Long Short-Term Memory Networks with Climate and Atmosphere Circulation</i>	Bi-LSTM, LSTM, Random Forest, SVM	Bi-LSTM mengungguli metode tradisional (RF & SVM) dalam prediksi kemunculan hama dan penyakit kapas. Model Bi-LSTM mencapai AUC = 0.95, lebih tinggi dan lebih stabil dibandingkan LSTM satu arah serta algoritma pembandingan lainnya, terutama dalam menangkap ketergantungan temporal jangka panjang data iklim.
2.	(Filho et al., 2020)	<i>Rice Crop Detection Using</i>	<i>Bi-LSTM, LSTM,</i>	Bi-LSTM mencatat performa tertinggi dengan

No.	Penulis	Judul	Metode	Hasil
		<i>LSTM, Bi-LSTM, and Machine Learning Models from Sentinel-1 Time Series</i>	<i>SVM, Random Forest</i>	<i>Overall Accuracy</i> mencapai 99.14% dan <i>Kappa Coefficient</i> 0.9900, yang secara spesifik menungguli LSTM standar (98.86%) serta algoritma <i>Machine Learning</i> tradisional seperti <i>Random Forest</i> (98.39%) dan <i>SVM</i> (98.28%) dalam mendeteksi tanaman padi.
3.	(Yadav et al., 2024)	<i>Predicting Yellow Stem Borer Occurrence in Rice Using Weather Parameters and LSTM</i>	Bi-LSTM, LSTM, Deep LSTM, RNN, GRU, RF, SVR	Bi-LSTM menunjukkan keunggulan signifikan dibandingkan metode tradisional dengan nilai RMSE 315.11, yang jauh lebih rendah daripada <i>Random Forest</i> (336.13) dan SVR (415.89), serta lebih baik dari RNN standar (331.42) dalam memprediksi kejadian hama berdasarkan cuaca.
4.	(Wahyono et al., 2020)	<i>Enhanced LSTM Multivariate Time Series Forecasting for Crop Pest Attack Prediction</i>	Bi-LSTM, Vanilla LSTM	Bi-LSTM menghasilkan tingkat kesalahan terendah dengan <i>Mean Squared Error</i> (MSE) sebesar 0.135 dan RMSE 0.368, yang terbukti lebih akurat dibandingkan varian <i>Vanilla LSTM</i> (MSE 0.136) maupun <i>Stack LSTM</i> dalam

No.	Penulis	Judul	Metode	Hasil
				peramalan deret waktu.
5.	(Silva & Gomedede, 2025)	<i>Temporal Deep Learning for Satellite-Based Sugarcane Monitoring and Age Estimation</i>	TCN, LSTM, GRU, Transformer, TSMixer	TCN terbukti menjadi model temporal terbaik dengan akurasi dari generalisasi mencapai 96.6%, yang secara signifikan mengungguli efisiensi model berbasis RNN seperti LSTM (93.5%) dan GRU (91.2%) pada pengujian dataset satelit tebu.
6.	(Ngo et al., 2021)	<i>Honey Bee Colony Population Daily Loss Rate Forecasting and An Early Warning Method Using Temporal Convolutional Networks</i>	TCN	Penerapan TCN menghasilkan performa prediksi yang handal dengan WMAPE sebesar 17.1% dan akurasi dari sistem peringatan dini mencapai 90.0%, menunjukkan stabilitas yang lebih baik untuk data jangka panjang dibandingkan metode konvensional..
7.	(Liu & Song, 2025)	<i>Accurate Crop Yield Prediction Via Temporal Convolutional Network and Variational Autoencoder</i>	TCN, LSTM, SVM	Pada pengujian dataset Eropa, TCN berhasil mencatatkan <i>Mean Squared Error</i> (MSE) sebesar 4.59, yang lebih rendah dibandingkan model LSTM (5.91) dan

No.	Penulis	Judul	Metode	Hasil
				SVM (6.42), yang membuktikan model TCN mampu dalam menangkap fitur temporal non-linear yang kompleks.
8.	(Betha et al., 2025)	<i>Crop Leaf Disease Prediction Using Graph Diffusion TCN with Fibroblast Optimization</i>	<i>Graph Diffusion TCN</i>	Model <i>Graph Diffusion</i> berbasis TCN ini mencapai efisiensi klasifikasi hampir sempurna sebesar 99.9% dengan sensitivitas 99.8%, mengungguli metode diagnosis penyakit daun tanaman lainnya yang ada saat ini.

## 2.2. Landasan Teori

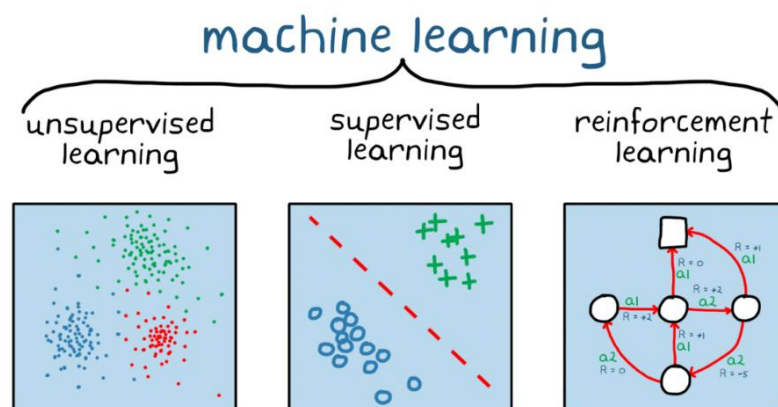
Dalam penelitian ini, landasan teori disusun ke dalam tiga bagian pokok, yaitu pembahasan mengenai konsep machine learning beserta algoritmanya, pemanfaatan bahasa pemrograman Python beserta pustaka pendukung yang digunakan, serta penjelasan terkait hama pada tanaman tebu yang menjadi fokus objek penelitian.

### 2.2.1. Machine Learning

*Machine Learning* (ML) atau pembelajaran mesin adalah cabang dari kecerdasan buatan (AI) yang fokusnya adalah pada pemanfaatan data untuk membangun sistem

yang nantinya bisa belajar secara mandiri tanpa perlu diprogram berulang kali oleh manusia (Alfarizi et al., 2023). Prinsip dasar dari *machine learning* adalah pengembangan sistem melalui proses pelatihan dengan *dataset* berskala sangat besar untuk menangani tugas-tugas dengan kompleksitas tinggi. Tujuan utama dari model yang dihasilkan adalah untuk memiliki kemampuan dalam mengambil keputusan, melakukan prediksi, atau mengeksekusi fungsi tertentu secara otomatis, tanpa bergantung pada pemrograman langsung. Oleh karena itu, model harus mampu memproduksi keluaran yang diinginkan saat diberikan suatu masukan (Rahmani et al., 2021).

Secara teori, *machine learning* terbagi ke dalam tiga kategori utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Pratama et al., 2025). *Supervised learning*, yang merupakan pendekatan paling dominan dan bekerja dengan data yang telah diberi label. Tujuan model adalah mempelajari korelasi antara fitur masukan dan label tersebut, di mana tugasnya dapat berupa Klasifikasi atau Regresi. Sebaliknya, seperti ditunjukkan pada Gambar 1, *unsupervised learning* diterapkan pada data tanpa label, di mana mesin ditugaskan untuk secara mandiri mengidentifikasi struktur atau pola tersembunyi di dalam data tanpa panduan apapun (Cheraghi et al., 2021). Sementara itu, *reinforcement learning* terinspirasi dari cara hewan belajar. Pembelajaran ini tidak bergantung pada data yang sudah ada, melainkan pada pengalaman yang diperoleh dari interaksi berkelanjutan dengan lingkungan dinamis melalui mekanisme *trial-and-error* untuk menemukan kebijakan optimal (Roihan et al., 2020).



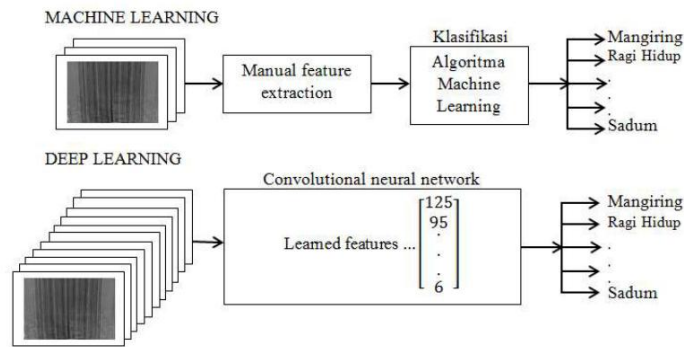
Gambar 1. *Supervised, Unsupervised, dan Reinforcement Learning*  
(Kumar, 2024)

### 2.2.2. Deep Learning

*Deep learning* (DL) adalah evolusi dari *machine learning* (ML), yang mana keduanya juga merupakan sub-bidang dari kecerdasan buatan (AI). Inti dari *Deep Learning* adalah penggunaan algoritma yang memungkinkannya belajar dari data yang ada dan melakukan tugas tertentu secara otomatis tanpa perlu diprogram secara eksplisit (Ahmed et al., 2023). *Deep learning* merupakan bidang dalam *Machine Learning* yang algoritmanya meniru struktur otak seorang manusia yang disebut dengan Jaringan Saraf Tiruan (*Artificial Neural Networks* atau ANN). Beberapa algoritma populer dalam *Deep learning* meliputi *Convolutional Neural Network* (CNN), *Long Short-Term Memory* (LSTM), *Recurrent Neural Network* (RNN), dan *Self-Organizing Maps* (SOM) (Alfarizi et al., 2023).

Umumnya, metode *deep learning* (DL) dapat dikelompokkan menjadi tiga kategori besar, yaitu *supervised*, *semi-supervised*, dan *unsupervised*. Ada pula pendekatan lain yang disebut *deep reinforcement learning* (DRL) atau RL, yang posisinya sering kali dianggap saling tumpang tindih dengan kategori *semi-supervised*, dan dalam beberapa kasus, juga dengan kategori *unsupervised* (Alzubaidi et al., 2021). Cara kerja *deep learning* bergantung pada arsitektur jaringan dan proses optimisasi yang digunakan. Data akan diproses melalui beberapa lapisan tersembunyi (*hidden layers*), di mana setiap lapisan melakukan Transformasi Non-Linear dengan menggunakan fungsi aktivasi. Proses ini bertujuan untuk terus menyesuaikan bobot (kekuatan koneksi antar *neuron*) hingga model mampu menghasilkan *output* yang paling akurat atau sesuai dengan target yang ingin dicapai (Ahmed et al., 2023).

Seperti yang ditunjukkan pada Gambar 2, perbedaan mendasar antara *deep learning* dan *machine learning* konvensional terletak pada otomatisasi ekstraksi fitur. *Machine learning* masih memerlukan intervensi manusia untuk menentukan fitur-fitur penting dari data secara manual. Sementara itu, *deep learning* dapat menemukan serta memanfaatkan fitur-fitur tersebut secara otomatis sebagai bagian dari proses pembelajarannya (Hutagalung & Sitompul, 2023).



Gambar 2. Perbedaan Proses *Machine Learning* dan *Deep Learning*  
(Hutagalung & Sitompul, 2023)

### 2.2.3. *Recurrent Neural Network*

*Recurrent Neural Network* (RNN) dikembangkan pada tahun 1980-an untuk memungkinkan jaringan saraf memproses data sekuensial, seperti teks atau deret waktu. Arsitektur ini berhasil mengatasi dua masalah utama, yaitu kemampuan dalam mengolah input dengan panjang yang tidak seragam serta menangkap hubungan antara data masa lalu dan masa depan. Hal ini dapat dicapai melalui mekanisme "*parameter sharing*", di mana model menggunakan bobot yang sama secara berulang di setiap langkah waktu, sehingga ia memiliki "memori" untuk membuat prediksi atau ramalan berdasarkan konteks sebelumnya (Fernández et al., 2024). Artinya, hasil keluaran di satu langkah tidak hanya bergantung pada masukan saat itu, tetapi juga pada hasil dari semua perhitungan sebelumnya. Panjang sekuens masukan secara langsung menentukan berapa kali proses ini akan diulang (Onan, 2022). Model dari RNN disajikan dalam Persamaan 1-2:

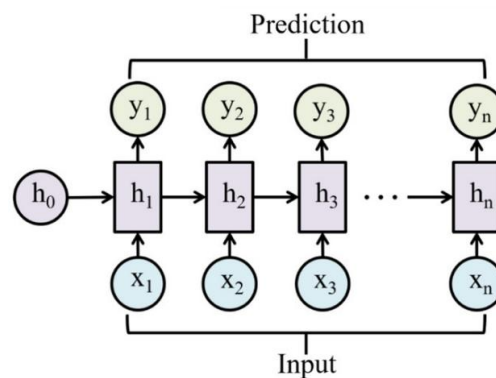
Arsitektur standar RNN terdiri dari *input layer*, *hidden layer*, dan *output layer*. Berbeda dengan jaringan saraf *feedforward*, RNN menggunakan koneksi *recurrent* untuk memungkinkan sirkulasi informasi secara internal, seperti yang digambarkan pada Gambar 3. Pada setiap langkah waktu  $t$ , RNN menerima vektor input  $x_t$  dan memperbarui kondisi tersembunyi (*hidden state*)  $h_t$  menggunakan Persamaan 1-2.

$$h_t = \sigma_h(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

di mana  $W_{hx}$  dan  $W_{hh}$  masing-masing adalah matriks bobot untuk koneksi input-tersembunyi dan koneksi rekuren,  $b_h$  adalah vektor bias, dan  $\sigma_h$  adalah fungsi aktivasi berupa tangen hiperbolik ( $\tanh$ ) atau ReLU (Mienye et al., 2024). Selanjutnya, output pada setiap langkah waktu  $t$  dihitung sebagai:

$$y_t = \sigma_y(W_{hy}h_t + b_y) \quad (2)$$

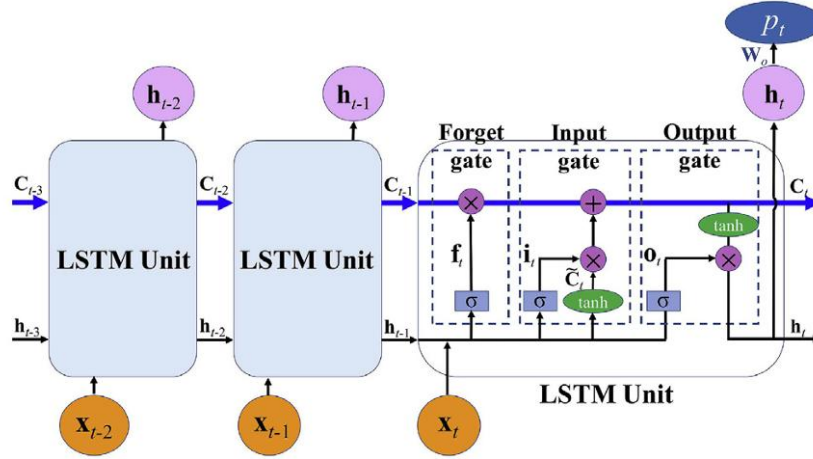
Di mana  $W_{hy}$  adalah matriks bobot lapisan tersembunyi ke output,  $b_y$  adalah vektor bias, dan  $\sigma_y$  adalah fungsi aktivasi lapisan output .



