

TUGAS AKHIR - EF234801

Peramalan Deret Waktu untuk Prediksi Parameter Operasional untuk Optimasi Kapasitas Sistem Penyimpanan Energi Baterai (BESS)

Rr. Diajeng Alfisyahrinnisa Anandha

NRP 5025211147

Dosen Pembimbing 1

Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D., IPM.

NIP 198106202005011003

Dosen Pembimbing 2

Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D.

NIP 198412102014042003

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025



TUGAS AKHIR - EF234801

Peramalan Deret Waktu untuk Prediksi Parameter Operasional untuk Optimasi Kapasitas Sistem Penyimpanan Energi Baterai (BESS)

Rr. Diajeng Alfisyahrinnisa Anandha

NRP 5025211147

Dosen Pembimbing 1

Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D., IPM.

NIP 198106202005011003

Dosen Pembimbing 2

Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D.

NIP 198412102014042003

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025



FINAL PROJECT - EF234801

**TIME SERIES FORECASTING FOR OPERATIONAL
PARAMETER *PREDICTION* TO OPTIMIZE BATTERY
ENERGY STORAGE SYSTEM (BESS) *CAPACITY***

Rr. Diajeng Alfisyahrinnisa Anandha

NRP 5025211147

Advisor

Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D., IPM.

NIP 198106202005011003

Co-Advisor

Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D.

NIP 198412102014042003

Undergraduate Study Program of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2025

LEMBAR PENGESAHAN

PERAMALAN DERET WAKTU UNTUK PREDIKSI PARAMETER OPERASIONAL UNTUK OPTIMASI KAPASITAS SISTEM PENYIMPANAN ENERGI BATERAI (BESS)

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh : Rr. Diajeng Alfisyahrinnisa Anandha
NRP. 5025211147

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Ary Mazharuddin Shiddiqi, S.Kom., Pembimbing
M.Comp.Sc., Ph.D., IPM.
2. Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. Ko-pembimbing
3. Dr. Kelly Rossa Sungkono, S.Kom., M.Kom. Penguji
4. Dr. Sarwosri, S.Kom. M.T Penguji

SURABAYA

Juli, 2025

[Halaman ini sengaja dikosongkan]

APPROVAL SHEET

TIME SERIES FORECASTING FOR OPERATIONAL PARAMETER *PREDICTION* TO OPTIMIZE BATTERY ENERGY STORAGE SYSTEM (BESS) *CAPACITY*

FINAL PROJECT

Submitted to fulfill one of the requirements
for obtaining a degree Computer Science at
Undergraduate Study Program of Informatics Engineering
Department of Informatics
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember

By: Rr. Diajeng Alfisyahrinnisa Anandha

NRP. 5025211147

Approved by Final Project Examiner Team:

1. Ir. Ary Mazharuddin Shiddiqi, S.Kom., Advisor
M.Comp.Sc., Ph.D., IPM.
2. Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. Co-Advisor
3. Dr. Kelly Rossa Sungkono, S.Kom., M.Kom. Examiner
4. Dr. Sarwosri, S.Kom. M.T Examiner

SURABAYA

July, 2025

[Halaman Ini Sengaja Dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Rr. Diajeng Alfisyahrinnisa Anandha / 5025211147
Program studi : Teknik Informatika
Dosen Pembimbing / NIP : Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc.,
Ph.D., IPM. / 198106202005011003
Dosen Ko-Pembimbing/ NIP : Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. /
198412102014042003

dengan ini menyatakan bahwa Tugas Akhir dengan judul "*Peramalan Deret Waktu Untuk Prediksi Parameter Operasional Untuk Optimasi Kapasitas Sistem Penyimpanan Energi Baterai (BESS)*" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 17 Juli 2025

Mengetahui
Dosen Pembimbing


Ir. Ary Mazharuddin Shiddiqi, S.Kom.,
M.Comp.Sc., Ph.D., IPM.
NIP. 198106202005011003

Mahasiswa


Rr. Diajeng Alfisyahrinnisa Anandha
NRP. 5025211147

Dosen Ko-Pembimbing


Ratih Nur Esti Anggraini, S.Kom., M.Sc.,
Ph.D.
NIP. 198412102014042003

[Halaman ini sengaja dikosongkan]

STATEMENT OF ORIGINALITY

The undersigned below:

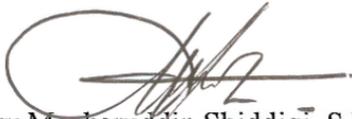
Name of student / NRP : Rr. Diajeng Alfisyahrinnisa Anandha / 5025211147
Department : Informatics Engineering
Advisor / NIP : Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc.,
Ph.D., IPM. / 198106202005011003
Co-Advisor / NIP : Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. /
198412102014042003

Hereby declare that Final Project with the title of "*Time Series Forecasting For Operational Parameter Prediction To Optimize Battery Energy Storage System (BESS) Capacity*" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, 17 July 2025.....

Acknowledge
Advisor



Ir. Ary Mazharuddin Shiddiqi, S.Kom.,
M.Comp.Sc., Ph.D., IPM.
NIP. 198106202005011003

Student



Rr. Diajeng Alfisyahrinnisa Anandha
NRP. 5025211147

Co-Advisor



Ratih Nur Esti Anggraini, S.Kom., M.Sc.,
Ph.D.
NIP. 198412102014042003

[Halaman Ini Sengaja Dikosongkan]

PERNYATAAN KODE ETIK PENGGUNAAN AI GENERATIF

Code of Conduct Statement: Generative AI or AI-Assisted Usage

Saya yang bertanda tangan di bawah ini:

I, the undersigned:

Nama Mahasiswa / : Rr. Diajeng Alfisyahrinnisa Anandha / 5025211147
NRP

Full Name / Student ID

Program Studi : S-1 Teknik Informatika

Study Program

Judul Tugas Akhir : Peramalan Deret Waktu Untuk Prediksi Parameter
Operasional Untuk Optimasi Kapasitas Sistem
Penyimpanan Energi Baterai (BESS)

Final Project Title

dengan ini menyatakan bahwa pada Tugas Akhir dengan judul di atas tersebut:

hereby declare that in the Final Project with the above title:

No.	Pernyataan <i>Statement</i>	
1	Saya hanya menggunakan AI generatif sebagai alat bantu untuk memperbaiki tata bahasa. AI generatif tidak digunakan untuk membuat isi Tugas Akhir. <i>I only used generative AI as a tool to improve the readability or language of the text in my Final Project. It was not used to generate a complete text of my work.</i>	✓
2	Saya telah memeriksa dan/atau memperbaiki seluruh bagian dari Tugas Akhir saya yang dibantu oleh AI generatif agar sesuai dengan baku mutu penulisan karya ilmiah. <i>I have reviewed and refined all aspects of my work that generative AI assists with, ensuring it adheres to the standards of academic writing.</i>	✓
3	Saya tidak menggunakan AI generatif untuk pembuatan data primer, grafik dan/atau tabel pada Tugas Akhir saya. <i>I did not use generative AI to generate primary data, figures, and/or tables in my work.</i>	✓
4	Saya telah memberikan atribusi/pengakuan terhadap alat AI yang digunakan, secara rinci pada suatu bagian pada lampiran. <i>I have acknowledged the use of generative AI in any part of the work in the specific appendix page.</i>	✓
5	Saya memastikan tidak ada plagiarisme, termasuk hal yang berasal dari penggunaan AI generatif. <i>I have ensured that there is no plagiarism issue in the work, including any parts generated by AI.</i>	✓

Surabaya, 22 Juli 2025

Mahasiswa



Rr. Diajeng Alfisyahrinnisa Anandha
NRP. 5025211147

[Halaman Ini Sengaja Dikosongkan]

ABSTRAK

PERAMALAN DERET WAKTU UNTUK PREDIKSI PARAMETER OPERASIONAL UNTUK OPTIMASI KAPASITAS SISTEM PENYIMPANAN ENERGI BATERAI (BESS)

Nama Mahasiswa / NRP : Rr. Diajeng Alfisyahrinnisa Anandha/5025211147
Departemen : Teknik Informatika FTEIC - ITS
Dosen Pembimbing : Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc.,
Ph.D., IPM.
Dosen Ko-Pembimbing : Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D.

Abstrak

Pada *off-grid power system*, penerapan *Photovoltaic* (PV) sangatlah menguntungkan karena sumber energi listrik dipasok melalui solar panel dan memperoleh penghematan biaya yang signifikan dari proses *de-dieselization*. Akan tetapi, apabila hanya bergantung pada PV system, maka akan terjadi berbagai masalah yaitu tidakseimbangan *supply-demand* dan hasil voltase yang bervariasi. PV perlu diintegrasikan dengan *Battery Energi Storage System* (BESS) yang mampu memitigasi daya fluktuasi untuk menangani masalah tidakseimbangan dalam *supply demand*.

Penelitian ini berfokus pada optimasi kapasitas untuk BESS dan PV dalam memenuhi permintaan listrik masyarakat dengan menggunakan algoritma *Marine Predator Algorithm* (MPA). Selain itu, penelitian ini memberikan rekomendasi berupa prediksi parameter operasional (*temperature*, *Global Horizontal Irradiance* [GHI], dan *demand* listrik) dengan metode peramalan deret waktu menggunakan model LSTM, *Hybrid LSTM with Transformer*, FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB *with* LSTM, dan Hybrid FEDFormer WEB *with* LSTM untuk menentukan waktu optimal bagi baterai dalam melakukan penyimpanan dan pelepasan energi. Dipilih satu model terbaik tiap parameter operasional menggunakan evaluasi metrik berupa Mean Absolute Error (MAE) dan Mean Squared Error (MSE).

Hasil evaluasi menunjukkan bahwa dari lima percobaan yang dilakukan, model FEDFormer WEB memiliki kinerja terbaik pada parameter *temperature* dengan nilai *average* MAE sebesar 0,156 dan nilai *average* MSE sebesar 0,039. Selain itu, model Hybrid FEDFormer WEB *with* LSTM memiliki kinerja terbaik pada parameter GHI dengan nilai *average* MAE sebesar 56,123 dan nilai *average* MSE sebesar 10053,324. Terakhir, model FEDFormer WEB memiliki kinerja terbaik pada parameter *demand* listrik dengan nilai *average* MAE sebesar 18,296 dan nilai *average* MSE sebesar 604,718. Hasil dari berbagai parameter tersebut dilakukan proses MPA dan memperoleh kapasitas PV sebesar 7303 kW dan kapasitas BESS sebesar 10689 kW yang menghasilkan total biaya minimum sebesar \$14.459.293,15 NTD.

Kata kunci: *BESS, FEDFormer, Marine Predator Algorithm, Off-grid System, Optimizing, Transformer*

[Halaman ini sengaja dikosongkan]

ABSTRACT

TIME SERIES FORECASTING FOR OPERATIONAL PARAMETER PREDICTION TO OPTIMIZE BATTERY ENERGY STORAGE SYSTEM (BESS) CAPACITY

Student Name / NRP : Rr. Diajeng Alfisyahrinnisa Anandha/5025211147
Department : Teknik Informatika FTEIC - ITS
Advisor : Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc.,
Ph.D., IPM.
Co-Advisor : Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D.

Abstract

In off-grid power systems, the implementation of Photovoltaic (PV) technology is highly beneficial as electricity is supplied via solar panels, leading to significant cost savings through the de-dieselization process. However, relying solely on PV systems poses several challenges, such as supply-demand imbalances and voltage fluctuations. To address these issues, PV systems need to be integrated with a Battery Energy Storage System (BESS), which can mitigate power fluctuations and ensure a stable power supply.

This study focuses on optimizing the capacity of both BESS and PV to meet electricity demands using the Marine Predator Algorithm (MPA). Additionally, this research provides operational parameter predictions (temperature, Global Horizontal Irradiance [GHI], and electricity demand) using time series forecasting models, including LSTM, Hybrid LSTM with Transformer, FEDFormer with Fourier Enhanced Block (FEDFormer FEB), FEDFormer with Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB with LSTM, and Hybrid FEDFormer WEB with LSTM. The best-performing model for each operational parameter was selected based on evaluation metrics, including Mean Absolute Error (MAE) and Mean Squared Error (MSE).

The evaluation results indicate that among five experimental trials, the FEDFormer WEB model yielded the best performance for the temperature parameter, achieving an average MAE of 0.156 and an average MSE of 0.039. Furthermore, the Hybrid FEDFormer WEB with LSTM model demonstrated the highest accuracy for the GHI parameter, with an average MAE of 56.123 and an average MSE of 10053.234. Lastly, the FEDFormer WEB model achieved the best results for electricity demand forecasting, with an average MAE of 18.296 and an average MSE of 604.718. Based on these predicted parameters, the MPA optimization process determined the optimal PV capacity of 7303 kW and BESS capacity of 10689 kW, resulting in a total minimum cost of \$14,459,293.15 NTD.

Keywords: *BESS, FEDFormer, Marine Predator Algorithm, Off-grid System, Optimizing, Transformer*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis sampaikan kepada Allah SWT atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian tugas akhir yang dirangkum dalam buku tugas akhir berjudul “*Peramalan Deret Waktu Untuk Prediksi Parameter Operasional Untuk Optimasi Kapasitas Sistem Penyimpanan Energi Baterai (BESS)*”. Penulisan buku ini tidak mungkin dapat terselesaikan tanpa adanya dukungan dan bantuan dari berbagai pihak. Maka dari itu, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penulis dalam menyelesaikan penulisan ini:

1. Allah SWT serta junjungan besar baginda Nabi Muhammad SAW, karena rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir dan juga perkuliahan di Teknik Informatika ITS,
2. Kedua orang tua saya, Ayah dan Ibu, yang selalu memberikan semangat, dukungan, dan motivasi dalam pengerjaan Tugas Akhir Ini,
3. Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, dan Institut Teknologi Sepuluh Nopember yang telah menerima, mendidik, dan menempa penulis,
4. Bapak Ir. Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D., IPM., Bapak Anindhita Dewabharata, PhD. dan Ibu Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. yang telah membimbing penelitian dan penyusunan buku tugas akhir ini,
5. Seluruh sahabat-sahabat penulis yaitu Anindya Gita, Rara Zerlina, Septiandini, Stevani, Hera, Nasywa, Rayssa Ravelia, Syarifah Talitha, Wan Sabrina, Osvaldo Kurniawan,
6. Seluruh teman-teman penulis yang tidak bisa disebutkan namanya satu per satu,
7. Choi Seungcheol, Hou Minghao, Liu Xueyi, Chen Zheyuan, Lee Donghae, dan Caleb yang menjadi *support system* penulis selama proses pengerjaan Tugas Akhir,
8. Serta seluruh pihak yang tidak bisa penulis sebutkan satu per satu.

