

**PEMELIHARAAN PREDIKTIF BERBASIS *ARTIFICIAL INTELLIGENCE*
PERANGKAT EDDY COVARIANCE STATION STATIK
BERDASARKAN DATASET HARIAN**

(Skripsi)

Oleh:

NUR AFNI LATIFATUL MUCHLISA

NPM 2115031024



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

**PEMELIHARAAN PREDIKTIF BERBASIS *ARTIFICIAL INTELLIGENCE*
PERANGKAT EDDY COVARIANCE STATION STATIK
BERDASARKAN DATASET HARIAN**

Oleh:

NUR AFNI LATIFATUL MUCHLISA

Skripsi

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

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRAK

PEMELIHARAAN PREDIKTIF BERBASIS *ARTIFICIAL INTELLIGENCE* PERANGKAT EDDY COVARIANCE STATION STATIK BERDASARKAN DATASET HARIAN

Oleh

Nur Afni Latifatul Muchlisa

Perangkat Eddy Covariance Station (ECS), yang dikembangkan oleh Tim Peneliti Universitas Lampung, terdiri dari Gas Analyzer, 3D Anemometer, dan Microclimate Station, telah digunakan dalam pengukuran emisi karbon di perkebunan kelapa sawit lahan gambut, tetapi masih menghadapi tantangan dari sisi keandalan operasional. Di lapangan, *downtime* instrumen berujung pada celah data dan bias estimasi fluks. Pendekatan *reactive maintenance* mahal dan terlambat; karena itu diperlukan pemeliharaan prediktif di tepi jaringan (*edge*) yang ringan-komputasi agar kontinuitas data perangkat ECS yang berfrekuensi tinggi tetap terjaga. Penelitian ini mengembangkan sistem pemeliharaan prediktif berbasis model *hybrid* CNN-LSTM dan LSTM *autoencoder* untuk mendeteksi anomali dan memprediksi kegagalan perangkat ECS. Model dilatih dengan dataset dari masing-masing perangkat, dan dievaluasi menggunakan MAE, MSE, RMSE, dan R^2 . Konfigurasi hibrida terbaik adalah CNN-32–LSTM-32 (Gas Analyzer) dan CNN-32–LSTM-64 (3D Anemometer, Microclimate). LSTM *autoencoder* terbaik menggunakan filter 64 untuk semua perangkat. Implementasi *on-device* pada sebuah *singleboard computer* diuji pada dua skenario. Pada sistem terintegrasi, penggunaan CPU berkisar 0%–54,20%, RAM 154,16–736,07 MB, dan waktu inferensi tercepat adalah 0,000766 detik (*hybrid* CNN-LSTM) serta 0,005214 detik (LSTM *autoencoder*). Pada sistem terdistribusi, CPU mencapai 100%, RAM hingga 1402,23 MB, dan waktu inferensi tercepat adalah 0,000777 detik (*hybrid* CNN-LSTM) dan 0,005074 detik (LSTM *autoencoder*). Seluruh latensi berada jauh di bawah ambang operasi 1–10 Hz, sehingga deteksi anomali *real-time* tercapai tanpa mengorbankan akuisisi. Hasil menunjukkan sistem mampu mendeteksi anomali secara *real-time* dengan efisiensi komputasi yang baik. Secara R&D, kontribusi utama meliputi: (1) arsitektur model yang ditenagai data dan *device-specific*, (2) *benchmarking* komputasi *edge* dengan dua pola penyebaran, dan (3) bukti kemampuan sistem memberi peringatan dini kegagalan instrumen untuk meningkatkan kontinuitas dan keterpercayaan data ECS.

Kata Kunci: Pemeliharaan Prediktif, model *hybrid* CNN-LSTM, model LSTM *autoencoder*, Python, TensorFlow, Raspberry Pi

ABSTRACT

ARTIFICIAL INTELLIGENCE-BASED PREDICTIVE MAINTENANCE FOR STATIC EDDY COVARIANCE STATION DEVICES USING DAILY DATASETS

By

Nur Afni Latifatul Muchlisa

The Eddy Covariance Station (ECS) developed by the University of Lampung research team—comprising a Gas Analyzer, a 3D anemometer, and a Microclimate Station—has been deployed to quantify carbon emissions in peatland oil-palm plantations. Operational reliability remains a critical bottleneck: instrument downtime creates data gaps and biases flux estimates. Reactive maintenance is costly and late by design; consequently, lightweight edge-based predictive maintenance is needed to preserve the continuity of high-frequency ECS measurements. This study develops a predictive-maintenance framework based on two model families: a hybrid CNN–LSTM for pattern extraction and sequence memory, and an LSTM autoencoder for reconstruction-based anomaly detection. Device-specific models are trained per subsystem and evaluated using MAE, MSE, RMSE, and R^2 . The best hybrid configurations are CNN-32–LSTM-32 for the Gas Analyzer and CNN-32–LSTM-64 for the 3D Anemometer and Microclimate Station; the best LSTM autoencoder uses 64 units across devices. On-device implementation on a single-board computer was assessed under integrated and distributed deployment scenarios. In the integrated setup, CPU usage ranged from 0% to 54.20%, RAM from 154.16 to 736.07 MB, and best inference latencies were 0.000766 s (hybrid) and 0.005214 s (autoencoder). In the distributed setup, CPU reached 100%, RAM up to 1,402.23 MB, with best latencies of 0.000777 s (hybrid) and 0.005074 s (autoencoder). All latencies are well below the 1–10 Hz operational window, enabling real-time anomaly detection without sacrificing data acquisition. R&D contributions include: (i) data-driven, device-specific architectures; (ii) edge-computing benchmarking across two deployment patterns; and (iii) empirical evidence of early fault warning that improves ECS data continuity and trustworthiness for flux estimation.

Keywords: Predictive Maintenance, Hybrid CNN-LSTM Model, LSTM Autoencoder Model, Python, TensorFlow, Raspberry Pi

Judul Skripsi

: **PEMELIHARAAN PREDIKTIF BERBASIS
ARTIFICIAL INTELLIGENCE PERANGKAT
EDDY COVARIANCE STATION STATIK
BERDASARKAN DATASET HARIAN**

Nama Mahasiswa

: **Nur Afni Latifatul Muchlisa**

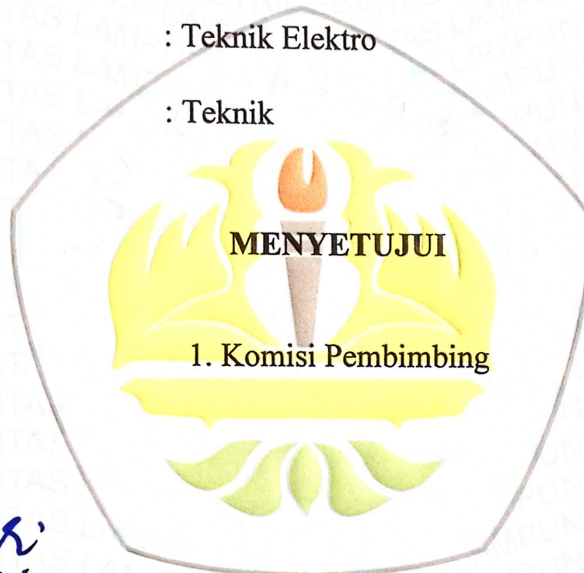
Nomor Pokok Mahasiswa : 2115031024

Jurusan

: Teknik Elektro

Fakultas

: Teknik



Ing. Melvi, S.T., M.T., Ph.D.
NIP. 197301182000032001

Aryanto, S.T., M.T.
NIP.199006212019031011

2. Mengetahui

Ketua Jurusan Teknik Elektro

Herlinawati, S.T., M.T.
NIP. 197103141999032001

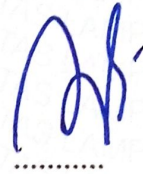
Ketua Program Studi Teknik Elektro

Sumadi, S.T., M.T.
NIP. 197311042000031001

MENGESAHKAN

1. Tim Penguji

Ketua : Ing. Ardian Ulvan, S.T., M.Sc., Ph.D.



Sekretaris : Aryanto, S.T., M.T.



Penguji : Ing. Melvi, S.T., M.T., Ph.D.



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. 1
NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi: 29 Juli 2025

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Nur Afni Latifatul Muchlisa

NPM : 2115031024

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang sepengetahuan saya tidak terdapat atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 29 Juli 2025



Nur Afni Latifatul Muchlisa

NPM. 2115031024

RIWAYAT HIDUP



Penulis dilahirkan di Kota Metro pada tanggal 18 Agustus 2003 sebagai anak pertama dari dua bersaudara, putri dari Bapak Siam Muchlisa dan Ibu Tri Rasmiati. Penulis memulai pendidikan dasar di SD Negeri 7 Metro Pusat pada tahun 2010 hingga 2016, kemudian melanjutkan ke jenjang menengah pertama di SMP Negeri 1 Metro pada tahun 2016 hingga 2019 dan jenjang menengah atas di SMA Negeri 1 Metro pada tahun 2019 hingga 2021.

Penulis melanjutkan studi di Prodi S1 Teknik Elektro, Jurusan Teknik Elektro, Universitas Lampung pada tahun 2021 melalui jalur SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri). Penulis aktif mengikuti organisasi di tingkat jurusan, Himatro (Himpunan Mahasiswa Teknik Elektro) sebagai anggota Divisi Pendidikan periode 2022 dan 2023. Selain itu, penulis juga aktif mengikuti organisasi di tingkat universitas, UKM Penelitian sebagai anggota divisi HLPM (Hubungan Luar dan Pengembangan Masyarakat) periode 2023 dan SRE Unila (*Society of Renewable Energy* Universitas Lampung) sebagai anggota departemen HR (*Human Resource*) periode 2023. Penulis mengambil konsentrasi Telti (Telekomunikasi dan Teknologi Informasi) dan aktif menjadi asisten praktikum di Laboratorium Telekomunikasi. Pada tahun 2023, penulis mengikuti PPK Ormawa (Program Penguatan Kapasitas Organisasi Mahasiswa) di Kampung Bumi Dipasena Agung, bertanggung jawab di pembuatan *website* bagian *front-end* pengenalan produk olahan, pembuatan buku soft skill, dan pendukung administrasi. PPK Ormawa ini lolos Abdidaya dan meraih juara 2 di kategori poster terlengkap. Pada tahun 2024, penulis mengikuti program Kampus Merdeka, berupa Studi Independen di Bangkit mengambil studi *Machine Learning*. Penulis aktif mengikuti penelitian di laboratorium, khususnya penelitian Dosen terkait *Carbon Monitoring* di perkebunan kelapa sawit dan mangrove, bertanggung jawab di bagian pemeriksaan dan pemeliharaan perangkat Eddy Covariance Station.

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah atas berkah rahmat dan karunia Allah SWT Skripsi ini bisa terselesaikan dengan baik.

Skripsi ini dipersembahkan dengan penuh rasa syukur, hormat, dan kasih sayang sebagai tanda terima kasih untuk yang selalu mendukung:

Kedua Orang Tua Tersayang

Bapak Siam Muchlisa dan Ibu Tri Rasmiati atas segala rasa kepercayaan dan dukungan dalam menempuh kuliah hingga menyelesaikan skripsi ini.

Adik Tersayang

Afifatur Rohma Muchlisa yang selalu menyemangati dan menghibur dalam menempuh kuliah hingga menyelesaikan skripsi ini.

Seluruh **keluarga besar, dosen, teman, dan almamater**

Atas segala dukungan dalam menyelesaikan skripsi.

MOTO HIDUP

“Kalau salah ya revisi, kalau gagal ya coba lagi.”

“Dream it. Believe it. Build it.”

“There’s nothing to catch up to. There’s no final version of you that unlock the rest of your life.”

(The Ash Files)

“Allah does not burden a soul beyond that it can bear.”

(Quran, 2:286)

“So surely with hardship comes ease.”

(Quran, 94:5-6)

SANWACANA

Segala puji dan syukur dipanjatkan ke hadirat Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Pemeliharaan Prediktif Berbasis *Artificial Intelligence* Perangkat Eddy Covariance Station Statik Berdasarkan Dataset Harian” tepat pada waktunya. Skripsi ini disusun sebagai salah satu syarat dalam memperoleh gelar Sarjana Teknik di Prodi S1 Teknik Elektro, Jurusan Teknik Elektro, Universitas Lampung. Pada penyusunan ini, penulis mendapat banyak dukungan, arahan, dan bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
4. Bapak Sumadi, S.T., M.T. selaku Ketua Program Studi Teknik Elektro Universitas Lampung.
5. Bapak Ing. Ardian Ulvan, S.T., M.Sc., Ph.D. selaku Kepala Laboratorium Telekomunikasi, sekaligus penguji utama pada pengerjaan skripsi ini yang telah banyak memberi arahan dan bimbingan, khususnya terkait fokus penelitian ini, serta memberi motivasi kepada penulis.
6. Ibu Ing. Melvi, S.T., M.T., Ph.D. selaku dosen pembimbing utama skripsi yang telah banyak memberi arahan dan bimbingan, khususnya terkait proses penulisan penelitian ini, serta memberi motivasi kepada penulis.
7. Bapak Aryanto, S.T., M.T., selaku dosen pembimbing pendamping yang telah banyak memberi arahan dan bimbingan, khususnya memberi perspektif lain terkait penelitian ini, serta memberi motivasi kepada penulis.

8. Bapak Dr. Eng. Lukmanul Hakim, S.T., M.Sc., selaku dosen pembimbing akademik yang telah banyak memberi bimbingan terkait perkuliahan.
9. Seluruh Dosen dan Civitas Jurusan Teknik Elektro yang memberikan banyak ilmu selama di perkuliahan ini dan membantu mengurus administrasi.
10. Keluargaku tersayang, Bapak Siam, Ibu Rasmi, dan Adik, yang memberi dukungan terbaik untuk penulis.
11. Teman terdekat Tim ECS, Sofya, Aldi, Rizky, dan Wisnu yang telah menjadi bagian senang dan susah penulis, serta menjadi teman cerita dan diskusi ketika mengerjakan projek dan skripsi.
12. Teman diskusi Tim MCS dan Pumma, Akmal, Rizkan, Ridho, Atika, Ilham, dan lainnya yang telah menjadi teman diskusi penulis ketika merasa bingung.
13. Teman penyemangat *vibe in five*, Sofya, Azizah, Armita, dan Sefia yang selalu mendukung penulis dengan canda dan tawa.
14. Teman terlama yang paham dengan penulis, Allamanda, Amrina, Afifah, dan Sofia yang memberi semangat untuk penulis menyelesaikan skripsi ini.
15. Tim ECS angkatan 2022 dan 2023, Joko, Rafi, Maharani, Dhimas, Dimas, Rizki, Ghaza, dan lainnya yang membantu membuat perangkat dan mendukung penulis untuk menyelesaikan skripsi ini.
16. Teman-teman konsentrasi Telti yang memberi semangat untuk terus menyelesaikan skripsi.
17. Keluarga besar Himatro Unila dan Teknik Elektro Angkatan 21 yang telah memberikan pengalaman dan cerita di perkuliahan ini.

Semua pihak yang terlibat dalam pengerjaan skripsi yang tidak dapat disebutkan satu persatu. Penulis menyadari bahwa skripsi masih terdapat banyak kekurangan, untuk itu diharapkan kritik dan saran yang dapat membangun kedepannya.

Bandar Lampung, 29 Juli 2025

Nur Afni Latifatul Muchlisa

DAFTAR ISI

	Halaman
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvii
I. PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	4
1.4. Ruang Lingkup dan Batasan Masalah.....	4
1.5. Keutamaan Penelitian.....	5
1.6. Sistematika Penulisan.....	6
II. TINJAUAN PUSTAKA.....	7
2.1. Penelitian Terdahulu	7
2.2. Analisis Fluks Karbon di Perkebunan Kelapa Sawit	8
2.3. Sistem Pemantauan Lingkungan Eddy Covariance Station	10
2.3.1. Sistem Perangkat Gas Analyzer	12
2.3.2. Sistem 3D Anemometer	12
2.3.3. Sistem Microclimate Station.....	13
2.4. Sistem Pemeliharaan	14
2.5. Pemeliharaan Prediktif Berbasis AI di Industri Algikultur.....	16
2.5.1. Persiapan Data.....	17
2.5.2. Pembuatan Model.....	18
2.6. Sistem Pemantauan dan Pengintegrasian Jarak Jauh	30
2.7. Pengukuran Kinerja Sistem.....	32
III. METODOLOGI PENELITIAN	33
3.1. Waktu dan Tempat Penelitian	33
3.2. <i>Capstone Project</i>	33
3.3. Perangkat Penelitian.....	34
3.4. Sumber Data.....	35
3.5. Tahapan Penelitian	36
3.6. Pengolahan Data.....	38
3.7. Pengembangan Model.....	39
3.7.1. Pemilihan Algoritma	39
3.7.2. Struktur Model	40

3.7.3.	Pelatihan Model	44
3.7.4.	Evaluasi Model.....	44
3.7.5.	Konversi Model TensorFlow ke TensorFlow Lite.....	44
3.8.	Implementasi Sistem	45
3.8.1.	Skenario Sistem Deteksi Kegagalan	45
3.8.2.	Skenario Pengujian Sistem.....	46
3.9.	Evaluasi Sistem	48
IV.	HASIL DAN PEMBAHASAN.....	49
4.1	Pengolahan Dataset	49
4.2	Pengembangan Model.....	55
4.2.1.	Model Hybrid CNN-LSTM.....	55
4.2.2.	Model LSTM Autoencoder.....	75
4.2.3.	Konversi Model ke Format TFLite	84
4.3	Implementasi Sistem	85
4.3.1.	Persiapan Implementasi Sistem.....	85
4.3.1.1.	Pengiriman Data Sensor	86
4.3.1.2.	Pengiriman Notifikasi Telegram.....	87
4.3.2.	Sistem Pendeteksi Anomali.....	87
4.3.3.	Pengaplikasian Sistem di Raspberry Pi.....	92
4.4	Evaluasi Sistem	93
4.4.1.	Evaluasi Sistem Terintegrasi.....	93
4.4.2.	Evaluasi Sistem Terdistribusi.....	102
V.	PENUTUP	120
5.1	Kesimpulan	120
5.2	Saran.....	121
	DAFTAR PUSTAKA	122
	LAMPIRAN.....	124