Gambar 3. Arsitektur Standar RNN  
(Mienye et al., 2024)

#### 2.2.4. Long Short-Term Memory

Jaringan *Long Short-Term Memory* (LSTM) adalah sebuah arsitektur jaringan saraf dalam (*deep neural network*) yang berbasis RNN di mana *forget gates* (gerbang lupa) dimanfaatkan untuk menghilangkan masalah *exploding* atau *vanishing gradient*. Berbeda dengan arsitektur jaringan saraf berulang tradisional, LSTM memungkinkan *error backpropagation* melalui sejumlah langkah waktu (*time steps*) yang terbatas (Onan, 2022). Seperti yang terlihat pada Gambar 4, setiap unit LSTM terdiri dari tiga gerbang. *Input Gate* menentukan informasi yang masuk ke dalam keadaan sel saat ini, *forget gate* mengontrol seberapa banyak memori sebelumnya yang disimpan, dan *output gate* mengatur informasi yang diteruskan ke langkah waktu berikutnya (Chen et al., 2020; Tarkus et al., 2020).



Gambar 4. Struktur *Long Short-Term Memory*

(Wei et al., 2021)

Aliran informasi melalui unit LSTM berlangsung dalam lima langkah berurutan sebagai berikut (Yadav et al., 2024).

Langkah 1. Keluaran dari unit sebelumnya dan *input* saat ini dimasukkan ke dalam *forget gate*, yang menghitung keluarannya menggunakan Persamaan 3.

$$f_t = \sigma\{W_f \times (h_{t-1} \times x_t)\} + b_f \quad (3)$$

Di mana  $W_f$  adalah bobot *forget gate*,  $b_f$  adalah bias,  $x_t$  adalah *input* saat ini, dan  $h_{t-1}$  adalah *hidden state* dari unit sebelumnya.

Langkah 2. *Output* unit sebelumnya dan *input* saat ini dimasukkan ke *input gate*, sementara *cell state* kandidat dihitung secara bersamaan menggunakan persamaan-persamaan berikut:

$$i_t = \sigma\{W_i \times (h_{t-1} \times x_t)\} + b_i \quad (4)$$

$$\tilde{C}_t = \tanh\{W_c \times (h_{t-1} \times x_t)\} + b_c \quad (5)$$

Di mana  $W_i$  adalah bobot *input gate*,  $b_i$  adalah bias, sedangkan  $W_c$  dan  $b_c$  masing-masing merupakan bobot dan bias dari *cell state* kandidat.

Langkah 3. Status sel saat ini diperbarui dengan menggabungkan *output forget gate* dan *input gate* pada Persamaan 6:

$$\tilde{C}_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (6)$$

Langkah 4. *Output gate* menerima  $h_{t-1}$  dan  $x_t$  sebagai masukan dan menghasilkan *output* sesuai dengan Persamaan 7.

$$o_t = \sigma\{W_o \times (h_{t-1} \times x_t)\} + b_o \quad (7)$$

Di mana  $W_o$  dan  $b_o$  masing-masing merupakan bobot dan bias *output gate*.

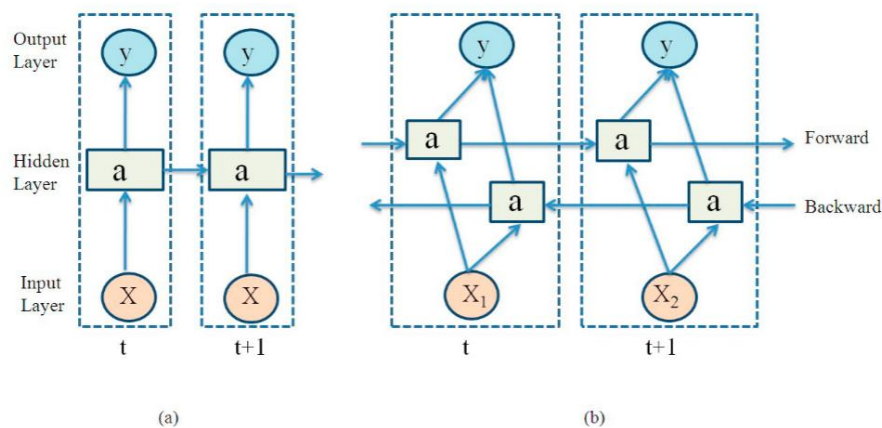
Langkah 5. *Hidden state* akhir dari unit LSTM diperoleh dengan menggabungkan hasil *output gate* dengan *cell state* saat ini, yang ditunjukkan pada Persamaan 8.

$$h_t = o_t \times \tanh(C_t) \quad (8)$$

Di mana  $C_t$  merupakan *cell state*,  $H_t$  Adalah keluaran dari LSTM pada waktu (Abduljabbar et al., 2021b).

### 2.2.5. Bidirectional Long Short-Term Memory

Bi-LSTM merupakan pengembangan dari LSTM yang telah terbukti andal dalam menangkap dependensi data dari dua arah (masa lalu dan masa depan), sehingga mampu memahami konteks sekuensial secara lebih utuh (Praveenkumar et al., 2025). Berbeda dengan LSTM, arsitektur *Bidirectional LSTM* (Bi-LSTM) menggunakan dua lapisan LSTM yang bekerja secara paralel, dimana satu lapisan memproses data dari awal hingga akhir (arah maju), dan lapisan lainnya memproses data dari akhir ke awal (arah mundur).



Gambar 5. (a) Konfigurasi LSTM, (b) Konfigurasi Bi-LSTM

(Mahadevaswamy & Swathi, 2022)

Seperti yang ditunjukkan pada Gambar 5, pendekatan dua arah ini memungkinkan model untuk mempertimbangkan konteks dari masa lalu dan masa depan secara bersamaan. Walaupun mirip cara melatihnya dengan LSTM standar, proses *back-propagation* pada Bi-LSTM memerlukan penanganan khusus karena pembaruan bobot untuk kedua arah tidak dapat dilakukan serentak (Mahadevaswamy & Swathi, 2022).

Meskipun LSTM standar memproses data berurutan hanya dalam satu arah maju, Bi-LSTM memperluas model LSTM standar dengan memproses urutan masukan secara bersamaan dalam arah maju dan mundur, sehingga memungkinkan model ini menangkap ketergantungan temporal dua arah dari konteks masa lalu dan masa depan secara lebih komprehensif (Mahadevaswamy & Swathi, 2022; Praveenkumar et al., 2025). Keluaran lapisan Bi-LSTM secara formal didefinisikan pada Persamaan 9 (Yadav et al., 2024).

$$(h_t, C_t) = LSTM([h_{t-1}, h_{t+1}, x_i], C_{t-1}, C_{t+1}, W) \quad (9)$$

Di mana  $W$  merupakan gabungan dari empat matriks bobot  $W_i$ ,  $W_f$ ,  $W_o$ , dan  $W_c$ . Lapisan *fully-connected* ditambahkan untuk menggabungkan dan mengurangi dimensi vektor *output* yang digabungkan, diikuti fungsi aktivasi *softmax* untuk menghasilkan hasil klasifikasi akhir, seperti yang ditunjukkan pada Persamaan 10.

$$prediction = softmax(W_{fc} \times y_i + b_{fc}) \quad (10)$$

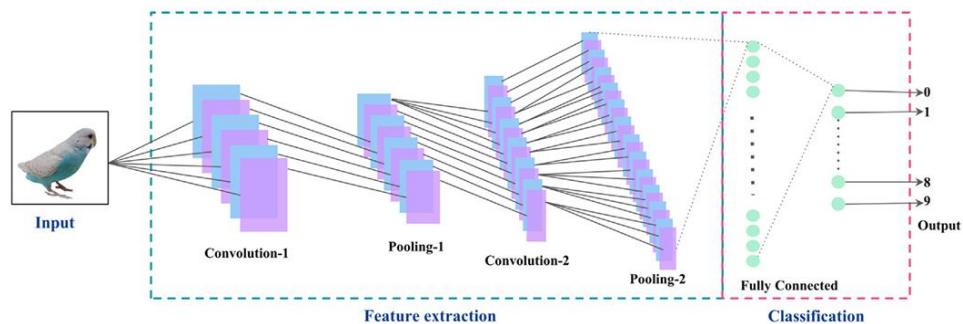
Di mana  $W_{fc}$  dan  $b_{fc}$  masing-masing merupakan matriks bobot dan bias lapisan FC, sedangkan prediksi  $\in \{0,1,2,3\}$  mewakili tingkat keparahan yang diklasifikasikan setelah pengkodean *one-hot*. Selain itu, lapisan dropout dimasukkan ke dalam blok Bi-LSTM untuk mencegah *overfitting* selama pelatihan.

### 2.2.6. Convolutional Neural Network (CNN)

Sebagai salah satu jenis jaringan saraf tiruan, *Convolutional Neural Network* (CNN) sangat unggul dalam tugas pengenalan dan analisis gambar karena meniru cara

kerja neuron visual. Keunikan utamanya terletak di operasi konvolusi, sebuah proses di mana *filter* digerakkan ke seluruh bagian gambar untuk mengidentifikasi pola-pola penting *feature map* (Peryanto et al., 2020). Meskipun sama-sama terinspirasi dari otak manusia, arsitektur CNN memiliki perbedaan mendasar dengan CNN biasa. *Neuron* pada CNN direpresentasikan dalam format dua dimensi agar sesuai dengan struktur data gambar. Operasi matematis utamanya bukanlah hanya perkalian matriks sederhana, tapi operasi konvolusi. Akibatnya, *weights* pada CNN tidak lagi berupa vektor satu dimensi, tapi sebuah tensor empat dimensi yang berupa kumpulan dari filter atau kernel konvolusi (Rasidi et al., 2022).

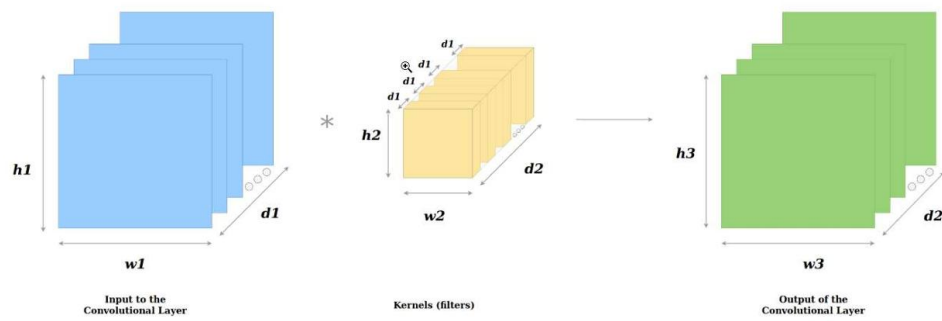
Seperti yang ditunjukkan pada Gambar 8, arsitektur dasar sebuah CNN dibangun di atas tiga jenis lapisan utama: Lapisan Konvolusi (*Convolutional Layers*), Lapisan Pooling (*Pooling layers*), dan Lapisan Terhubung Penuh (*Fully-connected layers*). Kinerja model sangat bergantung pada semua lapisan ini, yang akan dibahas secara mendalam selanjutnya (Raiaan et al., 2024).



Gambar 6. Arsitektur CNN  
(Raiaan et al., 2024)

a. *Convolutional Layers* (Lapisan Konvolusi)

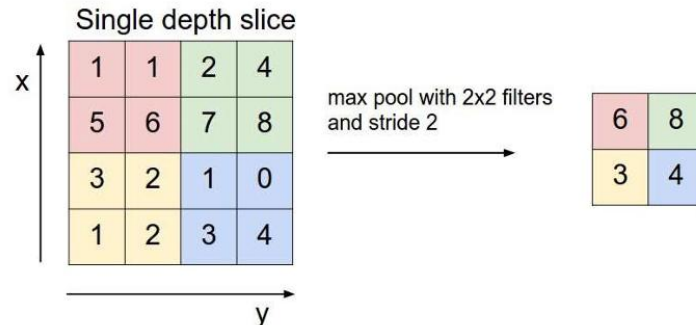
*Convolutional Layers* adalah ciri khas utama yang membuat CNN berbeda dari jaringan saraf lain. Seperti terlihat di Gambar 9, lapisan ini berisi sekumpulan filter untuk mendeteksi fitur (Hamza et al., 2022). Lapisan ini berguna sebagai pengekstrak fitur-fitur penting dari gambar dan mengubahnya menjadi matriks yang lebih ringkas (*feature maps*).

Gambar 7. *Convolutional Layers*

(Peryanto et al., 2020)

b. *Pooling layers* (Lapisan Pooling)

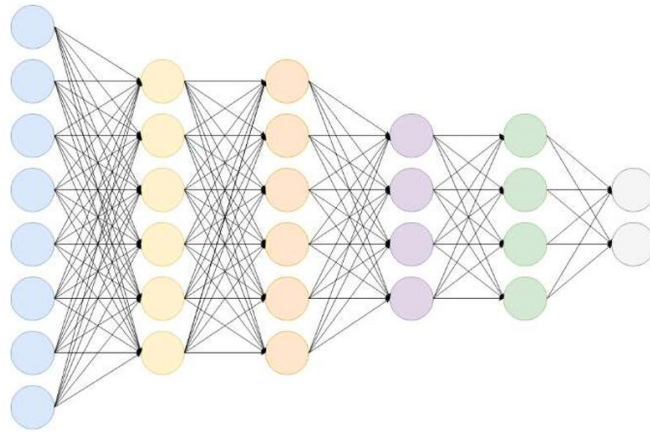
Setelah fitur-fitur diekstrak oleh lapisan konvolusi, lapisan *pooling* bertugas untuk meringkas dan serta menyederhanakan informasi tersebut. Proses ini juga disebut dengan *down-sampling* yang berfungsi untuk mengurangi ukuran peta fitur dengan hanya mempertahankan informasi yang paling dominan. Tujuannya adalah untuk mengurangi beban komputasi dan membuat model lebih efisien tanpa kehilangan fitur-fitur penting (Khan et al., 2020; Masuda & Yairi, 2023). Gambar 10 menunjukkan bagaimana proses *down-sampling* terjadi.

Gambar 8. *Pooling Layers*

(Peryanto et al., 2020)

c. *Fully Connected Layers* (Lapisan Terhubung Penuh)

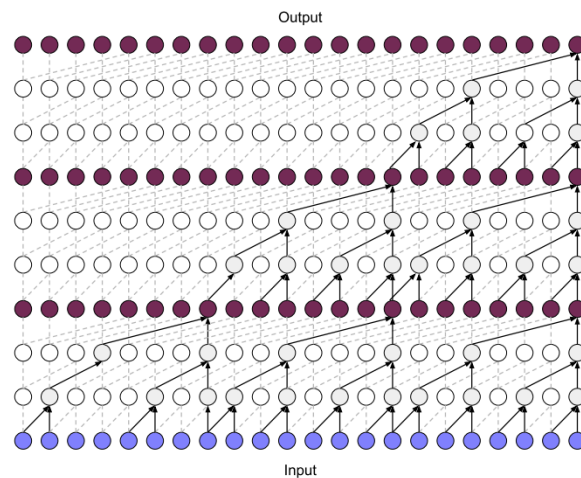
Sebagaimana ditunjukkan pada Gambar 11, *fully connected layers* adalah lapisan terakhir yang berfungsi sebagai pengambil keputusan. Lapisan ini mengintegrasikan semua fitur yang telah diekstrak dan diringkas sebelumnya ke dalam sebuah vektor tunggal untuk proses klasifikasi (Hamza et al., 2022). Karena kepadatan parameternya, lapisan ini membutuhkan sumber daya komputasi yang besar (Raiaan et al., 2024).