Penulis sadar bahwa buku tugas akhir ini masih jauh dari kata sempurna. Oleh sebab itu, penulis masih sangat terbuka kepada seluruh saran dan masukan yang datang untuk memperbaiki tulisan ini di masa yang akan datang. Penulis menyampaikan permohonan maaf yang sebesar-besarnya apabila ada konten di buku ini yang kurang berkenan bagi pembaca. Semoga buku tugas akhir dan penelitian ini dapat memberikan manfaat bagi ilmu bidang deep learning.

Surabaya,

Rr. Diajeng Alfisyahrinnisa Anandha

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN	I
APPROVAL SHEET	III
PERNYATAAN ORISINALITAS.....	V
STATEMENT OF ORIGINALITY	VII
PERNYATAAN KODE ETIK PENGGUNAAN AI GENERATIF.....	IX
ABSTRAK.....	XI
KATA PENGANTAR.....	XV
DAFTAR ISI	XVII
DAFTAR GAMBAR.....	XXI
DAFTAR TABEL	XXVII
DAFTAR KODE SEMU	XXXI
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Hasil Penelitian Terdahulu.....	5
2.2 Dasar Teori.....	7
2.2.1 Battery	7
2.2.2 Battery Energy Storage System (BESS)	8
2.2.3 Parameter Operasional	9
2.2.4 Timeseries Forecasting	9
2.2.5 Model Optimasi.....	11
2.2.6 Metriks Evaluasi	15
2.2.7 Uji T	16
BAB 3 METODOLOGI	19
3.1 Dataset.....	20
3.2 Pemilihan Fitur Terbaik	21
3.3 Preprocessing Data.....	23

3.4	Pembagian Dataset Metode Timeseries Forecasting	23
3.5	Training Model	29
3.6	Pengembangan Model Penelitian.....	29
3.6.1	LSTM	29
3.6.2	Hybrid LSTM With Transformer.....	31
3.6.3	Hybrid FEDFormer With LSTM	32
3.6.4	FEDFormer	40
3.6.5	Marine Predator Algorithm.....	42
3.7	Pengembangan Antarmuka Pengguna	46
3.8	Evaluasi Dan Perbandingan	51
3.8.1	Perangkat Keras	52
3.9	Spesifikasi Environment	52
3.9.1	Perangkat Lunak	52
BAB 4	HASIL DAN PEMBAHASAN.....	53
4.1	Parameter Penelitian	53
4.2	Skenario Eksperimen	54
4.3	Hasil Penelitian	54
4.3.1	Hasil Implementasi Skenario 1	55
4.3.2	Hasil Implementasi Skenario 2	69
4.3.3	Hasil Implementasi Skenario 3	84
4.3.4	Hasil Implementasi Skenario 4	98
4.3.5	Hasil Implementasi Skenario 5	111
4.3.6	Hasil Implementasi Marine Predator Algorithm.....	127
4.4	Pembahasan.....	134
4.4.1	Pembahasan Skenario 1	134
4.4.2	Pembahasan Skenario 2	137
4.4.3	Pembahasan Skenario 3	140
4.4.4	Pembahasan Skenario 4	142
4.4.5	Pembahasan Skenario 5	145
4.4.6	Pembahasan Hasil Marine Predator Algorithm	147
4.5	Diskusi	151
BAB 5	KESIMPULAN DAN SARAN.....	155
5.1	Kesimpulan	155

5.2	Saran.....	155
	DAFTAR PUSTAKA.....	157
	LAMPIRAN	161
	BIODATA PENULIS.....	173

[Halaman Ini Sengaja Dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Skala makro dan mikro pada off-grid system	8
Gambar 2.2 Arsitektur vanilla LSTM	10
Gambar 2.3 Arsitektur model Transformer	10
Gambar 2.4 Struktur Model FEDFormer	11
Gambar 3.1 Alur Proses Metodologi	19
Gambar 3.2 Grafik Karakteristik Fitur dalam Dataset NSRDB	20
Gambar 3.3 Grafik Karakteristik Dataset NTUST	21
Gambar 3.4 Hasil Correlation Matrix untuk Dataset	21
Gambar 3.5 Flowchart LSTM	30
Gambar 3.6 Flowchart Hybrid LSTM with Transformer	31
Gambar 3.7 Flowchart Awal FEDFormer	33
Gambar 3.8 Flowchart Training dan Validation pada FEDFormer	33
Gambar 3.9 Flowchart Testing pada FEDFormer	34
Gambar 3.10 Flowchart FEDFormer Fourier	35
Gambar 3.11 Flowchart FEDFormer Wavelet	38
Gambar 3.12 Rancangan Antarmuka Prediksi Temperature	47
Gambar 3.13 Rancangan Antarmuka Prediksi GHI	48
Gambar 3.14 Rancangan Antarmuka Prediksi Demand Listrik	48
Gambar 3.15 Usecase Diagram Antarmuka Pengguna	49
Gambar 3.16 Activity Diagram Upload File	50
Gambar 3.17 Activity Diagram Mendapatkan Hasil Prediksi	50
Gambar 4.1 Grafik Validation loss Temperature Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	58
Gambar 4.2 Grafik Validation loss Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	58
Gambar 4.3 Grafik Validation MAE Temperature Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	59
Gambar 4.4 Grafik Validation MAE Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	59
Gambar 4.5 Grafik Prediksi Temperature Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	60
Gambar 4.6 Grafik Prediksi Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	60
Gambar 4.7 Grafik Validation loss GHI Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	63
Gambar 4.8 Grafik Validation loss GHI Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	63
Gambar 4.9 Grafik Validation MAE GHI Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	63
Gambar 4.10 Grafik Validation MAE GHI Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	64
Gambar 4.11 Grafik Prediksi GHI Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	64

Gambar 4.12 Grafik Prediksi GHI Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	65
Gambar 4.13 Grafik Validation loss Demand Listrik Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	67
Gambar 4.14 Grafik Validation loss Demand Listrik Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	68
Gambar 4.15 Grafik Validation MAE Demand Listrik Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	68
Gambar 4.16 Grafik Validation MAE Demand Listrik Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	69
Gambar 4.17 Grafik Prediksi Demand Listrik Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB	69
Gambar 4.18 Grafik Prediksi Demand Listrik Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB	69
Gambar 4.19 Grafik Validation loss Temperature Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	72
Gambar 4.20 Grafik Validation loss Temperature Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	73
Gambar 4.21 Grafik Validation MAE Temperature Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	73
Gambar 4.22 Grafik Validation MAE Temperature Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	73
Gambar 4.23 Grafik Prediksi Temperature Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	74
Gambar 4.24 Grafik Prediksi Temperatue Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	74
Gambar 4.25 Grafik Validation loss GHI Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	77
Gambar 4.26 Grafik Validation loss GHI Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	78
Gambar 4.27 Grafik Validation MAE GHI Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	78
Gambar 4.28 Grafik Validation MAE GHI Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	78
Gambar 4.29 Grafik Prediksi GHI Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	79
Gambar 4.30 Grafik Prediksi GHI Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	79
Gambar 4.31 Grafik Validation loss Demand Listrik Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	82
Gambar 4.32 Grafik Validation loss Demand Listrik Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	82
Gambar 4.33 Grafik Validation MAE Demand Listrik Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	83
Gambar 4.34 Grafik Validation MAE Demand Listrik Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	83

Gambar 4.35 Grafik Prediksi Demand Listrik Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	84
Gambar 4.36 Grafik Prediksi Demand Listrik Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM	84
Gambar 4.37 Grafik Validation loss Temperature Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM	87
Gambar 4.38 Grafik Validation loss Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer with LSTM	87
Gambar 4.39 Grafik Validation MAE Temperature Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM	88
Gambar 4.40 Grafik Validation MAE Temperature Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM	88
Gambar 4.41 Grafik Prediksi Temperature Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	88
Gambar 4.42 Gambar Prediksi Temperature Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	89
Gambar 4.43 Grafik Validation loss GHI Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	91
Gambar 4.44 Grafik Validation loss GHI Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	92
Gambar 4.45 Grafik Validation MAE GHI Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	92
Gambar 4.46 Grafik Validation MAE GHI Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	93
Gambar 4.47 Grafik Prediksi GHI Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	93
Gambar 4.48 Grafik Prediksi GHI Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	93
Gambar 4.49 Grafik Validation loss Demand Listrik Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	96
Gambar 4.50 Grafik Validation loss Demand Listrik Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM	96
Gambar 4.51 Grafik Validation MAE Demand Listrik Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	97
Gambar 4.52 Grafik Validation MAE Demand Listrik Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	97
Gambar 4.53 Grafik Prediksi Demand Listrik Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM	97
Gambar 4.54 Grafik Prediksi Demand Listrik Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM.....	98
Gambar 4.55 Grafik Validation loss Temperature Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM.....	100
Gambar 4.56 Grafik Validation loss Temperature Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM.....	101
Gambar 4.57 Grafik Validation MAE Temperature Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM.....	101

Gambar 4.58 Grafik Validation MAE Temperature Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM.....	102
Gambar 4.59 Grafik Prediksi Temperature Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	102
Gambar 4.60 Grafik Prediksi Temperature Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	102
Gambar 4.61 Grafik Validation loss GHI Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	105
Gambar 4.62 Grafik Validation loss GHI Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	105
Gambar 4.63 Grafik Validation MAE GHI Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	106
Gambar 4.64 Grafik Validation MAE GHI Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	106
Gambar 4.65 Grafik Prediksi GHI Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	106
Gambar 4.66 Grafik Prediksi GHI Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	107
Gambar 4.67 Grafik Validation loss Demand Listrik Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	109
Gambar 4.68 Grafik Validation loss Demand Listrik Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM.....	110
Gambar 4.69 Grafik Validation MAE Demand Listrik Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	110
Gambar 4.70 Grafik Validation MAE Demand Listrik Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	110
Gambar 4.71 Grafik Prediksi Demand Listrik Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM.....	111
Gambar 4.72 Grafik Prediksi Demand Listrik Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM	111
Gambar 4.73 Grafik Validation loss Temperature Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB	115
Gambar 4.74 Grafik Validation loss Temperature Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	116
Gambar 4.75 Grafik Validation MAE Temperature Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	116
Gambar 4.76 Grafik Validation MAE Temperature Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	117
Gambar 4.77 Grafik Validation loss GHI Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM	121
Gambar 4.78 Grafik Validation loss GHI Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM	122
Gambar 4.79 Grafik Validation MAE GHI Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM	122
Gambar 4.80 Grafik Validation MAE GHI Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM	122

Gambar 4.81 Grafik Validation loss Demand Listrik Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	126
Gambar 4.82 Grafik Validation loss Demand Listrik Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	126
Gambar 4.83 Grafik Validation MAE Demand Listrik Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	127
Gambar 4.84 Grafik Validation MAE Demand Listrik Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB.....	127
Gambar 4.85 Grafik Hasil MPA	128
Gambar 4.86 Grafik Beban, Output PV, Charge dan Discharge Minggu Pertama	129
Gambar 4.87 Grafik Penjadwalan Charge dan Discharge BESS Minggu Pertama.....	129
Gambar 4.88 Grafik Beban, Output PV, Charge dan Discharge Minggu Kedua	130
Gambar 4.89 Grafik Penjadwalan Charge dan Discharge BESS Minggu Kedua.....	130
Gambar 4.90 Grafik Penjadwalan Charge dan Discharge BESS Minggu Ketiga	131
Gambar 4.91 Grafik Beban, Output PV, Charge dan Discharge Minggu Ketiga.....	131
Gambar 4.92 Grafik Beban, Output PV, Charge dan Discharge Minggu Keempat	132
Gambar 4.93 Grafik Penjadwalan Charge dan Discharge BESS Minggu Keempat.....	132
Gambar 4.94 Grafik Beban, Output PV, Charge dan Discharge Minggu Kelima.....	133
Gambar 4.95 Grafik Penjadwalan Charge dan Discharge BESS Minggu Kelima	133
Gambar 4.96 Hasil Prediksi Antarmuka Temperature.....	148
Gambar 4.97 Hasil Prediksi Antarmuka GHI	148
Gambar 4.98 Hasil Prediksi Antarmuka Demand Listrik.....	149
Gambar 4.99 Excel Perhitungan SG	149

[Halaman Ini Sengaja Dikosongkan]

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait.....	5
Tabel 3.1 Tabel Dataset Penelitian	20
Tabel 3.2 Tabel Nilai Correlation Fitur pada Target Demand Listrik (KW).....	22
Tabel 3.3 Tabel Nilai Correlation Fitur pada Target Temperature.....	22
Tabel 3.4 Tabel Nilai Correlation Fitur pada Target GHI	22
Tabel 3.5 Range Date pada Data Training LSTM	23
Tabel 3.6 Window pada Data Training LSTM	24
Tabel 3.7 Range Date pada Data Validation LSTM.....	24
Tabel 3.8 Window pada Data Validation LSTM.....	24
Tabel 3.9 Range Date pada Data Testing LSTM.....	25
Tabel 3.10 Window pada Data Testing LSTM.....	25
Tabel 3.11 Range Date pada Data Train FEDFormer.....	25
Tabel 3.12 Window Encoder pada Data Train FEDFormer	26
Tabel 3.13 Window Decoder pada Data Train FEDFormer	26
Tabel 3.14 Range Date pada Data Validation FEDFormer	27
Tabel 3.15 Window Encoder pada Data Validation FEDFormer	27
Tabel 3.16 Window Decoder pada Data Validation FEDFormer.....	27
Tabel 3.17 Range Date pada Data Testing FEDFormer	28
Tabel 3.18 Window Encoder pada Data Testing FEDFormer.....	28
Tabel 3.19 Window Decoder pada Data Testing FEDFormer.....	28
Tabel 3.20 Encoder Embedding.....	34
Tabel 3.21 Proses Fourier Block.....	35
Tabel 3.22 Proses Fourier Cross Attention	36
Tabel 3.23 Proses Decoder Embedding	36
Tabel 3.24 Proses MultiWaveletTransform.....	38
Tabel 3.25 Proses MultiWaveletCross.....	39
Tabel 4.1 Parameter Optimal Sizing	54
Tabel 4.2 Skenario Timeseries Forecasting	55
Tabel 4.3 Hasil MAE Skenario 1 Temperature Standar Scaler	56
Tabel 4.4 Hasil MAE Skenario 1 Temperature Skala Asli.....	56
Tabel 4.5 Hasil MSE Skenario 1 Temperature Standar Scaler	57
Tabel 4.6 Hasil MSE Skenario 1 Temperature Skala Asli.....	57
Tabel 4.7 Hasil MAE Skenario 1 GHI Standar Scaler.....	61
Tabel 4.8 Hasil MAE Skenario 1 GHI Skala Asli	61
Tabel 4.9 Hasil MSE Skenario 1 GHI Standar Scaler	62
Tabel 4.10 Hasil MSE Skenario 1 GHI Skala Asli.....	62
Tabel 4.11 Hasil MAE Skenario 1 Demand Listrik Standar Scaler	65
Tabel 4.12 Hasil MAE Skenario 1 Demand Listrik Skala Asli	66
Tabel 4.13 Hasil MSE Skenario 1 Demand Listrik Standar Scaler	66
Tabel 4.14 Hasil MSE Skenario 1 Demand Listrik Skala Asli.....	67
Tabel 4.15 Hasil MAE Skenario 2 Temperature Standar Scaler	70
Tabel 4.16 Hasil MAE Skenario 2 Temperature Skala Asli.....	71
Tabel 4.17 Hasil MSE Skenario 2 Temperature Standar Scaler.....	71