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Metode Eddy Covariance [8]	9
Gambar 2.2 Pengaplikasian Perangkat C-Flux Analyzer.....	11
Gambar 2.3 Pengaplikasian Perangkat Microclimate Station.....	11
Gambar 2.4 Sistem Gas Analyzer	12
Gambar 2.5 Sistem 3D Anemometer	13
Gambar 2.6 Sistem Microclimate Station	13
Gambar 2.7 Strategi Pemeliharaan Korektif [9]	14
Gambar 2.8 Strategi Pemeliharaan Preventif [9]	15
Gambar 2.9 Strategi Pemeliharaan Prediktif [9]	16
Gambar 2.10 Arsitektur Sel LSTM [14]	19
Gambar 2.11 Langkah-langkah Operasi di Sel LSTM [14].....	20
Gambar 2.12 Struktur Jaringan LSTM [14]	21
Gambar 2.13 Jaringan LSTM Restricted [14].....	22
Gambar 2.14 Ilustrasi Model Hybrid CNN-LSTM [15]	23
Gambar 2.15 Struktur Autoencoder [17]	24
Gambar 2.16 Overview Model LSTM Autoencoder [18].....	25
Gambar 2.17 Detail LSTM Encoder [18]	26
Gambar 2.18 Detail LSTM Decoder [18]	26
Gambar 2.19 Detail Reconstruction Loss [18].....	27
Gambar 3.1 Diagram Capstone Project Tim Eddy Covariance Station (ECS)	34
Gambar 3.2 Diagram Alir Tahapan Penelitian Keseluruhan	36
Gambar 3.3 Diagram Alir Pengolahan Data	38
Gambar 3.4 Skenario Sistem Prediksi-Deteksi	45
Gambar 3.5 Skenario Deteksi Langsung.....	46
Gambar 3.6 Diagram Alir Pengujian Sistem Terintegrasi	47
Gambar 3.7 Diagram Alir Pengujian Sistem Terdistribusi	47
Gambar 4.1 Pemeriksaan Missing Value Dataset Gas Analyzer dan 3D Anemometer.....	49
Gambar 4.2 Visualisasi Salah Satu Fitur Gas Analyzer.....	50
Gambar 4.3 Visualisasi Awal Salah Satu Fitur 3D Anemometer	51
Gambar 4.4 Visualisasi Setelah Imputasi Salah Satu Fitur 3D Anemometer	52
Gambar 4.5 Pemeriksaan Missing Value Dataset Microclimate Station	53
Gambar 4.6 Visualisasi Setelah Imputasi Salah Satu Fitur Microclimate Station	54
Gambar 4.7 Summary Model Hybrid CNN-LSTM Gas Analyzer	56
Gambar 4.8 Training dan Validation Loss Model Hybrid CNN-LSTM Gas Analyzer.....	57

Gambar 4.9 Perbandingan Aktual dan Prediksi Fitur Pressure.....	58
Gambar 4.10 Perbandingan Aktual dan Prediksi Fitur Temperature 1	58
Gambar 4.11 Perbandingan Aktual dan Prediksi Fitur Approximate Altitude	59
Gambar 4.12 Perbandingan Aktual dan Prediksi Fitur Humidity	59
Gambar 4.13 Perbandingan Aktual dan Prediksi Fitur Temperature 2	60
Gambar 4.14 Perbandingan Aktual dan Prediksi Fitur CO ₂	60
Gambar 4.15 Perbandingan Aktual dan Prediksi Fitur CH ₄	61
Gambar 4.16 Perbandingan Aktual dan Prediksi Fitur H ₂ O	61
Gambar 4.17 Summary Model Hybrid CNN-LSTM 3D Anemometer	63
Gambar 4.18 Training dan Validation Loss Model Hybrid CNN-LSTM 3D Anemometer.....	64
Gambar 4.19 Perbandingan Aktual dan Prediksi Fitur Utara	65
Gambar 4.20 Perbandingan Aktual dan Prediksi Fitur Barat.....	65
Gambar 4.21 Perbandingan Aktual dan Prediksi Fitur Timur	66
Gambar 4.22 Perbandingan Aktual dan Prediksi Fitur Selatan.....	66
Gambar 4.23 Perbandingan Aktual dan Prediksi Fitur Bawah	67
Gambar 4.24 Perbandingan Aktual dan Prediksi Fitur Atas	67
Gambar 4.25 Summary Model Hybrid CNN-LSTM Microclimate Station	69
Gambar 4.26 Training dan Validation Loss Model Hybrid CNN-LSTM Microclimate Station.....	70
Gambar 4.27 Perbandingan Aktual dan Prediksi Fitur Angle.....	71
Gambar 4.28 Perbandingan Aktual dan Prediksi Fitur Humidity	71
Gambar 4.29 Perbandingan Aktual dan Prediksi Fitur Pyrano	72
Gambar 4.30 Perbandingan Aktual dan Prediksi Fitur Rainfall	72
Gambar 4.31 Perbandingan Aktual dan Prediksi Fitur Temperature	73
Gambar 4.32 Perbandingan Aktual dan Prediksi Fitur Wind Speed.....	73
Gambar 4.33 Summary Model LSTM Autoencoder Gas Analyzer.....	76
Gambar 4.34 Training dan Validation Loss Model LSTM Autoencoder Gas Analyzer	77
Gambar 4.35 Distribusi MAE per Fitur Model LSTM Autoencoder Gas Analyzer	78
Gambar 4.36 Summary Model LSTM Autoencoder 3D Anemometer.....	79
Gambar 4.37 Training dan Validation Loss Model LSTM Autoencoder 3D Anemometer.....	80
Gambar 4.38 Distribusi MAE per Fitur Model LSTM Autoencoder 3D Anemometer.....	81
Gambar 4.39 Summary Model LSTM Autoencoder Microclimate Station.....	82
Gambar 4.40 Training dan Validation Loss Model LSTM Autoencoder Microclimate Station.....	83
Gambar 4.41 Distribusi MAE per Fitur Model LSTM Autoencoder Microclimate Station	84
Gambar 4.42 Skenario Sistem Pendeteksi Anomali Langsung.....	88

Gambar 4.43 Skenario Sistem Prediksi-Deteksi Anomali	90
Gambar 4.44 Visualisasi Eror Rekonstruksi Sistem Terintegrasi Model Gas Analyzer Deteksi Anomali Langsung	94
Gambar 4.45 Visualisasi Eror Rekonstruksi Sistem Terintegrasi Model Gas Analyzer Prediksi-Deteksi Anomali	96
Gambar 4.46 Penggunaan CPU Raspberry Pi Sistem Terintegrasi.....	98
Gambar 4.47 Penggunaan RAM Raspberry Pi Sistem Terintegrasi	99
Gambar 4.48 Notifikasi Telegram Sistem Terintegrasi Model Gas Analyzer Deteksi Anomali Langsung.....	101
Gambar 4.49 Notifikasi Telegram Sistem Terintegrasi Model Gas Analyzer Prediksi-Deteksi Anomali	101
Gambar 4.50 Visualisasi Eror Rekonstruksi Sistem Terdistribusi Model Gas Analyzer Deteksi Anomali Langsung	102
Gambar 4.51 Visualisasi Eror Rekonstruksi Sistem Terdistribusi Model Gas Analyzer Prediksi-Deteksi Anomali	104
Gambar 4.52 Visualisasi Eror Rekonstruksi Sistem Terdistribusi Model 3D Anemometer Deteksi Anomali Langsung.....	106
Gambar 4.53 Visualisasi Eror Rekonstruksi Sistem Terdistribusi Model 3D Anemometer Prediksi-Deteksi Anomali	107
Gambar 4.54 Visualisasi Eror Rekonstruksi Sistem Terdistribusi Model Microclimate Station Deteksi Anomali Langsung	109
Gambar 4.55 Visualisasi Eror Rekonstruksi Sistem Terdistribusi Model Microclimate Station Prediksi-Deteksi Anomali	110
Gambar 4.56 Penggunaan CPU Raspberry Pi Sistem Terdistribusi	113
Gambar 4.57 Penggunaan RAM Raspberry Pi Sistem Terdistribusi	114
Gambar 4.58 Notifikasi Telegram Sistem Terdistribusi Model Gas Analyzer Deteksi Anomali.....	117
Gambar 4.59 Notifikasi Telegram Sistem Terdistribusi Model Gas Analyzer Prediksi-Deteksi Anomali	117
Gambar 4.60 Notifikasi Telegram Sistem Terdistribusi Model 3D Anemometer Deteksi Anomali.....	118
Gambar 4.61 Notifikasi Telegram Sistem Terdistribusi Model 3D Anemometer Prediksi-Deteksi Anomali	118
Gambar 4.62 Notifikasi Telegram Sistem Terdistribusi Microclimate Station Deteksi Anomali.....	119
Gambar 4.63 Notifikasi Telegram Sistem Terdistribusi Microclimate Station Prediksi-Deteksi Anomali	119

DAFTAR TABEL

	Halaman
Tabel 3.1 Struktur Layer Hybrid CNN-LSTM	40
Tabel 3.2 Hyperparameter Model Hybrid CNN-LSTM	41
Tabel 3.3 Struktur Model LSTM Autoencoder.....	42
Tabel 3.4 Hyperparameter Model LSTM Autoencoder.....	43
Tabel 4.1 Beberapa Data Teratas Normalisasi Dataset Model Hybrid CNN-LSTM.	55
Tabel 4.2 Evaluasi Model Hybrid CNN-LSTM Gas Analyzer.....	62
Tabel 4.3 Evaluasi Model Hybrid CNN-LSTM 3D Anemometer.....	68
Tabel 4.4 Evaluasi Model Hybrid CNN-LSTM Microclimate Station.....	74
Tabel 4.5 Beberapa Data Teratas Normalisasi Dataset Model LSTM Autoencoder	75
Tabel 4.6 Evaluasi Model LSTM Autoencoder Gas Analyzer	77
Tabel 4.7 Evaluasi Model LSTM Autoencoder 3D Anemometer	80
Tabel 4.8 Evaluasi Model LSTM Autoencoder Microclimate Station	83
Tabel 4.9 Pengaplikasian Sistem di Raspberry Pi.....	92
Tabel 4.10 Ringkasan Error Rekonstruksi Sistem Terintegrasi Model Gas Analyzer Deteksi Anomali Langsung.....	95
Tabel 4.11 Ringkasan Error Rekonstruksi Sistem Terintegrasi Model Gas Analyzer Prediksi-Deteksi Anomali	97
Tabel 4.12 Waktu Eksekusi Model Sistem Terintegrasi	97
Tabel 4.13 Penggunaan CPU dan RAM per Script Python Sistem Terintegrasi	100
Tabel 4.14 Ringkasan Sensor Terdeteksi Anomali Sistem Terdistribusi Model Gas Analyzer Deteksi Anomali Langsung	103
Tabel 4.15 Ringkasan Sensor Terdeteksi Anomali Sistem Terdistribusi Model Gas Analyzer Prediksi-Deteksi Anomali	105
Tabel 4.16 Ringkasan Sensor Terdeteksi Anomali Sistem Terdistribusi Model 3D Anemometer Deteksi Anomali Langsung.....	107
Tabel 4.17 Ringkasan Sensor Terdeteksi Anomali Sistem Terdistribusi Model 3D Anemometer Prediksi-Deteksi Anomali	108
Tabel 4.18 Ringkasan Sensor Terdeteksi Anomali Sistem Terdistribusi Model Microclimate Station Deteksi Anomali Langsung	110
Tabel 4.19 Ringkasan Sensor Terdeteksi Anomali Sistem Terdistribusi Model Microclimate Station Prediksi-Deteksi Anomali	111

Tabel 4.20 Waktu Eksekusi Model Sistem Terdistribusi.....	112
Tabel 4.21 Penggunaan CPU dan RAM per Script Python Sistem Terdistribusi	115

I. PENDAHULUAN

1.1. Latar Belakang

Pemantauan dan pengukuran parameter gas rumah kaca di perkebunan kelapa sawit menjadi semakin penting untuk memahami dinamika ekosistem dan pola fluks karbon dalam sistem pertanian tropis. Pengukuran fluks karbon dioksida (CO_2), bersama dengan parameter lingkungan yang terkait, memberikan wawasan penting mengenai kemampuan penyerapan karbon dan dampak lingkungan dari budidaya kelapa sawit. Industri kelapa sawit, sebagai salah satu sektor pertanian terbesar di dunia, menghadapi tekanan yang semakin besar untuk menunjukkan keberlanjutan lingkungan dan manajemen jejak karbon [1]. Pemantauan parameter lingkungan yang akurat dan berkesinambungan, terutama fluks CO_2 , telah menjadi hal yang mendasar dalam mengatasi masalah ini.

Kelapa sawit memiliki kemampuan menyerap karbon neto sebesar 64,5 ton CO_2 per hektar per tahun, lebih tinggi dibandingkan hutan hujan tropis yang hanya menyerap 42,4 ton CO_2 per hektar per tahun [2]. Pengembangan teknologi pemantauan berbasis sensor menjadi krusial untuk memberikan data *real-time* mengenai emisi dan penyerapan karbon. Data ini dapat digunakan untuk mengidentifikasi praktik terbaik dalam memaksimalkan penyerapan karbon dan meminimalkan jejak karbon, mendukung langkah strategis menuju keberlanjutan.

Sistem pemantauan biasanya mengintegrasikan beberapa perangkat, termasuk pengukuran fluks gas karbon, anemometer tiga dimensi, dan stasiun cuaca otomatis, untuk menyediakan data lingkungan yang komprehensif. Peneliti di Jurusan Teknik

Elektro, Universitas Lampung, telah melakukan beberapa terobosan penting dalam pengembangan perangkat Eddy Covariance Station (ECS), yang dapat digunakan untuk pengukuran emisi karbon. Kemajuan terbaru dalam teknologi ECS untuk pemantauan lingkungan telah memungkinkan pengukuran berbagai parameter dengan presisi tinggi: konsentrasi CO₂, kandungan uap air, pola angin, dan parameter meteorologi tambahan seperti suhu udara, kelembapan, curah hujan, dan intensitas matahari. Pengukuran-pengukuran ini, jika dikumpulkan secara terus menerus dan akurat, akan memberikan informasi yang sangat penting mengenai perilaku ekosistem dan pola pertukaran karbon.

Namun, penerapan sistem pemantauan seperti ini di perkebunan kelapa sawit menghadirkan tantangan signifikan, terutama dalam hal keandalan dan perawatan perangkat. Lokasi instalasi yang terpencil sering kali menyulitkan pengoperasian dan perawatan sistem secara optimal. Kegagalan sistem pemantauan sensor akibat kurangnya prediksi dapat mengganggu pengumpulan data dan meningkatkan biaya operasional. Perangkat ECS telah menjalani pemeliharaan korektif, yaitu pemeliharaan yang dilakukan setelah perangkat mengalami gangguan atau tidak berfungsi secara normal. Namun, pendekatan ini memiliki kelemahan karena menyebabkan perangkat penurunan performa. Guna mengantisipasi hal tersebut, pemeliharaan preventif secara berkala juga telah diterapkan. Tetapi metode ini pun belum sepenuhnya efektif karena perangkat tetap berisiko mengalami gangguan sebelum jadwal pemeliharaan berikutnya dilakukan.

Penelitian oleh Hafidhoh *et al.* menyoroti pentingnya pemeliharaan prediktif untuk mengurangi kegagalan mesin [3]. Sebagai contoh, analisis dataset industri menunjukkan bahwa kegagalan sistem dapat diklasifikasikan dengan akurasi 96,87% [3]. Penerapan teknologi serupa dalam sistem pemantauan lingkungan dapat mencegah kegagalan perangkat, menjaga konsistensi data, dan mendukung keberlanjutan operasional. Oleh karena itu, pendekatan pemeliharaan prediktif menjadi sangat relevan untuk diterapkan. Pemeliharaan ini berfokus pada pemantauan kondisi perangkat secara *real-time* untuk memprediksi dan mencegah potensi kerusakan sebelum terjadi. Sistem ECS, yang terdiri dari penganalisis gas,

anemometer tiga dimensi, dan stasiun pengukuran iklim mikro, sangat bergantung pada keakuratan dan konsistensi data yang dihasilkannya. Melalui penerapan strategi pemeliharaan prediktif, risiko kegagalan fungsi perangkat dapat diminimalkan, sehingga kualitas data tetap terjaga.

Implementasi pemeliharaan prediktif ini memiliki peran penting dalam mendukung keberlangsungan penelitian ekosistem, khususnya dalam upaya memahami dinamika lingkungan di perkebunan kelapa sawit serta kontribusinya terhadap mitigasi emisi karbon. Berdasarkan urgensi tersebut, penelitian ini berfokus pada penerapan sistem pemeliharaan prediktif berbasis data pada perangkat Eddy Covariance Station sebagai upaya meningkatkan efektivitas dan keberlanjutan sistem monitoring lingkungan.

1.2. Rumusan Masalah

Adapun rumusan masalah yang terdapat pada penelitian ini adalah sebagai berikut:

1. Bagaimana membangun sistem pemeliharaan prediktif dengan membuat model prediksi dan deteksi anomali berbasis model *hybrid* CNN-LSTM dan LSTM *autoencoder* pada perangkat Eddy Covariance Station (Gas Analyzer, 3D Anemometer, Microclimate Station)?
2. Sejauh mana akurasi dan performa masing-masing model (dalam hal MAE, MSE, RMSE, dan R-Squared) dalam memprediksi data sensor dan mendeteksi anomali pada ketiga perangkat tersebut?
3. Bagaimana efisiensi penggunaan sumber daya komputasi (CPU, RAM, dan waktu inferensi) dari sistem prediksi dan deteksi yang diimplementasikan pada platform *edge computing* (Raspberry Pi), baik dalam skenario terintegrasi maupun sistem terdistribusi?

1.3. Tujuan Penelitian

Adapun tujuan penelitian yang terdapat pada penelitian ini adalah sebagai berikut:

1. Membangun dan mengimplementasikan sistem pemeliharaan prediktif berbasis model *hybrid* CNN-LSTM dan LSTM *autoencoder* pada perangkat Eddy Covariance Station (Gas Analyzer, 3D Anemometer, Microclimate Station)
2. Menguji dan membandingkan performa model prediksi dan deteksi anomali berdasarkan metrik evaluasi (MAE, MSE, RMSE, dan R^2) untuk masing-masing perangkat, guna menentukan model terbaik.
3. Menganalisis efisiensi penggunaan sumber daya komputasi (CPU, RAM, dan waktu inferensi) pada dua skenario sistem, yaitu sistem terintegrasi dan sistem terdistribusi.

1.4. Ruang Lingkup dan Batasan Masalah

Penelitian ini berfokus pada pengembangan sistem pemeliharaan prediktif berbasis kecerdasan buatan (AI) untuk stasiun pemantauan karbon perangkat Eddy Covariance Station. Sistem ini mencakup penggunaan dataset harian dari tiga perangkat utama: C-Flux Analyzer yang terdiri dari Gas Analyzer dan 3D Anemometer, serta Microclimate Station (MCS). Penelitian ini menggunakan data dari Gas Analyzer dan 3D Anemometer selama 9 hari (7 data per detik selama 30 menit) serta Microclimate Station selama 6 bulan (1 data per menit). Pengembangan model menggunakan model *hybrid* CNN-LSTM untuk prediksi dan model LSTM *autoencoder* untuk deteksi anomali, dengan konfigurasi 32 dan 64 filter. Evaluasi model dilakukan menggunakan MAE, MSE, RMSE, dan R^2 . Sistem diimplementasikan dalam Python dan TensorFlow Lite, serta diuji pada Raspberry Pi dalam dua skenario: sistem terintegrasi (sensor dan sistem pemeliharaan pada satu perangkat) dan sistem terdistribusi (sensor dan sistem pemeliharaan pada perangkat berbeda). Evaluasi mencakup penggunaan CPU, RAM, dan waktu

inferensi selama 3 hari. Sistem beroperasi secara real time dan terintegrasi dengan Telegram untuk notifikasi.

Adapun batasan penelitian pada penelitian ini adalah sebagai berikut:

1. Penelitian ini tidak mencakup analisis perangkat keras atau perbaikan fisik.
2. Penelitian ini tidak membandingkan performa model *hybrid* CNN-LSTM dan LSTM *autoencoder* dengan algoritma atau pendekatan lain.
3. Evaluasi sistem dilakukan hanya pada satu jenis perangkat komputasi, yaitu Raspberry Pi.
4. Validasi terhadap hasil deteksi anomali tidak menggunakan *ground truth* dari kejadian nyata, sehingga evaluasi terbatas pada metrik teknis seperti penggunaan CPU dan RAM, serta kecepatan inferensi.