Gambar 9. *Fully Connected Layers*  
(Peryanto et al., 2020)

### 2.2.7. *Temporal Convolutional Network (TCN)*

TCN (*Temporal Convolutional Network*) merupakan sebuah pendekatan berbasis konvolusi yang efektif untuk menganalisis data yang bersifat sekuensial, terutama untuk menemukan pola rentang waktu yang panjang. Mirip dengan RNN, TCN mampu memetakan input sekuensial ke *output* dengan panjang yang sama. Namun, keunikannya terletak pada penggunaan konvolusi kausal, yang memastikan bahwa pemrosesan data tidak “mengintip” ke masa depan, sehingga lebih cocok untuk prediksi deret waktu (Hewage et al., 2020).

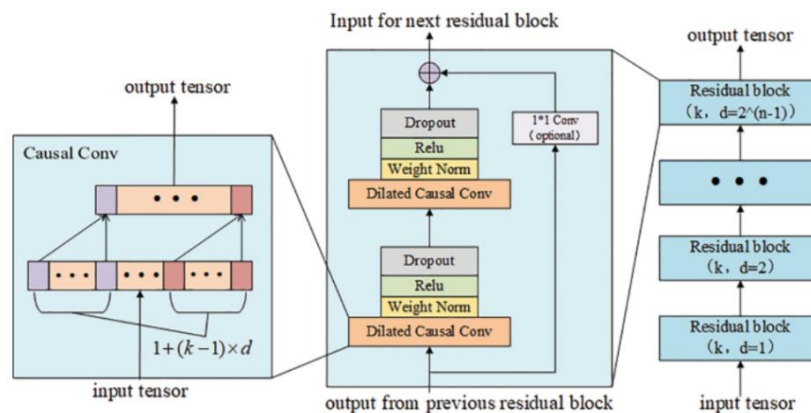


Gambar 10. *TCN Layers*  
(Lara-Benítez et al., 2020)

Seperti jaringan *deep learning* lainnya, arsitektur *Temporal Convolutional Network* (TCN) juga terdiri dari tiga lapisan utama: *input*, *hidden*, dan *output* seperti yang ditunjukkan pada Gambar 12. Pada TCN, lapisan tersembunyi (*hidden layer*) memegang peranan penting dalam melakukan operasi konvolusi satu dimensi (1D), di mana ia menerima masukan dari area terbatas lapisan sebelumnya yang disebut dengan *receptive field*. Berbeda dengan CNN konvensional yang menggunakan konvolusi dua dimensi untuk data gambar, TCN menerapkan konvolusi 1D *dilated convolution* yang bergerak sepanjang dimensi waktu, sehingga mampu menangkap pola tempooral dari data deret waktu secara efektif (Shaikh et al., 2023), yang operasinya dijelaskan pada Pesamaan 11 (He et al., 2022).

$$g(x) = \sum_{l=0}^{k-1} h(l)f(x - d \cdot l) \quad (11)$$

Di mana  $f(*)$  adalah input,  $g(*)$  adalah output dari *dilated convolution*,  $h(l)$  menunjukkan filter dengan panjang  $k$ ,  $d$  adalah faktor dilasi, dan  $x - d \cdot l$  melambangkan arah masa lalu dari elemen  $x$ .



Gambar 11. Arsitektur Utama TCN  
(Munandar, 2025)

Seiring bertambahnya kedalaman jaringan saraf, penumpukan lapisan konvolusi secara membabi buta sering kali menyebabkan masalah gradien yang meledak dan menghilang. Untuk mengatasi hal ini, TCN memperkenalkan blok residual, yang terdiri dari dua set lapisan konvolusi dilatasi, lapisan normalisasi bobot, fungsi aktivasi ReLU, dan lapisan Dropout yang dihubungkan secara berurutan seperti yang terlihat pada Gambar 13 (Ma et al., 2026). Hubungan antara  $input\ x^i$  dan

output  $x^{i+1}$  untuk *residual block* ke- $i$  dinyatakan pada Persamaan 12.

$$X^{i+1} = \text{Activation} \left( X^i + F(X^i) \right) \quad (12)$$

Di mana *Activation* adalah fungsi aktivasi dan  $F(*)$  adalah transformasi dari *residual block*.

### 2.2.8. Python

Python, yang pertama kali dirilis oleh Guido van Rossum pada tahun 1991, adalah bahasa pemrograman tingkat tinggi yang kini menjadi salah satu yang terpopuler di dunia. Popularitasnya didorong oleh sintaksnya yang bersih dan mudah dipelajari. Karena kemudahan penggunaannya, Python banyak diadopsi di berbagai bidang, terutama untuk ilmu data, pengembangan *web*, dan otomatisasi (Candra, 2025).

Python dipilih karena berbagai keunggulannya. Pertama, Python memiliki koleksi *library* yang sangat lengkap dan serbaguna, mendukung berbagai kebutuhan mulai dari analisis statistik hingga *deep learning*. Kedua, sintaksnya yang sederhana dan mudah dipahami menjadikannya bahasa yang ideal bagi pemula. Selain itu, Python unggul dalam visualisasi data. Beragamnya pilihan paket visualisasi memungkinkan pengguna untuk membuat berbagai grafik dan bagan interaktif. Hal ini sangat mempermudah proses analisis dan pemahaman data, menjadikannya alat yang sangat berguna dalam bidang ilmu data (Junaidi et al., 2023).

#### 2.2.8.1. Pandas

Pandas, yang diperkenalkan oleh Wes McKinney pada tahun 2008, adalah sebuah pustaka Python yang sangat populer di kalangan analis dan ilmuwan data karena menyediakan struktur data yang fleksibel dan alat analisis yang kuat. Pustaka ini memudahkan pengguna untuk memanipulasi dan menganalisis data terstruktur dari berbagai sumber seperti CSV, Excel, dan SQL. Salah satu keunggulan utamanya adalah kemampuan untuk menangani data yang hilang dan melakukan pemfilteran

secara efisien, sehingga berbagai tugas seperti pembersihan dan analisis data dapat dilakukan dengan cepat tanpa perlu menulis kode yang rumit, menjadikannya alat yang sangat praktis dan mudah digunakan (Candra, 2025).

### 2.2.8.2. NumPy

NumPy merupakan pustaka Python yang krusial untuk analisis data dan pemodelan matematika, dengan fokus utama pada pemrosesan data numerik yang efisien. Pustaka ini menyediakan struktur data berupa *array* yang kuat dan beragam fungsi untuk melakukan operasi komputasi, mulai dari aritmatika, trigonometri, hingga statistik, dengan performa tinggi. Kemampuan ini menjadikannya landasan penting bagi banyak aplikasi ilmiah dalam ekosistem Python (Surbakti et al., 2024).

### 2.2.8.3. Matplotlib

Matplotlib merupakan pustaka andalan di Python untuk kebutuhan visualisasi data. Library ini sangat populer karena kemampuannya untuk menyajikan data dalam bentuk array ke dalam berbagai format grafik plot 2D yang informatif (Galahartlambang et al., 2021). Sebagai komponen vital dalam analisis data modern, visualisasi data memungkinkan para peneliti untuk secara efektif mengidentifikasi pola, tren, dan anomali dalam sebuah *dataset*. Matplotlib telah menjadi salah satu pustaka visualisasi data yang paling populer di Python, digunakan secara luas oleh ilmuwan data dan peneliti untuk menciptakan grafik yang informatif dan menarik. Fleksibilitas dan kekuatan Matplotlib terbukti dari pemanfaatannya dalam berbagai penelitian, yang menunjukkan kemampuannya untuk beragam tujuan analisis (Kaestria et al., 2024). Matplotlib berfungsi untuk membuat beragam jenis visualisasi data, seperti *scatter plot*, *line chart*, *bar chart*, dan *pie chart*. Selain itu, pustaka ini juga menyediakan kontrol penuh untuk menyesuaikan tampilan grafik, termasuk menambahkan elemen seperti judul dan label (Surbakti et al., 2024).

#### 2.2.8.4. Scikit-Learn

Scikit-learn adalah pustaka *open-source* untuk *machine learning* yang berfokus pada penggunaan bahasa Python. Pustaka ini pertama kali dirancang oleh David Cournapeau melalui program *Google Summer of Code* dan dirilis pada Juni 2007. Dibangun di atas ekosistem python seperti NumPy, SciPy, Matplotlib, dan Pandas, Scikit-learn dapat terintegrasi sepenuhnya dengan Pustaka tersebut. Walaupun ditulis terutama dalam Python, beberapa algoritma inti ditulis menggunakan Cython agar kinerjanya lebih optimal. Beragam metode *machine learning*, baik *supervised* maupun *unsupervised*, dapat diimplementasikan menggunakan Scikit-learn, termasuk *Classification*, *Regression*, *SVM*, *Random Forest*, *K-Nearest Neighbors*, *Naïve Bayes*, *Decision Trees*, hingga *Clustering* (Gupta, 2025).

Selain menghadirkan berbagai metode pemodelan, pustaka ini juga memberikan kemudahan akses ke sejumlah *dataset* yang bermanfaat untuk pembelajaran, evaluasi kinerja, maupun penelitian eksperimental. Scikit-learn juga dapat mempermudah dalam *Exploratory Data Analysis* (EDA) dan visualisasi. Selain itu, Scikit-learn juga menyediakan mekanisme *pipeline* yang mengintegrasikan tahapan *preprocessing*, seperti *scaling* dan *encoding*, dengan pelatihan serta evaluasi model, sehingga mendukung alur kinerja yang konsisten dan terorganisir. Selain itu, kompatibilitasnya dengan pustaka Pandas dan NumPy memudahkan manipulasi data dan analisis eksploratif, yang mempercepat kegiatan Pendidikan maupun penelitian (Salehi & Zarei, 2025).

#### 2.2.9. Hama Tebu

Sebagai bahan baku utama industri gula, yang merupakan komoditas primer dengan permintaan tahunan yang terus meningkat, produktivitas tebu menghadapi tantangan serius. Salah satu kendala utamanya adalah serangan beragam hama yang mengakibatkan kerugian ekonomi yang signifikan. Kegagalan petani dalam upaya pengendalian hama sering kali disebabkan oleh kurangnya pemahaman dalam mengidentifikasi jenis hama dan mengenali gejala serangan secara akurat (Adrian

et al., 2019). Terdapat lebih dari 100 jenis hama yang dapat menyerang tanaman tebu, terutama hama dari jenis serangga (Subiyakto, 2017). Dalam penelitian ini, hama yang menjadi objek penelitian adalah penggerek pucuk (*Scirpophaga excerptalis* Walker), penggerek batang (*Chilo sacchariphagus* Bojer), dan kutu perisai (*Aulacaspis tegalensis* Zehntner).

### 2.2.9.1. Penggerek Pucuk (*Scirpophaga excerptalis* Walker)

Penggerek pucuk (*Scirpophaga excerptalis* Walker) adalah hama utama pada tanaman tebu yang menyebabkan kerusakan parah, sehingga mengakibatkan penurunan hasil panen dan rendemen gula secara signifikan. Dianggap sebagai kendala serius dalam produksi tebu, hama ini telah dilaporkan keberadaannya di berbagai industri gula di kawasan Asia Tenggara (Paudel et al., 2021).



Gambar 12. Penggerek Pucuk  
(Muliasari & Trilaksono, 2020)

Serangan hama penggerek pucuk umumnya terjadi pada awal masa tanam dan setelah ruas batang terbentuk. Tingkat serangannya pun cenderung sama, baik antar waktu pengamatan maupun antar varietas tebu yang berbeda (Gupta & Paul, 2023). Larva *S. Excerptalis* menyerang pucuk dan daun muda tanaman yang masih kuncup dengan cara menggereknya seperti yang terlihat pada Gambar 14. Akibatnya, ketika daun mekar, akan terlihat barisan lubang bekas gerakan. Setelah menyerang daun, larva ini akan memakan titik tumbuh tanaman yang menyebabkan pucuk tanaman

mati (mati puser). Jika ini terjadi pada tanaman muda, tanaman tersebut biasanya akan mati. Namun, jika serangan terjadi pada tanaman yang lebih tua, pertumbuhannya akan terhambat, tanaman tidak bias tumbuh tinggi lagi, dan sering kali akan muncul tunas-tunas baru dari batangnya (Muliasari & Trilaksono, 2020).

#### 2.2.9.2. Penggerek Batang (*Chilo sacchariphagus* Bojer)

Ada enam jenis penggerek batang tebu yang pernah ditemukan di Indonesia, yaitu penggerek batang berkilat, penggerek batang bergaris, penggerek batang abu-abu, penggerek batang kuning, penggerek batang raksasa, dan penggerek batang jambon (Muliasari & Trilaksono, 2020). Sebagaimana ditampilkan pada Gambar 15, penggerek batang tebu bergaris (*Chilo sacchariphagus* Bojer) merupakan salah satu hama utama yang menyebabkan kerugian ekonomi signifikan. Siklus serangannya dimulai pada fase larva muda yang memakan daun, kemudian berlanjut pada fase larva berikutnya yang menggerek masuk dan merusak ruas-ruas batang hingga mencapai titik tumbuh. Mengingat besarnya kerugian yang dapat ditimbulkan, diperlukan upaya untuk menekan perkembangan hama ini. Salah satu metode pengendalian hama yang dianggap efektif adalah pengendalian hayati dengan memanfaatkan musuh alami, seperti parasitoid, predator, pathogen (jamur, bakteri, virus), nematoda, maupun mikroorganisme bermanfaat lainnya (Nurhajjah, 2022).



Gambar 13. Penggerek Batang Tebu  
(Subiyakto, 2017a)

### 2.2.9.3. Kutu Perisai (*Aulacaspis tegalensis* Zehntner)

Kutu perisai (*Aulacaspis tegalensis* Zehntner) merupakan serangga berukuran sangat kecil, di mana Panjang tubuh dewasanya hanya berukuran 1,04 mm. Untuk melindungi dirinya, kutu jenis ini tinggal di dalam sebuah perisai sepanjang 2,90 mm yang dibentuk dari campuran lilin batang tebu dan zat yang dikeluarkannya sendiri. Terdapat perbedaan dalam siklus hidupnya, yaitu kutu betina hanya mengalami dua kali pergantian kulit, sementara kutu jantan mengalaminya sebanyak empat kali (Sunaryo & Hasibuan, 2003). Dalam kurun waktu beberapa dekade terakhir, terjadi peningkatan serangan kutu perisai secara signifikan di area perkebunan tebu di Pulau Sumatera (Pramono et al., 2020).