Tabel 4.18 Hasil MSE Skenario 2 Temperature Skala Asli.....	72
Tabel 4.19 Hasil MAE Skenario 2 GHI Standar Scaler.....	75
Tabel 4.20 Hasil MAE Skenario 2 GHI Skala Asli	75
Tabel 4.21 Hasil MSE Skenario 2 GHI Standar Scaler	76
Tabel 4.22 Hasil MSE Skenario 2 GHI Skala Asli.....	76
Tabel 4.23 Hasil MAE Skenario 2 Demand Listrik Standar Scaler	80
Tabel 4.24 Hasil MAE Skenario 2 Demand Listrik Skala Asli	80
Tabel 4.25 Hasil MSE Skenario 2 Demand Listrik Standar Scaler	81
Tabel 4.26 Hasil MSE Skenario 2 Demand Listrik Skala Asli.....	81
Tabel 4.27 Hasil MAE Skenario 3 Temperature Standar Scaler	85
Tabel 4.28 Hasil MAE Skenario 3 Temperature Skala Asli.....	85
Tabel 4.29 Hasil MSE Skenario 3 Temperature Standar Scaler.....	86
Tabel 4.30 Hasil MSE Skenario 3 Temperature Skala Asli.....	86
Tabel 4.31 Hasil MAE Skenario 3 GHI Standar Scaler.....	89
Tabel 4.32 Hasil MAE Skenario 3 GHI Skala Asli	90
Tabel 4.33 Hasil MSE Skenario 3 GHI Standar Scaler	90
Tabel 4.34 Hasil MSE Skenario 3 GHI Skala Asli.....	91
Tabel 4.35 Hasil MAE Skenario 3 Demand Listrik Standar Scaler	94
Tabel 4.36 Hasil MAE Skenario 3 Demand Listrik Skala Asli	94
Tabel 4.37 Hasil MSE Skenario 3 Demand Listrik Standar Scaler	95
Tabel 4.38 Hasil MSE Skenario 3 Demand Listrik Skala Asli.....	95
Tabel 4.39 Hasil MAE Skenario 4 Temperature Standar Scaler	98
Tabel 4.40 Hasil MAE Skenario 4 Temperature Skala Asli.....	99
Tabel 4.41 Hasil MSE Skenario 4 Temperature Standar Scaler.....	99
Tabel 4.42 Hasil MSE Skenario 4 Temperature Skala Asli.....	100
Tabel 4.43 Hasil MAE Skenario 4 GHI Standar Scaler.....	103
Tabel 4.44 Hasil MAE Skenario 4 GHI Skala Asli	103
Tabel 4.45 Hasil MSE Skenario 4 GHI Standar Scaler	104
Tabel 4.46 Hasil MSE Skenario 4 GHI Skala Asli.....	104
Tabel 4.47 Hasil MAE Skenario 4 Demand Listrik Standar Scaler	107
Tabel 4.48 Hasil MAE Skenario 4 Demand Listrik Skala Asli	108
Tabel 4.49 Hasil MSE Skenario 4 Demand Listrik Standar Scaler	108
Tabel 4.50 Hasil MSE Skenario 4 Demand Listrik Skala Asli.....	109
Tabel 4.51 Hasil MAE Skenario 5 Temperature Standar Scaler	112
Tabel 4.52 Hasil MAE Skenario 5 Temperature Skala Asli.....	113
Tabel 4.53 Hasil MSE Skenario 5 Temperature Standar Scaler.....	114
Tabel 4.54 Hasil MSE Skenario 5 Temperature Skala Asli.....	115
Tabel 4.55 Hasil MAE Skenario 5 GHI Standar Scaler.....	118
Tabel 4.56 Hasil MAE Skenario 5 GHI Skala Asli	118
Tabel 4.57 Hasil MSE Skenario 5 GHI Standar Scaler	119
Tabel 4.58 Hasil MSE Skenario 5 GHI Skala Asli.....	120
Tabel 4.59 Hasil MAE Skenario 5 Demand Listrik Standar Scaler	123
Tabel 4.60 Hasil MAE Skenario 5 Demand Listrik Skala Asli	124
Tabel 4.61 Hasil MSE Skenario 5 Demand Listrik Standar Scaler	125
Tabel 4.62 Hasil MSE Skenario 5 Demand Listrik Skala Asli.....	125
Tabel 4.63 Hasil Chromosome Terbaik	128

Tabel 4.64 Tabel pearson correlation.....	134
Tabel 4.65 Hasil T-Test Skenario 1 Temperature.....	135
Tabel 4.66 Hasil T-Test Skenario 1 GHI.....	135
Tabel 4.67 Hasil T-Test Skenario 1 Demand Listrik.....	136
Tabel 4.68 Hasil T-Test Skenario 2 Temperature.....	137
Tabel 4.69 Hasil T-Test Skenario 2 GHI.....	138
Tabel 4.70 Hasil T-Test Skenario 2 Demand Listrik.....	139
Tabel 4.71 Hasil T-Test Skenario 3 Temperature.....	140
Tabel 4.72 Hasil T-Test Skenario 3 GHI.....	141
Tabel 4.73 Hasil T-Test Skenario 3 Demand Listrik.....	142
Tabel 4.74 Hasil T-Test Skenario 4 Temperature.....	143
Tabel 4.75 Hasil T-Test Skenario 4 GHI.....	144
Tabel 4.76 Hasil T-Test Skenario 4 Demand Listrik.....	144
Tabel 4.77 Hasil Waktu Kapan Baterai Idle.....	150
Tabel 5.1 Pengakuan Penggunaan Alat Berbasis AI.....	171

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SEMU

Kode semu 3.1 Algoritma LSTM	30
Kode semu 3.2 Algoritma Hybrid LSTM with Transformer	32
Kode semu 3.3 Algoritma Transformer Encoder	32
Kode semu 3.4 Algoritma FEDFormer with LSTM	39
Kode semu 3.5 Algoritma FEDFormer without LSTM	40
Kode semu 3.6 Algoritma Decoder without LSTM	41
Kode semu 3.7 Algoritma MarinePredator Algorithm	43
Kode semu 3.8 Algoritma High Velocity Action	45
Kode semu 3.9 Algoritma Transition Action	46
Kode semu 3.10 Algoritma Low Velocity Action	46
Kode semu 3.11 Algoritma Levy Flight	46
Kode semu 3.12 Algoritma Brownian Motion	46
Kode semu 3.13 AntarMuka Pengguna	50
Kode semu 5.1 Algoritma Series Decomposition	161
Kode semu 5.2 Algoritma Moving Average	161
Kode semu 5.3 Algoritma DataEmbedding without Positional Encoding	162
Kode semu 5.4 Algoritma TokenEmbedding	162
Kode semu 5.5 Algoritma PositionalEmbedding	162
Kode semu 5.6 Algoritma TemporalEmbedding	163
Kode semu 5.7 Algoritma TimeFeatureEmbedding	163
Kode semu 5.8 Algoritma FourierBlock	163
Kode semu 5.9 Algoritma GetFrequencyModes	164
Kode semu 5.10 Algoritma FourierCrossAttention	164
Kode semu 5.11 Algoritma MultiWaveletTransform	165
Kode semu 5.12 Algoritma MultiWaveletCross	165
Kode semu 5.13 Algoritma AutoCorrelationLayer	166
Kode semu 5.14 Algoritma EncoderLayer	167
Kode semu 5.15 Algoritma Encoder	167
Kode semu 5.16 Algoritma DecoderLayer	168
Kode semu 5.17 Algoritma Decoder	169
Kode semu 5.18 Fitness Function	169

[Halaman ini sengaja dikosongkan]

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Indonesia ikut serta dalam Conference of The Parties (COP) yang bertujuan untuk membatasi perubahan iklim. Pada COP 27, Indonesia berkomitmen untuk mengurangi GHG (*Greenhouse Gas*) Emission dengan target 31,89% dengan upaya sendiri dan target 43,02% dengan bantuan internasional. Dengan menggunakan Analytic Hierarchy Process (AHP), prioritas tertinggi di Indonesia adalah transisi energi (Ali & Kim, 2024). Transisi energi sangatlah penting karena energi merupakan salah satu faktor yang paling penting bagi manusia karena menjadi kebutuhan pokok dalam bidang transportasi, perdagangan, manufaktur, perkantoran, dan utilitas manusia. Dalam penggunaan energi, Indonesia berusaha melakukan transisi energi dengan mengurangi peredaran gas karbon dari yang awalnya menggunakan *fossil fuels* dalam industri menjadi menggunakan *renewable energy* seperti solar, angin, dan *hydroelectric* (Erdiwansyah et al., 2024).

Sebagai badan usaha negara yang memasok energi listrik, PLN (Pembangkit Listrik Negara) menemukan bahwa pembangkit listrik menggunakan solar panel atau *Photovoltaic* (PV) dinilai sangat menguntungkan karena diperoleh penghematan biaya yang signifikan dari proses *de-dieselization* (Paradongan et al., 2024). Di Indonesia, terdapat dua *power system* yang diterapkan yaitu *on-grid power system* dan *off-grid power system*. *On-grid power system* berarti sistem menggunakan *energy* dari pemerintah yaitu PLN sebagai *power sources*, sedangkan *off-grid system* berarti energi yang didapatkan bukan berasal dari sumber energi pemerintah. Salah satu penerapan *off-grid system* yang memiliki kompetensi ekonomi di Indonesia adalah PV *power system* dengan *Energy Storage System* (ESS) dan biasanya, daerah terpencil menggunakan *off-grid system* karena tidak bisa mendapatkan energi langsung dari PLN karena dibatasi oleh gunung, hutan, bahkan pulau yang berbeda (Susilowati et al., 2023).

Power generation dari PV *system* sangat bergantung pada kondisi alam dan lokasi penempatan PV. Akibatnya, apabila hanya mengandalkan PV *system*, maka akan terjadi berbagai masalah, seperti terjadi ketidakseimbangan dalam *supply-demand*, hasil voltase yang bervariasi, dan lain sebagainya. Untuk menangani berbagai masalah tersebut, PV diintegrasikan dengan ESS. Varian ESS yang cocok untuk diintegrasikan dengan PV adalah BESS (*Battery Energy Storage System*) karena dapat memitigasi daya fluktuasi (Rana et al., 2022).

Upaya untuk melakukan optimasi kapasitas PV dan BESS yang terbaik digunakan *machine learning* dalam penerapannya. Terdapat penelitian sebelumnya yang menggunakan pendekatan *nonlinear programming method* dalam optimasi kapasitas baterai dan strategi berbagi manajemen energi dalam *microgrid* dengan tujuan meminimalkan biaya BESS (Symeonidou et al., 2021). Selain itu, terdapat penelitian lain yang menggunakan *mixed-integer linear programming* (MILP) model untuk menentukan ukuran optimal dan tahun untuk mengganti baterai (Amini et al., 2021). Dari penelitian-penelitian yang sudah ada, dalam melakukan optimasi kapasitas PV dan BESS, penelitian tugas akhir ini akan menggunakan *Marine Predator Algorithm* (MPA).

Penelitian tugas akhir yang dilakukan tidak hanya berfokus pada optimasi kapasitas PV dan BESS. Penelitian ini juga akan berfokus pada prediksi *charging* dan *discharging* untuk sistem PV-BESS. Dalam menentukan prediksi *charging* dan *discharging*, digunakan *timeseries forecasting* untuk mengetahui waktu yang tepat bagi BESS dalam *charging* dan *discharging*. Pada penelitian terdahulu, digunakan CNN-LSTM *neural network* model untuk prediksi *charging* dan *discharging* menggunakan metode *timeseries forecasting* pada BESS yang

memiliki hasil bahwa modelnya tidak realistis karena tidak mempertimbangkan *battery life* dalam memprediksi waktu untuk *charging* dan *discharging* pada lokasi yang diteliti yaitu Baja California Sur (BCS), Mexico (Tovar Rosas et al., 2022). Selain itu, terdapat penelitian yang menggunakan RNN-LSTM arsitektur dalam melakukan *timeseries forecasting* yang hasilnya adalah dalam menggunakan dataset summer, menghasilkan *best result*. Akan tetapi, pada winter season, arsitektur tersebut gagal dalam prediksi *peak period* (Hernando C. S, 2023). Penelitian tugas akhir yang dilakukan akan menggunakan beberapa model *machine learning* berupa LSTM, *Hybrid LSTM with Transformer*, FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB *with LSTM*, dan Hybrid FEDFormer WEB *with LSTM* untuk prediksi deret waktu pada *target temperature*, *Global Horizontal Irradiance* (GHI), dan *demand* listrik (KW) dengan mempertimbangkan *battery life*.