1.5. Keutamaan Penelitian

Adapun keutamaan penelitian yang terdapat pada penelitian ini adalah sebagai berikut:

1. Penelitian ini memberikan kontribusi yang signifikan pada aspek ilmiah dan praktis dari pemantauan lingkungan di lingkungan pertanian. Pengembangan sistem pemeliharaan prediktif berbasis AI mengatasi kesenjangan kritis dalam paradigma operasional stasiun pemantauan lingkungan jarak jauh.
2. Pada perspektif ekonomi, keberhasilan implementasi sistem ini dapat secara substansial mengurangi biaya pemeliharaan sekaligus meningkatkan kualitas dan kontinuitas data.
3. Metodologi dan temuan dari penelitian ini dapat diadaptasi untuk aplikasi pemantauan jarak jauh serupa di berbagai konteks penelitian pertanian dan lingkungan. Potensi untuk memperluas solusi ini ke skenario pemantauan jarak jauh lainnya merupakan kontribusi yang signifikan terhadap bidang teknologi pemantauan lingkungan.

1.6. Sistematika Penulisan

BAB I. PENDAHULUAN

Bab pendahuluan ini berisi latar belakang, tujuan penelitian, rumusan masalah, ruang lingkup dan batasan masalah, keutamaan penelitian, serta sistematika penulisan pada penelitian ini.

BAB II. TINJAUAN PUSTAKA

Bab tinjauan pustaka ini berisi penelitian terdahulu, analisis fluks karbon di perkebunan kelapa sawit, sistem perangkat pemantauan lingkungan Eddy Covariance Station, sistem pemeliharaan, pemeliharaan prediktif berbasis AI, sistem pemantauan dan pengintegrasian jarak jauh, dan pengukuran kinerja sistem yang menjelaskan teori dan pengertian materi yang berkaitan dengan penelitian yang dilakukan.

BAB III. METODOLOGI PENELITIAN

Bab metodologi penelitian ini berisi tentang metodologi yang dilakukan pada penelitian ini, seperti waktu dan tempat penelitian, *capstone project*, perangkat penelitian, sumber data, tahapan penelitian, pengolahan data, pengembangan model, implementasi sistem, dan evaluasi sistem.

BAB IV. HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan ini membahas hasil dan bahasan setiap proses penelitian yang dilakukan pada penelitian ini, seperti hasil dan pembahasan dari pengolahan data, pengembangan model, implementasi sistem, dan evaluasi sistem.

BAB V. PENUTUP

Bab penutup ini membahas kesimpulan dari penelitian yang dilakukan ini, beserta saran untuk penelitian selanjutnya terkait pengembangan dari penelitian ini.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian ini dibuat dengan mempertimbangkan penelitian-penelitian terdahulu. Penelitian yang berjudul “*Optimization of Design Parameters in LSTM Model for Predictive Maintenance*” oleh Do-Gyun Kim dan Jin-Young Choi pada tahun 2021, bertujuan untuk meningkatkan akurasi prediksi degradasi *bearing* dengan mengintegrasikan analisis korelasi fitur dan algoritma genetika untuk optimasi *hyperparameter LSTM* [4]. Penelitian menggunakan dataset NASA dan memilih indikator kesehatan (*Health Indicator* atau HI) melalui analisis korelasi untuk mengecualikan fitur yang redundan penyebab multikolinearitas antarfitur. Fitur seperti *kurtosis*, *skewness*, dan *crest factor* terbukti menghasilkan akurasi validasi tertinggi sebesar 98,14%. Selain itu, terdapat algoritma genetika yang dirancang untuk menentukan konfigurasi *hyperparameter* yang optimal. Namun, penelitian ini tidak mengeksplorasi pendekatan berbasis *autoencoder* yang dapat mendeteksi pola anomali dalam data.

Selanjutnya, terdapat penelitian yang berjudul “*A Hybrid of Convolutional Neural Network and Long Short-Term Memory Network Approach to Predictive Maintenance*” oleh Ahmed Nasser dan Huthaifa Al-Khazraji pada tahun 2022, yang bertujuan untuk meningkatkan akurasi prediksi dalam sistem pemeliharaan prediktif dengan memadukan kemampuan ekstraksi fitur dari *Convolutional Neural Network* (CNN) dan analisis data deret waktu dari *Long Short-Term Memory* (LSTM) [5]. Penelitian ini menggunakan data sensor historis dari mesin industri yang diperoleh dari repositori GitHub Microsoft untuk membangun model *hybrid*

CNN-LSTM. Model tersebut dibandingkan dengan LSTM reguler, menghasilkan peningkatan akurasi rata-rata sebesar 4,44% (dari 93,34% menjadi 97,48%). Hasil ini menunjukkan bahwa pendekatan *hybrid* tidak hanya lebih akurat tetapi juga mengurangi kompleksitas model. Namun, penelitian ini tidak membahas pendeteksi anomali data.

Kemudian, terdapat penelitian yang berjudul “*LSTM-Autoencoder Deep Learning Model for Anomaly Detection in Electric Motor*” oleh Fadhila Lachekhab, Messouada Benzaoui, Sid Ahmed Tadjer, Abdelkrim Bensmaine, dan Hichem Hamma pada tahun 2024, bertujuan untuk mendeteksi anomali pada data getaran motor listrik menggunakan LSTM *Autoencoder* [6]. Data diambil dari sensor akselerometer pada tiga sumbu (aksial, radial, dan tangensial) untuk kondisi normal dan tidak seimbang. Model LSTM *Autoencoder* berhasil merekonstruksi pola data normal dengan eror rekonstruksi signifikan lebih rendah dibandingkan *autoencoder* biasa (0.0003 vs. 0.4) dan MSE anomali (10^{-16} vs. 10^{-7}). Studi ini menunjukkan efektivitas LSTM *Autoencoder* dalam mendeteksi anomali pada data. Namun, penelitian ini tidak mencakup memprediksi hasil keluaran mesin.

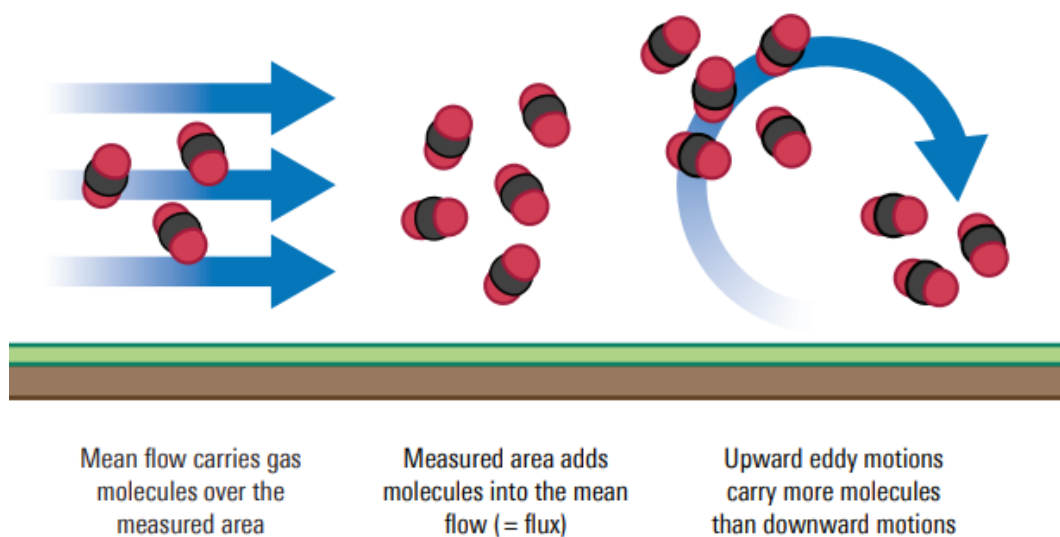
Berdasarkan penelitian-penelitian terdahulu, beberapa di antaranya menunjukkan potensi LSTM *Autoencoder* dan model berbasis LSTM khususnya *hybrid* CNN-LSTM dalam menangani data *time-series* yang kompleks. Namun, terdapat beberapa gap penelitian pada setiap jurnalnya. Penelitian ini mengisi gap tersebut dengan memanfaatkan dataset dari perangkat dan membangun model berbasis LSTM *Autoencoder* untuk mendeteksi anomali dan *hybrid* CNN-LSTM untuk memprediksi dengan akurasi yang lebih.

2.2. Analisis Fluks Karbon di Perkebunan Kelapa Sawit

Perkebunan kelapa sawit merupakan salah satu ekosistem yang signifikan dalam siklus karbon global. Ekosistem ini dapat berpotensi menyerap karbon dioksida melalui proses fotosintesis, tetapi juga dapat melepas emisi karbon melalui proses

respirasi, dekomposisi bahan organik, dan pengolahan limbah [7]. Dinamika fluks di daerah perkebunan kelapa sawit dapat dipantau melalui metode pengukuran *eddy covariance*.

Metode *eddy covariance* merupakan teknik untuk mengukur pertukaran (*flux*) gas, uap air, atau energi antara permukaan dan atmosfer [8]. Prinsip utamanya terletak pada pengamatan pusaran udara kecil (*eddies*) yang terbentuk akibat aliran angin tidak seragam di dekat permukaan. Pusaran ini mengangkut molekul gas dan uap air naik serta turun, sehingga menciptakan sirkulasi dinamis di atas suatu wilayah. Cara kerja metode ini melibatkan pengukuran secara simultan dan berkelanjutan terhadap konsentrasi gas, misalnya CO₂, serta komponen kecepatan angin, termasuk kecepatan vertikalnya, seperti ditunjukkan pada Gambar 2.1 Metode *Eddy Covariance*.



Gambar 2.1 Metode *Eddy Covariance* [8]

Pada Gambar 2.1 Metode *Eddy Covariance* terdapat proses yang menjelaskan bahwa aliran udara utama membawa molekul gas melewati area pengukuran yang menambah molekul gas ke dalam aliran tersebut, sehingga tercipta *flux*. Akibatnya, pusaran udara yang bergerak ke atas membawa lebih banyak molekul dibandingkan dengan pusaran yang bergerak ke bawah, menunjukkan adanya pertukaran gas dari area tersebut ke atmosfer. Ketika *eddy* bergerak ke atas, udara yang mengandung

CO₂ dari dekat permukaan terbawa menuju bagian atas atmosfer; sebaliknya, pergerakan *eddy* ke bawah membawa udara dengan konsentrasi CO₂ yang berbeda turun kembali mendekati permukaan. Melalui catatan perubahan konsentrasi gas dan variasi kecepatan angin dalam jangka waktu tertentu, dapat dihitung seberapa besar *net flux* gas tersebut yang mengalir melewati bidang pengukuran.

Secara sederhana, ketika lebih banyak CO₂ dibawa ke atas daripada ke bawah, wilayah tersebut terpantau sebagai sumber CO₂, sedangkan jika kebalikannya terjadi, wilayah tersebut dianggap menyerap CO₂ dari atmosfer [8]. Melalui pengamatan pergerakan *eddies* dan konsentrasi gas ini, metode *eddy covariance* memberikan informasi tentang arah serta besarnya pertukaran gas antara permukaan dan atmosfer.

2.3. Sistem Pemantauan Lingkungan Eddy Covariance Station

Sistem pemantauan lingkungan adalah pendekatan berbasis teknologi yang dirancang untuk mengukur dan memantau parameter lingkungan dengan tujuan mendukung pengelolaan ekosistem, mitigasi perubahan iklim, dan penelitian ilmiah. Sistem ini memanfaatkan berbagai sensor untuk mengumpulkan data fisik dan kimia dari lingkungan, seperti suhu, kelembapan, tekanan udara, kecepatan angin, dan konsentrasi gas rumah kaca. Pada penerapannya, teknologi *Internet of Things* (IoT) menjadi elemen penting yang mengintegrasikan sensor, perangkat keras, dan jaringan komunikasi untuk memastikan pengumpulan, pengolahan, dan analisis data yang efisien.

Sistem pemantauan lingkungan berbasis IoT dipasang di PT Persada Dinamika Lestari untuk memantau lingkungan di perkebunan kelapa sawit. Perangkat Eddy Covariance Station ini terdiri dari C-Flux Analyzer yaitu Gas Analyzer dan 3D Anemometer, serta Microclimate Station (MCS). Komponen-komponen ini bekerja bersama untuk menyediakan data yang akurat dan terintegrasi guna mendukung pengambilan keputusan berbasis data.



Gambar 2.2 Pengaplikasian Perangkat C-Flux Analyzer

Pemasangan perangkat C-Flux Analyzer di PT Persada Dinamika Lestari ditunjukkan pada Gambar 2.2 Pengaplikasian Perangkat C-Flux Analyzer. Kerangka pelindung berbentuk kanopi sejumlah 4 buah adalah kerangka pelindung perangkat Gas Analyzer, sedangkan kerangka pelindung berbentuk kubus adalah kerangka pelindung perangkat 3D Anemometer. Pemasangan kedua perangkat ini berada di tempat dengan ketinggian yang sama sesuai dengan syarat perhitungan metode *Eddy Covariance*. Masing-masing terdapat dua perangkat stasiun, stasiun yang pertama berada di ketinggian 3 meter dan stasiun yang kedua berada di ketinggian 14,5 meter.

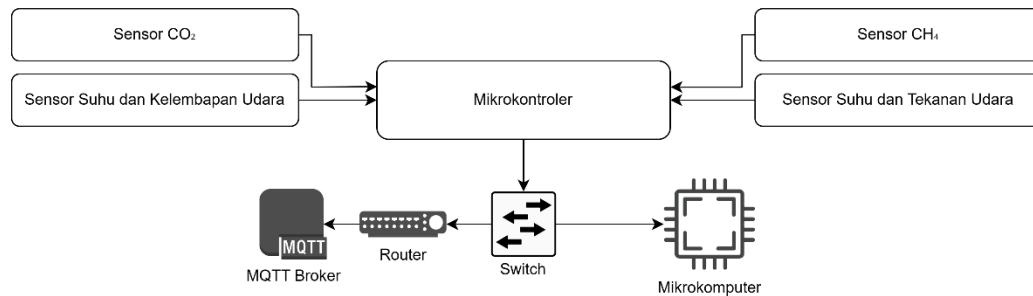


Gambar 2.3 Pengaplikasian Perangkat Microclimate Station

Sedangkan pemasangan perangkat Microclimate Station di PT Persada Dinamika Lestari ditunjukkan pada Gambar 2.3 Pengaplikasian Perangkat Microclimate Station. Perangkat Microclimate Station dipasang untuk mengukur parameter cuaca di perkebunan kelapa sawit tersebut.

2.3.1. Sistem Perangkat Gas Analyzer

Perangkat Gas Analyzer adalah perangkat yang memiliki tujuan utama untuk mendeteksi dan mengukur konsentrasi gas di udara. Sistem perangkat Gas Analyzer dapat dilihat pada Gambar 2.4 Sistem Gas Analyzer.

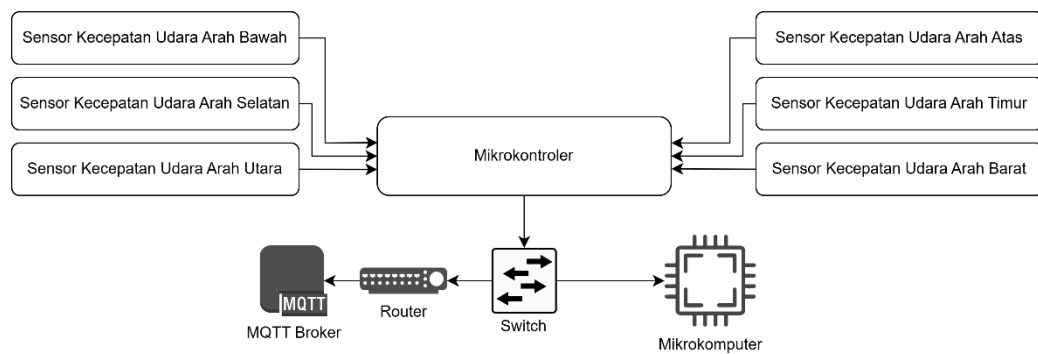


Gambar 2.4 Sistem Gas Analyzer

Pada Gambar 2.4 Sistem Gas Analyzer terdapat blok diagram sistem perangkat Gas Analyzer bekerja. Perangkat bekerja dengan mendeteksi dan mengukur konsentrasi gas di lingkungan menggunakan sensor karbon dioksida (CO_2), sensor metana (CH_4), sensor untuk suhu dan tekanan udara, sensor suhu dan kelembapan udara di lingkungan. Data dari sensor-sensor ini diproses oleh mikrokontroler sebelum diteruskan ke Raspberry Pi, yang bertugas mengelola pengiriman data ke *cloud* menggunakan MQTT broker.

2.3.2. Sistem 3D Anemometer

Perangkat 3D Anemometer digunakan untuk mengukur arah dan kecepatan angin. Sistem perangkat 3D Anemometer ditunjukkan pada Gambar 2.5 Sistem 3D Anemometer.

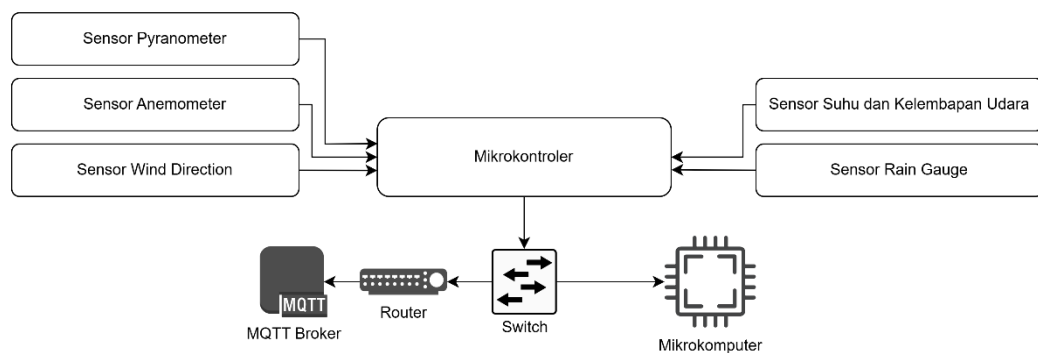


Gambar 2.5 Sistem 3D Anemometer

Pada Gambar 2.5 Sistem 3D *Anemometer* terdapat blok diagram sistem bekerja, perangkat bekerja dengan menggunakan 6 Digital Airspeed Sensors yang mengukur kecepatan angin dari enam arah utama, yaitu: utara, timur, selatan, barat, atas, dan bawah. Data dari keenam sensor ini dikumpulkan dan diproses oleh mikrokontroler, kemudian diteruskan ke Raspberry Pi, yang bertugas mengelola pengiriman data ke *cloud* menggunakan MQTT broker.

2.3.3. Sistem Microclimate Station

Perangkat Microclimate Station adalah perangkat dengan sistem terintegrasi yang menggabungkan berbagai sensor untuk memantau kondisi cuaca di perkebunan kelapa sawit. Sistem perangkat Microclimate Station dapat dilihat pada Gambar 2.6 Sistem Microclimate Station.