Gambar 14. Serangan Kutu Perisai  
(Girsang & Siagian, 2023)

Kutu perisai cenderung menyerang tanaman yang telah berumur lebih dari satu tahun. Gejala awal serangannya dapat dideteksi dari kehadiran semut yang mengerubungi tanaman, Serangan ini menyebabkan kerusakan bertahap, mulai dari daun yang menguning, layu, mengering, hingga akhirnya menyebabkan kematian tanaman. Untuk pengendaliannya, dapat dilakukan beberapa cara tergantung tingkat populasi: jika masih rendah, kutu dapat dimatikan secara mekanis dengan menekannya. Metode lain adalah dengan memangkas daun yang sudah terinfeksi. Namun, jika populasi sudah tinggi, dianjurkan menggunakan insektisida dengan bahan aktif streptomisin sulfat (Faj'r et al., 2018). Namun, pengendalian kutu

perisai pada tanaman tebu sulit dilakukan karena lokasinya yang tersembunyi seperti pada Gambar 16. Serangga ini melekat pada batang di bawah pelepah daun, di mana pelepah yang tidak dibersihkan justru menjadi tempat ideal bagi mereka untuk berkembang biak (Pramono et al., 2020).

### 2.2.10. Confusion Matrix

*Confusion matrix*, atau matriks kebingungan adalah metode paling sederhana untuk menilai performa suatu algoritma klasifikasi dengan membandingkan data positif yang diprediksi benar atau salah serta data negatif yang diprediksi benar atau salah. Terdapat berbagai pandangan mengenai penggunaan matriks ini, namun pada dasarnya *confusion matrix* memiliki peran penting dalam mengevaluasi efektivitas algoritma klasifikasi. Sebagaimana yang terlihat pada Gambar 17, dalam penyajiannya, baris pada matriks menunjukkan label aktual, sedangkan kolom mempresentasikan hasil prediksi dari algoritma klasifikasi (Lonang et al., 2023; Tjahjadi et al., 2023).

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) <i>Type I Error</i>
	0 (Negative)	<b>FN</b> (False Negative) <i>Type II Error</i>	<b>TN</b> (True Negative)

Gambar 15. Tabel *Confusion Matrix*

(Nugroho, 2019)

*Confusion Matrix* terdiri atas beberapa komponen utama yang digunakan untuk menilai kinerja suatu model klasifikasi. *True Positive* (TP) terjadi ketika model berhasil mengenali data sebagai kelas positif dengan benar. *True Negative* (TN)

menunjukkan keberhasilan model dalam memprediksi data yang memang berasal dari kelas negatif. Sebaliknya, *False Positive* (FP) muncul saat model salah mengklasifikasikan data ke dalam kelas positif padahal sebenarnya negatif. Adapun, *False Negative* (FN) terjadi ketika model keliru memprediksi data positif sebagai negatif. Keseluruhan elemen ini bersama-sama memberikan gambaran menyeluruh mengenai kemampuan model dalam menangani data uji (Ridwan et al., 2025).

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (12)$$

$$Presisi = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

Di mana, akurasi pada Persamaan 12 adalah rasio prediksi benar dengan keseluruhan data. Presisi pada Persamaan 13 merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang diprediksi positif. Terakhir, *Recall* pada Persamaan 14 merupakan rasio prediksi benar positif yang dibandingkan dengan keseluruhan data aktual positif (Amaliah et al., 2022). Terakhir, *F1-Score* pada Persamaan 15 merupakan rata-rata harmonik dari *precision* dan *recall*, di mana nilai 1 menunjukkan performa terbaik, sedangkan nilai 0 menunjukkan performa yang buruk (Hutagalung & Sitompul, 2023).

### 2.2.11. ROC AUC

Kurva ROC (*Receiver Operating Characteristic*) adalah metode analisis visual yang berfungsi untuk mengevaluasi efektivitas dan performa dari sebuah klasifikasi biner dalam pengujian diagnostik (Nahm, 2022). Komunitas *machine learning* mulai beralih menggunakan ROC *curve* karena metrik akurasi biasa dianggap kurang representatif dalam menggambarkan performa model seutuhnya. Keunggulan utama analisis ROC terletak pada kemampuannya menangani data

yang tidak seimbang (*imbalanced data*) serta mempertimbangkan konsekuensi biaya yang berbeda dari setiap kesalahan prediksi. Dalam membandingkan berbagai model klasifikasi, menyederhanakan ROC *curve* menjadi satu nilai tunggal sangatlah membantu untuk melihat gambaran umum performanya. Metode yang paling sering digunakan adalah menghitung luas di bawah kurva tersebut, atau yang dikenal sebagai AUC (*Area Under Curve*). Saat ini, AUC sudah menjadi standar utama dalam mengevaluasi dan membandingkan kualitas antar model (Gajowniczek & Ząbkowski, 2021).

$$TPR = \frac{TP}{TP+FN} \quad (16)$$

$$FPR = \frac{FP}{FP+TN} \quad (17)$$

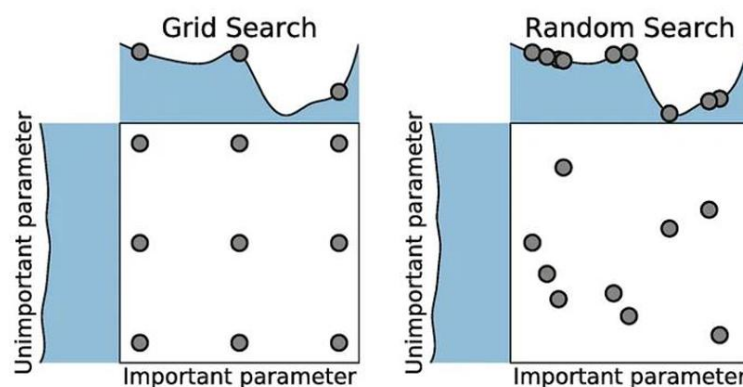
ROC-AUC berfokus pada kemampuan model dalam membedakan antara kelas positif dan negatif dengan memperhatikan *trade-off* antara tingkat *True Positive Rate* (TPR) dan tingkat *False Positive Rate* (FPR) seperti pada Persamaan 16 dan Persamaan 17. TPR (*True Positive Rate*) yang juga dikenal sebagai *sensitivity* atau *recall* berfungsi untuk mengukur sejauh mana model mampu mengidentifikasi kasus positif dengan benar. Sedangkan, FPR (*False Positive Rate*) digunakan untuk mengukur sejauh mana model memprediksi kasus kasus negative sebagai positif (Monaghan et al., 2021).

### 2.2.12. Hyperparameter Tuning

Dalam pengembangan model *machine learning*, *hyperparameter* berperan sebagai konfigurasi eksternal yang mengatur bagaimana sebuah algoritma belajar. Parameter-parameter ini harus ditetapkan sebelum proses dimulai karena berfungsi sebagai panduan utama bagi model, baik pada fase *training* maupun fase *testing* (Rom et al., 2024). Oleh karena itu, *hyperparameter tuning* merupakan aspek penting dalam penerapan algoritma *machine learning*. Dengan mengidentifikasi konfigurasi parameter yang paling tepat, efektivitas dan akurasi model yang

dihasilkan dapat ditingkatkan secara signifikan. Hingga saat ini, teknik *grid search* dan *random search* masih menjadi dua metode yang paling sering diandalkan untuk menyempurnakan berbagai model *machine learning* (Raji et al., 2022).

Dalam *hyperparameter tuning*, *grid search* masih dianggap sebagai metode standar utama yang digunakan sebagai titik acuan atau pembanding dalam strategi optimasi (Yarally et al., 2023). Seperti yang ditunjukkan pada Gambar 18, *grid search* bekerja dengan cara menguji seluruh kombinasi ruang parameter yang tersedia untuk menjamin temuan solusi dengan tingkat akurasi tertinggi (Ogunsanya et al., 2023). Namun, kelemahan utama dari pendekatan ini adalah kerentanannya terhadap masalah *curse of dimensionality*, yaitu beban komputasi yang meningkat secara drastic seiring bertambahnya jumlah parameter yang dioptimalkan (Yu & Zhu, 2020).



Gambar 16. Perbedaan *Grid Search* dan *Random Search*

(Rom et al., 2024)

Teknik *random search* hadir sebagai solusi menjanjikan untuk mengatasi hambatan yang ada pada *grid search*. Berbeda dengan metode pendahulunya, *random search* bekerja dengan cara memilih nilai *hyperparameter* secara acak dalam ruang pencarian tertentu. Sebagaimana ditunjukkan pada Gambar 18, keunggulan metode ini terletak pada mekanisme kerjanya yang sederhana dan implementasi yang relatif mudah. Kelebihan lain dari *random search* adalah kemampuannya dalam memberikan hasil yang lebih optimal melalui eksplorasi ruang konfigurasi yang lebih luas dan bervariasi. Selain dapat dijalankan secara otomatis, teknik ini bergantung pada informasi gradien ataupun pengetahuan spesifik mengenai

karakteristik ruang optimasinya. Karakteristik tersebut menjadikan *random search* sangat efektif dalam menangani persoalan dengan tingkat kompleksitas tinggi yang sulit dipetakan (Rom et al., 2024). Oleh karena itu, penelitian ini menerapkan *random search* sebagai metode utama dalam proses *hyperparameter tuning*.

### 2.2.13. Multi-Label Classification

*Multi-Label Classification* (MLC) merupakan salah satu paradigma *machine learning* yang paling sering diteliti, terutama karena kemampuannya untuk menyesuaikan diri dengan beragam permasalahan dunia nyata (Lv et al., 2021). *Multi-Label Classification* (MLC) adalah bentuk lanjutan dari klasifikasi Tunggal yang memungkinkan satu data memiliki lebih dari satu label sekaligus (Tarekegn et al., 2021). *Single-label classification* membatasi setiap data pada satu kelas, berbeda dengan *multi-label classification* yang memungkinkan satu *instance* memiliki lebih satu label sekaligus. Dengan demikian, *single-label classification* dapat dipandang sebagai bentuk khusus dari *multi-label classification*, sedangkan *multi-label classification* merupakan pendekatan yang lebih umum dan menyeluruh dibandingkan *single-label classification* (Saidabad et al., 2024).

Salah satu pendekatan yang sering digunakan dalam *multi-label classification* adalah membangun *classifier* individu untuk tiap kategori, kemudian menentukan hasil melalui teknik *ranking* atau *threshold*. Meskipun efektif, metode ini hanya cocok ketika antar-kategori tidak saling bergantung, sebab tidak mempertimbangkan korelasi antar-label. Faktanya, pada banyak bidang, label justru memiliki keterkaitan yang kuat. Oleh karena itu, *multi-label classification* banyak dimanfaatkan di berbagai *domain*, mulai dari pengolahan teks, citra, media digital lainnya, hingga bioinformatika (Saidabad et al., 2024).

### **III. METODE PENELITIAN**

#### **3.1. Tempat dan Waktu Penelitian**

##### **3.1.1. Tempat Penelitian**

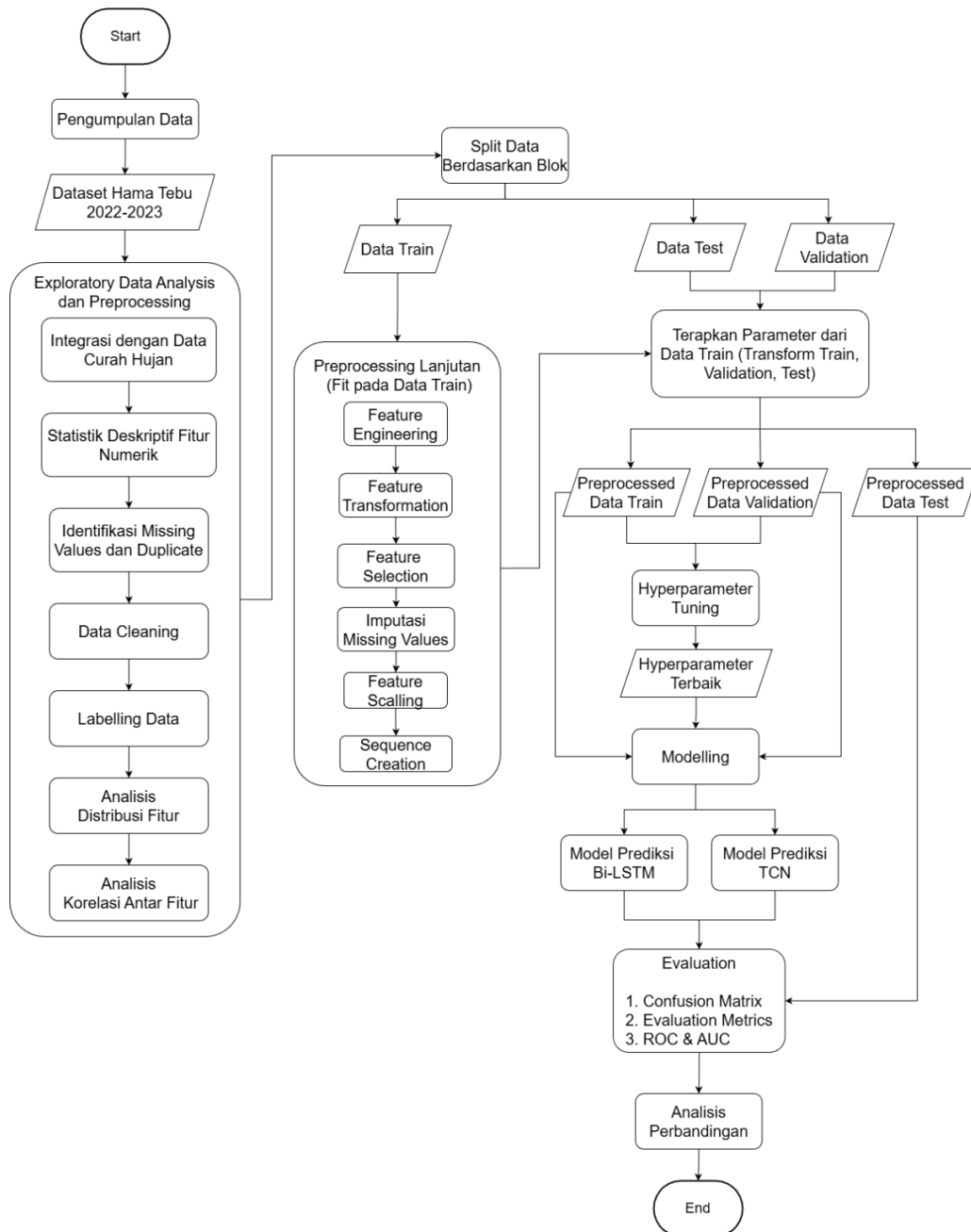
Penelitian ini dilaksanakan di Laboratorium Komputasi Dasar, Program Studi Ilmu Komputer, Universitas Lampung. Pemilihan lokasi penelitian didasarkan pada ketersediaan fasilitas yang memadai untuk mendukung jalannya penelitian, meliputi perangkat komputer dengan spesifikasi yang sesuai, perangkat lunak analisis data, serta jaringan internet untuk menunjang kebutuhan literatur maupun pengolahan informasi. Selain itu, Laboratorium Komputasi Dasar juga dipilih karena memiliki lingkungan akademik yang kondusif, sehingga dapat mendukung dalam menjalankan aktivitas penelitian secara terarah, sistematis, dan sesuai dengan tujuan yang telah ditetapkan.