Berdasarkan deret waktu dalam penjadwalan operasi PV dan BESS, penelitian tugas akhir ini akan memprediksi waktu yang tepat saat BESS melakukan *charging* atau *discharging* dengan mempertimbangkan permintaan energi listrik dari masyarakat. Selain itu, dibutuhkan kapasitas PV dan BESS yang terbaik agar dapat memenuhi kebutuhan energi listrik masyarakat. Dengan menggunakan sistem ini, diharapkan pengeluaran *electricity supplier* dalam menyediakan PV dan BESS membutuhkan biaya yang seminimal mungkin dengan kapasitas PV dan BESS se-optimal mungkin, serta *life cycle* dalam baterai pun meningkat karena memiliki sistem *charging* dan *discharging* yang tepat.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana mendesain metode untuk prediksi waktu yang tepat untuk BESS melakukan *charging* dan *discharging*?
2. Bagaimana kinerja metode prediksi waktu untuk BESS dalam melakukan *charging* dan *discharging*?
3. Bagaimana melakukan optimasi untuk mendapatkan kapasitas BESS terbaik agar dapat memenuhi permintaan listrik masyarakat dengan biaya minimum?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan yaitu:

1. Dataset dari *National Solar Radiation Database* (NSRDB) yang berisi kondisi cuaca, serta radiasi selama tahun 2019 yang berpusat pada Kota Taipei, Taiwan.
2. Dataset dari National Taiwan University of Science and Technology (NTUST) berupa *demand* listrik di *educational building* pada Kota Taipei, Taiwan.
3. Metode optimasi yang digunakan adalah *Marine Predator Algorithm*.
4. Metode peramalan deret waktu pada musim panas Taiwan yang digunakan adalah LSTM, *Hybrid LSTM with Transformer*, FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB *with LSTM*, dan Hybrid FEDFormer WEB *with LSTM*.
5. Metrik evaluasi yang digunakan pada penelitian ini adalah *Mean Absolute Error* dan *Mean Squared Error* dalam menentukan performa model peramalan deret waktu.
6. Digunakan 1 solar panel dan 1 *battery energy storage system*
7. Prediksi *discharge* dan *charge* pada baterai dilakukan untuk Bulan Agustus 2019 pada musim panas Taiwan, sesuai dengan periode percobaan pada penelitian (Hernando C. S, 2023) yang menjadi acuan penelitian ini.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Mendesain metode optimasi untuk mendapatkan kapasitas BESS agar dapat memenuhi permintaan listrik masyarakat dengan biaya minimum
2. Mendesain metode untuk prediksi nilai radiasi, *temperature*, dan kebutuhan listrik per jam selama satu bulan.
3. Mendesain metode untuk prediksi waktu yang tepat bagi BESS dalam melakukan *charging* dan *discharging*.

1.5 Manfaat

Manfaat dari pengerjaan tugas akhir ini dan dibuatnya sistem ini adalah terbentuknya suatu sistem yang dapat memberikan nilai kapasitas optimal yang tepat untuk PV dan BESS, serta memberikan kapan waktu yang tepat untuk BESS melakukan *charging* dan *discharging*. Hal ini dapat membantu dalam pemberdayaan sumber energi listrik terbarukan menggunakan sistem PV-BESS.

[Halaman ini sengaja dikosongkan]

BAB 2 TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Dalam penyusunan Tugas Akhir ini, terdapat beberapa penelitian terkait yang menjadi inspirasi dan referensi. Berikut adalah beberapa penelitian yang terkait dengan penyusunan Tugas Akhir yang terdapat pada Tabel 2.1. Dari keseluruhan penelitian terkait, dilakukan pengujian menggunakan model Hybrid LSTM dengan Transformer, serta menggunakan model FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB *with* LSTM, dan Hybrid FEDFormer WEB *with* LSTM dalam melakukan peramalan deret waktu. Hasil prediksi dari berbagai model tersebut akan diproses menggunakan *Marine Predator Algorithm* untuk mendapatkan kapasitas BESS dan PV terbaik. Penelitian ini berfokus dalam memaksimalkan penggunaan energi pada PV dan BESS agar dapat memenuhi kebutuhan listrik masyarakat.

Tabel 2.1 Penelitian Terkait

Judul	Hasil Penelitian	Pembeda Penelitian
<i>Optimization of battery energy storage system (BESS) sizing in different electricity market types considering BESS utilization mechanisms and ownerships</i> (Zhang et al., 2024).	Pada penelitian ini, terdapat model untuk menentukan ukuran BESS yang optimal dalam berbagai skenario pada <i>electricity market types</i> yaitu <i>peer-to-grid</i> , <i>peer-to-peer</i> , dan <i>energy storage sharing</i> . Referensi yang dapat diambil dari penelitian ini adalah model matematika yang digunakan untuk skenario (P2G-IDUO-BESS) yaitu penggunaan PV untuk pemenuhan listrik, setelah itu sisanya disimpan dalam BESS sebelum dijual ke <i>utility grid</i> .	Perbedaan dengan penelitian ini adalah skenario dari penyebaran energi listrik yang berbeda. Penelitian pada Tugas Akhir tidak menjual energi listrik ke <i>utility grid</i> , melainkan memaksimalkan penggunaan energi listrik pada PV dan BESS agar dapat memenuhi kebutuhan listrik masyarakat.
<i>Optimal Sizing and Scheduling Battery Storage System and Solar Photovoltaic Grid Connection based on Levelized Cost of Electricity using a Mathematical Model and Recurrent Neural Network</i> (Hernando C. S, 2023)	Pada penelitian ini, terdapat model arsitektur RNN-LSTM yang digunakan untuk prediksi <i>timeseries forecasting</i> . Hasilnya adalah untuk musim panas, masih terdapat deviasi pada beberapa <i>peak</i> , termasuk untuk <i>variable temperature</i> dan <i>Global Horizontal Irradiance (GHI)</i> pada titik-titik ektrim. Selain itu, dilakukan perhitungan optimasi dengan menggunakan AMPL <i>with</i> LINDOglobal server yang menghasilkan bahwa pada	Perbedaan dengan penelitian ini adalah perhitungan optimasi akan dilakukan menggunakan <i>Marine Predator Algorithm (MPA)</i> dengan menggunakan data dari hasil prediksi <i>timeseries forecasting</i> .

	musim panas, dibutuhkan kapasitas BESS yang besar.	
<i>Development, validation, and testing of advanced mathematical models for the optimization of BESS operation</i> (Bovera et al., 2023)	Pada penelitian ini menggunakan <i>mixed-integer linear programming</i> (MILP) model menggunakan <i>non-discrete decision variables</i> dengan <i>linear objective functions</i> dan <i>constrains</i> untuk melakukan optimasi kapasitas BESS dalam memenuhi kebutuhan listrik. Referensi yang dapat diambil dari penelitian ini adalah penggunaan <i>objective function</i> yang dapat digunakan dalam pembuatan <i>fitness function</i> pada <i>Marine Predator Algorithm</i> .	Perbedaan dengan penelitian ini adalah penggunaan <i>Marine Predator Algorithm</i> untuk melakukan optimasi kapasitas BESS dalam memenuhi kebutuhan listrik.
<i>Hybrid LSTM-Transformer model for emotion recognition from speech audio files</i> (Andayani et al., 2022).	Pada penelitian ini menggunakan <i>hybrid LSTM network</i> dan <i>transformer encoder</i> untuk mempelajari <i>long-term dependencies</i> pada <i>speech signals</i> dan <i>classify emotions</i> . Referensi yang dapat diambil dari penelitian ini adalah penggunaan <i>hybrid LSTM</i> dan <i>transformer encoder</i> untuk <i>long-term dependencies</i> .	Adapun perbedaan dengan penelitian ini adalah dataset yang digunakan pada Tugas Akhir ini adalah dataset <i>timeseries</i> .
<i>Itineraries for charging and discharging a BESS using energy predictions based on a CNN-LSTM neural network model in BCS, Mexico</i> (Tovar Rosas et al., 2022).	Pada penelitian ini, terdapat model yang menggunakan <i>hybrid Convolutional Long-Short Time Memory (CNN-LSTM Neural Network)</i> dalam <i>energy time series predictions</i> pada <i>demand</i> dan <i>generation</i> . Referensi yang dapat diambil dari penelitian ini yaitu menggunakan LSTM sebagai <i>base</i> awal dalam percobaan.	Perbedaan dengan penelitian ini adalah dilakukan pengujian menggunakan model Hybrid LSTM dengan Transformer, serta menggunakan model FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB <i>with</i> LSTM, dan Hybrid FEDFormer WEB <i>with</i> LSTM dalam melakukan peramalan deret waktu.
<i>Life cycle cost optimization analysis of battery storage system for residential photovoltaic</i>	Pada penelitian ini, peneliti membuat model menggunakan AIMMS software dan <i>Nonlinear Programming Approach</i> (NLP)	Perbedaan dengan penelitian ini yaitu menambah <i>constraint</i> dalam model matematika dengan

<p><i>panels (Symeonidou et al., 2021).</i></p>	<p>dalam menentukan <i>energy cost</i>, <i>initial investment</i>, <i>maintenance cost</i> dari baterai, dan profit yang akan didapatkan dari menjual kelebihan energi ke <i>main grid</i>. Referensi yang dapat diambil yaitu model matematika dalam menghitung kapasitas baterai yang optimal dengan mempertimbangkan <i>total initial cost</i>, <i>total maintenance cost</i>, <i>total replacement cost</i>, dan <i>total energy cost</i>.</p>	<p>mempertimbangkan <i>total</i> kebutuhan energi yang dibutuhkan masyarakat agar kapasitas baterai lebih optimal.</p>
<p><i>An analysis of multi objective energy scheduling in PV-BESS system under prediction uncertainty (Nair et al., 2021)</i></p>	<p>Pada penelitian ini, peneliti menggunakan <i>Model Predictive Control (MPC)</i> untuk melakukan <i>decision making</i> menggunakan data <i>forecast</i> yang dapat meminimalkan degradasi baterai dengan tetap memaksimalkan <i>self-consumption</i>. Referensi yang dapat diambil yaitu perhitungan model matematika untuk meminimalkan degradasi baterai.</p>	<p>Perbedaan dengan penelitian ini adalah penggunaan berbagai model yaitu LSTM, <i>Hybrid LSTM with Transformer</i>, FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB with LSTM, dan Hybrid FEDFormer WEB with LSTM untuk melakukan <i>forecast</i> serta penggunaan <i>Marine Predator Algorithm</i> untuk mendapatkan kapasitas optimal baterai.</p>

2.2 Dasar Teori

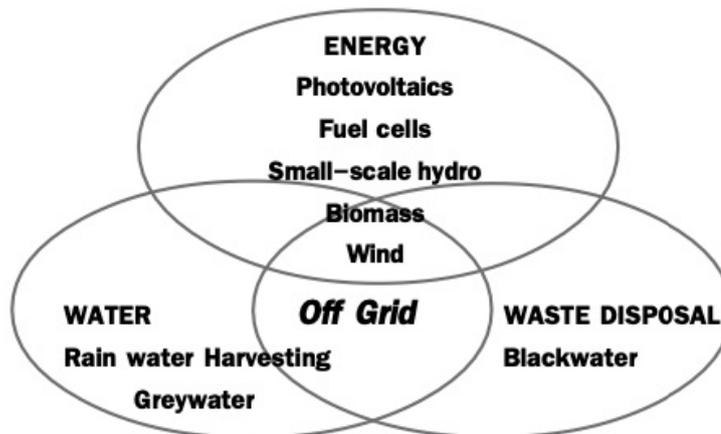
2.2.1 Battery

Baterai adalah sumber utama yang digunakan untuk menggerakkan kendaraan atau *supply energy* untuk kebutuhan listrik masyarakat. Banyaknya kebutuhan energi yang harus di-supply maka dibutuhkan kapasitas baterai yang besar. Berbagai *battery storage system* yang digunakan adalah *Lead Acid Batteries* yang telah digunakan di industri otomotif selama lebih dari sepuluh dekade. Baterai ini digunakan pada instalasi PV yang lebih kecil. Selain itu, terdapat *Nickel Based Batteries* yang biasanya digunakan untuk kondisi iklim khusus seperti musim dingin dan membutuhkan waktu *cycle time* yang lebih lama, serta membutuhkan *faster-charging*. Terdapat juga *Sodium based batteries* yang digunakan untuk penyimpanan energi stasioner karena memiliki efisiensi tinggi, umur siklus yang lebih panjang, dan kerapatan energi yang tinggi. Namun, baterai ini memerlukan perlindungan suhu dan listrik yang lebih baik, sehingga membutuhkan kabel isolasi yang lebih unggul untuk mencegah panas berlebih. Akibatnya,

biaya baterai ini lebih mahal dibandingkan dengan teknologi baterai lainnya (Mugemanyi et al., 2023; Sankaran & Venkatesan, 2022).

2.2.2 Battery Energy Storage System (BESS)

Berbagai teknologi baterai sebagai sistem penyimpanan energi merupakan elemen penting dalam *off-grid system*. BESS memiliki berbagai manfaat, salah satunya adalah menyeimbangkan *supply* energi listrik ke masyarakat dengan bantuan *photovoltaic system* (Zolfaghari et al., 2019). *Off-grid system* memiliki arti bahwa suatu bangunan mendapatkan energi listrik yang bukan berasal dari sistem utilitas publik, melainkan dari pengelolaan mandiri menggunakan *renewable energy*. Ruang lingkup *off-grid system* harus diperhatikan dari skala makro dan skala mikro. Pada skala makro, sistem ini mencakup energi, air, dan limbah. Pada Gambar 2.1 terlihat aspek pada skala makro dan mikro. Di dalamnya, terlihat tiga komponen utama dalam *off-grid system* yaitu *energy*, *water*, dan *waste disposal*. Pada bagian *energy*, teknologi yang digunakan mencakup *Photovoltaics* (panel surya), *Fuel cells*, *Small-scale hydro* (pembangkit listrik tenaga air skala kecil), *Biomass*, dan *Wind* (tenaga angin). Untuk aspek *water*, *off-grid system* memanfaatkan *Rainwater Harvesting* (penampungan air hujan) dan pengelolaan *Greywater* (air bekas pakai non-toilet). Sementara itu, pada aspek *waste disposal*, pengelolaan difokuskan pada *Blackwater* (limbah toilet). Di tengah diagram terdapat tulisan *off-grid*, yang menandakan integrasi dari ketiga sistem tersebut menjadi satu kesatuan yang mandiri dan berkelanjutan (Barrow, 2018).



Gambar 2.1 Skala makro dan mikro pada off-grid system

Pada sistem penyimpanan baterai, tentunya memiliki *cycle life* yang merupakan indikator penting yang dapat menunjukkan jumlah waktu atau jumlah siklus *charging* dan *discharging* sebelum mengalami *degradation*. Ini berarti proses pengisian baterai hingga penuh dan kemudian mengosongkannya kembali. *Charging* dalam BESS berarti melakukan pengisian daya pada baterai karena kebutuhan energi listrik di masyarakat telah terpenuhi oleh sumber energi listrik lainnya seperti panel surya. Sebaliknya, *discharging* pada BESS berarti melakukan pengosongan pada baterai akibat *high demand* dan panel surya tidak mampu untuk memenuhi kebutuhan listrik masyarakat (Bowen et al., 2019). Dalam menjaga umur baterai, dilakukan pembatasan presentase baterai dari 20% hingga 80% sehingga apabila baterai berada dalam ambang batas 20%, maka tidak akan melakukan *discharging*. Sebaliknya, apabila baterai dalam ambang batas 80%, maka tidak akan melakukan *charging* (Hernando C. S, 2023). Pada Tugas Akhir ini, *off-grid system* yang dilakukan adalah sistem yang berfokus pada energi menggunakan *photovoltaic* dan *battery energy storage system*.