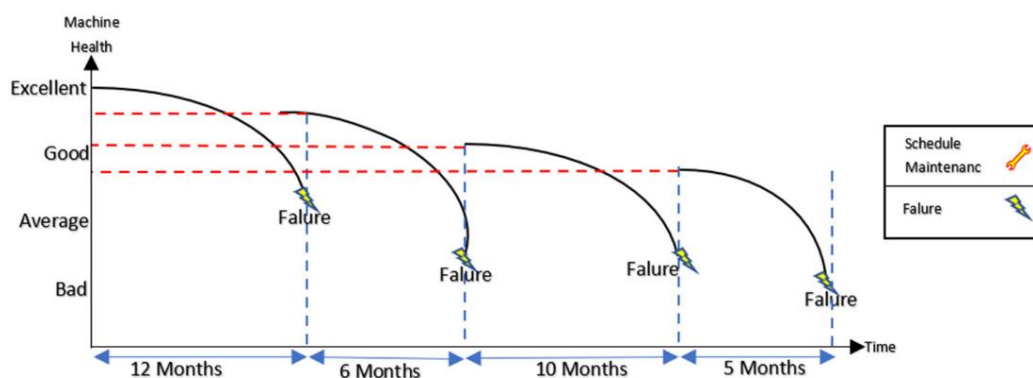
Gambar 2.6 Sistem Microclimate Station

Pada Gambar 2.6 Sistem Microclimate Station terdapat blok diagram sistem bekerja, perangkat bekerja dengan mengintegrasikan berbagai sensor untuk memantau parameter lingkungan. Terdapat sensor pyranometer atau sensor pengukur radiasi matahari, sensor anemometer pengukur kecepatan angin, sensor wind direction pendeteksi arah angin, sensor suhu dan kelembapan udara, serta sensor rain gauge pengukur curah hujan. Data dari keenam sensor ini dikumpulkan dan diproses oleh mikrokontroler, kemudian diteruskan ke Raspberry Pi, yang bertugas mengelola pengiriman data ke *cloud* menggunakan MQTT broker.

2.4. Sistem Pemeliharaan

Pemeliharaan atau *maintenance* berarti menjaga sesuatu agar tetap di kondisi yang baik. Sesuatu di sini dapat berupa perangkat elektronik, seperti perangkat Eddy Covariance Station (ECS). Perangkat ECS juga memerlukan pemeliharaan agar komponen yang berada di dalamnya dapat terjaga dengan baik dan dapat melakukan kerja dengan optimal. Umumnya sistem pemeliharaan dibagi menjadi tiga, yaitu pemeliharaan korektif, pemeliharaan prefentif, dan pemeliharaan prediktif [9].

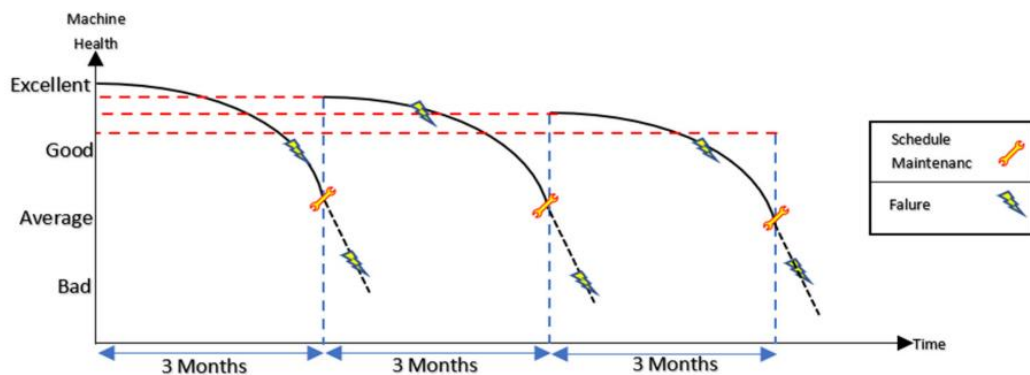
Pemeliharaan korektif atau *corrective maintenance* dilakukan ketika perangkat mengalami kegagalan sistem dan berhenti kerja. Strategi ini bertujuan untuk memperbaiki atau mengganti komponen yang rusak agar peralatan kembali berfungsi.



Gambar 2.7 Strategi Pemeliharaan Korektif [9]

Berdasarkan Gambar 2.7 Strategi Pemeliharaan Korektif, perbaikan dilakukan setelah kegagalan terjadi. Namun, perbaikan yang dilakukan ketika mesin tersebut rusak dapat menyebabkan kesehatan perangkat berangsur-angsur menurun dengan waktu produktif sistem bekerja tidak menentu di setiap masanya, sehingga tidak dapat diprediksi waktu perbaikannya. Perangkat Eddy Covariance Station melakukan sistem pemeliharaan ini pada awalnya yang menyebabkan salah satu sensor mengalami penurunan performa.

Sistem pemeliharaan selanjutnya adalah pemeliharaan preventif. Pemeliharaan preventif atau *preventive maintenance* dilakukan secara berkala untuk mencegah kerusakan tidak terduga pada perangkat.

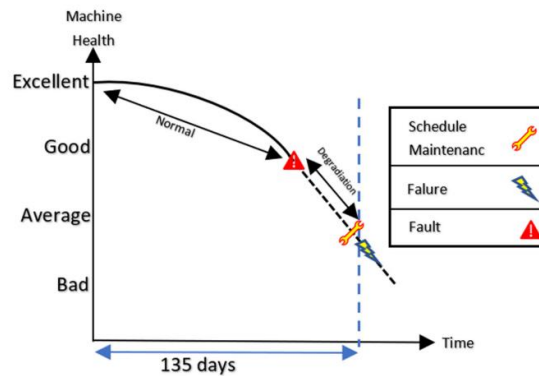


Gambar 2.8 Strategi Pemeliharaan Preventif [9]

Berdasarkan Gambar 2.8 Strategi Pemeliharaan Preventif, perbaikan dilakukan secara terjadwal selama 3 bulan untuk menjaga kesehatan mesin tetap optimal dan menghindari penurunan signifikan. Meskipun begitu, kerusakan perangkat masih bisa terjadi sebelum atau sesudah jadwal pemeliharaan [9]. Perangkat ECS juga telah melakukan pemeliharaan preventif setelah melakukan pemeliharaan korektif, tetapi perangkat mengalami kerusakan sebelum jadwal pemeliharaan.

Sistem pemeliharaan yang terakhir adalah pemeliharaan prediktif. Pemeliharaan prediktif atau *predictive maintenance* memanfaatkan teknologi untuk memantau

kondisi perangkat dan memprediksi kapan kegagalan akan terjadi, sehingga dapat melakukan pemeliharaan sebelum kegagalan terjadi.



Gambar 2.9 Strategi Pemeliharaan Prediktif [9]

Berdasarkan Gambar 2.9 Strategi Pemeliharaan Prediktif, kesehatan mesin awalnya dalam kondisi sangat baik, lalu secara bertahap menurun hingga mencapai ambang kegagalan. Pemeliharaan prediktif ini mendeteksi degradasi dan memberikan peringatan dini agar tindakan pemeliharaan dapat dilakukan tepat sebelum kegagalan terjadi. Salah satu penerapan pemeliharaan prediksi yang bisa digunakan adalah pemeliharaan prediksi berbasis AI.

2.5. Pemeliharaan Prediktif Berbasis AI di Industri Algikultur

Penerapan *Artificial Intelligence* (AI) di industri agrikultur bertujuan meningkatkan efisiensi, produktivitas, dan keberlanjutan melalui berbagai aplikasi berbasis data. AI digunakan untuk menganalisis data sensor seperti dari tanah, irigasi, dan cuaca untuk mendeteksi anomali atau pola yang mengindikasikan potensi masalah, seperti kekeringan, penyakit tanaman, atau kerusakan peralatan [10].

Pemeliharaan prediktif berbasis kecerdasan buatan (AI) adalah pendekatan inovatif yang menggunakan teknologi AI untuk mendeteksi potensi kerusakan pada mesin sebelum terjadi kegagalan. Pada penelitian ini, AI digunakan untuk menganalisis

data sensor perangkat Eddy Covariance Station yang berupa *time series*, mendeteksi anomali, dan memprediksi potensi kerusakan perangkat.

Artificial Intelligence (AI) mencakup keseluruhan *machine learning*, sedangkan *deep learning* secara keseluruhan berada pada cakupan *machine learning*. *Artificial Intelligence* atau AI merupakan metode yang membuat komputer meniru kecerdasan manusia, *machine learning* berfungsi agar komputer memiliki kemampuan dalam pembelajaran data baru tanpa perlu diprogram secara eksplisit, dan *deep learning* merupakan bagian dari *machine learning* yang dibangun menggunakan jaringan saraf tiruan atau *neural network*.

Pendekatan *deep learning*, seperti *hybrid CNN-LSTM* (*Convolutional Neural Network - Long Short-Term Memory*) dan *LSTM autoencoder*, menjadi pilihan utama karena kemampuannya dalam menangani data yang memiliki dependensi temporal. Ada dua tahapan penting pada proses ini, yang pertama adalah persiapan data dan kedua adalah pembuatan model.

2.5.1. Persiapan Data

Data divisualisasi melalui grafik *line chart*. Selanjutnya, data dibersihkan (*data cleaning*). Proses pembersihan data ini mencakup penanganan *outlier*. Data yang hilang juga diatasi menggunakan metode imputasi, seperti *linear interpolation* agar data siap digunakan [11].

Linear Interpolation (LI) adalah metode estimasi nilai hilang dengan menghubungkan dua titik data yang diketahui menggunakan garis lurus. Persamaan *interpolasi linear* ditujukan pada persamaan (2.1) [11].

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \quad (2.1)$$

Keterangan:

x_1 : Koordinat titik awal sumbu horizontal

y_1 : Koordinat titik awal sumbu vertikal

x_2 : Koordinat titik akhir sumbu horizontal

y_2 : Koordinat titik akhir sumbu vertikal

x : Nilai x antara x_1 dan x_2 yang ingin dihitung nilai y -nya.

y : Nilai interpolasi yang dihitung untuk x .

Sebelum data sensor digunakan dalam pemodelan *hybrid* CNN-LSTM dan LSTM *autoencoder*, data harus terlebih dahulu dinormalisasi menggunakan *Min-Max Scalar*. Teknik ini bertujuan untuk mengubah setiap nilai X dalam dataset ke dalam rentang $[0, 1]$ berdasarkan nilai minimum dan maksimum dari distribusi data, sesuai dengan persamaan (2.2) [12].

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.2)$$

Keterangan:

X_{scaled} : Nilai setelah dinormalisasi (*scaled value*).

X : Nilai mentah yang akan dicari nilai standarnya

X_{min} : Nilai minimum dalam dataset.

X_{max} : Nilai maximum dalam dataset.

2.5.2. Pembuatan Model

Fokus pada pemeliharaan prediktif didasari dengan mendeteksi anomali dan memprediksi kapan model akan mengalami kerusakan. Model LSTM *autoencoder* dapat digunakan untuk mendeteksi anomali sedangkan model *hybrid* CNN-LSTM digunakan untuk model prediksi. Model *hybrid* CNN-LSTM dan LSTM *autoencoder* dapat dibangun di Python, khususnya menggunakan *framework* TensorFlow [13].

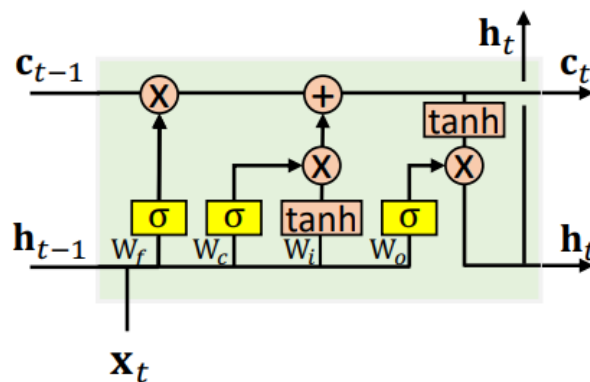
CNN (*Convolutional Neural Network*) adalah jenis jaringan saraf tiruan (*neural network*) yang dirancang khusus untuk memproses data yang memiliki struktur grid, seperti data *time-series*. Model CNN terdiri dari beberapa lapisan yang

berbeda, masing-masing dengan fungsi spesifik untuk mengekstraksi dan memproses fitur dari data input.

Convolutional 1D Layer (Conv1D) adalah salah satu jenis lapisan dalam jaringan saraf *convolutional* yang dirancang untuk memproses data satu dimensi, seperti *time series* atau deret waktu. Parameter penting pada Conv1D meliputi *filters*, *kernel size*, dan *padding*. *filters* menentukan jumlah fitur output yang dihasilkan, *kernel size* menunjukkan panjang jendela konvolusi untuk menangkap pola lokal dalam data, dan *padding = causal* menjaga agar *output* hanya dipengaruhi oleh data saat ini dan sebelumnya, sehingga cocok untuk prediksi *time-series* tanpa kebocoran informasi dari masa depan.

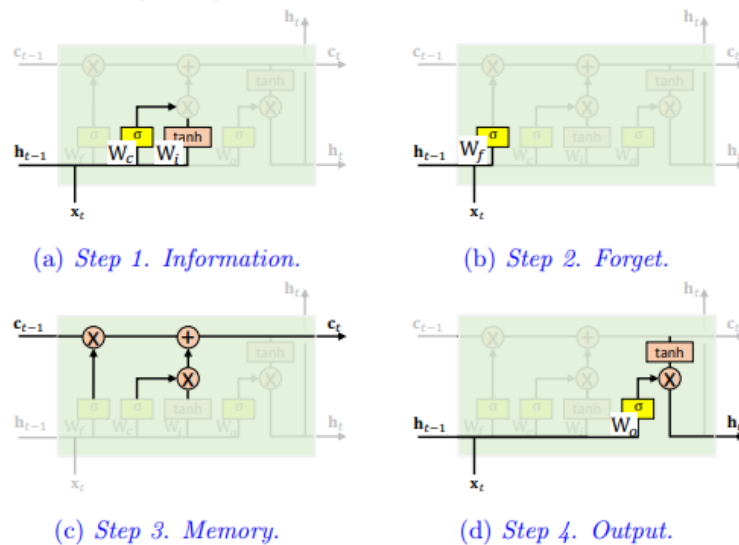
MaxPooling1D adalah lapisan yang digunakan untuk mengurangi dimensi temporal dari *output* Conv1D dengan mengambil nilai maksimum dari setiap segmen kecil data. Proses ini membantu menyederhanakan data sambil tetap mempertahankan informasi penting. Parameter utama pada MaxPooling1D adalah *pool size*, yaitu ukuran segmen yang diproses sekaligus.

LSTM (*Long Short-Term Memory*) merupakan salah satu jenis RNN (*Recurrent Neural Network*) yang digunakan untuk membuat model dari data *time-series*, khususnya pada pemeliharaan prediktif karena kemampuannya dalam memahami pola dan korelasi jangka panjang [6].



Gambar 2.10 Arsitektur Sel LSTM [14]

Arsitektur pada LSTM secara keseluruhan ditunjukkan pada Gambar 2.10 Arsitektur Sel LSTM. Bagian dalam sebuah sel LSTM terdiri dari tiga gerbang, yaitu *input* (i), *output* (o), dan *forget* (f), yang menggunakan fungsi aktivasi *sigmoid* (σ) ditunjukkan dengan kotak kuning, serta mendapatkan informasi relevan melalui fungsi aktivasi *tanh* yang ditunjukkan dengan kotak jingga. Sel LSTM tersebut memproses satu observasi pada satu waktu dalam *sequence window*. Ketika sel tersebut memproses *timestep* x_t , input x_t mengalir ke dalam sel, diproses melalui jalur yang ditunjukkan pada Gambar 2.10 Arsitektur LSTM dengan mempertimbangkan keluaran sebelumnya h_{t-1} dan keadaan sel sebelumnya c_{t-1} , sehingga menghasilkan keluaran yang diperbarui h_t dan sel yang diperbarui c_t [14].

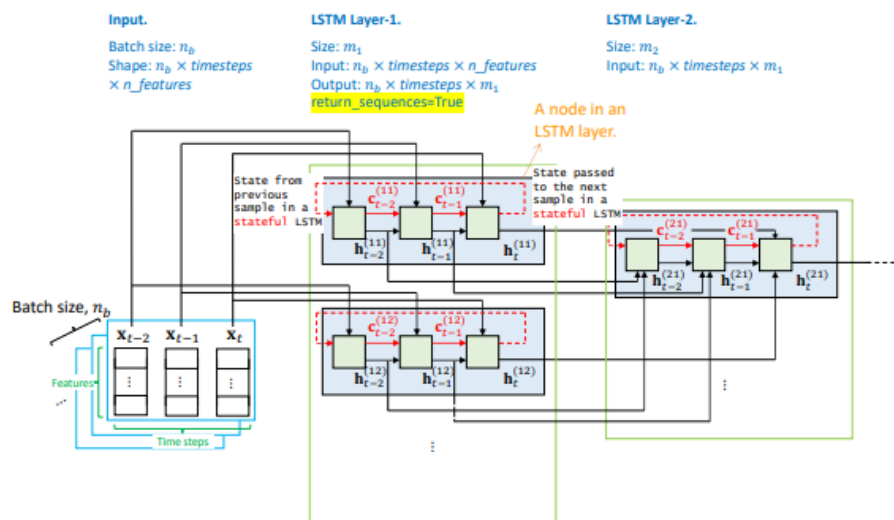


Gambar 2.11 Langkah-langkah Operasi di Sel LSTM [14]

Pada Gambar 2.11 Langkah-langkah Operasi di Sel LSTM terdapat empat langkah operasi yang terjadi pada sel LSTM. Langkah pertama *information*, langkah kedua *forget*, langkah ketiga *memory*, dan yang terakhir *output* [14].