##### **3.1.2. Waktu Penelitian**

Pelaksanaan penelitian direncanakan mulai dari bulan Agustus 2025 sampai dengan Januari 2026. Rentang waktu tersebut mencakup tahap persiapan, pengumpulan data, analisis, serta penyusunan laporan penelitian. Jadwal pelaksanaan penelitian secara rinci dapat dilihat pada Lampiran 1.

### 3.2. Tahapan Penelitian

Untuk memberikan gambaran yang komprehensif mengenai kerangka kerja yang digunakan, penelitian ini dilaksanakan melalui serangkaian langkah sistematis. Tahapan-tahapan tersebut, mulai dari pengumpulan data hingga evaluasi model, divisualisasikan dalam *flowchart* pada Gambar 19 berikut.



Gambar 17. *Flowchart* Metode Penelitian

### 3.2.1. Pengumpulan *Dataset*

*Dataset* yang digunakan dalam penelitian ini memiliki 9.954 baris dengan 26 kolom, yang didalamnya mencakup informasi identitas lahan, waktu pengamatan, karakteristik penanaman, hingga tingkat serangan hama dan perlakuan pengendalian. Pada Tabel 2 dijabarkan atribut-atribut yang dimiliki oleh *dataset* yang digunakan dalam penelitian.

Tabel 2. *Dataset Features*

No.	Atribut	Keterangan
1.	<i>Unique Code</i>	Kode unik untuk tanaman tebu yang sedang diamati. masing-masing kode terdiri dari nomor divisi, tahun pengamatan, varietas, bulan pengamatan, dan angka acak.
2.	Divisi	Nama divisi ditandai dengan angka romawi dari I–VII.
3.	Tahun Pengamatan	Pengamatan tebu dimulai pada pertengahan tahun 2022 hingga pertengahan tahun 2023.
4.	Bulan Pengamatan	Pengamatan tebu dilakukan selama 12 bulan penuh yang diawali pada bulan Agustus. Kolom ini berisi nilai dengan tipe data integer 1-12.
5.	Kawasan	Nama kawasan ditandai dengan angka biasa dari 1-31, yang membuat kolom ini memiliki tipe data <i>float</i> .
6.	Varietas Tebu	Tebu yang diamati terdiri dari 15 varietas di antaranya G11, GMP 3, GMP 5, GMP 6, GMP 7, R02, R06, R08, R2, R34, R54, R69, R7, R86, dan Lain-lain
7.	Bulan Tanam2	Berisi nilai dengan tipe data integer dari 5 – 11 di mana masa penanaman tebu di mulai

No.	Atribut	Keterangan
		pada bulan Mei
8.	Nama Blok	Terdiri dari 1.230 nilai unik kombinasi huruf dan angka.
9.	Tanggal Pengamatan	Berisi tanggal pengamatan tebu dalam format MM/DD/YYYY.
10.	Luas Tanah Produktif (Ha)	Berisi nilai dengan tipe data float yang merepresentasikan luas lahan perkebunan dalam satuan hektar.
11.	Bulan Tanam	Berisi tanggal penanaman tebu dalam format MM/DD/YYYY.
12.	Tahun Tanam	Tahun ketika tebu ditanam hanya tersedia di tahun 2022.
13.	<i>Category</i> Tanam	Berisi 5 kategori tebu yang dibedakan berdasarkan masa panen. PC mewakili tebu yang ditanam dengan bibit baru. RC mewakili tebu yang sudah di panen dan dibiarkan tumbuh lagi setelah penebangan. RC sendiri terdiri dari RC1 – RC5, yang menandai intensitas penebangan yang dilakukan pada tebu tersebut.
14.	<i>Category</i> 2	Data Duplikat dari <i>Category</i> Tanam
15.	Umur	Terdiri dari usia tebu ketika dilakukan pengamatan, yaitu mulai usia 3 – 11 bulan.
16.	Ruas Terserang <i>Stem Borer</i> (%)	Persentase ruas batang tebu yang terserang hama <i>stem borer</i> . Nilai berupa angka desimal dengan tipe data <i>float</i> .
17.	Batang Terserang <i>Top Borer</i> (%)	Persentase batang tebu yang mengalami serangan hama <i>top borer</i> . Data disajikan dalam bentuk <i>float</i> .
18.	Populasi Kutu Perisai (%)	Persentase populasi kutu perisai yang menyerang tanaman tebu. Nilai berbentuk

No.	Atribut	Keterangan
		desimal dengan tipe data <i>float</i> .
19.	Dosis pias	Jumlah pelepasan atau aplikasi agen pengendali hayati pias yang diberikan, dengan tipe data bilangan bulat ( <i>integer</i> ).
20.	Dosis lalat	Banyaknya pelepasan lalat sebagai agen pengendali, dicatat dalam satuan bilangan bulat ( <i>integer</i> ).
21.	Dosis Telenomus	Jumlah penggunaan parasitoid <i>Telenomus</i> untuk pengendalian hama, dinyatakan dalam <i>integer</i> .
22.	Dosis Tetras	Banyaknya pelepasan parasitoid <i>Tetrastichus</i> pada tanaman tebu. Data bertipe <i>integer</i> .
23.	Dosis cecopet	Takaran atau dosis penggunaan cecopet (agen hayati tertentu), dicatat dalam nilai desimal dengan tipe data <i>float</i> .
24.	Curah Hujan	Intensitas hujan setiap bulan selama Agustus 2022 hingga Juli 2023.

### 3.2.2. *Exploratory Data Analysis*

Sebelum melakukan tahap pemodelan inti, dilakukan tahap eksplorasi terhadap data untuk memahami karakteristik, distribusi, serta pola tersembunyi di dalam *dataset*. Tahapan ini penting untuk memastikan kualitas data dan mengidentifikasi potensi keberadaan anomali yang dapat mempengaruhi performa model. Adapun langkah-langkah eksplorasi yang dilakukan meliputi:

#### a. Statistik Deskriptif

Analisis statistik deskriptif dilakukan untuk mendapatkan gambaran umum serta ringkasan kuantitatif dari *dataset*. Hal ini mencakup perhitungan nilai rata-rata (mean), median, standar deviasi, hingga nilai minimum dan maksimum yang memberikan landasan awal dalam memahami variabilitas data sebelum

dilakukan tahap *preprocessing data*.

b. Identifikasi duplikasi data dan *missing values*

Identifikasi data berfungsi untuk memastikan integritas *dataset* dengan menemukan baris-baris yang identik secara keseluruhan. Selain itu, dilakukan juga identifikasi *missing values* yang dilakukan untuk mendeteksi adanya data yang hilang atau kosong (NaN).

c. Analisis Distribusi Fitur

Analisis korelasi menggunakan *heatmap* menunjukkan hubungan yang beragam antara variabel numerik dalam *dataset*. Korelasi antar jenis serangan hama menunjukkan hubungan yang positif moderat, dengan serangan penggerek pucuk memiliki korelasi cukup kuat mengindikasikan kedua hama ini muncul bersamaan, sementara kutu perisai menunjukkan korelasi lemah.

d. Analisis Korelasi Antarfitur

Tahap ini dilakukan untuk memvalidasi urutan waktu dan memastikan tidak terdapat anomali pada alur temporal *dataset*. Hasilnya, terdapat blok yang memiliki observasi tidak lengkap dan memiliki data observasi yang hilang. Terdapat juga pola yang menunjukkan peningkatan serangan hama seiring bertambahnya umur tebu.

### 3.2.3. *Preprocessing*

Setelah melalui tahap eksplorasi, data kemudian memasuki tahap *preprocessing* di mana *dataset* dibersihkan dan ditransformasikan ke dalam format representasi vektor yang sesuai dengan kebutuhan input algoritma pembelajaran mesin. Tujuan utama dari proses ini adalah untuk meningkatkan integritas data dan mematangkan informasi agar proses pelatihan (*training*) model dapat berjalan lebih efisien dan akurat. Langkah *preprocessing* yang diimplementasikan dalam penelitian ini meliputi:

a. Integrasi Data Curah Hujan

Mengintegrasikan atau menggabungkan hama tebu periode 2022-2023 Perusahaan Gula A dengan data curah hujan yang ada di Lampung Tengah. Integrasi ini dapat memperkaya konteks lingkungan yang ada, karena curah hujan dapat mempengaruhi banyak sedikitnya populasi hama. Hasil penggabungan memastikan tiap observasi mempunyai informasi cuaca berdasarkan periode waktunya. Kode program integrasi data curah hujan dapat dilihat pada Lampiran 2.

b. *Data Cleaning*

Proses *data cleaning* dimulai dengan *rename* kolom agar lebih konsisten dan mudah dipahami, dilanjutkan dengan penghapusan kolom-kolom yang tidak relevan. Data varietas dengan nilai 'lain-lain' dihapus karena kurang spesifik untuk analisis, sementara *missing values* pada kolom kawasan ditangani melalui imputasi dengan menggunakan nilai yang paling sering muncul (modus) dalam kelompok blok yang sama, termasuk perbaikan nilai anomali pada blok tertentu yang memiliki nilai Kawasan yang tidak sesuai. Duplikasi data yang ada dihapus berdasarkan kombinasi antara nama blok dan umur tanaman, kemudian dilakukan *filtering* untuk hanya memakai blok yang diamati minimal umur sembilan bulan. Data observasi yang hanya diamati sampai umur sembilan atau sepuluh bulan akan ditambahkan hingga umur sebelas bulan dengan teknik *forward fill* yang diambil dari pengamatan terakhir yang tersedia.

c. *Labelling Data*

Proses pelabelan data dilakukan untuk mengkategorikan tingkat serangan hama. Setiap jenis hama memiliki *threshold* yang disesuaikan dengan umur tanaman dalam bulan. Untuk masing-masing hama, diberikan label kategori hijau untuk tingkat serangan rendah, merah untuk tingkat serangan sedang, dan hitam untuk tingkat serangan parah. Pelabelan dilakukan pada tiga jenis hama yang diteliti, yaitu penggerek batang, penggerek pucuk dan kutu perisai, dengan nilai yang meningkat seiring dengan bertambahnya umur yang mencerminkan toleransi serangan yang berbeda pada tiap fase pertumbuhan tebu, sebagaimana pada Tabel 3 dan Tabel 4.

Tabel 3. Aturan Label Serangan Penggerek Batang dan Penggerek Pucuk

Kelas	Umur Tanaman									
	1	2	3	4	5	6	7	8	9	10
<b>Tinggi</b>	> 5.5	> 6.0	> 6.5	> 7.0	> 7.5	> 8.0	> 8.5	> 9.0	> 9.5	> 10
<b>Sedang</b>	0.5 - 5.5	1.0 - 6	1.5 - 6.5	2.0 - 7.0	2.5 - 7.5	3.0 - 8.0	3.5 - 8.5	4.0 - 9.0	4.5 - 9.5	5.0 - 10
<b>Rendah</b>	< 0.5	< 1.0	< 1.5	< 2.0	< 2.5	< 3.0	< 3.5	< 4.0	< 4.5	< 5.0

Tabel 4. Aturan Label Serangan Kutu Perisai

Kelas	Umur Tanaman									
	1	2	3	4	5	6	7	8	9	10
<b>Tinggi</b>	> 10	> 10	> 10	> 10	> 10	> 10.5	> 11.0	> 15	> 20	> 40
<b>Sedang</b>	0.1 - 10.0	0.1 - 10.0	0.1 - 10.0	0.1 - 10.0	0.1 - 10.0	0.5 - 10.5	1.0 - 11.0	5.0 - 15.0	10.0 - 20.0	20.0 - 40.0
<b>Rendah</b>	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.5	< 1	< 5	< 10	< 20

### 3.2.4. *Split Data Berdasarkan Blok*

*Split Data Berdasarkan Blok* adalah metode pembagian *dataset* yang memisahkan data berdasarkan blok, bukan secara acak per baris. Pendekatan ini sangat penting untuk menghindari kebocoran informasi (*data leakage*) dalam prediksi *time-series*, karena pengamatan dari blok yang sama cenderung memiliki keterkaitan waktu dan lokasi yang kuat. Jika data dari blok yang sama tersebar ke set pelatihan dan pengujian, model bisa 'menghafal' pola spesifik dari blok tersebut, yang mengakibatkan performa evaluasi yang terlalu optimis serta tidak mencerminkan kemampuan prediksi modelnya. Kode program *splitting data* dapat dilihat pada Lampiran 3. Langkah-langkah pembagian datanya adalah sebagai berikut.

#### a. Identifikasi Unit Blok

Sistem mengidentifikasi dan mengekstrak semua blok lahan unik yang ada dalam *dataset*. Setiap blok merepresentasikan satu area lahan produktif yang memiliki karakteristik dan riwayat pengamatan sendiri. Proses ini penting untuk memastikan bahwa setiap blok diperlukan sebagai satu kesatuan yang utuh dan tidak akan dipecah ke berbagai set yang berbeda.

b. Pengacakan Blok

Daftar blok-blok yang telah diidentifikasi tersebut kemudian diacak secara random untuk memastikan distribusi yang tidak bias dan representatif. Pengacakan ini menggunakan nilai *seed* yang tetap agar hasil pembagian dapat direproduksi dengan konsisten di setiap eksekusi program. Langkah ini dapat memastikan bahwa tidak ada pola sistematis atau urutan tertentu yang mempengaruhi pemilihan blok untuk setiap set data.

c. Penentuan Proporsi dan Alokasi Blok

Proporsi pembagian data mengikuti rasio standar *machine learning* yaitu 70% blok-blok urutan awal untuk set pelatihan, 15% blok-blok selanjutnya untuk set validasi, dan 15% blok-blok urutan terakhir untuk set pengujian. Selain itu, dilakukan juga 2 skenario lain dengan rasio 60:20:20 dan 80:10:10. Dengan cara ini, setiap blok hanya akan masuk ke satu set data saja untuk memastikan tidak ada *overlap* atau tumpang tindih antar set. Distribusi pembagian data berdasarkan target dapat dilihat pada Lampiran 4, 5, dan 6.

d. Pemisahan Data Berdasarkan Blok

Setelah blok-blok dialokasikan ke masing-masing set, sistem kemudian mengumpulkan seluruh barisan data yang berasal dari blok-blok di set pelatihan dan menggabungkannya menjadi satu *dataset* pelatihan yang utuh. Proses yang sama dilakukan untuk set validasi dan set pengujian. Dengan proses ini, nantinya semua observasi historis dari satu blok akan berada dalam satu set yang sama untuk menjaga integritas temporal dan mencegah kebocoran informasi antar set.