2.2.3 Parameter Operasional

Parameter operasional adalah variabel-variabel yang menjadi *input* dalam model *machine learning* untuk melakukan prediksi kebutuhan listrik di masa mendatang. Parameter yang dapat menjadi *input* untuk melakukan prediksi kebutuhan listrik, antara lain, *temperature*, *humidity*, *pressure*, dan *precipitation*. Penggunaan parameter operasional sangat krusial karena kebutuhan energi listrik bersifat fluktuatif dan sangat bergantung pada kondisi eksternal tersebut. Dengan mempertimbangkan parameter ini, model prediksi dapat menangkap pola konsumsi energi yang lebih realistis dan sesuai dengan kondisi aktual (Kamoona et al., 2023).

Parameter operasional yang menjadi fokus dan target prediksi pada penelitian ini adalah *temperature*, GHI yang merepresentasikan intensitas radiasi matahari, dan *demand* listrik (KW). Ketiga parameter ini menjadi target prediksi menggunakan *machine learning* karena berperan dalam menentukan *fitness function* untuk optimasi menggunakan *Marine Predator Algorithm* (MPA) yang akan dijelaskan pada Subbab 2.2.5.

2.2.4 Timeseries Forecasting

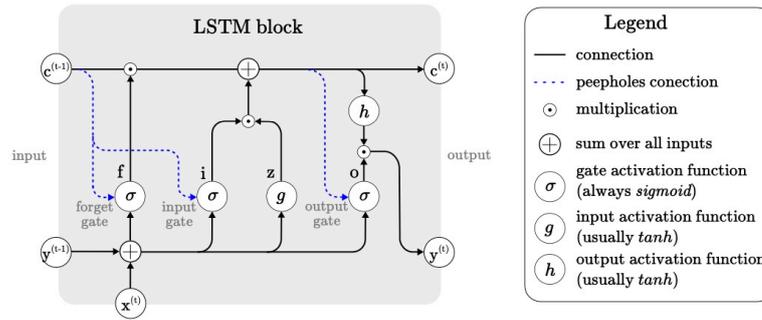
Timeseries merupakan data yang dikumpulkan pada interval waktu, seperti harga saham per hari, konsumsi listrik per jam di suatu kota, atau penjualan mingguan dari suatu toko. *Timeseries* ini penting karena dapat menganalisis dinamika dan pola yang muncul seiring waktu. Dalam mengolah data *timeseries*, penting untuk memahami berbagai aspek, seperti siklus periodik yang terjadi secara teratur, tren yang terbentuk seiring waktu, pola yang konsisten, serta lonjakan mendadak yang mungkin terjadi secara tidak terduga. *Timeseries* sering digunakan dalam *forecasting*. Hal ini berarti melakukan prediksi kejadian yang akan datang berdasarkan data historis. Contohnya adalah melakukan *forecast* penggunaan listrik beberapa jam ke depan yang dapat membantu perusahaan untuk mengantisipasi permintaan distribusi listrik secara lebih efektif (Chollet, 2021). Pada melakukan *timeseries forecasting*, terdapat dua jenis rentang waktu prediksi yaitu *one-step ahead forecasting* dan *multiple-step ahead forecasting*. Dari kedua jenis rentang waktu tersebut, *one-step ahead forecast* memiliki nilai akurasi yang lebih tinggi dan nilai *MEAN* yang lebih rendah, dibanding dengan *multi-step ahead forecasting* yang menurunkan nilai akurasi dan memiliki nilai *MEAN* yang lebih besar (Suradhaniwar et al., 2021). Pada percobaan untuk memprediksi *temperature*, terbukti bahwa *one-step ahead forecasting* memiliki hasil yang lebih akurat (Roy et al., 2024).

Pada penelitian Tugas Akhir ini, berbagai model *timeseries forecasting* yang digunakan adalah LSTM, *Hybrid LSTM with Transformer*, FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB *with LSTM*, dan Hybrid FEDFormer WEB *with LSTM* untuk menentukan waktu yang tepat bagi BESS untuk melakukan *charging* dan *discharging*. Rentang waktu *timeseries forecasting* yang digunakan pada penelitian ini adalah *one-step ahead forecasting*. Penjelasan lebih lengkap mengenai LSTM, Transformer, dan FED Former akan dibahas pada poin berikutnya.

a. Long-Short Term Memory (LSTM)

Model LSTM adalah suatu sistem *recurrent neural* yang *powerful* dan dirancang khusus untuk mengatasi masalah *exploding* atau *vanishing gradient* yang biasanya muncul ketika *learning long-term dependencies*. Sebenarnya, masalah *exploding* atau *vanishing gradient* dapat dicegah menggunakan *Constant Error Carousel* (CEC), dimana ini akan menjaga sinyal eror diantara masing-masing sel unit. Beberapa sel merupakan jaringan rekuren dengan arsitektur yang menggunakan CEC dan memiliki tambahan fitur yaitu *input gate* dan *output gate* yang bergabung menjadi *memory cell* (Van Houdt et al., 2020).

Pada Gambar 2.2, terlihat bahwa unit *vanilla* LSTM terdiri dari suatu sel, *input gate*, *output gate*, dan *forget gate*. *Forget gate* memperbolehkan jaringan untuk mengatur ulang keadaannya. Sel bertugas untuk mengingat *values* selama interval waktu yang berubah-ubah. Tiga gerbang tersebut (*input gate*, *output gate*, dan *forget gate*) bertugas untuk mengatur aliran informasi yang berhubungan dengan sel. Secara singkat, arsitektur LSTM terdiri dari sekumpulan koneksi yang terhubung secara rekuren pada sub-jaringan, yang dikenal sebagai *memory blocks*. Ide dalam *memory block* adalah untuk mempertahankan keadaannya dari waktu ke waktu dan mengatur aliran informasi secara *non-linear* gerbang linear (Van Houdt et al., 2020).

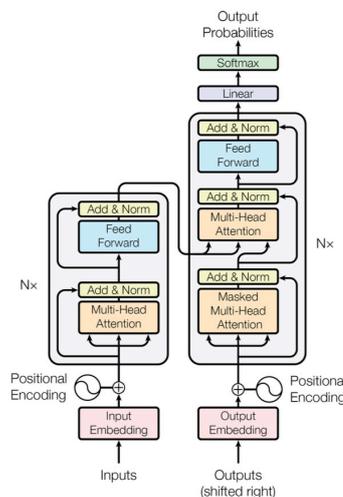


Gambar 2.2 Arsitektur vanilla LSTM

b. Transformer

Transformer merupakan arsitektur model yang menghindari *recurrence* dan mengandalkan pada *attention mechanism* untuk menarik ketergantungan *global* antara *input* dan *output*. *Transformer* memungkinkan untuk paralelisasi yang lebih besar dan dapat mencapai kecanggihan baru dalam kualitas *translation* (Vaswani et al., 2017).

Pada Gambar 2.3 terlihat model arsitektur dari *Transformer* yang terdiri dari dua komponen utama yaitu *encoder* dan *decoder*. Pada bagian *encoder*, *input* berupa token yang akan melalui proses *input embedding* dan *positional encoding*. Setiap *encoder layer* memiliki dua blok utama yaitu *multi-head attention* dan *feed-forward network* yang masing-masing diikuti oleh proses *Add & Norm*. Pada bagian *decoder*, mengalami proses yang mirip dengan *encoder* tetapi dengan tambahan *masked multi-head attention* karena *transformer* adalah model *auto-regressive*. Setelah melewati beberapa *stack layer*, *output decoder* diproses dengan *linear projection* dan *softmax* untuk menghasilkan probabilitas *output* berupa token prediksi. (Vaswani et al., 2017):



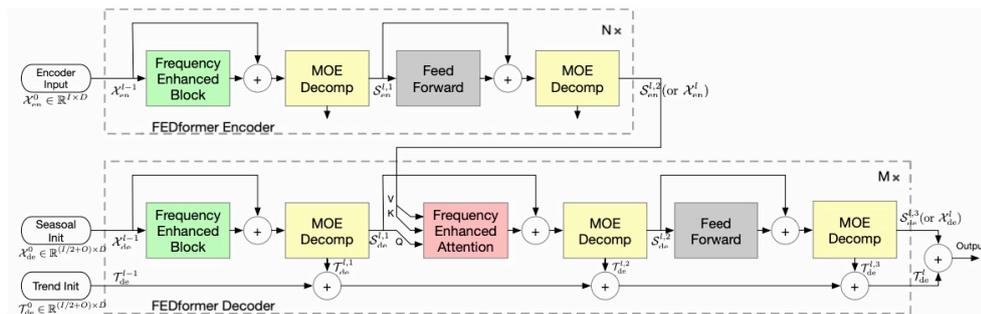
Gambar 2.3 Arsitektur model Transformer

c. FED Former

FEDFormer merupakan suatu model kombinasi dari transformer dengan *seasonal-trend decomposition method* dimana metode dekomposisi ini menangkap *global profile* dari *timeseries* dimana *transformer* itu lebih menangkap detail strukturnya. FEDFormer dikembangkan untuk mengatasi masalah *gradient vanishing* atau *exploding* yang dialami oleh metode RNN ataupun LSTM. Selain itu, juga mengurangi kompleksitas komputasi tinggi dan kebutuhan memori besar yang menjadi hambatan dalam penerapan Transformer pada *timeseries* model untuk *long term*. FEDFormer akan menggabungkan dengan *fourier analysis* pada *transformer-based method* untuk *frequency domain* karena membantu *transformer* untuk menangkap *global properties* dari *timeseries* yang lebih baik. Dengan adanya domain frekuensi juga pola *global* seperti tren dan siklus terlihat lebih jelas (Zhou et al., 2022).

Pada Gambar 2.4 terlihat struktur model dari FEDFormer yang terdiri dari bagian *encoder* dan *decoder*. Pada bagian FEDFormer *Encoder*, akan didapatkan *Encoder Input* yang berupa data *time series* lalu akan melalui *Frequency Enhanced Block* (FEB) untuk dilakukan ekstrasi fitur penting dalam domain frekuensi menggunakan *Fourier Transform*. Setelah itu, hasil ekstrasi akan dikombinasikan dengan *input* data *timeseries* lalu memasuki tahap dekomposisi menggunakan *MOE Decomp* untuk memisahkan tren dan *seasonal*. Setelah selesai proses dekomposisi, maka dilakukan proses *Feed Forward* lalu hasilnya didekomposisi lagi dalam *MOE Decomp* dan hasil akhirnya akan masuk ke dalam proses FEDFormer *Decoder* (Zhou et al., 2022).

Pada bagian FEDFormer *Decoder*, *input* berupa komponen *seasonal* dan tren yang diinisialisasi secara terpisah dan akan mengalami proses serupa dengan Encoder melalui FEB, MOEDecomp, Frequency Enhanced Attention (FEA) yang mengintegrasikan informasi dari encoder melalui mekanisme *cross attention* berbasis frekuensi, lalu dilakukan kembali proses MOE Decomp setelah itu *Feed Forward* dan melakukan *MOE Decomp* lagi. Hasil dari Decoder ini berupa komponen tren dan *seasonal* yang digabung menjadi *output* akhir. Ini menghasilkan prediksi *timeseries* melalui pemodelan interaksi antara komponen frekuensi, tren, dan *seasonal* secara efektif (Zhou et al., 2022).



Gambar 2.4 Struktur Model FEDFormer

2.2.5 Model Optimasi

Model optimasi digunakan untuk *regression*, *classification*, *clustering*, *deep learning*, dan *adversarial learning*. Model optimasi menggunakan *mathematical programming* yang memiliki kemampuan untuk mendapatkan nilai optimal dari sesuatu yang diteliti (Gambella et al., 2021). Pada penelitian Tugas Akhir ini, model optimasi yang digunakan adalah *Marine Predator Algorithm* (MPA).

Marine Predator Algorithm (MPA) adalah salah satu algoritma *metaheuristic* yang meniru tingkah laku predator laut ketika menyerang mangsanya. MPA telah terbukti unggul dalam hal kecepatan konvergensi dan akurasi karena memiliki implementasi yang fleksibel (Mugemanyi et al., 2023). MPA direkomendasikan karena merupakan salah satu algoritma terbaik untuk mencari kandidat potensial pada *real-world optimization problems* (Alorf, 2023). Terdapat dua strategi mencari makan (*foraging*) yang digunakan MPA *random walks* yaitu *brownian movements* dan *levy flight* (Mugemanyi et al., 2023).

a. Brownian movements

Dalam model stokastik, untuk panjang *step* diberikan oleh fungsi probabilitas yang menggunakan distribusi Gaussian. Fungsi probabilitas *density* untuk *Brownian Movements* terdapat pada Persamaan 2.1 (Mugemanyi et al., 2023).

$$f_B(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)} = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x^2}{2}\right)} \quad (2.1)$$

Dimana $f_B(x; \mu, \sigma)$ adalah fungsi densitas probabilitas dengan x adalah titik atau nilai yang diamati dalam model lalu memiliki parameter μ (nilai tengah) dan σ (standar deviasi). Pada $\frac{1}{\sqrt{2\pi\sigma^2}}$ adalah koefisien normalisasi yang memastikan *total* probabilitas adalah 1. Selain itu, $e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$ adalah bagian eksponensial yang menentukan bentuk distribusi Gaussian. Ketika $\sigma^2 = 1$ dan $\mu = 0$ maka rumus disederhanakan menjadi Gaussian standar yang terdapat pada Persamaan 2.2 (Mugemanyi et al., 2023)

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (2.2)$$

b. Levy flight

Rumus integral pada *Levy stable model* terdapat pada Persamaan 2.3.

$$f_L(x; \alpha, \gamma) = \frac{1}{\pi} \int_0^\infty \exp(-\gamma q^\alpha) \cos(xq) dq \quad (2.3)$$

Dimana α mengindikasikan index dari distribusi yang dimanfaatkan untuk menyesuaikan ukuran skala dari model dan γ ditugaskan untuk memilih unit skala. Biasanya, nilai α berada di rentang $0 < \alpha \leq 2$. Jika $\alpha = 2$ maka distribusi berubah menjadi distribusi gaussian. Jika $\alpha < 2$ maka distribusi memiliki tail yang berat yang berarti langkah-langkah yang sangat besar semakin sering muncul. Untuk $\exp(-\gamma q^\alpha)$ menggambarkan bagaimana peluang pergerakan berkurang seiring dengan meningkatnya jarak. Untuk $\cos(xq) dq$ berarti merepresentasikan sifat periodik dalam ruang fourier yang digunakan untuk transformasi dari domain frekuensi ke domain langkah aktual (Mugemanyi et al., 2023).