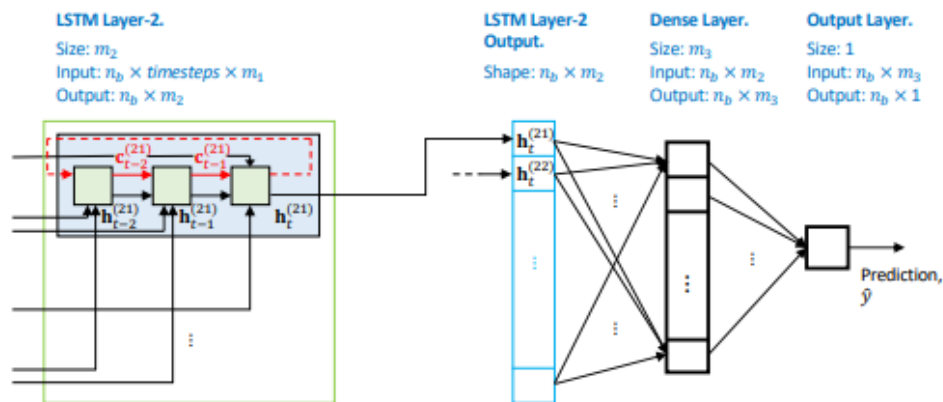
Langkah pertama adalah mempelajari informasi dari input x_t dan *output* sebelumnya h_{t-1} . Informasi relevan dalam x_t dihitung dengan fungsi aktivasi *tanh* $(-1, 1)$, sedangkan relevansi informasi diukur dengan fungsi aktivasi *sigmoid* $(0, 1)$, yang berarti mendekati 0 informasi tidak relevan dan mendekati 1 berarti relevan[14]. Langkah kedua, *forget*, informasi baru x_t dapat membuat memori

sebelumnya tidak relevan dan dilupakan. Keputusan ini diambil dengan fungsi aktivasi *sigmoid*, menghasilkan indikator antara 0 dan 1. Jika indikator mendekati nol, memori lama digantikan oleh informasi baru. Jika mendekati satu, memori lama diteruskan, meskipun informasi dalam x_t tetap bisa masuk ke dalam memori [14]. Langkah ketiga, *memory*, melibatkan dua komponen utama: pertama, memori lama c_{t-1} diputuskan untuk diteruskan atau dilupakan oleh gerbang *forget* f_t ; kedua, informasi baru dari x_t dihitung sebagai $\tilde{c}_t$ menggunakan *tanh* dan diperiksa relevansinya dengan indikator i_t . Informasi relevan ($i_t \cdot \tilde{c}_t$) kemudian digabungkan dengan memori lama ($f_t \cdot c_{t-1}$) untuk menghasilkan memori sel baru (c_t). Proses ini menjaga nilai memori dalam rentang $[-1,1]$ untuk mencegah inflasi [14]. Pada langkah terakhir, *output* h_t berasal dari dua komponen: gerbang *output* o_t (rentang 0-1) dan *state* sel c_t yang diaktifkan dengan *tanh*, menjadikannya memori jangka pendek (*short state*) [14]. Hasilnya: (a) jika $|\tilde{c}_t| \approx 0$ dan $i_t \approx 0$, tidak ada informasi baru di x_t ; (b) jika $|\tilde{c}_t| \approx 1$ dan $i_t \approx 0$, x_t memiliki informasi redundan; (c) jika $|\tilde{c}_t| \approx 0$ dan $i_t \approx 1$, model mungkin tidak konsisten; (d) jika $|\tilde{c}_t| \approx 1$ dan $i_t \approx 1$, x_t mengandung informasi baru yang relevan [14]. Proses ini memungkinkan LSTM untuk memanfaatkan pola jangka panjang dan pendek dalam data, sehingga cocok untuk pemodelan *time-series*. Kemampuan LSTM ketika mempertahankan informasi relevan dalam memori dan melupakan informasi yang tidak relevan membuatnya efektif dalam menangkap hubungan temporal yang kompleks.



Gambar 2.12 Struktur Jaringan LSTM [14]

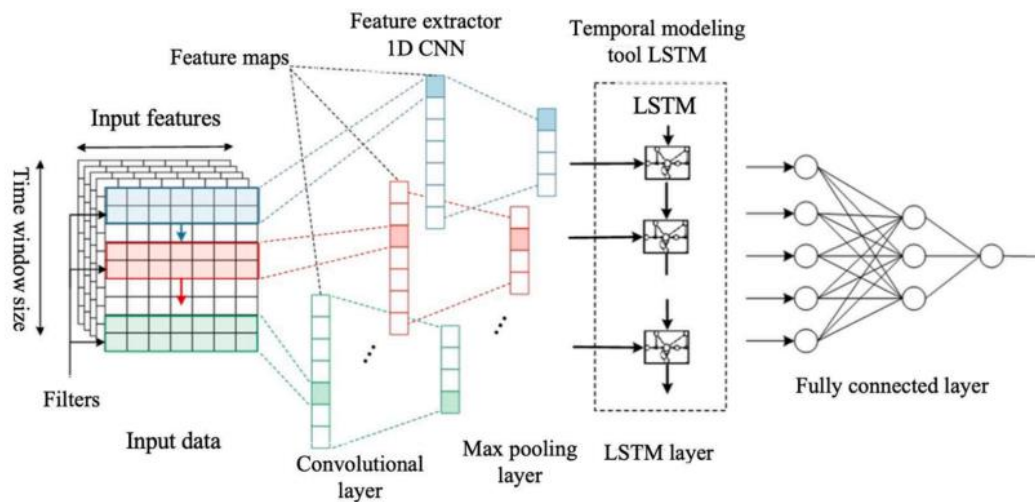
Pada Gambar 2.12 Struktur Jaringan *LSTM*, jaringan *LSTM* memproses input dalam bentuk *batch* data 3D yang mencakup *batch*, jendela waktu (*time-window*), dan fitur, dengan setiap sampel dalam *batch* berupa *array* 2D. Sel dalam lapisan tersembunyi diuraikan berdasarkan iterasi langkah waktu, dan informasi ditransmisikan setiap langkah waktu. Lapisan *LSTM* pertama menghasilkan *output* sekuensial (dengan *return_sequences=True*), yang diproses lebih lanjut oleh lapisan kedua. Jika model bersifat *stateful*, *cell state* dari *batch* sebelumnya disimpan untuk *batch* berikutnya, mempertahankan informasi lintas *batch* [14].



Gambar 2.13 Jaringan *LSTM Restricted* [14]

Pada jaringan *LSTM restricted* yang ditunjukkan pada Gambar 2.13 Jaringan *LSTM Restricted*, lapisan terakhir hanya mengeluarkan keluaran tersembunyi pada langkah waktu terakhir h_t , yang membentuk vektor peta fitur untuk diumpankan ke lapisan *dense*, dengan *return_sequences=False*.

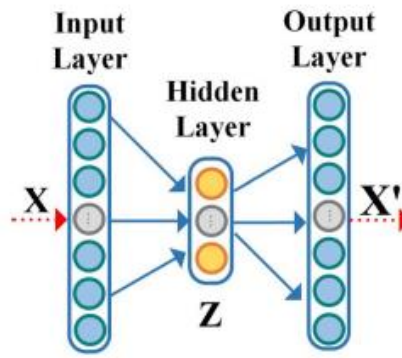
Model *hybrid* CNN-LSTM adalah kombinasi dari *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) yang dirancang untuk memanfaatkan keunggulan masing-masing model dalam memproses data. CNN dikenal efektif untuk mengekstraksi fitur spasial, sedangkan LSTM unggul dalam menangkap pola temporal pada data sekuensial. Kombinasi ini memungkinkan model mengolah data secara lebih holistik, baik dari aspek lokal maupun temporal.



Gambar 2.14 Ilustrasi Model *Hybrid CNN-LSTM* [15]

Ilustrasi terdapat pada Gambar 2.14 Ilustrasi Model *Hybrid CNN-LSTM*. Pada langkah awal, CNN digunakan untuk mengekstraksi fitur lokal dari data input. Pada data *time-series*, CNN membantu mengenali pola atau tren lokal yang mungkin berulang. Hasil ekstraksi ini berupa representasi fitur yang lebih kompak dan informatif, yang kemudian diteruskan ke LSTM. LSTM kemudian menerima fitur-fitur yang telah diekstraksi oleh CNN. LSTM memiliki kemampuan mengingat informasi urutan data, menganalisis hubungan temporal antarfitur, memungkinkan model untuk memahami pola yang melibatkan waktu atau sekuensial. Langkah ini penting untuk aplikasi seperti prediksi atau deteksi pola dalam data *time-series*. Setelah pemrosesan oleh LSTM, hasil akhirnya diteruskan ke lapisan *fully connected* untuk menghasilkan prediksi atau klasifikasi. Melalui hal ini, model *hybrid CNN-LSTM* mampu mengolah data yang kompleks secara efisien, menjadikannya cocok untuk tugas seperti prediksi *time-series*.

Autoencoder adalah jenis algoritma yang memiliki tujuan utama untuk mempelajari representasi data yang informatif sehingga dapat digunakan untuk berbagai aplikasi, dengan cara belajar merekonstruksi serangkaian pengamatan input dengan cukup baik [16].



Gambar 2.15 Struktur *Autoencoder* [17]

Pada Gambar 2.15 Struktur *Autoencoder* menggambarkan struktur sebuah *autoencoder*, X mewakili data input pada lapisan input, Z mewakili data pada lapisan tersembunyi (*hidden layer*), dan X' mewakili data *output* yang direkonstruksi pada lapisan *output*. Fungsi objektif dari *autoencoder* dapat dituliskan sebagai persamaan (2.3) [17].

$$\min_{\theta} J_{AE}(\theta) = \min_{\theta} \sum_{i=1}^n l(x_i, x'_i) = \min_{\theta} \sum_{i=1}^n l(x_i, g_{\theta}(f_{\theta}(x'_i))) \quad (2.3)$$

Keterangan:

J_{AE} : Fungsi objektif *autoencoder* yang menggambarkan total eror rekonstruksi.

$\min_{\theta}$: Proses minimasi untuk menemukan parameter θ yang menghasilkan eror rekontruksi kecil.

θ : Parameter mencakup bobot (W, W') dan bias (b, b') dari fungsi f_{θ} dan g_{θ} .

f_{θ} : Fungsi *encoding*.

g_{θ} : Fungsi *decoding*.

x_i : Mewakili dimensi ke- i dari sampel pelatihan.

x'_i : Mewakili dimensi ke- i dari data *output* hasil rekontruksi.

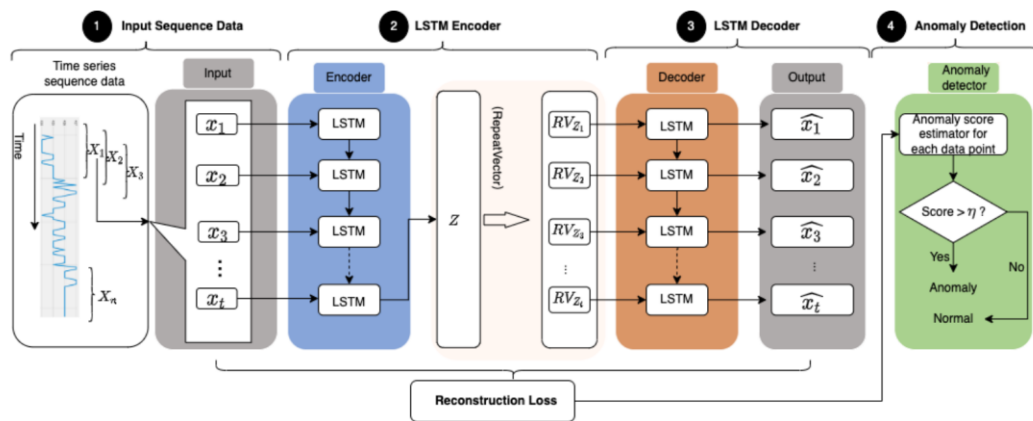
n : Jumlah total data pelatihan.

l : Error rekontruksi antara input dan *output*.

Persamaan ini menunjukkan bahwa *autoencoder* dilatih untuk meminimalkan eror rekonstruksi di seluruh dataset dengan cara mencari parameter θ terbaik. Fungsi

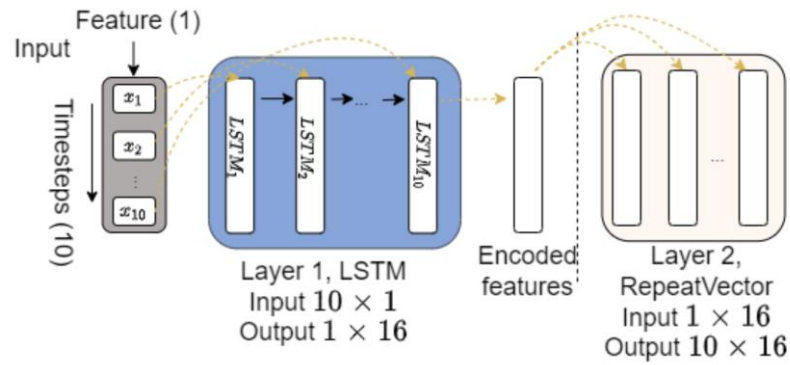
encoding (f_{θ}) dan *decoding* (g_{θ}) bekerja sama untuk merepresentasikan data input dan merekonstruksinya kembali ke *output* x' yang mirip dengan input asli. Konsep ini menjadi dasar dalam mendeteksi anomali [17].

Autoencoder dilatih untuk merekonstruksi data yang normal atau data yang merepresentasikan pola umum. Jika ada data input yang sangat berbeda dari data yang dilatih (*anomalous data*), hasil rekonstruksinya cenderung buruk, sehingga error rekonstruksi (I) menjadi tinggi. Data dengan error rekonstruksi yang tinggi dapat dianggap sebagai anomali, karena tidak sesuai dengan pola umum yang dipelajari oleh model [17].



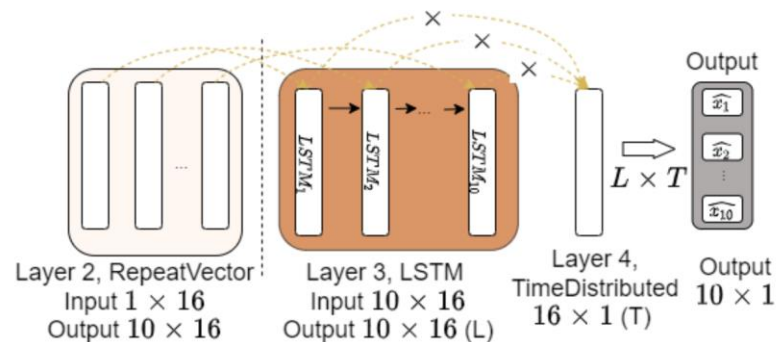
Gambar 2.16 Overview Model LSTM Autoencoder [18]

LSTM *Autoencoder* adalah variasi lanjutan dari *autoencoder* berulang (*recurrent autoencoder*), yang dirancang khusus untuk menangkap representasi dari data berurutan [17]. Ilustrasi model ini ditunjukkan pada Gambar 2.16 Overview Model LSTM *Autoencoder*. Pada arsitektur ini, baik komponen *encoder* maupun *decoder* dibangun menggunakan unit LSTM. *Encoder* LSTM menerima input berupa urutan vektor, yang dapat mewakili fitur, sebaliknya, *decoder* LSTM merekonstruksi urutan input asli, sering kali dalam urutan terbalik [17]. Jika perbedaan antara data asli dan hasil rekonstruksi terlalu besar, maka data tersebut dapat dianggap sebagai anomali [18].



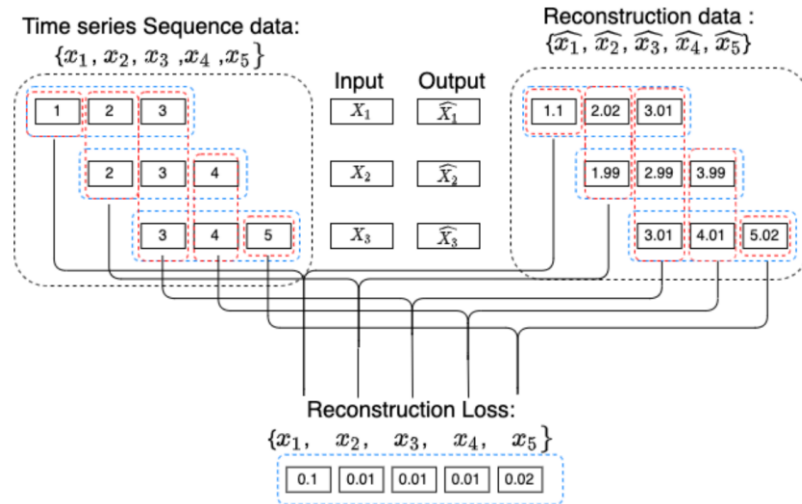
Gambar 2.17 Detail LSTM Encoder [18]

Berdasarkan Gambar 2.17 Detail LSTM Encoder, LSTM encoder mengubah data berdimensi tinggi menjadi representasi laten yang lebih kecil. Data dalam bentuk sekuensial dengan panjang jendela waktu tertentu diproses oleh unit LSTM secara berurutan. Hasil akhirnya adalah *encoded feature* dengan dimensi 1×16 . Guna memungkinkan proses rekonstruksi, *RepeatVector* digunakan untuk mengulang *encoded feature* sebanyak jumlah *timestep* (misalnya 10). Ini memastikan bahwa representasi laten dapat diproses kembali dalam bentuk sekuensial oleh *decoder*.



Gambar 2.18 Detail LSTM Decoder [18]

Berdasarkan Gambar 2.18 Detail LSTM Decoder, LSTM decoder merekonstruksi data dari *encoded feature*. Input ke *decoder* berupa 10×16 , yang dihasilkan oleh *RepeatVector*. Setiap unit LSTM memproses satu langkah waktu dan meneruskan informasi ke unit berikutnya. Setelah semua unit selesai, lapisan *TimeDistributed Dense* memastikan keluaran memiliki dimensi yang sama dengan input asli, yaitu 10×1 .

Gambar 2.19 Detail *Reconstruction Loss* [18]

Berdasarkan Gambar 2.19 Detail *Reconstruction Loss*, bagian kiri gambar menunjukkan bagaimana data *time-series* diproses dalam bentuk *sequence* dengan jendela waktu tertentu. Dataset awal terdiri dari lima sampel: $\{x_1, x_2, x_3, x_4, x_5\}$. Data ini kemudian diubah menjadi beberapa *sequence* dengan panjang jendela waktu 3 timestep. Pada ilustrasi, terdapat tiga *sequence* yang terbentuk: $X_1 = [x_1, x_2, x_3]$, $X_2 = [x_2, x_3, x_4]$, dan $X_3 = [x_3, x_4, x_5]$. Setiap *sequence* akan diproses oleh *LSTM Encoder* untuk mengekstrak pola yang ada dalam data. Bagian kanan gambar menunjukkan hasil rekonstruksi dari setiap *sequence* setelah melalui *LSTM Decoder*. *Decoder* mencoba membangun kembali data asli berdasarkan informasi yang telah diproses oleh *encoder*. Hasil rekonstruksi dari masing-masing *sequence* adalah $\hat{X}_1 = [1.1, 2.02, 3.01]$, $\hat{X}_2 = [1.99, 2.99, 3.99]$, dan $\hat{X}_3 = [3.01, 4.01, 5.02]$. Terlihat bahwa nilai rekonstruksi ($\hat{x}$) sedikit berbeda dari nilai asli (x), yang akan digunakan untuk menghitung *reconstruction loss*. Setiap sampel dalam dataset mungkin muncul di lebih dari satu *sequence*, sehingga perhitungan *reconstruction loss* dilakukan dengan menghitung rata-rata kesalahan rekonstruksi dari semua *sequence* yang mengandung sampel tersebut. Sebagai contoh, x_2 muncul dalam *sequence* X_1 dan X_2 , sehingga *loss* dihitung sebagai $(|2.02 - 2| + |1.99 - 2|) / 2 = 0.01$. Proses yang sama dilakukan untuk semua sampel, menghasilkan *reconstruction loss* sebagai berikut: $\{0.1, 0.01, 0.01, 0.01, 0.02\}$. Setelah perhitungan *reconstruction loss*, model menetapkan *threshold* berdasarkan nilai maksimum dari data normal. Jika *reconstruction loss* suatu sampel lebih besar dari

threshold ini, maka sampel tersebut dianggap sebagai anomali. Pada contoh ini, nilai terbesar adalah 0.1, yang kemungkinan akan digunakan sebagai *threshold* untuk mendeteksi apakah suatu data menyimpang dari pola normal.

LSTM dan LSTM *Autoencoder* dikompilasi menggunakan *optimizer* Adam. *Optimizer* adalah algoritma yang digunakan untuk memperbarui bobot model selama pelatihan dengan tujuan meminimalkan fungsi *loss*. Adam (*Adaptive Moment Estimation*) adalah algoritma optimasi yang digunakan untuk memperbarui bobot model berdasarkan data pelatihan. *Optimizer* Adam (*Adaptive Moment Estimation*) digunakan karena dapat memperbarui parameter secara adaptif dengan mempertimbangkan momen pertama dan kedua dari gradien, mempercepat pelatihan jaringan seperti *LSTM*, meskipun keterbatasan *learning rate global* tetap dapat diatasi dengan penyesuaian selama pelatihan [19].