### 3.2.5. *Preprocessing* Lanjutan

*Preprocessing* lanjutan baru diterapkan setelah data dibagi menjadi data latih uji, dan tes pada tahap *splitting* berdasarkan blok. Proses ini bertujuan untuk memastikan fitur-fitur data berada dalam kondisi optimal saat diproses oleh model *machine learning*. Proses-proses ini dilakukan dengan prinsip *fit on training data*, di mana hanya diterapkan atau dihitung hanya dari *data train* dan kemudian akan

diterakan secara konsisten pada *data validation* dan *data test*. Hal ini dilakukan untuk mencegah *data leakage* dan untuk memastikan bahwa informasi statistik dari *data valid* dan *data test* tidak bocor ke *preprocessing training*. Langkah-langkah *preprocessing* lanjutan yang dilakukan adalah sebagai berikut.

a. *Feature Engineering*

Untuk meningkatkan sensitivitas model terhadap pola temporal dan hubungan non-linear dalam data serangan hama, dilakukan tahapan *feature engineering* sebagai bagian krusial dari tahap *preprocessing*. Melalui pendekatan yang sistematis, data mentah diproses menjadi serangkaian fitur turunan yang dirancang khusus untuk merepresentasikan karakteristik temporal dan dinamika fluktuasi serangan hama secara lebih akurat.

b. *Feature Transformation*

Setelah *feature engineering*, dilakukan tahap *feature transformation* untuk mengkonversi fitur kategorikal menjadi representasi numerik yang dapat diolah oleh model *machine learning*. Variabel kategorikal seperti 'Divisi' dan 'Kawasan' ditransformasi menggunakan *label encoding* yang memetakan data ke dalam nilai *integer* berurutan. Sementara itu, untuk variabel dengan kardinalitas tinggi seperti 'Category' dan 'Varietas Tebu' akan diterapkan *One-Hot Encoding* yang menghasilkan variabel biner untuk setiap kategori unik.

c. *Feature Selection*

Untuk memastikan model hanya memproses fitur yang relevan dan bermakna, dilakukan tahapan *feature selection* sebagai bagian krusial dari *preprocessing pipeline*. Melalui pendekatan berbasis karakteristik dan pola serangan spesifik setiap hama, fitur-fitur dipilih secara ketat berdasarkan relevansi dan keterkaitan langsung terhadap masing-masing target prediksi.

d. *Imputation Missing Values*

Setelah melalui tahap *feature selection*, nilai yang hilang pada saat pembuatan *lag features*, *rolling statistics*, dan *delta* akan diimputasi. Misalnya, pada pengamatan bulan pertama untuk suatu blok, fitur *lag\_1m*, *lag\_2m*, dan *lag\_3m* akan memiliki *missing values* karena belum ada observasi sebelumnya, begitu

juga dengan *rolling statistics* yang memerlukan minimal 3 bulan data untuk dihitung 'min\_periods=window'. Untuk menangani *missing values*, digunakan SimpleImputer dengan strategi *median* yang menggantikan setiap nilai yang hilang dengan rata-rata dari fitur yang sama sama pada *training set*. Pemilihan ini didasarkan pada *median* yang memiliki keunggulan dalam menangani *outliers* dan distribusi data yang tidak simetris, yang umum terjadi pada data serangan hama di mana terdapat variasi ekstrem antara serangab rendah dan tinggi. Berbeda dengan *mean* yang sensitif terhadap nilai ekstrem, *median* memberikan representasi yang lebih *robust* terhadap nilai *outlier* dan lebih stabil untuk data dengan distribusi *skewed*, sehingga imputasi tidak akan terdistorsi oleh observasi serangan hama yang sangat tinggi atau anomali pada beberapa blok tertentu.

e. *Feature Scaling*

Mengingat fitur-fitur dalam *dataset* memiliki skala dan rentang nilai yang sangat bervariasi, mulai dari dosis pengendalian biologis dalam satuan kecil hingga curah hujan dalam ratusan milimeter, maka dilakukan tahap *feature scaling* untuk menormalkan seluruh fitur numerik ke dalam skala sebanding. Teknik *StandardScaler* dari *library scikit-learn* diterapkan, yang mentransformasikan setiap fitur sehingga memiliki rata-rata (*mean*) nol dan standar deviasi satu.

f. *Sequence Creation*

Tahap akhir dari *preprocessing* lanjutan adalah *sequence creation*, yang merupakan proses penting untuk mentransformasi data tabular menjadi format *time series* yang sesuai untuk *input* model *deep learning*. Proses ini dilakukan dengan membuat *sliding window* sepanjang *max\_lookback* (3 bulan) untuk setiap blok lahan, di mana setiap sekuens *input* terdiri dari observasi 3 bulan terakhir yang digunakan untuk memprediksi kondisi *forecast\_horizon* (1 bulan ke depan). Jika panjang *history* kurang dari *max\_lookback* (seperti pada bulan awal pengamatan blok), Bi-LSTM menggunakan nilai -999 yang kemudian di mask oleh *Masking Layer* di dalam model sehingga tidak mempengaruhi perhitungan gradien, sedangkan untuk TCN akan menggunakan *Zero Padding*

(0.0) karena TCN tidak memerlukan mekanisme *masking*. Proses ini dilakukan per blok secara independen untuk menjaga integritas temporal, menghasilkan *dataset* sekuens dengan *shape* (jumlah\_sampel, max\_lookback, jumlah\_fitur) yang siap digunakan sebagai *input* untuk model Bi-LSTM dan TCN, sambil mempertahankan label target yang sesuai untuk *supervised learning*.

### 3.2.6. Hyperparameter Tuning

Pencarian hyperparameter dilakukan dengan strategi *random search* untuk menemukan konfigurasi optimal. Pencarian dilakukan dengan jumlah trial maksimum, yang mana setiap *trial* mengevaluasi satu kombinasi *hyperparameter* berdasarkan akurasi validasi. Kombinasi yang terbaik akan dipilih untuk pelatihan final. Proses ini dilakukan terpisah untuk setiap target prediksi (*stem borer*, *top borer*, dan kutu perisai) karena karakteristik dari masing-masing label berbeda yang menghasilkan total 6 proses *tuning* (3 target x 2 model) dengan konfigurasi yang dijelaskan pada Tabel 5 dan 6.

Tabel 5. *Hyperparameter Search Space* untuk Model Bi-LSTM

<i>Hyperparameter</i>	<i>Range/Nilai</i>	<i>Step</i>	<i>Keterangan</i>
<code>lstm_units_1</code>	32 – 128	32	Jumlah unit LSTM <i>layer</i> pertama
<code>lstm_units_2</code>	16 – 64	16	Jumlah unit LSTM <i>layer</i> kedua
<code>dense_units</code>	16 – 64	16	Jumlah unit pada <i>Dense layer</i>
<code>dropout_rate</code>	0.2 – 0.5	0.1	Tingkat <i>dropout</i> untuk regularisasi
<code>l2_weight</code>	0.001 – 0.005	0.001	Bobot L2 <i>regularization</i>
<code>learning_rate</code>	[0.0001, 0.0003, 0.0005, 0.001	-	<i>Learning rate optimizer</i> Adam
<code>focal_gamma</code>	2.0 – 4.0	0.5	Parameter gamma untuk <i>Focal Loss</i>

Tabel 6. *Hyperparameter Search Space* untuk Model TCN

<i>Hyperparameter</i>	<i>Range/Nilai</i>	<i>Step</i>	<i>Keterangan</i>
<i>nb_filters_1</i>	32 – 128	32	Jumlah filter TCN layier pertama
<i>nb_filters_2</i>	16 – 64	16	Jumlah filter TCN <i>layer</i> kedua
<i>kernel_size</i>	2 – 4	1	Ukuran <i>kernel</i> konvolusi
<i>dense_units</i>	16 – 64	16	Jumlah unit pada <i>Dense layer</i>
<i>dropout_rate</i>	0.2 – 0.5	0.1	Tingkat <i>dropout</i> untuk regularisasi
<i>l2_weight</i>	0.001 – 0.005	0.001	Bobot L2 <i>regularization</i>
<i>Learning_rate</i>	[0.0001, 0.0003, 0.0005, 0.001	-	<i>Learning rate optimizer</i> Adam
<i>focal_gamma</i>	2.0 – 4.0	0.5	Parameter gamma <i>Focal Loss</i>

Konfigurasi *tuning* dirancang untuk memberikan keseimbangan optimal antara eksplorasi ruang pencarian *hyperparameter* dan efisiensi waktu komputasi. Setelah melakukan beberapa eksperimen, dipilih konfigurasi dengan 10 *trials* dan 50 *tuning epochs* yang memberikan hasil yang stabil untuk setiap kombinasi *hyperparameter* yang diuji. Konfigurasi *tuning parameters* dijelaskan seperti pada Tabel 7.

Tabel 7. Konfigurasi *Hyperparameter Tuning*

<i>Hyperparameter</i>	<i>Nilai</i>	<i>Keterangan</i>
<i>max_trials</i>	10	Jumlah maksimal kombinasi <i>hyperparameter</i> yang diuji
<i>executions_per_trial</i>	1	Jumlah unit LSTM <i>layer</i> kedua
<i>tuning_epochs</i>	50	Jumlah <i>epoch training</i> selama <i>tuning</i>
<i>objective</i>	'val_accuracy'	Metrik yang dioptimalkan
<i>overwrite</i>	True	Menimpa hasil <i>tuning</i> sebelumnya

### 3.2.7. Modelling

Pada tahap ini, data latih yang telah didapatkan dari proses *splitting* sebelumnya akan digunakan untuk membangun model prediksi menggunakan dua algoritma berbasis *Deep Learning*, yaitu *Bidirectional Long Short-Term Memory* (Bi-LSTM) dan *Temporal Convolutional Network* (TCN). Penelitian ini berfokus pada permasalahan *multi-class classification*, di mana model ditugaskan untuk mengklasifikasikan jenis serangan hama tebu yang terdiri dari tiga kelas utama, yaitu penggerek batang, penggerek pucuk, dan kutu perisai. Tahapan *modelling*-nya adalah sebagai berikut.

#### a. Pembangunan Arsitektur Model

Penelitian ini menggunakan dua arsitektur *deep learning* untuk prediksi Tingkat serangan hama. Arsitektur pertama menggunakan *Bidirectional Long Short-Term Memory* (Bi-LSTM), yang menerima *input* berupa sekuens dengan *shape* (*max\_lookback*, *n\_fitur*), di mana data telah di-*padding* dengan nilai -999.0 untuk observasi yang kurang dari 3 bulan. Struktur Bi-LSTM terdiri dari dua *layer* bertumpuk, di mana *layer* pertama dengan '*return\_sequence=True*' mengembalikan sekuens lengkap untuk diproses *layer* berikutnya, sedangkan *layer* kedua '*return\_sequence=False*' hanya mengembalikan *output* terakhir. Setiap *layer* Bi-LSTM dilengkapi dengan *Batch Normalization* dan *Dropout* untuk stabilitas *training* dan pencegahan *overfitting*. Setelah *layer* Bi-LSTM, ditambahkan *Dense Layer* dengan aktivasi ReLU yang diikuti *Dropout*, dan kemudian *layer output* dengan aktivasi *softmax* untuk prediksi tiga kelas.

Arsitektur kedua menggunakan *Temporal Convolutional Network* (TCN) yang juga menerima *input* dengan dengan (*max\_lookback*, *n\_fitur*), namun menggunakan *zero padding* (0.0) untuk observasi yang kurang dari 3 bulan. TCN memanfaatkan konvolusi kausal berdilasi (*dilated causal convolution*). Struktur TCN menggunakan beberapa *layer* bertumpuk dengan dilasi [1,2,4] pada *layer* pertama dan [1,2] pada *layer* kedua. Setelah *layer* TCN, dilakukan *batch normalization*, *dense layer*, *dropout*, *output layer*, dengan *softmax* untuk prediksi *multi-class*. Dalam pembangunan model berbasis algoritma TCN dan

Bi-LSTM, terdapat sejumlah parameter penting yang dapat disesuaikan untuk meningkatkan kualitas prediksi. Parameter ini mengatur aspek-aspek seperti jumlah lapisan, ukuran *filter* atau jumlah unit tersembunyi, panjang urutan input, laju pembeajaran, serta teknik regulasi untuk mengurangi risiko dari *overfitting*. Beberapa parameter penting ini dirangkum pada Tabel 8.

Tabel 8. Parameter Model TCN dan Bi-LSTM

Algoritma	Parameter	Keterangan
Shared Parameters	dense_unit	Jumlah unit pada <i>dense layer</i> untuk transformasi non-lienar sebelum <i>output layer</i> .
	dropout_rate	Tingkat probabilitas <i>neuron</i> yang dinonaktifkan selama <i>training</i> untuk regularisasi.
	learning_rate	Tingkat pembelajaran <i>optimizer</i> Adam yang mengontrol besar <i>update weight</i> .
	focal_gamma	Parameter gamma pada <i>focal loss</i> untuk fokus pada kelas minoritas.
	max_lookback	Jumlah <i>timestep</i> historis yang digunakan sebagai input sekuens.
	forecast_horizon	Jumlah <i>timestep</i> ke depan yang diprediksi (n bulan ke depan).
	final_epochs	Jumlah maksimum <i>epoch</i> unruk pelatihan model final.
	tuning_epochs	Jumlah untuk evaluasi setiap kombinasi parameter saat <i>tuning</i> .
	max_trials	Jumlah maksimum kombinasi <i>hyperparameter</i> yang dievaluasi saat <i>tuning</i> .
	clipnorm	Nilai maksimum <i>gradient clipping</i> untuk mencegah <i>exploding gradient</i> .

Algoritma	Parameter	Keterangan
TCN	tcn_filters_1	Jumlah filter konvolusi pada <i>layer</i> TCN pertama.
	tcn_filters_2	Jumlah filter konvolusi pada <i>layer</i> TCN kedua.
	kernel_size	Ukuran <i>kernel</i> konvolusi yang menentukan <i>receptive field</i> lokal.
	dilations (L1)	Tingkat dilatasi <i>layer</i> pertama yang meningkat secara eksponensial.
	dilations(L2)	Tingkat dilatasi <i>layer</i> kedua untuk representasi lebih abstrak.
	use_skip_connections	Koneksi residual untuk membantu aliran gradien.
	return_sequences (L1)	<i>Layer</i> pertama mengembalikan <i>output</i> pada setiap <i>timestep</i> .
return_sequences (L2)	<i>Layer</i> kedua menghasilkan representasi final untuk klasifikasi.	
Bi-LSTM	lstm_units_1	Jumlah unit LSTM pada <i>bidirectional layer</i> pertama.
	Bidirectional	Jumlah unit LSTM pada <i>bidirectional layer</i> kedua.
	return_sequences (L1)	<i>Layer</i> pertama mengembalikan <i>output</i> pada setiap <i>timestep</i> .
	return_sequences (L2)	<i>Layer</i> kedua mengembalikan <i>output</i> pada setiap <i>timestep</i> .
	kernel_regularizer	Regularisasi L2 pada <i>kernel/weight matrix</i> setiap <i>layer</i> LSTM.
	mask_value	Nilai <i>padding</i> yang diabaikan dalam perhitungan model.
	BatchNormalization	Normalisasi setelah setiap <i>layer</i> LSTM untuk stabilitas <i>training</i> .

b. Penanganan Ketidakseimbangan Kelas

*Dataset* yang digunakan memiliki distribusi kelas yang sangat tidak seimbang dengan kelas Rendah mendominasi. Untuk mengatasi masalah ini, digunakan *focal loss function* untuk mengurangi bobot dari kelas mayoritas dan meningkatkan fokus pada kelas minoritas, serta parameter *gamma* yang ada digunakan untuk mengontrol seberapa kuat penurunan bobot untuk kelas mayoritas. Selain itu, diterapkan juga *adaptive class weights/alpha*, di mana kelas Sedang diberikan bobot tambahan karena penting untuk deteksi dini. Bobot-bobot ini nantinya dinormalisasi dan diintegrasikan ke dalam *focal loss* untuk memberikan penalti lebih besar pada kesalahan prediksi kelas minoritas.

c. Pelatihan Independen untuk Tiga Target Prediksi

Setelah *hyperparameter* optimal ditemukan, model final dilatih dengan konfigurasi terbaik, di mana dilakukan independen untuk setiap target (*stem borer*, *top borer*, dan kutu perisai). Setiap target mempunyai set fitur yang disesuaikan, *hyperparameter* yang berbeda, model terpisah, serta bobot kelas yang spesifik. Hal ini memungkinkan setiap model berspesialisasi pada pola serangan hama tertentu untuk hasil yang lebih akurat.

d. Penerapan Teknik Regularisasi untuk Mencegah *Overfitting*

Berbagai teknik regularisasi akan diterapkan secara bersamaan: *Dropout* yang menonaktifkan sebagian neuron secara *random* untuk pembelajaran *robust*, *L2 Regularization* yang menambahkan penalti pada *magnitude weight*, *batch normalization* yang memberikan efek regularisasi ringan, dan *early stopping* yang mencegah pelatihan berlebihan. Kombinasi teknik-teknik ini dapat memastikan model dapat menggeneralisasi dengan baik pada data baru untuk prediksi hama yang lebih baik.