Selain itu, dalam MPA, terdapat tiga fase utama untuk optimisasi dalam MPA, yaitu: (Mugemanyi et al., 2023)

1. Fase pertama: Fase Eksplorasi

Pada fase ini, mangsa bergerak cepat dan melakukan fase eksploratori untuk mencari makanan menggunakan *brownian motion*. Predator hanya diam dan memantau pergerakan mangsa

2. Fase kedua: Fase transisi antara fase eksplorasi dan fase eksploitasi

Pada fase ini, kedua-duanya yaitu baik predator maupun mangsa bergerak pada langkah yang sama. Ini adalah fase dimana fase eksploratori itu berubah menjadi fase eksploitasi. Mangsa melakukan eksploitasi menggunakan *Levy flight*, sedangkan predator melakukan fase eksploratori menggunakan *brownian motion*.

3. Fase ketiga: Fase Eksploitasi

Pada fase ini, predator bergerak lebih cepat daripada mangsa. Predator melakukan fase eksploitasi menggunakan *Levy flight* untuk menangkap mangsa. Hasil akhir dari keseluruhan fase ini adalah berupa *chromosome* yang berisi *fitness function* terbaik.

Dalam MPA, untuk menentukan *fitness function* terbaik agar menemukan model optimasi baterai terbaik dengan *total cost* minimum pada penelitian ini, maka ditetapkan berdasarkan rumus matematika yang mengacu pada (Hernando C. S, 2023). Hasil dari *timeseries* forecasting berupa prediksi target *temperature*, GHI, dan *demand* listrik (KW) akan masuk dalam perhitungan untuk mendapatkan *fitness function* terbaik. *Off-grid system* yang akan dimodelkan memiliki rumus seperti pada Persamaan 2.4. Arti dari $P_{PV,out}$ adalah energi yang dihasilkan oleh solar PV dalam kW dan $P_{BESS}(t)$ adalah energi transmisi dari atau ke sistem baterai.

$$Load(t) = P_{PV,out}(t) - P_{BESS}(t) \quad (2.4)$$

Lalu, untuk mendapatkan *value* dari PBESS, maka dapat dihitung menggunakan rumus pada Persamaan 2.5. Jika nilai $P_{BESS}(t)$ *negative* maka baterai melakukan *discharging* karena $P_{PV,out}(t)$ tidak mampu mengisi kebutuhan energi listrik sendirian.

$$P_{BESS}(t) = P_{PV,out}(t) - Load(t) \quad (2.5)$$

Rumus dari $P_{PV,out}(t)$ terdapat pada Persamaan 2.6. Arti dari SG adalah *hourly solar generated profile* dalam kWh/kWp dan Cap^{PV} adalah kapasitas solar PV dalam kW.

$$P_{PV,out}(t) = SG(t) \times Cap^{PV} \quad (2.6)$$

Rumus dari SG terdapat pada Persamaan 2.7. Arti dari SL adalah *system loss* berdasarkan referensi adalah 14% (Hernando C. S, 2023). GHI adalah *Global Horizontal Irradiance* dalam kWh/m² yang didapatkan dari hasil *timeseries forecasting* sebelumnya. T_h^{env} adalah *environment temperature* dalam *celcius* (yang didapatkan dari hasil *timeseries forecasting* sebelumnya) dan $(1 - 0.005(T_h^{env} - 25))$ merupakan perhitungan untuk penurunan *output* dari solar panel ketika suhu lingkungan lebih dari 25°C.

$$SG = (1 - SL) \times GHI(t) \times (1 - 0.005(T_h^{env} - 25)), t \in \{1,8760\} \quad (2.7)$$

Goals dari penelitian ini adalah untuk mendapatkan kapasitas solar PV dan BESS terbaik dengan biaya minimum, maka dalam *Marine Predator Algorithm* nanti, akan dicari nilai

terbaiknya berdasarkan *fitness function*. Perhitungan *total cost* dari kapasitas solar PV dan kapasitas baterai terdapat pada Persamaan 2.8. Arti dari C_{PV} adalah *cost* dari solar PV dan C_{BESS} adalah *cost* dari Battery Energy Storage System.

$$TC(\$ NTD) = \sum_{t=1}^T (C_{PV} + C_{BESS}) \quad (2.8)$$

Rumus dari C_{PV} terdapat pada Persamaan 2.9. Arti dari $O\&M^{PV}$ adalah nilai *constant* dari biaya *operation* dan *maintenance* dari PV dan IC_{PV} adalah nilai *constant* dari *initial cost* untuk PV.

$$C_{PV}(\$ NTD) = \left(\sum_{t=1}^T SG(t) \times Cap^{PV} \times O\&M^{PV} \right) + IC_{PV} + Cap^{PV} \quad (2.9)$$

Selain itu, rumus dari C_{BESS} terdapat pada Persamaan 2.10. Biaya dari BESS berisi *Initial Cost* (IC) yang merupakan biaya instalasi pertama kali dan *Battery Degeneration Cost* (DC) dihitung *per-hour time period*.

$$C_{BESS}(\$ NTD) = Initial\ cost\ (IC) + Degeneration\ Cost\ (DC) \quad (2.10)$$

Rumus dari IC terdapat pada Persamaan 2.11. Arti dari C_{PR} adalah biaya *power rating*, $C_{O\&M}$ adalah biaya *operation* dan *maintenance*, C_{ER} adalah biaya energi *rating*, dan C_{INS} adalah biaya instalasi dan semuanya menggunakan NTD.

$$IC(\$ NTD) = (P_{BESS, mx} \times \{C_{PR} + C_{O\&M}\}) + (E_{BESS} \times \{C_{ER} + C_{INS}\}) \quad (2.11)$$

Untuk menghitung *Battery Degeneration Cost* (DC), maka digunakan Persamaan 2.12. Fungsi yang terdapat pada $C_{B,t}$ dimodelkan menggunakan actual *Sate of Charge* (SoC) pada t time.

$$DC(\$ NTD) = \sum_c C_{B,t} \quad (2.12)$$

Rumus dari fungsi tersebut terdapat pada Persamaan 2.13. Arti dari $C_E(SOC)$ adalah biaya *degeneration* setiap satu siklus baterai yang terdapat pada Persamaan 2.14.

$$C_{B,t} = \begin{cases} C_E(SOC_t) - C_E(SOC_{t-1}), & P_{discharge}(t) > 0 \\ 0, & else \end{cases} \quad (2.13)$$

$$C_E(SOC_t) = \frac{C_{cap}}{N_{cycle}} \quad (2.14)$$

N_{cycle} pada Persamaan 2.15 adalah *best-fit curve equation* yang digunakan untuk membuat model *lifetime* dari baterai karena *battery lifetime* itu bergantung pada seberapa dalam Depth of Discharge (DoD) yang digunakan.

$$N_{cycle} = \beta_0 \times DOD^{-\beta_1} \times e^{\beta_2(1-DOD)} \quad (2.15)$$

Perhitungan DoD didapatkan dari Persamaan 2.16. Untuk perhitungan Battery State of Charge (SOC_{BESS}) terdapat pada Persamaan 2.17.

$$DoD = 1 - SoC \quad (2.16)$$

$$SOC_{BESS}(t + \Delta t) = SOC_{BESS}(t) \{1 - \sigma_i \Delta t\} + \frac{P_{charge}(t) \times \Delta t \times eff}{E_{BESS}} - \frac{P_{charge}(t) \times \Delta t}{E_{BESS} \times eff} \quad (2.17)$$

Battery system (BESS) memiliki nilai *constraint* limit penggunaan untuk menjaga *maximum efficiency* per-cycle yang terdapat pada Persamaan 2.18, Persamaan 2.19, dan Persamaan 2.20.

$$0 \leq P_{charge}(t) \leq P_{BESS,max} \quad (2.18)$$

$$0 \leq P_{discharge}(t) \leq P_{BESS,max} \quad (2.19)$$

$$0,2 \leq SOC_{BESS}(t) \leq 0,8 \quad (2.20)$$

Pada SOC BESS ditetapkan *range* dari 0,2 hingga 0,8. Nilai dibawah 0,2, berarti adalah building blackout dimana kekurangan daya untuk supply energi listrik dan akan masuk ke *surge SOC list*. Nilai diatas 0,8 berarti baterai sudah hampir penuh dan berpotensi mengalami *overcharging* dan akan masuk ke *surge SOC list*. Dari berbagai perhitungan diatas, *fitness function* akan berisi *Total Cost* (\$ NTD) dan dibuat program yang akan menjadi *fitness function* dalam *Marine Predator Algorithm*. Kode semu untuk *fitness function* terdapat pada *Kode semu 5.18*.

2.2.6 Metriks Evaluasi

Pada pembuatan model *machine learning* untuk LSTM, *Hybrid LSTM with Transformer*, FEDFormer dengan Fourier Enhanced Block (FEDFormer FEB), FEDFormer dengan Wavelet Enhanced Block (FEDFormer WEB), Hybrid FEDFormer FEB *with* LSTM, dan Hybrid

FEDFormer WEB *with* LSTM, dibutuhkan suatu pengukuran untuk menghitung berapa nilai *error* antara nilai *predict* dengan nilai yang sebenarnya. Hal ini diperlukan untuk mengetahui seberapa bagus model yang dibuat. Model yang memiliki nilai *error* yang kecil merupakan model yang baik. Terdapat berbagai *evaluation metrics* yang dapat digunakan untuk mengevaluasi model yaitu *Mean Absolute Error* (MAE) dan *Mean Squared Error* (MSE). Untuk evaluasi proses dalam Marine Predator Algorithm, akan digunakan MAE. Perbedaan diantara MAE dan MSE adalah nilai pada MSE akan semakin besar untuk *large errors* (Rainio et al., 2024).

Rumus pada MAE dan MSE terdapat pada Persamaan 2.21 dan 2.22 (Rainio et al., 2024).

1. MAE:

$$\frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (2.21)$$

Dimana (n) adalah jumlah observasi, (x_i) adalah *true value* dan (y_i) adalah *predicted value*

2. MSE:

$$\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (2.22)$$

Dimana (n) adalah jumlah observasi, (x_i) adalah *true value* dan (y_i) adalah *predicted value*

2.2.7 Uji T

Uji T adalah pengujian untuk mencari perbedaan rata-rata diantara dua sampel. Prosedur pengujian perbedaan rata-rata pada uji t dilihat dari dua hipotesis. H_0 atau hipotesis pertama terjadi jika semua rata-rata dianggap sama. H_1 atau hipotesis kedua adalah rata-rata bahwa sample pertama tidak sama dengan rata-rata kedua statistik uji (Khalis Syahril Suryana et al., 2024).

Rumus *separated varians* untuk varians yang heterogen terdapat pada Persamaan 2.23. Arti dari t adalah distribusi pada sample, \bar{x} adalah rata-rata, s^2 adalah varians, dan n adalah sampel. Selain itu, untuk rumus varians homogen terdapat pada Persamaan 2.24.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (2.23)$$

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}} \quad (2.24)$$

Selain itu, pada uji T-test terdapat dua skenario pengujian yaitu *one-tail* dan *two-tail*. *One-tail* adalah uji satu arah yang digunakan untuk membuktikan bahwa rata-rata dari sample pertama

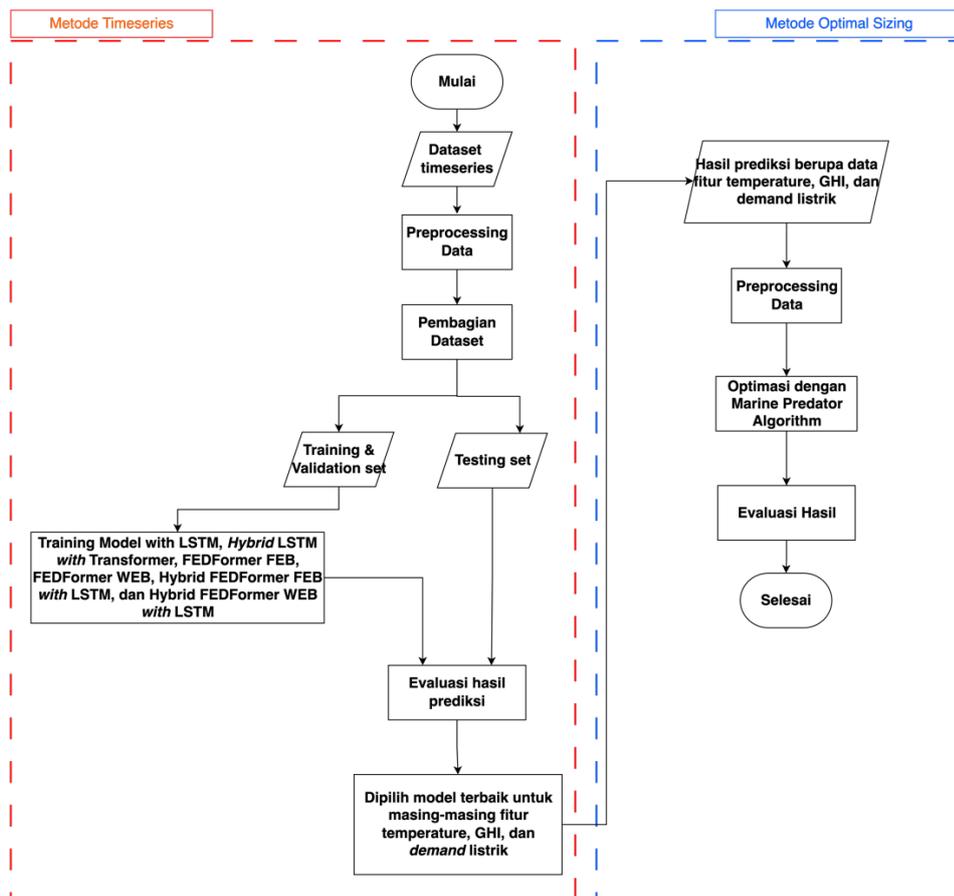
lebih besar atau lebih kecil dari rata-rata sampel kedua. Sementara itu, *Two-tail* adalah uji dua arah yang digunakan ketika ingin mengetahui apakah rata-rata kedua kelompok itu berbeda (bisa lebih besar, lebih kecil, dan tidak spesifik ke salah satu arah) (Khalis Syahril Suryana et al., 2024).