Loss function atau fungsi kerugian mengukur seberapa baik atau buruk model memprediksi target sebenarnya. *Loss function* yang digunakan adalah loss function Huber untuk model Hybrid CNN-LSTM dan MAE untuk model LSTM Autoencoder. Huber Loss menggabungkan loss absolut dan loss kuadrat untuk menghasilkan sebuah fungsi yang dapat diturunkan (*differentiable*) seperti *loss* kuadrat, dan tidak terlalu sensitif terhadap *outlier* seperti *loss* absolut, ditunjukkan pada persamaan (2.4) [20].

$$L(\theta, y) = \frac{1}{n} \sum_{i=1}^n \begin{cases} \frac{1}{2} (y_i - \theta)^2 & |y_i - \theta| \leq \gamma \\ \gamma \left(|y_i - \theta| - \frac{1}{2} \gamma \right) & \text{otherwise} \end{cases} \quad (2.4)$$

Keterangan:

L : Nilai *loss*.

y_i : Nilai sebenarnya dari data.

θ : Nilai prediksi model.

γ : Nilai ambang batas.

n : Jumlah data.

Metrics adalah ukuran yang digunakan untuk mengevaluasi kinerja model yang menunjukkan seberapa baik model bekerja. Pada tahap ini terdapat metrik evaluasi, yaitu: *MSE*, *MAE*, *RMSE*, dan *R-Squared*.

Mean Squared Error (MSE) menghitung rata-rata selisih kuadrat antara nilai sebenarnya dan nilai prediksi, ditunjukkan pada persamaan (2.5) [21].

$$MSE = \frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2 \quad (2.5)$$

Keterangan:

Y_i : Data input asli untuk data ke- i .

X_i : Data hasil rekontruksi (prediksi) dari *autoencoder* untuk data ke- i .

m : Jumlah total data (sampel) dalam dataset.

Mean Absolute Error (MAE) mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual, memberikan gambaran kesalahan rata-rata dalam satuan asli data. Persamaan *MAE* ditunjukan pada persamaan (2.6) [21].

$$MAE = \frac{1}{m} \sum_{i=1}^m |X_i - Y_i| \quad (2.6)$$

Keterangan:

Y_i : Data input asli untuk data ke- i .

X_i : Data hasil rekontruksi (prediksi) dari *autoencoder* untuk data ke- i .

m : Jumlah total data (sampel) dalam dataset.

Root Mean Squared Error (RMSE) adalah akar kuadrat dari MSE, yang juga memberikan ukuran rata-rata kesalahan tetapi tetap dalam satuan asli data, menjadikannya lebih mudah diinterpretasikan. Persamaan RMSE ditunjukan pada persamaan (2.7) [21].

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2} \quad (2.7)$$

Keterangan:

Y_i : Data input asli untuk data ke- i .

X_i : Data hasil rekontruksi (prediksi) dari *autoencoder* untuk data ke- i .

m : Jumlah total data (sampel) dalam dataset.

R-squared (R^2) adalah metrik yang menunjukkan proporsi variabilitas data yang dapat dijelaskan oleh model, nilai mendekati 1 menunjukkan model yang sangat baik dalam menjelaskan data. Persamaan *R-squared* ditujukan pada persamaan (2.8) [21].

$$R^2 = \frac{1}{m} \sum_{i=1}^m \frac{(X_i - Y_i)^2}{(\bar{Y} - Y_i)^2} \quad (2.8)$$

Keterangan:

Y_i : Data input asli untuk data ke- i .

$\bar{Y}$: Rata-rata nilai input asli data ke- i .

X_i : Data hasil rekontruksi (prediksi) dari *autoencoder* untuk data ke- i .

m : Jumlah total data (sampel) dalam dataset.

2.6. Sistem Pemantauan dan Pengintegrasikan Jarak Jauh

Perangkat dapat dipantau dan diintegrasikan dalam jarak jauh. Raspberry Pi, sebagai perangkat komputasi yang kecil, dapat diubah menjadi sistem pemantauan jarak jauh yang handal dengan mengintegrasikan perangkat lunak yang tepat. Sistem ini memungkinkan pengguna untuk mengakses, memantau, dan mengontrol Raspberry Pi dari mana saja dengan memanfaatkan kombinasi berbagai teknologi.

Setiap perangkat di jaringan internet diberi nomor unik yang dikenal sebagai alamat *IP* (*internet protocol*) perangkat. Alamat *IP* ini digunakan untuk mengidentifikasi

perangkat dan memfasilitasi pengiriman data antarperangkat dalam jaringan virtual. Salah satu solusi untuk berada di jaringan yang sama meskipun lokasi fisik perangkat berada di lokasi geografis yang berbeda ialah menggunakan ZeroTier.

ZeroTier merupakan solusi jaringan modern berbasis perangkat lunak (*Software-Defined Networking*) yang memfasilitasi konektivitas jaringan yang fleksibel, memungkinkan pengguna untuk membangun jaringan virtual mirip *virtual private network* (VPN) konvensional, sehingga perangkat dapat saling terhubung dengan mudah tanpa memandang lokasi fisik mereka [22]. ZeroTier berfungsi sebagai saklar *ethernet* virtual yang menghubungkan perangkat dalam jaringan lokal yang luas dan terintegrasi, serta mereplikasi fungsionalitas *ethernet* layer 2, termasuk fitur *multipath* (menggunakan beberapa jalur untuk mengoptimalkan pengiriman data), *multicast* (pengiriman data ke beberapa perangkat sekaligus), dan *bridging* (menghubungkan beberapa jaringan), yang memungkinkan pembentukan jaringan mirip *LAN* (*Local Area Network*), tetapi dengan kemampuan menghubungkan perangkat di seluruh dunia [22].

Pada implementasi sistem ini, *Secure Shell* (SSH) digunakan sebagai metode akses jarak jauh (*remote access*) yang aman untuk mengelola dan memantau perangkat Raspberry Pi yang tergabung dalam jaringan ZeroTier. SSH memungkinkan pengguna untuk melakukan perintah terminal, mengakses file, dan melakukan konfigurasi sistem dari lokasi manapun melalui koneksi terenkripsi.

Node-RED adalah alat pemrograman berbasis alur untuk menghubungkan perangkat keras, API, dan layanan daring [23]. Node-RED berperan sebagai *platform* integrasi yang memungkinkan Raspberry Pi mengumpulkan, memproses, dan mengirim data sensor ke *database*. Pada perangkat *Eddy Covariance Station*, Node-RED menangani data *real-time* yang dihasilkan oleh sensor yang terhubung ke Raspberry Pi.

2.7. Pengukuran Kinerja Sistem

Pengukuran kinerja sistem dalam penelitian ini difokuskan pada evaluasi efisiensi pemrosesan model pada perangkat *edge*, khususnya Raspberry Pi, melalui tiga parameter utama: penggunaan CPU (*Central Processing Unit*), penggunaan RAM (*Random Access Memory*), dan waktu inferensi model. Penggunaan CPU (%) mencerminkan seberapa besar kapasitas prosesor digunakan saat proses inferensi berjalan. Nilai yang terlalu tinggi dapat menunjukkan beban yang berat. Penggunaan RAM (MB) menggambarkan konsumsi memori selama model dimuat dan dijalankan. Efisiensi penggunaan memori penting agar sistem tetap stabil dalam jangka panjang. Waktu inferensi (detik) mengukur durasi yang dibutuhkan oleh model untuk menghasilkan *output* dari satu input. Semakin cepat waktu inferensi, semakin cocok model digunakan untuk sistem *real-time*. Ketiga parameter ini merepresentasikan beban kerja sistem selama menjalankan proses prediksi secara *real-time* dan digunakan untuk menilai apakah perangkat mampu menjalankan model secara optimal dalam batas sumber daya yang tersedia.

Ketiga parameter ini merepresentasikan beban kerja aktual yang ditanggung oleh sistem selama menjalankan proses prediksi, dan digunakan untuk menilai apakah perangkat mampu menjalankan model secara optimal dalam batas sumber daya yang tersedia. Pada implementasinya, sistem inferensi dijalankan menggunakan skrip Python yang dieksekusi langsung melalui terminal Raspberry Pi. Guna memastikan sistem dapat berjalan secara otomatis dan terjadwal, digunakan mekanisme manajemen layanan berbasis *systemd*. Melalui perintah *systemctl*, dapat mengatur, mengaktifkan, atau menonaktifkan layanan inferensi yang ditentukan dalam file konfigurasi *.service*. Penjadwalan eksekusi dilakukan dengan *.timer*, yang memungkinkan sistem menjalankan proses inferensi secara berkala sesuai waktu yang telah ditentukan.

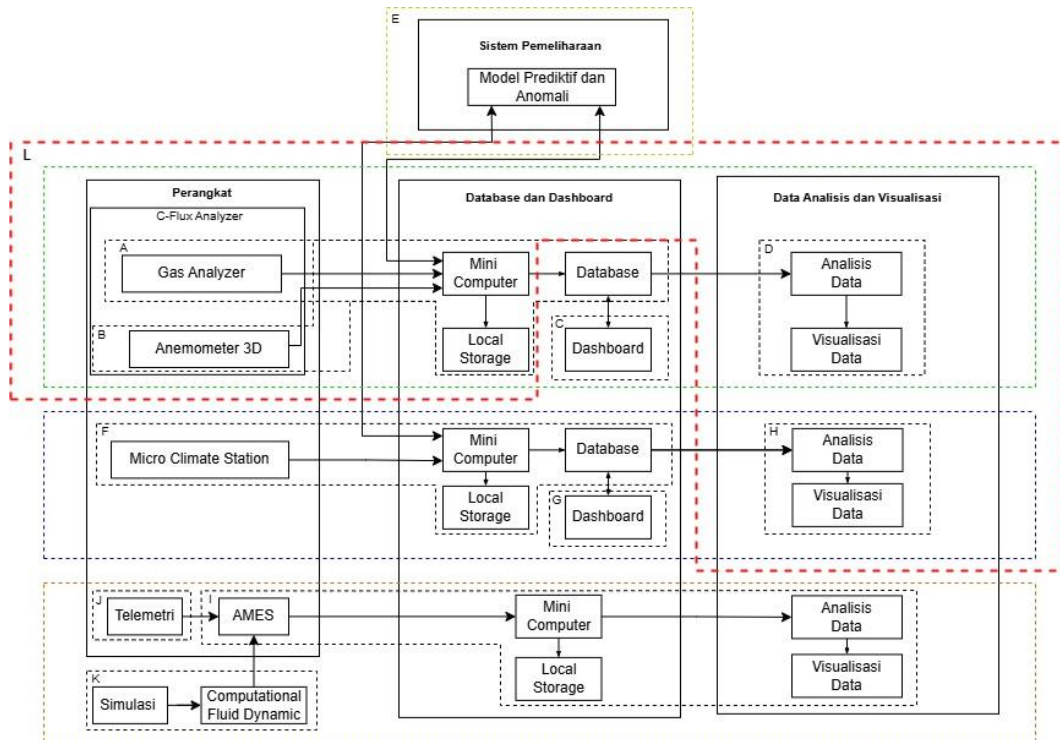
III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian tugas akhir ini dilaksanakan selama enam bulan di Laboratorium Telekomunikasi, Jurusan Teknik Elektro, Universitas Lampung, serta di PT Persada Dinamika Lestari sebagai sumber data tempat perangkat Eddy Covariance Station dipasang.

3.2. *Capstone Project*

Penelitian ini merupakan bagian dari penelitian utama tim Eddy Covariance Station. Berdasarkan Gambar 3.1 Diagram *Capstone Project* Tim Eddy Covariance Station (ECS), perangkat Eddy Covariance Station terdiri dari tiga perangkat, yaitu perangkat C-Flux Analyzer yang terdiri dari Gas Analyzer dan 3D Anemometer ditunjukkan dengan kolom berwarna hijau, perangkat Microclimate Station (MCS) ditunjukkan dengan kolom berwarna biru, dan perangkat Aerial Measurement Eddy Covariance System (AMES) ditunjukkan dengan kolom berwarna jingga. Bagian yang dikerjakan pada penelitian ini adalah membuat model pemeliharaan prediktif di sistem pemeliharaan berdasarkan data harian yang direkam di perangkat Eddy Covariance Station, yang ditunjukan pada kolom bagian E berwarna kuning.



Gambar 3.1 Diagram *Capstone Project* Tim Eddy Covariance Station (ECS)

Berdasarkan Gambar 3.1 Diagram *Capstone Project* Tim Eddy Covariance Station (ECS). Penelitian lain pada kolom bagian A merupakan perangkat C-Flux Analyzer, bagian B merupakan perangkat 3D Anemometer, bagian C berisi *dashboard* C-Flux Analyzer dan 3D Anemometer, bagian D merupakan analisis data perangkat, bagian F merupakan perangkat MCS, bagian G merupakan *dashboard* MCS, bagian H merupakan analisis data MCS, bagian I merupakan perangkat AMES, bagian J merupakan telemetri perangkat AMES, bagian K merupakan simulasi perangkat AMES, dan yang terakhir bagian L membahas keseluruhan kecuali perangkat AMES.

3.3. Perangkat Penelitian

Pengerjaan penelitian ini menggunakan perangkat keras serta komponen dan perangkat lunak sebagai berikut:

1. Laptop dengan prosesor Intel® Core™ i3 Generasi ke-11.

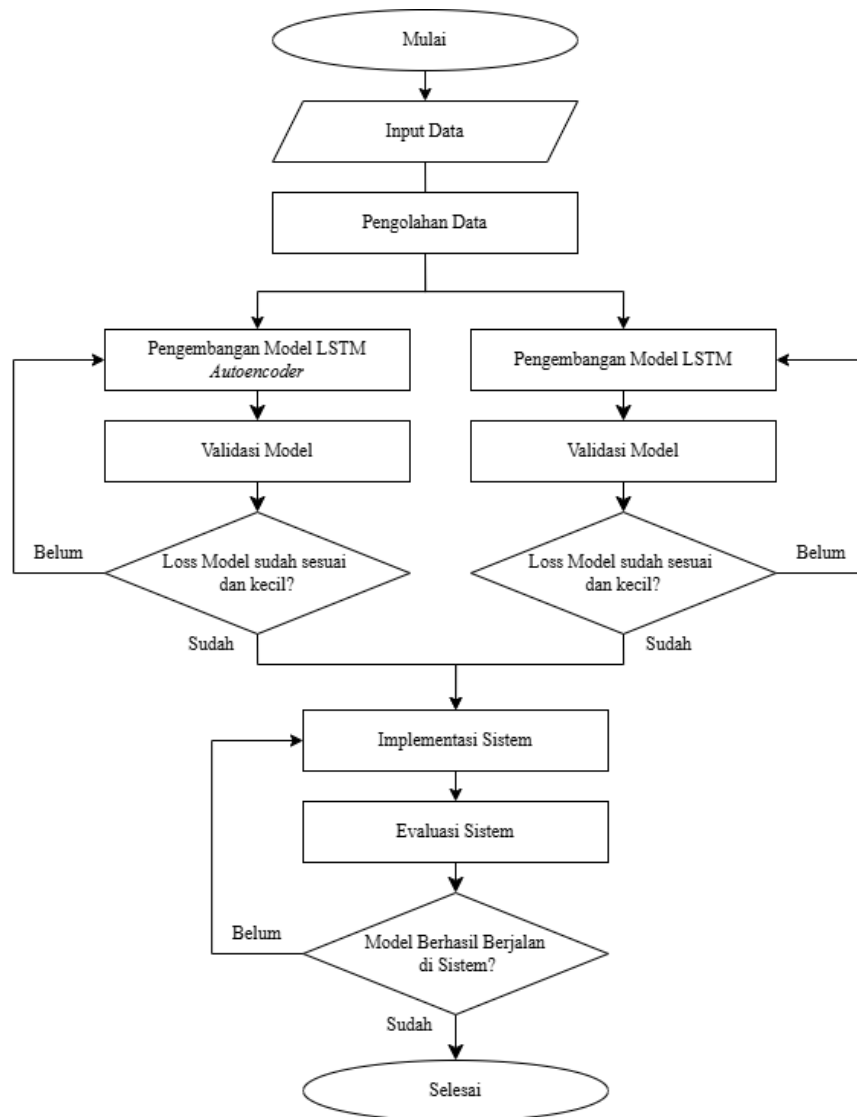
2. Komputer dengan prosesor Intel® Core™ i9 Generasi ke-15 dan GPU Nvidia® RTX 4090.
3. Raspberry Pi 4 Model B (RAM 8GB) digunakan untuk pengujian dengan data dari beberapa perangkat yang berbeda, dan Raspberry Pi 4 Model B (RAM 2GB) untuk pengujian pada sistem terintegrasi.
4. Visual Studio Code (VS Code) digunakan untuk mengolah data, melatih model, dan mengevaluasi model menggunakan bahasa pemrograman Python.
5. WinSCP digunakan untuk memindahkan file antara laptop dan Raspberry Pi.
6. Command Prompt (CMD) digunakan untuk menjalankan sistem agar berjalan, serta untuk menjalankan perintah sistem, seperti menjalankan skrip Python dan melihat penggunaan CPU dan RAM.
7. ZeroTier yang digunakan untuk menghubungkan Raspberry Pi ke jaringan virtual (VPN) sehingga bisa diakses dari jarak jauh meskipun tidak berada pada jaringan lokal yang sama
8. Telegram yang digunakan sebagai media untuk mengirimkan notifikasi otomatis apabila sistem mendeteksi anomali pada data sensor, melalui bot Telegram yang telah dikonfigurasi.

3.4. Sumber Data

Sumber data diperoleh dari perangkat Eddy Covariance Station (ECS), berupa C-Flux Analyzer yang terdiri dari Gas Analyzer dan 3D Anemometer, serta Microclimate Station (MCS). Perangkat Gas Analyzer dan 3D Anemometer mengambil data 6-7 data per detik selama 30 menit per jam, sedangkan perangkat MCS menghasilkan data satu data per menit. Parameter yang digunakan adalah hasil dari perangkat C-Flux Analyzer berupa sensor gas karbon dioksida, sensor gas metana, *water vapour*, suhu, dan kelembapan udara, serta altitude, suhu, dan tekanan udara, perangkat 3D Anemometer berupa enam Digital Airspeed Sensor yang masing-masing mendeteksi kecepatan angin di arah timur, selatan, barat, utara, atas, dan bawah perangkat, serta MCS berupa parameter temperature, humidity, rainfall, angle, windspeed, dan pyrano.

3.5. Tahapan Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang dijelaskan pada Gambar 3.2 Diagram Alir Tahapan Penelitian Keseluruhan.



Gambar 3.2 Diagram Alir Tahapan Penelitian Keseluruhan

1. Input data

Tahapan pertama dalam penelitian ini adalah input data. Data dikumpulkan dari sensor-sensor yang dipasang pada perangkat. Data yang diinput dijelaskan pada sub bab 3.3 Sumber Data. Data MCS yang digunakan adalah data selama 6 Bulan. Sedangkan, untuk Gas Analyzer dan 3D Anemometer digunakan data selama 9 hari.