### 3.2.8. Evaluasi Model

Evaluasi model dilakukan untuk mengukur kualitas prediksi yang dihasilkan oleh kedua arsitektur *deep learning* pada setiap target hama. Karena penelitian ini

memiliki karakteristik *time-series* dan ketidakseimbangan kelas yang signifikan, proses evaluasi dirancang secara komprehensif untuk menilai performa model dari berbagai sudut pandang. Evaluasi ini dilaksanakan menggunakan *temporal holdout test set* hasil dari *split data* berdasarkan blok, sehingga model benar-benar diuji pada blok yang belum pernah dilihat selama proses pelatihan. Dengan cara seperti ini, performa model nantinya dapat mencerminkan kemampuan prediksi yang realistis pada kondisi lapangan baru. Model dievaluasi menggunakan beberapa metrik untuk memberikan pemahaman menyeluruh tentang performanya, diantaranya:

- a. *Evaluation Metrics*: Digunakan untuk menampilkan secara rinci *accuracy*, *precision*, *recall*, dan *F1-Score* untuk setiap kelas hama (Rendah, Sedang, Tinggi). Laporan ini memberikan pemahaman mendalam tentang bagaimana model berkinerja pada masing-masing tingkat serangan.
- b. *Confusion Matrix*: Berfungsi untuk memvisualisasikan kesalahan model secara detail dengan menunjukkan perbandingan antara data aktual dengan data prediksi, sehingga nantinya bisa terlihat performa model untuk prediksi.
- c. ROC-AUC: Kurva yang membantu memahami kemampuan model dalam membedakan antar kelas pada berbagai tingkat *confidence* dan *Area Under Curve* yang memberikan gambaran komprehensif tentang seberapa baik model dapat memisahkan distribusi kelas yang berbeda.

## V. SIMPULAN DAN SARAN

### 5.1. Simpulan

Berdasarkan hasil penelitian, pengujian, dan evaluasi yang telah dilakukan terhadap model *Bidirectional LSTM* dan *Temporal Convolutional Network* untuk memprediksi serangan hama *stem borer*, *top borer*, dan kutu perisai, dapat ditarik beberapa simpulan sebagai berikut:

1. Penelitian ini berhasil membangun model prediksi *multi-label* tingkat keparahan serangan hama tebu untuk tiga jenis hama, yaitu *stem borer*, *top borer*, dan kutu perisai dengan memanfaatkan arsitektur *deep learning Bidirectional LSTM* (Bi-LSTM) dan *Temporal Convolutional Network* (TCN). Pendekatan pembelajaran berbasis target hama yang independen memungkinkan setiap model menangkap pola temporal dan karakteristik serangan secara lebih spesifik untuk masing-masing jenis hama.
2. Secara keseluruhan, TCN menunjukkan performa yang sedikit lebih baik pada metrik *F1-Macro* dan akurasi, mengindikasikan kemampuan yang lebih seimbang dalam dalam klasifikasi di semua kelas. Berdasarkan nilai ROC-AUC pada berbagai skema pembagian data, TCN juga menunjukkan menunjukkan keunggulan yang lebih konsisten dibandingkan Bi-LSTM pada hampir semua skenario di seluruh rasio yang ada. Dari segi efisiensi komputasi, TCN memiliki waktu *training* dan prediksi yang jauh lebih cepat dibandingkan Bi-LSTM yang menjadikannya lebih efisien untuk implementasi sistem prediksi *real-time*. Sementara itu, Bi-LSTM menunjukkan performa yang baik dan kompetitif dengan hasil yang tidak jauh berbeda bahkan bisa lebih tinggi dari TCN pada

beberapa kasus. Kedua model ini memiliki karakteristik dan kekuatan masing-masing yang dapat dimanfaatkan sesuai kebutuhan implementasi. Dengan mempertimbangkan konsistensi performa, keunggulan pada kelas kritis, dan efisiensi komputasi yang signifikan, TCN direkomendasikan sebagai model utama untuk sistem pendukung keputusan untuk manajemen hama, sedangkan Bi-LSTM dapat digunakan sebagai model alternatif atau pelengkap pada kondisi tertentu.

3. Penggunaan *Area Under the Receiver Operating Characteristic Curve* (ROC-AUC) sebagai metrik evaluasi terbukti krusial dalam memvalidasi performa model pada *dataset* serangan hama yang memiliki ketidakseimbangan kelas (*class imbalance*). Dalam kondisi di mana kelas mayoritas sangat mendominasi, metrik akurasi konvensional cenderung bias dan memberikan estimasi performa yang terlalu optimis hanya dengan memprediksi kelas mayoritas. Sebaliknya, ROC-AUC memberikan gambaran yang lebih objektif mengenai kemampuan diskriminatif model dalam memisahkan kelas positif dan negatif. Oleh karena itu, skor ROC-AUC yang tinggi pada penelitian ini menegaskan bahwa model mampu mendeteksi kejadian serangan hama secara efektif tanpa terkecoh oleh dominasi data normal, menjadikannya indikator performa yang dapat diandalkan dibandingkan akurasi semata.

## 5.2. Saran

Berdasarkan keterbatasan yang ditemukan selama penelitian serta potensi pengembangan masa depan, diajukan beberapa saran sebagai berikut:

1. Penelitian ini menggunakan *dataset* dengan rentang waktu terbatas yaitu dari tahun 2022 sampai 2023 saja. Untuk penelitian selanjutnya, disarankan untuk memperpanjang periode pengamatan data historis (misalnya 3-5 tahun) agar model dapat menangkap pola musiman tahunan yang lebih kompleks.
2. Disarankan agar pencatatan data lapangan dilakukan secara lebih teratur dan lengkap untuk meminimalisir celah pada data deret waktu. Konsistensi dalam

pencatatan parameter kunci seperti umur tanaman dan tanggal pengamatan sangat krusial untuk mengurangi beban imputasi data. Dengan data historis yang lebih utuh, model prediksi diharapkan dapat mempelajari karakteristik temporal hama dengan lebih presisi tanpa terganggu oleh *noise* dari data buatan.

## DAFTAR PUSTAKA

- Abduljabbar, R. L., Dia, H., & Tsai, P. W. (2021). Unidirectional and bidirectional LSTM models for short-term traffic prediction. *Journal of Advanced Transportation*, 2021. <https://doi.org/10.1155/2021/5589075>
- Adrian, R., Nasamsir, & Meilin, A. (2019a). Survei Serangan Hama pada Perkebunan Tebu (*Saccharum officinarum* L.) di Provinsi Jambi. *Jurnal Media Pertanian*, 4(1), 1–7.
- Adrian, R., Nasamsir, & Meilin, A. (2019b). Survei Serangan Hama pada Perkebunan Tebu (*Saccharum officinarum* L.) di Provinsi Jambi. *Jurnal Media Pertanian*, 4(1), 1–7.
- Ahmed, S. F., Alam, M. S. Bin, Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., Mofijur, M., Shawkat Ali, A. B. M., & Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, 56(11), 13521–13617. <https://doi.org/10.1007/s10462-023-10466-8>
- Alfarizi, M. R. sirfatullah, Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman Untuk Machine Learning Dan Deep Learning. *Karimah Tauhid*, 2(1).
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Amaliah, S., Nusrang, M., & Aswi. (2022). Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa

- Bantaeng. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, 4(3), 121–127.  
<https://doi.org/10.35580/variansiunm31>
- Anggarda, M. F., Kustiawan, I., Nurjanah, D. R., & Hakim, N. F. A. (2023). Pengembangan Sistem Prediksi Waktu Penyiraman Optimal pada Perkebunan: Pendekatan Machine Learning untuk Peningkatan Produktivitas Pertanian. *Jurnal Budidaya Pertanian*, 19(2), 124–136.  
<https://doi.org/10.30598/jbdp.2023.19.2.124>
- Betha, S. K., Pallavi, L., Upadhyay, S. K., Satheesh Kumar, S., Lakshmanarao, A., & Krishna Prasad, B. (2025). Crop Leaf Disease Prediction Using Graph Diffusion TCN with Fibroblast Optimization. *Journal of Machine and Computing*, 5(3), 1736–1744.  
<https://doi.org/10.53759/7669/jmc202505137>
- Bonnu, C. H. (2025). Potensi dan Tantangan Tanaman Transgenik bagi Ketahanan Pangan Indonesia (Sebuah Tinjauan Literatur). *JIA (Jurnal Ilmiah Agribisnis): Jurnal Agribisnis Dan Ilmu Sosial Ekonomi Pertanian*, 10(2), 132–143. <https://doi.org/10.37149/jia.v10i2.1940>
- Candra, A. P. (2025). Analisis Data Menggunakan Python: Memperkenalkan Pandas dan NumPy. *Journal of Information System and Education Development*, 3(1), 11–16.
- Chen, P., Xiao, Q., Zhang, J., Xie, C., & Wang, B. (2020). Occurrence prediction of cotton pests and diseases by bidirectional long short-term memory networks with climate and atmosphere circulation. *Computers and Electronics in Agriculture*, 176.  
<https://doi.org/10.1016/j.compag.2020.105612>
- Cheraghi, Y., Kord, S., & Mashayekhizadeh, V. (2021). Application of machine learning techniques for selecting the most suitable enhanced oil recovery method; challenges and opportunities. *Journal of Petroleum Science and Engineering*, 205.  
<https://doi.org/10.1016/j.petrol.2021.108761>
- Faj'r, I. A., Hidayat, N., Sihombing, D., Studi, P., Informatika, T., Komputer, I., Brawijaya, U., Pengkajian, B., Pertanian, T., & Timur, J. (2018).

*Identifikasi Hama Dan Penyakit Pada Tanaman Sedap Malam Menggunakan Metode K-Nearest Neighbor* (Vol. 2, Number 11). <http://j-ptiik.ub.ac.id>

- Fernández, J., Chiachío, J., Barros, J., Chiachío, M., & Kulkarni, C. S. (2024). Physics-guided recurrent neural network trained with approximate Bayesian computation: A case study on structural response prognostics. *Reliability Engineering and System Safety*, 243. <https://doi.org/10.1016/j.ress.2023.109822>
- Filho, H. C. de C., Júnior, O. A. de C., de Carvalho, O. L. F., de Bem, P. P., de Moura, R. dos S., de Albuquerque, A. O., Silva, C. R., Ferreira, P. H. G., Guimarães, R. F., & Gomes, R. A. T. (2020). Rice crop detection using LSTM, Bi-LSTM, and machine learning models from Sentinel-1 time series. *Remote Sensing*, 12(16). <https://doi.org/10.3390/RS12162655>
- Gajowniczek, K., & Ząbkowski, T. (2021). ImbTreeAUC: An R package for building classification trees using the area under the ROC curve (AUC) on imbalanced datasets. *SoftwareX*, 15. <https://doi.org/10.1016/j.softx.2021.100755>
- Galahartlambang, Y., Khotiah, T., & Jumain. (2021). View of Visualisasi Data Dari Dataset COVID-19 Menggunakan Pemrograman Python. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, 3(1), 58–64. <https://doi.org/https://doi.org/10.46772/intech.v3i01.417>
- Girsang, N. S. S., & Siagian, F. A. (2023, April 27). *Pengendalian Kutu Perisai Aspidiotus destructor pada Tanaman Kelapa (Cocos nucifera)*. <https://Balaimedan.Ditjenbun.Pertanian.Go.Id/Pengendalian-Kutu-Perisai-Aspidiotus-Destructor-Pada-Tanaman-Kelapa-Cocos-Nucifera/>.
- Gupta, A. (2025). Machine Learning and Deep Learning: A Comprehensive Overview. *International Journal for Research in Applied Science and Engineering Technology*, 13(6), 1620–1626. <https://doi.org/10.22214/ijraset.2025.72470>
- Gupta, S., & Paul, N. (2023). Infestation of early shoot borer, top borer and root borer in sugarcane varieties in Shahjahanpur, Uttar Pradesh.