[Halaman Ini Sengaja Dikosongkan]

BAB 3 METODOLOGI

Pada penelitian ini, metode yang akan digunakan ada dua yaitu metode *timeseries* dan *metode optimal sizing* yang terlihat pada Gambar 3.1. Metode *timeseries* menggunakan enam model yaitu LSTM, *Hybrid LSTM with Transformer*, FEDFormer FEB, FEDFormer WEB, *Hybrid FEDFormer FEB with LSTM*, dan *Hybrid FEDFormer WEB with LSTM*. Untuk metode *optimal sizing* menggunakan *Marine Predator Algorithm*. Langkah pertama dimulai dari metode *timeseries* dimana didapatkan berbagai dataset untuk *weather*, *solar radiation*, dan *demand listrik (KW)*. Setelah itu, dilakukan *preprocessing data* menggunakan *standar scaler* yang akan dijelaskan lebih lanjut pada Subbab 3.3. Kemudian, dilakukan pembagian dataset menjadi *training*, *validation*, dan *testing*. Untuk data *train* dan *validation* digunakan untuk pelatihan dan validasi model *timeseries forecasting*. Untuk data *test* digunakan agar menghasilkan hasil *predict*. Selanjutnya, dilakukan evaluasi menggunakan parameter MAE dan MSE dari hasil prediksi.

Hasil dari evaluasi dipilih model terbaik untuk masing-masing fitur *temperature*, GHI, dan *demand listrik* dari keenam model yang diuji. Hasil prediksi tersebut digunakan untuk proses dalam metode *optimal sizing*. Setelah didapatkan hasil prediksi, dilakukan *preprocessing data* untuk fitur *temperature* dan GHI karena fitur-fitur tersebut digunakan dalam perhitungan untuk mendapatkan SG yang terdapat pada Persamaan 2.7. Setelah didapatkan data SG, dilakukan proses optimasi menggunakan *Marine Predator Algorithm* dengan parameter berupa SG dan *demand listrik*. Selanjutnya, dilakukan evaluasi hasil untuk mendapatkan *fitness* terbaik dengan *total cost* terendah.



Gambar 3.1 Alur Proses Metodologi

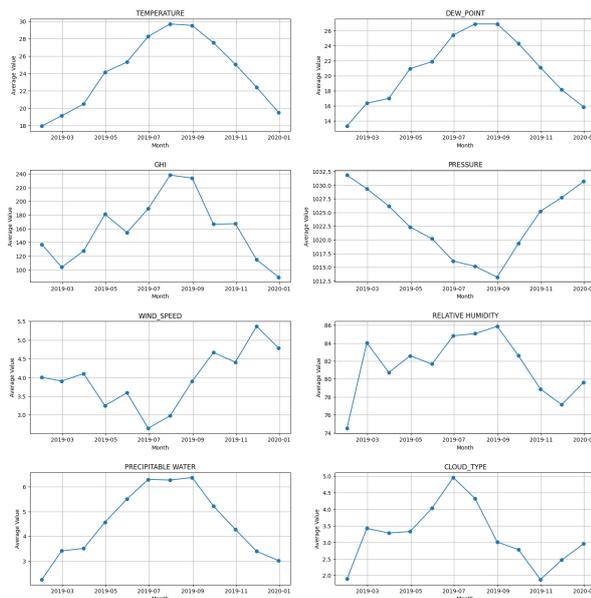
3.1 Dataset

Pada Tabel 3.1, terdapat tabel untuk dataset yang digunakan dalam penelitian ini. Dataset yang akan digunakan pada penelitian ini adalah dataset pada tahun 2019 yang memiliki jumlah baris 8232 (berisi dataset 24 jam per hari selama 1 tahun dari 23 Januari 2019 hingga 31 Desember 2019). Dataset yang digunakan untuk *timeseries* forecasting diambil dari *National Solar Radiation Database (NSRDB)* Taiwan dan data dari NTUST untuk *demand* listrik pada *educational building* di Taipei, Taiwan.

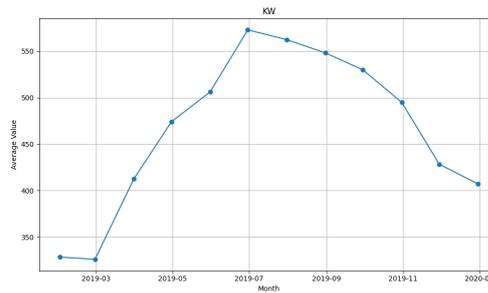
Tabel 3.1 Tabel Dataset Penelitian

Sumber Dataset	NSRDB	NTUST
Jumlah Fitur	8	1
Jumlah Baris	8232	8232
Tipe Data	Numerical	Numerical
Uraian	Dataset radiasi matahari dan cuaca di kota Taipei, Taiwan pada tahun 2019. Dataset ini direkam per jam selama satu tahun penuh. Isi fitur dalam dataset ini mencakup <i>Temperature</i> , <i>dew point</i> , <i>GHI</i> , <i>pressure</i> , <i>wind speed</i> , <i>relative humidity</i> , <i>precipitable water</i> , dan <i>cloud type</i>	Dataset kebutuhan energi listrik per jam di kota Taipei, Taiwan selama tahun 2019. Dataset ini direkam selama satu tahun penuh. Dataset ini berisi <i>demand</i> listrik (KW)

Dilakukan *Exploratory Data Analysis (EDA)* untuk mengetahui karakteristik dari dataset yang digunakan. Untuk dataset dari NSRDB terdapat pada *Gambar 3.2*. Disini terlihat bahwa masing-masing fitur menunjukkan pola yang fluktuatif, yaitu mengalami kenaikan dan penurunan nilai per hari sepanjang tahun. Hal ini menandakan bahwa tidak terdapat pola yang sepenuhnya konstan, melainkan bersifat dinamis dan variatif. Hal tersebut juga berlaku sama untuk dataset dari NTUST yang terdapat pada *Gambar 3.3*. Disini juga terlihat bahwa pola pergerakan data harian menunjukkan fluktuasi yang mencerminkan sifat tidak konstan pada masing-masing fitur.



Gambar 3.2 Grafik Karakteristik Fitur dalam Dataset NSRDB

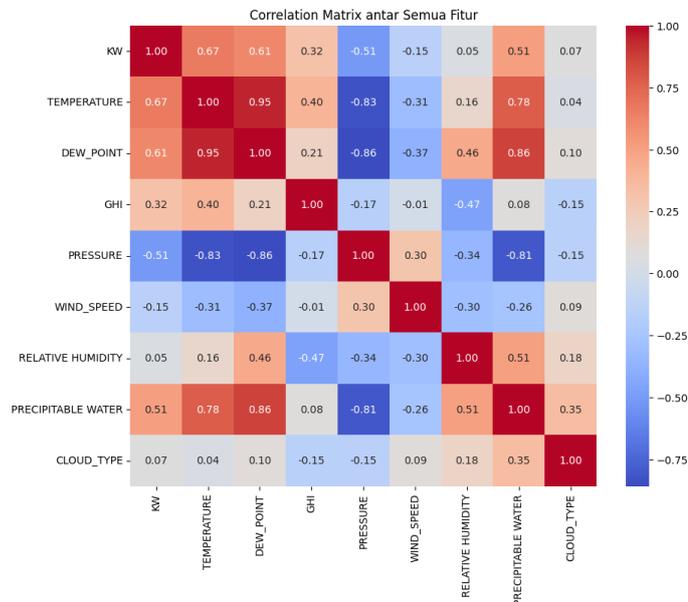


Gambar 3.3 Grafik Karakteristik Dataset NTUST

Dari kedua dataset tersebut akan digabung menjadi satu berupa dataset besar yang berisi fitur *temperature*, *dew point*, *GHI*, *pressure*, *wind speed*, *relative humidity*, *precipitable water*, *cloud type*, dan *demand listrik (KW)*. Keseluruhan fitur akan mengalami proses standarisasi dulu menggunakan *standard scaler* karena datanya sangat bervariasi. Selanjutnya, dilakukan pemisahan dataset menjadi tiga yaitu *training*, *validation*, dan *testing*. Aturan dari pembagian dataset akan dijelaskan pada Subbab 3.4.

3.2 Pemilihan fitur terbaik

Untuk pemilihan fitur terbaik dari dataset besar, dilakukan analisis korelasi menggunakan *correlation matrix*. Hasil analisis korelasi antar semua fitur terdapat pada Gambar 3.4. Pada gambar tersebut, terlihat berbagai warna. Untuk warna merah tua artinya adalah korelasi positif sangat kuat dan mendekati nilai +1. Untuk warna merah muda, korelasi positif sedang dengan nilai sekitar +0,5. Untuk warna putih berarti tidak ada korelasi dengan nilai mendekati 0. Untuk warna biru muda, berarti korelasi negatif sedang dengan nilai sekitar -0,5. Terakhir, untuk warna biru tua, berarti korelasi negatif sangat kuat dengan nilai mendekati -1.



Gambar 3.4 Hasil *Correlation Matrix* untuk Dataset

Dari keseluruhan hasil *correlation matrix* tersebut, kita fokus pada target yaitu target kolom *demand listrik (KW)*, *temperature*, dan *GHI* karena pada *time series forecasting* yang diprediksi adalah ketiga target tersebut.

1. Untuk target *demand* listrik (KW), fitur-fitur terbaik yang berkorelasi terdapat pada Tabel 3.2.

Tabel 3.2 Tabel Nilai *Correlation* Fitur pada Target *Demand* Listrik (KW)

Fitur	Nilai korelasi
<i>Temperature</i>	0,666033
<i>Dew point</i>	0,613370
<i>Precipitable water</i>	0,506498
GHI	0,320896
<i>Cloud type</i>	0,074982
Relative humidity	0,046703
<i>Wind speed</i>	-0,149321
<i>Pressure</i>	-0,512292

Dari berbagai fitur pada Tabel 3.2, dipilih fitur yang memiliki nilai korelasi absolut $\geq 0,2$ karena menurut paper (Zaki Abdullah et al., 2024), akan memberikan kompromi optimal antara akurasi prediksi dan performa sistem. Berdasarkan kriteria tersebut, fitur-fitur yang akan digunakan untuk prediksi target *demand* listrik (KW) adalah *temperature*, *dew point*, *precipitable water*, GHI, dan *pressure*.

2. Untuk target *temperature*, fitur-fitur terbaik yang berkorelasi terdapat pada Tabel 3.3.

Tabel 3.3 Tabel Nilai *Correlation* Fitur pada Target *Temperature*

Fitur	Nilai korelasi
<i>Dew point</i>	0,947692
<i>Precipitable water</i>	0,780527
<i>Demand</i> listrik(KW)	0,666033
GHI	0,403617
Relative humidity	0,156180
<i>Cloud type</i>	0,041042
<i>Wind speed</i>	-0,311811
<i>Pressure</i>	-0,832715

Dari berbagai fitur pada Tabel 3.3, juga akan dipilih nilai korelasi *absolute* $\geq 0,2$ berdasarkan paper (Zaki Abdullah et al., 2024), sehingga fitur-fitur yang akan dipilih adalah *dew point*, *precipitable water*, *demand* listrik (KW), GHI, *wind speed*, dan *pressure*.

3. Untuk target GHI, fitur-fitur terbaik yang berkorelasi terdapat pada Tabel 3.4.

Tabel 3.4 Tabel Nilai *Correlation* Fitur pada Target GHI

Fitur	Nilai korelasi
<i>Temperature</i>	0,403617
<i>Demand</i> listrik (KW)	0,320896
<i>Dew point</i>	0,208560
<i>Precipitable water</i>	0,083732
<i>Wind speed</i>	-0,008674
<i>Cloud type</i>	-0,154457
<i>Pressure</i>	-0,166859
Relative humidity	-0,471357

Dari berbagai fitur pada Tabel 3.4, juga akan dipilih nilai korelasi *absolute* $\geq 0,2$ berdasarkan paper (Zaki Abdullah et al., 2024), yaitu *temperature*, *demand* listrik (KW), *dew point*, dan relative humidity.

Alasan mengenai pemilihan target berupa *temperature*, GHI, dan *demand* listrik (KW) adalah ketiga fitur tersebut krusial di dalam metode *optimal sizing*. Ketiganya akan masuk ke dalam rumus untuk menentukan optimasi kapasitas baterai. Dengan memprediksi nilai-nilai tersebut secara akurat, sistem dapat melakukan perencanaan waktu *discharging* dan *charging* dengan lebih baik yang telah dijelaskan pada Subbab 2.2.5.

3.3 Preprocessing data

Dalam metode *timeseries* forecasting, dataset akan dilakukan standarisasi menggunakan *Standard Scaler* atau *Z-score normalization* untuk *rescale* data. *Standard scaler* ini berguna untuk mengurangi efek dari data outlier. Formulasi dari *standard scaler* terdapat pada Persamaan 3.1.

$$x'_{i,n} = \frac{x_{i,n} - \mu_i}{\sigma_i} \quad (3.1)$$

Dimana μ_i adalah *mean* dan σ_i adalah standar deviasi dari masing-masing data ke-*i*. Setelah di normalisasi, dia akan memiliki rata-rata bernilai 0 dan standar deviasi bernilai 1. Hasil rata-rata dan standar deviasi untuk *training* akan disimpan karena akan digunakan untuk normalisasi *test* dan *validation*. Seluruh nilai yang terdapat dalam *input* akan dilakukan normalisasi dengan cara yang sama. Ada proses tambahan dalam *preprocessing* data untuk fitur target. Nilai *mean* dan standar deviasi dari fitur target akan disimpan untuk melakukan *inverse transform* karena hasil *timeseries forecasting* ini membutuhkan nilai *riil* sebelum masuk ke metode *optimal sizing*.

3.4 Pembagian dataset metode *timeseries* forecasting

Akan dilakukan pembagian menjadi dua bagian yaitu *training data*, *validation data* dan *testing data*. Pada penelitian ini, data akan dibagi berdasarkan bulan karena penelitian ini akan berfokus pada musim panas di Taiwan yaitu Bulan April hingga Agustus tahun 2019. Untuk data *training*, akan diambil data dari Bulan April 2019 hingga Juni 2019. Untuk data *validation*, akan diambil data pada Bulan Juli 2019. Selain itu, untuk data *testing*, akan diambil data pada Bulan Agustus 2019. Keseluruhan data ini akan memiliki *history length* sebesar 24 dan *prediction length* sebesar 1 yang berarti melihat 24 jam ke belakang untuk memprediksi 1 jam ke depan. Pembagian dataset berdasarkan *training*, *testing*, dan *validation* ini berdasarkan penelitian sebelumnya (Hernando C. S, 2023).