2. Pengolahan Data

Setelah data terkumpul, langkah berikutnya adalah pengolahan data. Kelengkapan data dicek untuk mengetahui langkah selanjutnya pada tahap pengembangan model. Pada tahap pengolahan data ini, hal yang dilakukan adalah mengatasi data *outlier* dan mengatasi *missing value* melalui visualisasi data. Kemudian normalisasi data dan pembuatan *window* tepat sebelum pembuatan model.

3. Pengembangan Model

Tahap pengembangan model mencakup proses pemilihan algoritma, perancangan struktur, pelatihan, validasi model, dan konversi model ke TensorFlow Lite. Penelitian ini menggunakan *hybrid CNN-LSTM (Convolutional Neural Network - Long Short-term Memory)* untuk prediksi waktu kerusakan dan LSTM (*Long Short-Term Memory*) *Autoencoder* digunakan untuk mendeteksi pola kegagalan atau anomali. Setelah model selesai dikembangkan, langkah berikutnya adalah validasi model untuk memastikan kinerja model. Percobaan pembuatan model dilakukan dengan filter yang berbeda supaya dapat menemukan model dengan mempertimbangkan evaluasi model.

4. Implementasi Sistem

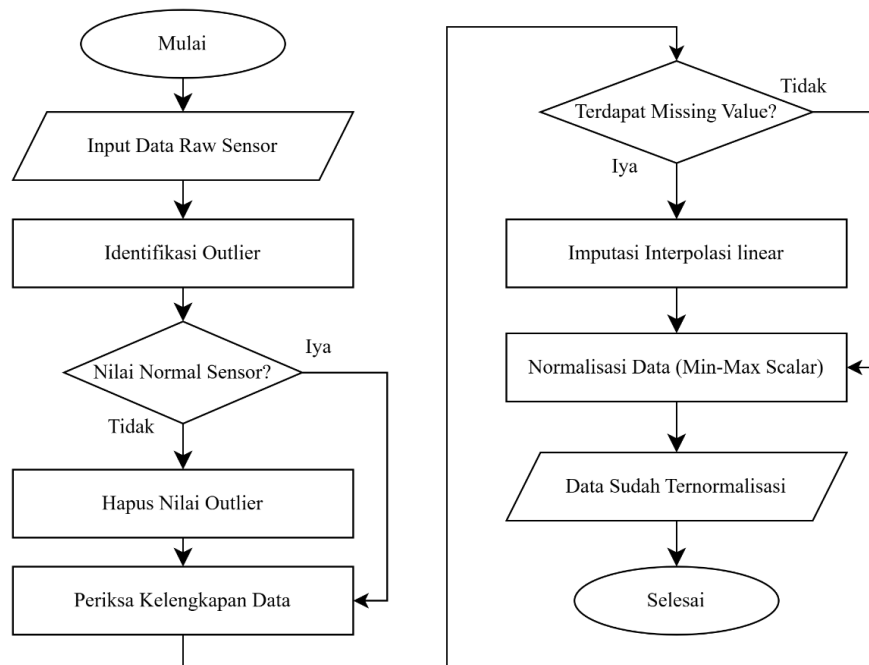
Tahapan implementasi sistem bertujuan untuk mengintegrasikan model yang telah divalidasi ke dalam lingkungan operasional untuk *monitoring*. Model dikonversi ke format tensorflow lite agar bisa diintegrasikan di Python Raspberry Pi. Pengembangan dan pengujian dilakukan menggunakan bahasa pemrograman Python dengan dua sistem pendekatan pendeteksian pola kegagalan, yaitu sistem prediksi-deteksi menggunakan kombinasi model prediksi *hybrid CNN-LSTM* dan model deteksi pola kegagalan menggunakan LSTM *autoencoder*, serta sistem deteksi langsung menggunakan model LSTM *autoencoder*. Sistem diimplementasikan dan diuji pada perangkat Raspberry Pi dengan dua skenario pengujian, yaitu sistem prediksi dan perangkat sensor berjalan pada Raspberry Pi yang sama, dan sistem prediksi berjalan pada Raspberry Pi terpisah yang menerima data dari Raspberry Pi lain di lokasi berbeda.

5. Evaluasi Sistem

Tahapan terakhir dalam penelitian ini adalah evaluasi sistem. Evaluasi sistem dilakukan selama 3 hari mencakup penggunaan CPU dan RAM, waktu inferensi model, serta waktu pengiriman notifikasi melalui Telegram.

3.6. Pengolahan Data

Pengolahan data mencakup mengatasi data *outlier*, cek *incomplite* data, dan mengatasi missing value. Langkah-langkah ini dirancang untuk menghasilkan dataset yang siap digunakan dalam pelatihan model prediksi *hybrid* CNN-LSTM dan deteksi anomali LSTM *autoencoder*. Proses pengolahan data ditunjukkan pada Gambar 3.3 Diagram Alir Pengolahan Data.



Gambar 3.3 Diagram Alir Pengolahan Data

Pengolahan data dimulai dengan identifikasi dan penanganan *outlier*, yaitu data yang memiliki nilai sangat berbeda dari data lainnya dan dapat menyebabkan distorsi dalam analisis maupun pelatihan model, khususnya untuk model *LSTM*

autoencoder. Identifikasi *outlier* dilakukan melalui visualisasi data menyeluruh dan dari pemahaman *output default* sensor. Sensor yang mengeluarkan data rentang jauh di atas data normal, tetapi merupakan nilai *default* dari sensor akan tetap disimpan dan digunakan, sedangkan data yang bukan nilai *default* akan dihapus dari data. Langkah selanjutnya dalam pengolahan data adalah pemeriksaan terhadap *incomplete data*, yang mencakup *missing value* atau nilai yang hilang. *Incomplete data* dapat memengaruhi akurasi model yang akan dikembangkan, sehingga penanganan *missing value* menjadi langkah penting untuk memastikan kualitas data. Salah satu metode yang sering digunakan untuk mengatasi *missing value* adalah interpolasi linear. Tahap terakhir adalah normalisasi data yang berfungsi untuk membantu model menghindari bias dalam pembelajaran akibat skala data yang berbeda. Normalisasi yang digunakan adalah *min-max scalar*.

3.7. Pengembangan Model

Proses pengembangan model ini melibatkan pemilihan algoritma yang tepat, perancangan struktur model, pelatihan model, validasi untuk memastikan kinerja yang optimal, serta konversi model ke TensorFlow Lite.

3.7.1. Pemilihan Algoritma

Tahap pertama dalam pengembangan model adalah pemilihan algoritma yang sesuai dengan tujuan analisis. Algoritma *hybrid CNN-LSTM* (*Convolutional Neural Network – Long Short-Term Memory*) dipilih untuk prediksi waktu kerusakan karena kemampuannya yang unggul dalam menangani data berurutan atau *time-series* data. CNN digunakan untuk mengekstraksi fitur spasial dari data dengan mendeteksi pola penting dalam jendela waktu, sementara LSTM mampu memproses data secara berurutan untuk menangkap hubungan temporal jangka panjang. Kombinasi ini memberikan kemampuan untuk mengenali pola kompleks yang terjadi dalam waktu tertentu, menjadikannya sangat sesuai untuk tugas prediksi.

LSTM *Autoencoder* dipilih untuk mendeteksi anomali karena strukturnya yang dapat mempelajari pola normal dalam data dengan sangat baik. LSTM *Autoencoder* terdiri dari *encoder* yang merangkum data ke representasi latennya dan *decoder* yang mencoba merekonstruksi data asli. Jika data memiliki perbedaan signifikan dengan pola normal (anomali), maka rekonstruksinya akan memiliki eror yang tinggi, sehingga memudahkan identifikasi anomali. Pendekatan ini sangat cocok untuk data *time-series* yang kompleks dan dinamis, karena LSTM dapat memahami urutan waktu secara mendalam.

3.7.2. Struktur Model

Struktur model merujuk pada struktur dan desain dari model pembelajaran mesin yang digunakan untuk memecahkan masalah menggunakan TensorFlow atau Keras. TensorFlow adalah sebuah *library open source* yang dikembangkan oleh Google untuk komputasi numerik dan pembelajaran mesin berbasis *deep learning*. Keras adalah sebuah *high level API (Application Programming Interface)* dalam TensorFlow yang menyediakan antarmuka sederhana untuk membangun dan melatih model *deep learning* dengan mudah. Pada hal ini, model yang digunakan adalah *hybrid CNN-LSTM* untuk prediksi waktu kerusakan dan LSTM *autoencoder* untuk deteksi anomali.

Tabel 3.1 Struktur Layer Hybrid CNN-LSTM

Layer	Deskripsi
Input(shape=(batch_size, filters, feature))	Menerima data sensor dalam bentuk (batch_size, timesteps, feature)
Conv1D(filters, kernel_size, activation, padding)	Mengekstraksi fitur lokal dari data <i>time-series</i>
MaxPooling1D(pool_size)	Mengurangi dimensi fitur setelah CNN
LSTM(units, activation, return_sequences)	Menangkap pola temporal dalam data sensor
Dense()	Output prediksi berdasarkan pola yang dipelajari

Struktur model prediksi ini ditunjukkan dengan Tabel 3.1 Struktur Layer Hybrid CNN-LSTM. Model ini dimulai dengan Input Layer, yang menerima data dalam format $(batch_size, timesteps, feature)$, $batch_size$ adalah jumlah sampel dalam satu *batch*, $timesteps$ adalah panjang urutan waktu, dan $feature$ adalah jumlah fitur dalam satu sampel bersamaan dengan Conv1D digunakan untuk mengekstraksi pola lokal dalam data *time-series*. Kemudian diikuti oleh MaxPooling1D untuk mengurangi dimensi fitur yang diproses. Setelah fitur utama diekstraksi oleh CNN, LSTM Layer bertugas menangkap pola temporal antar-*timestep* dalam data sensor, membantu model memahami hubungan antara nilai sebelumnya dan nilai yang akan datang. Akhirnya, *Dense Output Layer* menghasilkan prediksi nilai sensor berdasarkan pola yang telah dipelajari oleh model.

Tabel 3.2 *Hyperparameter Model Hybrid CNN-LSTM*

Hyperparameter	Deskripsi	Nilai Rekomendasi
<i>Conv1D Filters</i>	Jumlah filter pada layer <i>CNN</i>	32, 64
<i>Kernel Size</i>	Ukuran filter <i>CNN</i> untuk ekstraksi fitur	5
<i>Pooling Size</i>	Ukuran <i>pooling</i> setelah <i>CNN</i>	2
<i>Padding</i>	Mengontrol <i>output size</i> setelah konvolusi	<i>causal</i>
<i>LSTM Units</i>	Jumlah unit pada <i>LSTM</i> setelah <i>CNN</i>	32, 64
<i>Return Sequences</i>	Mengontrol <i>output layer LSTM</i>	<i>False</i>
<i>Activation Function</i>	Fungsi aktivasi dalam <i>CNN</i> dan <i>LSTM</i>	ReLU, TanH
<i>Optimizer</i>	Mengatur model memperbarui bobot	Adam
<i>Loss Function</i>	Mengatur eror prediksi dan nilai asli	Huber
<i>Batch Size</i>	Jumlah sampel dalam satu <i>batch training</i>	512
<i>Epochs</i>	Jumlah iterasi <i>training</i>	100
<i>Learning Rate</i>	Kecepatan update bobot model	0.0001

Tabel 3.2 *Hyperparameter Model Hybrid CNN-LSTM* merangkum *hyperparameter* yang dirancang untuk menangani data *time-series* dengan kombinasi ekstraksi fitur melalui *Convolutional Neural Network* (CNN) dan pemodelan dependensi temporal dengan *Long Short-Term Memory* (LSTM). Model ini menggunakan lapisan Conv1D dengan jumlah filter yang bervariasi antara 32

dan 64, *kernel size* dengan nilai 5 untuk menangkap pola dalam data sensor, serta *padding* bisa diatur ke *causal* untuk menjaga agar output hanya dipengaruhi oleh data saat ini dan sebelumnya. *Pooling size* 2 digunakan untuk mengurangi dimensi fitur, sedangkan *LSTM units* diatur dengan variasi percobaan 32 dan 64 untuk menangkap hubungan jangka panjang dalam data dengan *return sequences false* hanya mengembalikan *output* dari *timestamp* terakhir karena layer LSTM hanya ada satu layer. Model ini menggunakan fungsi aktivasi ReLu untuk layer CNN dan *Tanh* untuk layer LSTM, *optimizer* Adam, dan *loss function* Huber untuk mengoptimalkan proses pembelajaran. *Batch size* sebesar 512, jumlah *epoch* 100, dan learning rate untuk optimasi sebesar 0.0001.

Tabel 3.3 Struktur Model LSTM *Autoencoder*

Lapisan	Deskripsi
Input(<i>shape</i> =(<i>batch_size</i> , <i>timesteps</i> , <i>feature</i>))	Menerima data sensor dalam bentuk (<i>batch_size</i> , <i>timesteps</i> , <i>feature</i>)
LSTM(<i>units</i> , <i>activation</i> , <i>return_sequences</i>)	Encoder mengompresi data menjadi representasi laten
RepeatVector(<i>timesteps</i>)	Mengulang representasi laten sebanyak jumlah <i>timestep</i> input
LSTM(<i>units</i> , <i>activation</i> , <i>return_sequences</i>)	Decoder merekonstruksi kembali data dari ruang laten
Dense(<i>feature</i>)	Output rekonstruksi untuk dibandingkan dengan input awal

Struktur model anomali ditunjukkan pada Tabel 3.3 Struktur Model *LSTM Autoencoder* yang digunakan untuk mendeteksi anomali dalam data sensor dengan cara merekonstruksi input dan membandingkannya dengan *output*. Model ini dimulai dengan Input Layer, yang menerima data dalam format (*batch_size*, *timesteps*, *feature*) bersamaan dengan LSTM Encoder bertugas mengompresi data input menjadi representasi laten, yaitu bentuk ringkas dari pola utama dalam data. *RepeatVector* digunakan untuk mengulang representasi laten sebanyak jumlah *timesteps*, sehingga informasi dari *encoder* dapat diproses kembali oleh LSTM Decoder. LSTM Decoder kemudian berusaha merekonstruksi kembali data dari

ruang laten, menghasilkan *output* yang seharusnya mendekati input asli. Akhirnya, *Dense Output Layer* digunakan untuk menghasilkan hasil rekonstruksi, yang akan dibandingkan dengan input awal. Jika selisih antara input dan *output* terlalu besar (*loss tinggi*), maka data tersebut merupakan anomali. *Hyperparameter* yang digunakan dalam model LSTM *Autoencoder*, yang berfungsi untuk mendeteksi anomali dalam data *time-series* dengan cara merekonstruksi input dan membandingkan dengan *output* dirangkum di Tabel 3.4 *Hyperparameter Model LSTM Autoencoder*.

Tabel 3.4 *Hyperparameter Model LSTM Autoencoder*

Hyperparameter	Deskripsi	Nilai Rekomendasi
<i>LSTM Encoder Units</i>	Jumlah unit pada <i>LSTM Encoder</i>	32, 64
<i>LSTM Decoder Units</i>	Jumlah unit pada <i>LSTM Decoder</i>	32, 64
<i>Repeat Vector</i>	Jumlah kali input diulang di <i>decoder</i>	<i>Input sequence</i>
<i>Activation Function</i>	Fungsi aktivasi dalam <i>LSTM</i>	<i>Tanh</i>
<i>Return Sequences</i>	Mengontrol <i>output layer LSTM</i>	<i>False</i>
<i>Optimizer</i>	Mengatur model memperbarui bobot	Adam
<i>Loss Function</i>	Mengatur eror prediksi dan nilai asli	MAE
<i>Batch Size</i>	Jumlah sampel satu <i>batch training</i>	512
<i>Epochs</i>	Jumlah iterasi <i>training</i>	100
<i>Learning Rate</i>	Kecepatan <i>update</i> bobot model	0.001

Model ini menggunakan *LSTM Encoder* dan *LSTM Decoder* dengan jumlah unit 32 dan 64, yang memungkinkan model untuk menangkap pola kompleks dalam data. *Repeat Vector* digunakan untuk mengulang representasi laten sebelum dikirim ke *decoder*, memastikan rekonstruksi yang optimal dengan input sesuai nilai *input sequence*. Fungsi aktivasi *Tanh* digunakan karena cocok untuk data *time-series*. Model ini juga menggunakan *optimizer Adam* untuk mengatur pembaruan bobot dengan efisien, serta *loss function MAE* untuk mengukur kesalahan rekonstruksi. Pengaturan *batch size* sebesar 512, sementara jumlah *epoch* sebesar 100. *Learning rate* ditetapkan antara 0.001 atau nilai default.

3.7.3. Pelatihan Model

Pelatihan model dilakukan setelah struktur dan parameter dasar ditentukan. Proses pelatihan dimulai dengan membagi dataset menjadi tiga bagian (data pelatihan, data validasi, dan data test) untuk model *hybrid* CNN-LSTM, serta dua bagian (data pelatihan dan data validasi) untuk LSTM *Autoencoder*. Data pelatihan digunakan untuk mengajarkan model mengenali pola dalam data, data validasi digunakan untuk memonitor kinerja model pada data yang tidak terlihat sebelumnya, dan data test digunakan untuk tes hasil kinerja model. Selama pelatihan, beberapa parameter seperti *learning rate*, *batch size*, dan jumlah *epoch* diatur agar model dapat belajar dengan optimal dan menghindari *overfitting*.

3.7.4. Evaluasi Model

Uji evaluasi model ini menggunakan metrik evaluasi berupa MSE, MAE, RMSE, dan R-squared. MSE (*Mean Squared Error*) untuk mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual, semakin kecil, semakin baik. MAE (*Mean Absolute Error*) untuk menghitung rata-rata selisih absolut antara prediksi dan nilai aktual, lebih tahan terhadap *outlier* dibanding MSE. RMSE (*Root Mean Squared Error*) merupakan akar dari MSE, lebih mudah diinterpretasikan karena satuannya sama dengan data asli. R-squared (R^2) untuk mengukur seberapa baik model menjelaskan variansi data, dengan nilai antara 0 hingga 1, semakin mendekati 1, semakin baik modelnya.

3.7.5. Konversi Model TensorFlow ke TensorFlow Lite

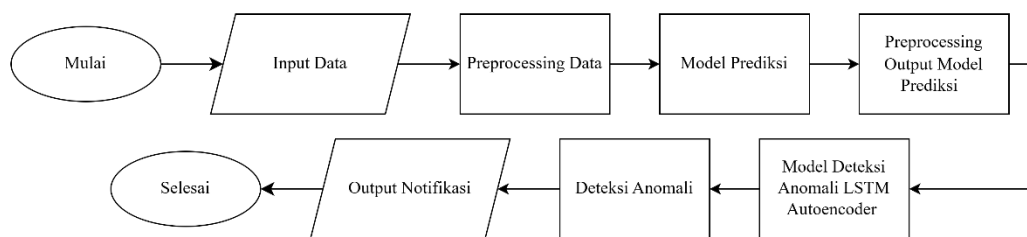
Setelah model selesai dievaluasi, langkah selanjutnya adalah mengonversinya ke format TensorFlow Lite agar dapat dijalankan pada perangkat dengan sumber daya terbatas, seperti Raspberry Pi. Proses konversi dimulai dengan menyimpan model yang sudah dilatih dalam format SavedModel atau HDF5 di TensorFlow Python. Kemudian, model tersebut dikonversi menggunakan TensorFlow Lite Converter, yang menghasilkan file dengan ekstensi *.tflite*.