- International Journal of Science and Research Archive*, 9(2), 447–451.  
<https://doi.org/10.30574/ijjsra.2023.9.2.0589>
- Hamza, A., Khan, M. A., Wang, S.-H., Alqahtani, A., Alsubai, S., Binbusayyis, A., Hussein, H. S., Martinetz, T. M., & Alshazly, H. (2022). COVID-classification using chest X-ray images: A framework of CNN-LSTM and improved max value moth flame optimization. *Frontiers in Public Health* *Frontiersin.Org*.
- He, Z., Zhong, Y., & Pan, J. (2022). An adversarial discriminative temporal convolutional network for EEG-based cross-domain emotion recognition. *Computers in Biology and Medicine*, 141. <https://doi.org/10.1016/j.combiomed.2021.105048>
- Heryanto, M. A., Pardian, P., & Nugraha, A. (2024). Analisis Wilayah Unggulan Perkebunan: Komoditas Tebu (*Saccharum officinarum* Linn). *Agricore: Jurnal Agribisnis Dan Sosial Ekonomi Pertanian Unpad*, 9(2).
- Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F., & Liu, Y. (2020). Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24(21), 16453–16482. <https://doi.org/10.1007/s00500-020-04954-0>
- Hutagalung, E. F. S., & Sitompul, P. (2023). Implementasi Deep Learning Menggunakan Metode Cnn Untuk Klasifikasi Jenis Ulos Batak Toba. *Student Scientific Creativity Journal*, 1(4), 01–19. <https://doi.org/10.55606/sscj-amik.v1i4.1541>
- Jeni, B. J. J., Solekhah, B. A., Sholikah, D. H., & Rosmala, K. I. (2025). Kesesuaian Lahan untuk Komoditas Tebu, Matoa, Vanili dan Jagung di Wilayah Wonosalam, Jombang, Jawa Timur. *Agroteknika*, 8(2), 231–247. <https://doi.org/10.55043/agroteknika.v8i2.486>
- Junaidi, S., Devegi, M., & Kurniawan, H. (2023). Pelatihan Pengolahan dan Visualisasi Data Penduduk menggunakan Python. *ADMA : Jurnal Pengabdian Dan Pemberdayaan Masyarakat*, 4(1), 151–162. <https://doi.org/10.30812/adma.v4i1.2963>
- Kaestria, R., Faiqotul Himmah, E., & Irawan, R. (2024). Penerapan Matplotlib

- dalam Visualisasi Data untuk Analisis Hubungan Penggunaan Gadget dan Hasil Belajar. *Journal of Digital Business and Information Technology*, 1(1), 29–39. <https://doi.org/10.23971/jobit.v1i1.204>
- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- Kumar, R. (2024). *Machine Learning: Compare Supervised Learning Vs Unsupervised Learning Vs Reinforcement Learning*. DevOpsSchool. <https://www.devopsschool.com/blog/machine-learning-compare-supervised-learning-vs-unsupervised-learning-vs-reinforcement-learning/>
- Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2020). Temporal convolutional networks applied to energy-related time series forecasting. *Applied Sciences (Switzerland)*, 10(7). <https://doi.org/10.3390/app10072322>
- Larasati, L., Saadah, S., & Yunanto, P. E. (2024). Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Methods to Forecast Daily Turnover at BM Motor Ngawi. *Indonesian Journal of Artificial Intelligence and Data Mining*, 7(1), 141. <https://doi.org/10.24014/ijaidm.v7i1.27643>
- Liu, L., & Song, X. (2025). Accurate crop yield prediction via temporal convolutional network and variational autoencoder. *Turkish Journal of Agriculture and Forestry*, 49(3), 612–623. <https://doi.org/10.55730/1300-011X.3290>
- Lonang, S., Yudhana, A., & Biddinika, M. K. (2023). Analisis Komparatif Kinerja Algoritma Machine Learning untuk Deteksi Stunting. *Jurnal Media Informatika Budidarma*, 7(4), 2109. <https://doi.org/10.30865/mib.v7i4.6553>
- Lv, J., Wu, T., Peng, C., Liu, Y., Xu, N., & Geng, X. (2021). Compact learning for multi-label classification. *Pattern Recognition*, 113. <https://doi.org/10.1016/j.patcog.2021.107833>

- Ma, D., Wen, Y., Zhao, C., & Zhang, C. (2026). Study on Temporal Convolutional Network Rainfall Prediction Model and Its Interpretability Guided by Physical Mechanisms. *Hydrology*, 13(1), 38. <https://doi.org/10.3390/hydrology13010038>
- Mahadevaswamy, U. B., & Swathi, P. (2022). Sentiment Analysis using Bidirectional LSTM Network. *Procedia Computer Science*, 218, 45–56. <https://doi.org/10.1016/j.procs.2022.12.400>
- Masuda, N., & Yairi, I. E. (2023). Multi-Input CNN-LSTM deep learning model for fear level classification based on EEG and peripheral physiological signals. *Frontiers in Psychology*, 14. <https://doi.org/10.3389/fpsyg.2023.1141801>
- Mendrofa, J. S., Zentrato, M. W., Halawa, N., Zalukhu, E. E., & Lase, N. K. (2024). Peran Teknologi dalam Meningkatkan Efisiensi Pertanian. *Tumbuhan : Publikasi Ilmu Sosiologi Pertanian Dan Ilmu Kehutanan*, 1(3), 01–12. <https://doi.org/10.62951/tumbuhan.v1i3.111>
- Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. In *Information (Switzerland)* (Vol. 15, Number 9). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/info15090517>
- Monaghan, T. F., Rahman, S. N., Agudelo, C. W., Wein, A. J., Lazar, J. M., Everaert, K., & Dmochowski, R. R. (2021). Foundational statistical principles in medical research: Sensitivity, specificity, positive predictive value, and negative predictive value. In *Medicina (Lithuania)* (Vol. 57, Number 5). MDPI AG. <https://doi.org/10.3390/medicina57050503>
- Muliasari, A. A., & Trilaksono, R. (2020). Insidensi Hama dan Penyakit Utama Tebu (*Saccharum officinarum* L) di PT PG Rajawali II Jatitujuh Majalengka. *Jurnal Sains Terapan*, 10(1), 40–52. <https://doi.org/10.29244/jst.10.1.40>
- Munandar, A. (2025). Prediksi Harga Mata Uang Kripto Menggunakan Algoritma Temporal Convolutional Network. *Jurnal Nusantara Of Engineering*, 8. <https://ojs.unpkediri.ac.id/index.php/noe>

- Nahm, F. S. (2022). Receiver operating characteristic curve: overview and practical use for clinicians. *Korean Journal of Anesthesiology*, 75(1), 25–36. <https://doi.org/10.4097/kja.21209>
- Ngo, T. N., Rustia, D. J. A., Yang, E. C., & Lin, T. Te. (2021). Honey bee colony population daily loss rate forecasting and an early warning method using temporal convolutional networks. *Sensors*, 21(11). <https://doi.org/10.3390/s21113900>
- Nugroho, K. S. (2019, November 13). *Confusion Matrix untuk Evaluasi Model pada Supervised Learning*. <https://Ksnugroho.Medium.Com/Confusion-Matrix-Untuk-Evaluasi-Model-Pada-Unsupervised-Machine-Learning-Bc4b1ae9ae3f>.
- Nurhajjah. (2022). Pengaruh Pemberian Nipagin Terhadap Perkembangan Penggerak Batang Tebu Bergaris (*Chilo Sacchariphagus*). *Jurnal Ilmu Pertanian*, 1(1).
- Ogunsanya, M., Isichei, J., & Desai, S. (2023). Manufacturing Letters Grid Search Hyperparameter Tuning in Additive Manufacturing Processes-NC-ND license. *Manufacturing Letters*, 1031–1042. [www.sciencedirect.com](http://www.sciencedirect.com)
- Onan, A. (2022). Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *Journal of King Saud University - Computer and Information Sciences*, 34(5), 2098–2117. <https://doi.org/10.1016/j.jksuci.2022.02.025>
- Paudel, K., Dangi, N., Aryal, S., & Regmi, R. (2021). Evaluation of chemical pesticides for the management of Top Borer (*Scirpophaga excerptalis* Walker) in sugarcane. *Journal of Agriculture and Natural Resources*, 4(1), 282–290. <https://doi.org/10.3126/janr.v4i1.33289>
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. In *Journal of Applied Informatics and Computing (JAIC)* (Vol. 4, Number 1). <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Pramono, S., Wagiman, F. X., Trisyono, Y. A., & Witjaksono. (2020). The

- Preferences and Distribution of Sugarcane Scale Insect *Aulacaspis tegalensis* (Hemiptera: Diaspididae) on Different Clones of Sugarcane. *Journal of Tropical Plant Pests and Diseases*, 20(1), 46–53. <https://doi.org/10.23960/j.hptt.12046-53>
- Pratama, A. R., Wabula, F., Imandry, H., Isabela, M. L., Raharjo, M., & Sianipar, R. (2025). Literature Review The Impact of Machine Learning in Modern Industries. *Nian Tana Sikka : Jurnal Ilmiah Mahasiswa*, 3(1), 177–182. <https://doi.org/10.59603/niantanasikka.v3i1.680>
- Praveenkumar, A., Jha, G. K., Madival, S. D., Lama, A., & Kumar, R. R. (2025). Deep Learning Approaches for Potato Price Forecasting: Comparative Analysis of LSTM, Bi-LSTM, and AM-LSTM Models. *Potato Research*, 68(2), 1941–1963. <https://doi.org/10.1007/s11540-024-09823-z>
- Rahmani, A. M., Yousefpoor, E., Yousefpoor, M. S., Mehmood, Z., Haider, A., Hosseinzadeh, M., & Ali Naqvi, R. (2021). Machine learning (ML) in medicine: Review, applications, and challenges. In *Mathematics* (Vol. 9, Number 22). MDPI. <https://doi.org/10.3390/math9222970>
- Raiaan, M. A. K., Sakib, S., Fahad, N. M., Mamun, A. Al, Rahman, M. A., Shatabda, S., & Mukta, M. S. H. (2024). A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. In *Decision Analytics Journal* (Vol. 11). Elsevier Inc. <https://doi.org/10.1016/j.dajour.2024.100470>
- Raji, I. D., Bello-Salau, H., Umoh, I. J., Onumanyi, A. J., Adegboye, M. A., & Salawudeen, A. T. (2022). Simple Deterministic Selection-Based Genetic Algorithm for Hyperparameter Tuning of Machine Learning Models. *Applied Sciences (Switzerland)*, 12(3). <https://doi.org/10.3390/app12031186>
- Rasidi, A. I., Pasaribu, Y. A. H., Ziqri, A., & Adhinata, F. D. (2022). Klasifikasi Sampah Organik dan Non-Organik Menggunakan Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sistem Informasi*, 8(1). <https://doi.org/10.28932/jutisi.v8i1.4314>
- Ridwan, Handayani, H. H., Lestari, S. A. P., & Cahyana, Y. (2025). Evaluasi

- Kinerja Algoritma Random Forest Dan Gradient Boosting Untuk Klasifikasi Penyakit Jantung. *Jurnal Komtika (Komputasi Dan Informatika)*, 9(1). <https://doi.org/10.31603/komtika.v9i1.13450>
- Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82. <https://doi.org/10.31294/ijcit.v5i1.7951>
- Rom, A. R. M., Jamil, N., & Ibrahim, S. (2024). Multi objective hyperparameter tuning via random search on deep learning models. *Telkomnika (Telecommunication Computing Electronics and Control)*, 22(4), 956–968. <https://doi.org/10.12928/TELKOMNIKA.v22i4.25847>
- Saidabad, M. Y., Hassanzadeh, H., Seyed Ebrahimi, S. H., Khezri, E., Rahimi, M. R., & Trik, M. (2024). An efficient approach for multi-label classification based on Advanced Kernel-Based Learning System. *Intelligent Systems with Applications*, 21. <https://doi.org/10.1016/j.iswa.2024.200332>
- Salehi, M. M., & Zarei, H. (2025). A Review of the Structure and Application of Scikit-Learn Datasets in Machine Learning Model Development. *International Journal of Operations Research and Artificial Intelligence*, 1(2), 88–8.
- Shaikh, A. K., Nazir, A., Khalique, N., Shah, A. S., & Adhikari, N. (2023). A new approach to seasonal energy consumption forecasting using temporal convolutional networks. *Results in Engineering*, 19. <https://doi.org/10.1016/j.rineng.2023.101296>
- Silva, L. P. de S., & Gomedede, E. (2025). *Temporal Deep Learning for Satellite-Based Sugarcane Monitoring and Age Estimation*. <https://browser.dataspace.copernicus.eu/>
- Sinuraya, J. F., Suryana, E. A., Shaffitri, L. R., Suharyono, S. R., & Hermawan, H. R. (2024). Kinerja Industri dan Dinamika Kebijakan Komoditas Gula Kristal Putih Nasional. *Indonesian Sugar Research Journal*, 4(2), 68–79. <https://doi.org/10.54256/isrj.v4i2.129>
- Subiyakto. (2017a). Hama Penggerek Tebu dan Perkembangan Teknik

- Pengendaliannya. *Jurnal Penelitian Dan Pengembangan Pertanian*, 35(4), 179. <https://doi.org/10.21082/jp3.v35n4.2016.p179-186>
- Subiyakto, S. (2017b). Hama Penggerek Tebu dan Perkembangan Teknik Pengendaliannya. *Jurnal Penelitian Dan Pengembangan Pertanian*, 35(4), 179. <https://doi.org/10.21082/jp3.v35n4.2016.p179-186>
- Sunaryo, & Hasibuan, R. (2003). Perkembangan Populasi Kutu Perisai *Aulacaspis tegalensis* Zehntner (Homoptera: Diaspididae) dan Pengaruh Tingkat Serangannya terhadap Penurunan Hasil Tebu di PT Gunung Madu Plantations, Lampung Tengah. *J. Hama Dan Penyakit Tumbuhan Tropika*, 3(1).
- Surbakti, N. M., Angelyca Angelyca, Anita Talia, Cecilia Br Perangin-Angin, Dina Olivia Nainggolan, Nia Devi Friskauly, & Sikap Ruth Br Tumorang. (2024). Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel. *Algoritma : Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumihan Dan Angkasa*, 2(3), 98–107. <https://doi.org/10.62383/algoritma.v2i3.67>
- Syahnaz, C., Soedarto, T., & Yuliati, N. (2022). Analisis Perkembangan dan Faktor-Faktor yang Mempengaruhi Impor Gula di Indonesia. *Jurnal Pendidikan Tambusai*, 6(2).
- Tarekegn, A. N., Giacobini, M., & Michalak, K. (2021). A review of methods for imbalanced multi-label classification. In *Pattern Recognition* (Vol. 118). Elsevier Ltd. <https://doi.org/10.1016/j.patcog.2021.107965>
- Tarkus, D., Sompie, S. R. U. A., & Jacobus, A. (2020). Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh. *Jurnal Teknik Informatika*, 15(2), 137–144.
- Tjahjadi, V. E., Santoso, B., & Serwin. (2023). Klasifikasi Malware Menggunakan Teknik Machine Learning. *Jurnal Ilmiah Ilmu Komputer*, 2(1). <https://www.kaggle.com/datasets/amauricio/pe-files-malwares>.
- Wahyono, T., Heryadi, Y., Soeparno, H., & Abbas, B. S. (2020). Enhanced lstm multivariate time series forecasting for crop pest attack prediction. *ICIC Express Letters*, 14(10), 943–949. <https://doi.org/10.24507/icicel.14.10.943>

- Wedastra, M. S., Suartha, I. D. G., Catharina, T. S., Marini, I. A. K., Meikapasa, N. W. P., & Nopiari, I. A. (2020). Pengendalian Hama Penyakit Terpadu untuk Mengurangi Kerusakan pada Tanaman Padi di Desa Mekar Sari Kecamatan Gunung Sari. *Jurnal Gema Ngabdi*, 2(1), 88–94. <https://doi.org/10.29303/jgn.v2i1.68>
- Wei, X., Zhang, L., Yang, H. Q., Zhang, L., & Yao, Y. P. (2021). Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. *Geoscience Frontiers*, 12(1), 453–467. <https://doi.org/10.1016/j.gsf.2020.04.011>
- Yadav, S. K., Pawar, D., Yadav, L., Yadav, A., Mishra, P., Tripathi, S., & Yadav, S. K. (2024). Predicting Yellow Stem Borer Occurrence in Rice Using Weather Parameters and LSTM. *Library Progress International*, 44(3), 25926–25940. [www.bpasjournals.com](http://www.bpasjournals.com)
- Yarally, T., Cruz, L., Feitosa, D., Sallou, J., & van Deursen, A. (2023). *Uncovering Energy-Efficient Practices in Deep Learning Training: Preliminary Steps Towards Green AI*. <http://arxiv.org/abs/2303.13972>
- Yu, T., & Zhu, H. (2020). *Hyper-Parameter Optimization: A Review of Algorithms and Applications*. <http://arxiv.org/abs/2003.05689>
- Zahra, A., & Aulia, S. I. (2025). Challenges and Processes of Agribusiness Management in the Agricultural Sector in Indonesia (Tantangan dan Proses Manajemen Agribisnis pada Sektor Pertanian di Indonesia). *Jurnal Ilmiah Ekonomi Dan Manajemen*, 3(3), 406–417. <https://doi.org/10.61722/jiem.v3i3.4199>