3.8. Implementasi Sistem

Tahap implementasi sistem bertujuan untuk mengintegrasikan model *Artificial Intelligence (AI)* yang telah dikembangkan ke dalam sistem *monitoring real-time*. Sistem ini dirancang agar mampu menerima data sensor secara *real-time*, memproses data yang diterima, serta melakukan analisis menggunakan model untuk memprediksi nilai sensor dan mendeteksi anomali. Sistem dapat memberikan peringatan dini apabila terjadi penyimpangan dalam pola data sensor.

3.8.1. Skenario Sistem Deteksi Kegagalan

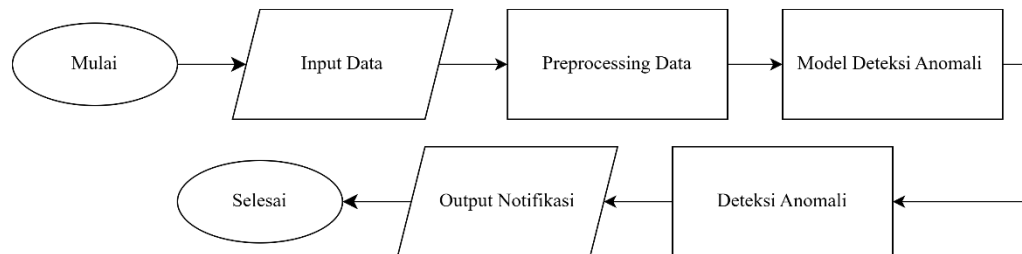
Ada dua skenario sistem untuk deteksi kegagalan yang digunakan, skenario pertama penggabungan model *hybrid CNN-LSTM* untuk memprediksi data dan model *LSTM autoencoder* untuk deteksi anomali sehingga dapat diprediksi kapan sistem mengalami kerusakan serta skenario kedua model *LSTM autoencoder* secara langsung untuk mendeteksi anomali secara langsung.



Gambar 3.4 Skenario Sistem Prediksi-Deteksi

Skenario pertama terdapat pada Gambar 3.4 Skenario Sistem Prediksi-Deteksi. Sistem ini dirancang untuk mendeteksi anomali berbasis prediksi dengan memanfaatkan data sensor yang dikirim secara *real-time* melalui protokol MQTT. Data yang diterima akan melalui tahap *preprocessing*, dimulai dari penanganan nilai hilang menggunakan penanda khusus -9999, dilanjutkan dengan pembentukan sekuens menggunakan teknik *Sliding Window*, serta normalisasi data dengan *min-max scaler*. Data yang telah diproses kemudian dimasukkan ke dalam model *hybrid CNN-LSTM* untuk memprediksi nilai sensor berikutnya berdasarkan pola historis.

Hasil prediksi *hybrid* CNN-LSTM kemudian disimpan dalam *buffer* sebelum digunakan oleh LSTM *Autoencoder* untuk deteksi anomali. Jika perbedaan antara nilai prediksi dan nilai aktual terlalu besar, maka kondisi ini dianggap sebagai anomali, dan sistem akan mengirimkan peringatan ke pengguna melalui notifikasi Telegram.

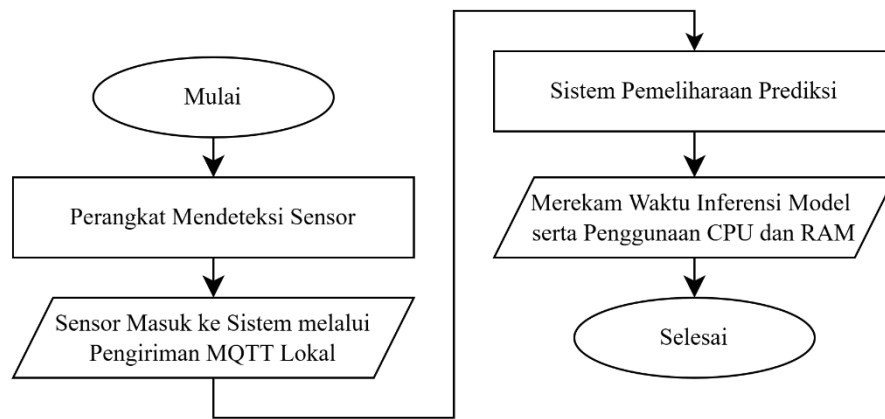


Gambar 3.5 Skenario Deteksi Langsung

Skenario kedua pada Gambar 3.4 Skenario Deteksi Langsung. Pada skenario kedua, sistem langsung mendeteksi anomali tanpa melakukan prediksi nilai sensor sebelumnya dengan memanfaatkan data sensor yang dikirim secara *real-time* melalui protokol MQTT. Data yang diterima akan melalui tahap *preprocessing*, dimulai dari penanganan nilai hilang menggunakan penanda khusus -9999, dilanjutkan dengan pembentukan sekuens menggunakan teknik *Sliding Window*, serta normalisasi data dengan *min-max scaler*. Data yang telah diproses kemudian dimasukkan ke dalam model LSTM *autoencoder* untuk deteksi anomali. Jika perbedaan antara nilai prediksi dan nilai aktual terlalu besar, maka kondisi ini dianggap sebagai anomali, dan sistem akan mengirimkan peringatan ke pengguna melalui notifikasi Telegram.

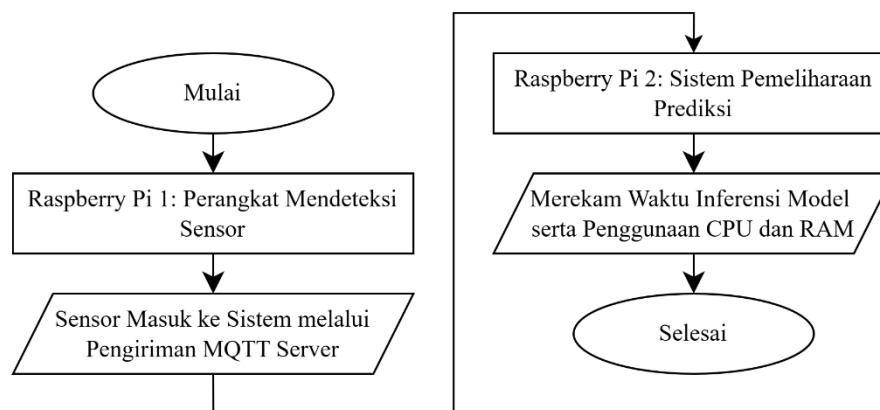
3.8.2. Skenario Pengujian Sistem

Terdapat dua pengujian sistem, yaitu: sistem terintegrasi ketika prediksi dan perangkat sensor berjalan pada Raspberry Pi yang sama dan sistem terdistribusi ketika prediksi berjalan pada Raspberry Pi terpisah yang menerima data dari Raspberry Pi lain di lokasi berbeda.



Gambar 3.6 Diagram Alir Pengujian Sistem Terintegrasi

Pada pengujian pertama ditunjukkan pada Gambar 3.6 Diagram Alir Pengujian Sistem Terintegrasi, sistem terintegrasi, sistem prediksi dan sensor berjalan pada Raspberry Pi yang sama dengan menggunakan MQTT lokal sebagai protokol komunikasi. Pada skenario ini, Raspberry Pi mengumpulkan data dari sensor lingkungan secara periodik, lalu mengirimkan data tersebut melalui MQTT lokal untuk diproses oleh model prediksi yang juga dijalankan pada Raspberry Pi yang sama. Tujuan pengujian ini adalah untuk memastikan integrasi sistem yang berjalan di perangkat tunggal, mengukur waktu inferensi model, serta penggunaan sumber daya Raspberry Pi seperti CPU dan RAM.



Gambar 3.7 Diagram Alir Pengujian Sistem Terdistribusi

Pada pengujian kedua ditunjukkan pada Gambar 3.7 Diagram Alir Pengujian Sistem Terdistribusi, sistem terdistribusi, sistem prediksi berjalan pada Raspberry Pi yang terpisah dan menerima data dari Raspberry Pi lain di lokasi berbeda melalui server MQTT yang terpusat. Raspberry Pi pertama mengumpulkan data sensor dan mengirimkan data tersebut ke server MQTT, sementara Raspberry Pi kedua menerima data melalui server dan menjalankan model prediksi. Tujuan dari skenario ini adalah untuk memastikan bahwa model prediksi dapat memproses data yang diterima secara *real-time* dengan baik. Evaluasi hasil pengujian ini meliputi mengukur waktu inferensi model, serta penggunaan sumber daya Raspberry Pi seperti CPU dan RAM.

3.9. Evaluasi Sistem

Evaluasi sistem dilakukan selama 3 hari dengan fokus pada: penggunaan RAM dan CPU, serta kecepatan inferensi model. Penggunaan RAM dan CPU dipantau untuk memastikan bahwa sistem berjalan dengan efisien dan tidak membebani perangkat lebih dari yang diperlukan. Kecepatan inferensi model diukur untuk memastikan bahwa sistem dapat memproses data sensor dan menghasilkan prediksi secara *real-time* tanpa penundaan yang signifikan.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, didapat beberapa kesimpulan pada penelitian ini:

1. Model yang digunakan pada perangkat Gas Analyzer: model hybrid CNN-LSTM filter CNN 32 dan LSTM 32 dengan MAE 2.2065, MSE 0.4095, RMSE 1.4854, dan R-Squared 0.9208, serta model LSTM *autoencoder* filter LSTM 64 dengan MAE 0.001010, MSE 0.000017, RMSE 0.004146, dan R-Squared 0.999845. Model yang digunakan pada perangkat 3D Anemometer: model hybrid CNN-LSTM filter CNN 32 dan LSTM 64 dengan MAE 0.1302, MSE 0.1322, RMSE 0.3608, dan R-Squared 0.9994, serta model LSTM *autoencoder* filter LSTM 64 dengan MAE 0.009898, MSE 0.000680, RMSE 0.026069, dan R-Squared 0.986413. Model yang digunakan pada perangkat Microclimate Station: model hybrid CNN-LSTM filter CNN 32 dan LSTM 64 dengan MAE 1440.2135, MSE 8.8093, RMSE 37.9501, dan R-Squared 0.6622, serta model LSTM *autoencoder* filter LSTM 64 dengan MAE 0.017724, MSE 0.003332, RMSE 0.057727, dan R-Squared 0.968167. Secara keseluruhan, perangkat 3D Anemometer menghasilkan model *hybrid* CNN-LSTM terbaik dengan nilai MAE, MSE, dan RMSE terkecil serta R-Squared tertinggi, sedangkan Gas Analyzer menghasilkan model LSTM *autoencoder* terbaik dengan nilai MAE, MSE, dan RMSE terkecil serta R-Squared tertinggi.
2. Sistem terintegrasi berhasil dilakukan dengan secara keseluruhan penggunaan CPU Raspberry Pi berada di rentang 0% hingga 54,20%, penggunaan RAM berada di rentang 154.16 MB hingga 736.07 MB, waktu inferensi model rata-

rata terkecil untuk model LSTM *autoencoder* pada sistem prediksi-deteksi Gas Analyzer sebesar 0.005214 detik dan untuk model *hybrid* CNN-LSTM pada sistem prediksi-deteksi Gas Analyzer sebesar 0.000766 detik.

3. Sistem terdistribusi berhasil dilakukan dengan secara keseluruhan penggunaan CPU berada di rentang 0% hingga 100%, penggunaan RAM berada di rentang 182.71 MB hingga 1402.23 MB, waktu inferensi model rata-rata terkecil untuk model LSTM *autoencoder* pada sistem prediksi-deteksi Gas Analyzer sebesar 0.005074 detik dan untuk model *hybrid* CNN-LSTM pada sistem prediksi-deteksi Gas Analyzer sebesar 0.000777 detik.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, didapat beberapa saran pada penelitian ini:

1. Penelitian ini menggunakan model CNN-LSTM dan LSTM Autoencoder dengan arsitektur sederhana, yaitu satu lapisan dengan dua variasi jumlah filter. Untuk meningkatkan akurasi dan kemampuan generalisasi model, disarankan pada penelitian selanjutnya untuk mengeksplorasi arsitektur yang lebih kompleks, seperti penggunaan beberapa lapisan (*multi-layer*).
2. Validasi hasil deteksi anomali dalam penelitian ini belum melibatkan data kejadian nyata atau catatan gangguan perangkat. Guna meningkatkan keandalan sistem, disarankan pada penelitian mendatang agar menyertakan pencatatan kejadian aktual di lapangan, seperti log pemeliharaan atau dokumentasi kerusakan perangkat, yang dapat dijadikan *ground truth*.
3. Sistem deteksi anomali yang dikembangkan menunjukkan sensitivitas yang tinggi, yang ditandai dengan jumlah deteksi anomali yang cukup besar. Meskipun hal ini mencerminkan kemampuan model dalam mengenali pola yang menyimpang, tingginya jumlah deteksi dapat mengindikasikan kemungkinan *false positive* yang tinggi. Penelitian selanjutnya disarankan untuk melakukan kalibrasi ambang batas deteksi dan menyusun batas toleransi deviasi alami sensor.

DAFTAR PUSTAKA

- [1] E. Ardelia, "Proyeksi Penerapan Pajak Karbon dalam Upaya Menekan Emisi Gas Rumah Kaca Pada Sektor Pertanian dan Perkebunan di Indonesia," *Innov. J. Soc. Sci. Res.*, vol. 3, no. 4, pp. 9070–9080, 2023, doi: <https://doi.org/10.31004/innovative.v3i4>.
- [2] S. Anggraini and Y. Wiratama Arifin, "Analysis Of Palm Oil Carbon Stock Generating Plant Phase (TM <20 Years) In Silk Land With 40-60 Cm," *AGRITEPA J. Ilmu dan Teknol. Pertan.*, vol. 8, no. 1, pp. 1–8, 2021, doi: <https://doi.org/10.37676/agritepa.v8i1.1290>.
- [3] N. Hafidhoh, A. P. Atmaja, G. N. Syaifuddiin, I. B. Sumafta, S. M. Pratama, and H. N. Khasanah, "Machine Learning untuk Prediksi Kegagalan Mesin dalam Predictive Maintenance System," *J. Masy. Inform.*, vol. 15, no. 1, pp. 56–66, 2024, doi: <https://doi.org/10.14710/jmasif.15.1.63641>.
- [4] D. G. Kim and J. Y. Choi, "Optimization of Design Parameters in LSTM Model for Predictive Maintenance," *Appl. Sci.*, vol. 11, no. 14, pp. 1–16, 2021, doi: <https://doi.org/10.3390/app11146450>.
- [5] A. Nasser and H. Al-Khazraji, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, pp. 721–730, 2022, doi: <https://doi.org/10.11591/ijece.v12i1.pp721-730>.
- [6] F. Lachehab, M. Benzaoui, S. A. Tadjer, A. Bensmaine, and H. Hamma, "LSTM-Autoencoder Deep Learning Model for Anomaly Detection in Electric Motor," *Energies*, vol. 17, no. 10, pp. 1–18, 2024, doi: <https://doi.org/10.3390/en17102340>.
- [7] Paspi, "Carbon Trading dan Potensi Perkebunan Sawit Indonesia," *J. Anal. Palm Oil Strateg. Issues*, vol. IV, no. 10, pp. 807–814, 2023, [Online]. Available: https://www.palmoilina.asia/wp-content/uploads/2023/02/4.10.-CARBON-TRADING-AND-THE-POTENTIAL-OF-INDONESIAS-OIL-PALM-PLANTATIONS_.pdf.
- [8] G. Burba, *Eddy Covariance Method for Scientific, Regulatory, and Commercial Applications*, 2022nd ed. Lincoln: LI-COR Biosciences, 2022.
- [9] J. Maktoubian, M. S. Taskhiri, and P. Turner, "Intelligent Predictive Maintenance (IPdM) in Forestry: A Review of Challenges and Opportunities Jamal," *Forests*, vol. 12, no. 11, pp. 1–26, 2021, doi: <https://doi.org/10.3390/f12111495>.
- [10] R. Ben Ayed and M. Hanana, "Artificial Intelligence to Improve the Food and Agriculture Sector," *J. Food Qual.*, vol. 2021, no. 1, pp. 1–7, 2021, doi: <https://doi.org/10.1155/2021/5584754>.
- [11] A. Flores, H. Tito-Chura, O. Cuentas-Toledo, V. Yana-Mamani, and D.

- Centty-Villafuerte, “PM2.5 Time Series Imputation with Moving Averages, Smoothing, and Linear Interpolation,” *Computers*, vol. 13, no. 12, pp. 1–20, 2024, doi: <https://doi.org/10.3390/computers13120312>.
- [12] E. Hvitfeldt, “Feature Engineering A-Z.” pp. 1–146, 2024, [Online]. Available: <https://feaz-book.com/>.
- [13] TensorFlow Developers, “TensorFlow.” Google, 2025, doi: <https://doi.org/10.5281/zenodo.4724125>.
- [14] C. Ranjan, *Understanding Deep Learning Application in Rare Event Prediction*, First Edit. Independently Published, 2020.
- [15] H. Dehghan Shoorkand, M. Nourelfath, and A. Hajji, “A hybrid CNN-LSTM model for joint optimization of production and imperfect predictive maintenance planning,” *Reliab. Eng. Syst. Saf.*, vol. 241, pp. 1–11, 2024, doi: <https://doi.org/10.1016/j.ress.2023.109707>.
- [16] U. Michelucci, “An Introduction to Autoencoders,” *arXiv preprint*. pp. 1–26, 2022, doi: <https://doi.org/10.48550/arXiv.2201.03898>.
- [17] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, “Autoencoders and their applications in machine learning: a survey,” *Artif. Intell. Rev.*, vol. 57, no. 28, pp. 1–52, 2024, doi: <https://doi.org/10.1007/s10462-023-10662-6>.
- [18] Y. Wei, J. Jang-Jaccard, W. Xu, F. Sabrina, S. Camtepe, and M. Boulic, “LSTM-Autoencoder-Based Anomaly Detection for Indoor Air Quality Time-Series Data,” *IEEE Sens. J.*, vol. 23, no. 4, pp. 3787–3800, 2023, doi: <https://doi.org/10.1109/JSEN.2022.3230361>.
- [19] J. Huang, G. Niu, H. Guan, and S. Song, “Ultra-Short-Term Wind Power Prediction Based on LSTM with Loss Shrinkage Adam,” *Energies*, vol. 16, no. 9, pp. 1–13, 2023, doi: <https://doi.org/10.3390/en16093789>.
- [20] S. Lau, J. Gonzalez, and D. Nolan, “Learning Data Science.” O’Reilly Media, Inc., Sebastopol, pp. 1–596, 2023, [Online]. Available: <https://learningds.org/>.
- [21] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–24, 2021, doi: <https://doi.org/10.7717/PEERJ-CS.623>.
- [22] D. F. Hritcan, A. Graur, and D. Balan, “Securing IoT Environments Using ZeroTier and OPNsense,” in *2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2024, pp. 1–4, doi: <https://doi.org/10.1109/RoEduNet64292.2024.10722755>.
- [23] Y. Zhang *et al.*, “Implementing and Deploying an ML Pipeline for IoT Intrusion Detection with Node-RED,” in *CPS-IoT Week ’23: Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, 2023, pp. 247–253, doi: <https://doi.org/10.1145/3576914.3589807>.
- [24] D. K. Barua, “Beaufort Wind Scale,” in *Encyclopedia of Coastal Science*, 2nd ed., Charles W. Finkl and C. Makowski, Eds. Gewerbestrasse: Springer Cham, 2019, pp. 315–317.