Pengolahan data antara model berbasis FEDFormer dengan model berbasis LSTM memiliki cara yang berbeda karena model berbasis FEDFormer memiliki *label length*. Pengolahan data *training*, *validation*, dan *testing* pada model berbasis LSTM (LSTM dan Hybrid LSTM *with* Transformer) terdapat pada poin berikut:

a. Pengolahan data *training* basis LSTM

Data *training* dibentuk dari data pada bulan 4 hingga bulan 6 pada tahun 2019. Hal ini terlihat pada *Tabel 3.5*. Panjang baris data yang dimiliki oleh data *training* adalah 2184. Kemudian, data akan dipisah pada masing-masing *window* yang akan digunakan untuk masuk ke dalam model. Jumlah maksimal *window* yang dapat terbentuk adalah 2160 yang didapat dari (*panjang dataset* – *sequence length* – *prediction length* + 1).

Tabel 3.5 Range Date pada Data Training LSTM

Data train	
start	01/04/19 00.00
end	30/06/19 23.00
seq len	24

Data train	
pred len	1

Pada *Tabel 3.6* merupakan contoh pengambilan data berdasarkan *date*. Untuk *window 1*, data akan dimulai dari indeks 0 hingga indeks 23 yang artinya data akan diambil dari Tanggal 1 April 2019 pada pukul 00:00 hingga 1 April 2019 pukul 23:00. Hal itu juga berlanjut hingga *window* terakhir.

Tabel 3.6 Window pada Data Training LSTM

window	start index	range waktu	label (target index)	waktu label
window 1	[0, 1, 2, ..., 23]	01/04/2019 00.00 hingga 01/04/2019 23.00	[24]	02/04/19 00.00
window 2	[1, 2, 3, ..., 24]	01/04/2019 01.00 hingga 02/04/2019 00.00	[25]	02/04/19 01.00
window 3	[2, 3, 4, ..., 25]	01/04/2019 02.00 hingga 02/04/2019 01.00	[26]	02/04/19 02.00
window 4	[3, 4, 5, ..., 26]	01/04/2019 03.00 hingga 02/04/2019 02.00	[27]	02/04/19 03.00
....				
window 2160	[2159, 2160, 2161, ..., 2182]	29/06/2019 23.00 hingga 30/06/2019 22.00	[2183]	30/06/19 23.00

b. Pengolahan data *validation* basis LSTM

Data *validation* akan dibentuk dari hari terakhir pada bulan 6 hingga keseluruhan data pada bulan 7. Hal ini terlihat pada *Tabel 3.7*. Panjang baris yang dimiliki oleh data *validation* adalah 768. Selanjutnya, data akan dipisah berdasarkan masing-masing *window* yang akan digunakan untuk masuk ke proses dalam model. Jumlah maksimal *window* yang dapat terbentuk adalah 744 yang didapat dari (*panjang dataset – sequence length – prediction length + 1*).

Tabel 3.7 Range Date pada Data Validation LSTM

Data validation	
start	30/06/19 00.00
end	31/07/19 23.00
seq len	24
pred len	1

Pada *Tabel 3.8* merupakan contoh pengambilan data berdasarkan *date*. Untuk *window 1*, data akan dimulai dari indeks 0 hingga indeks 23 yang artinya data akan diambil dari Tanggal 30 Juni 2019 pukul 00:00 hingga 30 Juni 2019 pukul 23:00. Hal ini terus berlanjut hingga *window* terakhir.

Tabel 3.8 Window pada Data Validation LSTM

window	start index	range waktu	label (target index)	waktu label
window 1	[0, 1, 2, ..., 23]	30/06/2019 00.00 hingga 30/06/2019 23.00	[24]	01/07/19 00.00
window 2	[1, 2, 3, ..., 24]	30/06/2019 01.00 hingga 01/07/2019 00.00	[25]	01/07/19 01.00
window 3	[2, 3, 4, ..., 25]	30/06/2019 02.00 hingga 01/07/2019 01.00	[26]	01/07/19 02.00
window 4	[3, 4, 5, ..., 26]	30/06/2019 03.00 hingga 01/07/2019 02.00	[27]	01/07/19 03.00
....				
window 743	[742, 743, 744, ..., 766]	30/07/2019 23.00 hingga 31/07/2019 22.00	[767]	31/07/19 23.00

c. Pengolahan data *testing* basis LSTM

Data *testing* akan dibentuk dari hari terakhir bulan 7 hingga keseluruhan data pada bulan 8. Hal ini terlihat pada *Tabel 3.9*. Panjang baris yang dimiliki oleh data *testing* adalah 768. Selanjutnya, data juga akan dipisah berdasarkan masing-masing *window* yang akan digunakan untuk masuk ke proses dalam model. Jumlah maksimal *window* yang dapat terbentuk adalah 744 yang didapat dari (*panjang dataset – sequence length – prediction length + 1*).

Tabel 3.9 Range Date pada Data Testing LSTM

Data testing	
start	31/07/19 00.00
end	31/08/19 23.00
seq len	24
pred len	1

Pada *Tabel 3.10* merupakan contoh pengambilan data berdasarkan *date*. Untuk *window 1*, data akan dimulai dari indeks 0 hingga indeks 23 yang artinya adalah data akan diambil dari Tanggal 31 Juli 2019 pukul 00:00 hingga 31 Juli 2019 pukul 23:00. Hal ini akan terus berlanjut hingga *window* terakhir.

Tabel 3.10 Window pada Data Testing LSTM

window	start index	range waktu	label (target index)	waktu label
window 1	[0, 1, 2, ..., 23]	31/07/2019 00.00 hingga 31/07/2019 23.00	[24]	01/08/19 00.00
window 2	[1, 2, 3, ..., 24]	31/07/2019 01.00 hingga 01/08/2019 00.00	[25]	01/08/19 01.00
window 3	[2, 3, 4, ..., 25]	31/07/2019 02.00 hingga 01/08/2019 01.00	[26]	01/08/19 02.00
window 4	[3, 4, 5, ..., 26]	31/07/2019 03.00 hingga 01/08/2019 02.00	[27]	01/08/19 03.00
....				
window 743	[742, 743, 744, ..., 766]	30/08/2019 23.00 hingga 31/08/2019 22.00	[767]	31/08/19 23.00

Pengolahan data *training*, *validation*, dan *testing* pada model basis FEDFormer (FEDFormer FEB, FEDFormer WEB, Hybrid FEDFormer FEB with LSTM, dan Hybrid FEDFormer WEB with LSTM) terdapat pada poin berikut:

a. Pengolahan data *training* basis FEDFormer

Data *training* akan dibentuk dari data pada bulan 4 hingga bulan 6. Hal ini terlihat pada *Tabel 3.11*. Panjang baris data yang dimiliki oleh data *training* adalah 2184. Selanjutnya, data akan dipisah berdasarkan masing-masing *window* yang akan digunakan untuk masuk ke proses dalam model. Jumlah maksimal *window* yang dapat terbentuk adalah 2160 yang didapat dari (*panjang dataset – sequence length – prediction length + 1*).

Tabel 3.11 Range Date pada Data Train FEDFormer

Data train	
start	01/04/19 00.00
end	30/06/19 23.00
seq len	24
label len	24
pred len	1

Pada *Tabel 3.12* merupakan contoh pengambilan data pada *encoder* berdasarkan *date*. Untuk *window* 1, data akan dimulai dari indeks 0 hingga sebelum indeks 24 yang artinya data akan diambil dari Tanggal 1 April 2019 pada pukul 00:00 hingga 1 April 2019 pukul 23:00. Hal itu juga berlanjut hingga *window* terakhir.

Tabel 3.12 Window Encoder pada Data Train FEDFormer

window	start index enc	end index enc (start index enc + seq_len)	Range start date - end date enc
window 1 for encoder	0	24	01/04/2019 00.00 hingga 01/04/2019 23.00
window 2 for encoder	1	25	01/04/2019 01.00 hingga 02/04/2019 00.00
window 3 for encoder	2	26	01/04/2019 02.00 hingga 02/04/2019 01.00
window 4 for encoder	3	27	01/04/2019 03.00 hingga 02/04/2019 02.00
....			
window 2160 for encoder	2159	2183	29/06/2019 23.00 hingga 30/06/2019 22.00

Pada *Tabel 3.13* merupakan contoh pengambilan data berdasarkan *date* namun untuk *decoder*. Disini, pada setiap *window*, *decoder* akan mengambil data dari indeks awal hingga indeks akhir yang memiliki panjang sesuai dengan pertambahan antara *label length* dengan *prediction length*. Contoh dari *window* pertama, dimulai dari indeks 0 hingga sebelum indeks 25 dengan *range date* yaitu Tanggal 1 April 2019 pukul 00:00 hingga 2 April 2019 pukul 00:00. Hal ini akan berlanjut hingga akhir *window* yang dapat dihasilkan yaitu 2160.

Tabel 3.13 Window Decoder pada Data Train FEDFormer

window	start index dec (end index enc - label_len)	end index dec (start index dec + label_len + pred_len)	Range start date - end date
window 1 for decoder	0	25	01/04/2019 00.00 hingga 02/04/2019 00.00
window 2 for decoder	1	26	01/04/2019 01.00 hingga 02/04/2019 01.00
window 3 for decoder	2	27	01/04/2019 02.00 hingga 02/04/2019 02.00
window 4 for decoder	3	28	01/04/2019 03.00 hingga 02/04/2019 03.00
....			
window 2160 for decoder	2159	2184	29/06/2019 23.00 hingga 30/06/2019 23.00

b. Pengolahan data *validation* basis FEDFormer

Data *validation* akan dibentuk dari hari terakhir pada bulan 6 hingga keseluruhan data pada bulan 7. Hal ini terlihat pada *Tabel 3.14*. Panjang baris yang dimiliki oleh data *validation* adalah 768. Selanjutnya, data akan dipisah berdasarkan masing-masing *window* yang akan digunakan untuk masuk ke proses dalam model. Jumlah maksimal *window* yang dapat terbentuk adalah 744 yang didapat dari (*panjang dataset* – *sequence length* – *prediction length* + 1).

Tabel 3.14 Range Date pada Data Validation FEDFormer

Data val	
start	30/06/19 00.00
end	31/07/19 23.00
seq len	24
label len	24
pred len	1

Pada *Tabel 3.15* merupakan contoh pengambilan data pada *encoder* berdasarkan *date*. Untuk *window 1*, data akan dimulai dari indeks 0 hingga sebelum indeks 24 yang artinya data akan diambil dari Tanggal 30 Juni 2019 pukul 00:00 hingga 30 Juni 2019 pukul 23:00. Hal ini terus berlanjut hingga *window* terakhir.

Tabel 3.15 Window Encoder pada Data Validation FEDFormer

window	start index enc	end index enc (start index enc + seq_len)	Range start date - end date enc
window 1 for encoder	0	24	30/06/2019 00.00 hingga 30/06/2019 23.00
window 2 for encoder	1	25	30/06/2019 01.00 hingga 01/07/2019 00.00
window 3 for encoder	2	26	30/06/2019 02.00 hingga 01/07/2019 01.00
window 4 for encoder	3	27	30/06/2019 03.00 hingga 01/07/2019 02.00
....			
window 744 for encoder	743	767	30/07/2019 23.00 hingga 31/07/2019 22.00

Pada *Tabel 3.16* merupakan contoh pengambilan data pada *decoder* berdasarkan *date*. Disini, pada setiap *window*, *decoder* akan mengambil data dari indeks awal hingga indeks akhir yang memiliki panjang sesuai dengan penambahan *label length* dengan *prediction length*. Contoh dari *window* pertama, dimulai dari indeks 0 hingga sebelum indeks 25 dengan *range date* yaitu Tanggal 30 Juni 2019 pukul 00:00 hingga 1 Juli 2019 pukul 00:00. Hal ini akan berlanjut hingga *window* terakhir.

Tabel 3.16 Window Decoder pada Data Validation FEDFormer

window	start index dec (end index enc - label_len)	end index dec (start index dec + label_len + pred_len)	Range start date - end date
window 1 for decoder	0	25	30/06/2019 00.00 hingga 01/07/2019 00.00
window 2 for decoder	1	26	30/06/2019 01.00 hingga 01/07/2019 01.00
window 3 for decoder	2	27	30/06/2019 02.00 hingga 01/07/2019 02.00
window 4 for decoder	3	28	30/06/2019 03.00 hingga 01/07/2019 03.00
....			
window 744 for decoder	743	768	30/07/2019 23.00 hingga 31/07/2019 23.00

c. Pengolahan data *testing*

Data *testing* akan dibentuk dari hari terakhir bulan 7 hingga keseluruhan data pada bulan 8. Hal ini terlihat pada *Tabel 3.17*. Panjang baris yang dimiliki oleh data *testing* adalah 768. Selanjutnya, data juga akan dipisah berdasarkan masing-masing *window* yang akan digunakan untuk masuk ke proses dalam model. Jumlah maksimal *window* yang dapat terbentuk adalah 744 yang didapat dari (*panjang dataset – sequence length – prediction length + 1*).

Tabel 3.17 Range Date pada Data Testing FEDFormer

Data test	
start	31/07/19 00.00
end	31/08/19 23.00
seq len	24
label len	24
pred len	1

Pada *Tabel 3.18* merupakan contoh pengambilan data pada *encoder* berdasarkan *date*. Untuk *window 1*, data akan dimulai dari indeks 0 hingga sebelum indeks 24 yang artinya adalah data akan diambil dari Tanggal 31 Juli 2019 pukul 00:00 hingga 31 Juli 2019 pukul 23:00. Hal ini akan terus berlanjut hingga *window* terakhir.

Tabel 3.18 Window Encoder pada Data Testing FEDFormer

window	start index enc	end index enc (start index enc + seq_len)	Range start date - end date enc
window 1 for encoder	0	24	31/07/2019 00.00 hingga 31/07/2019 23.00
window 2 for encoder	1	25	31/07/2019 01.00 hingga 01/08/2019 00.00
window 3 for encoder	2	26	31/07/2019 02.00 hingga 01/08/2019 01.00
window 4 for encoder	3	27	31/07/2019 03.00 hingga 01/08/2019 02.00
....			
window 744 for encoder	743	767	30/08/2019 23.00 hingga 31/08/2019 22.00

Pada *Tabel 3.19* merupakan contoh pengambilan data pada *decoder* berdasarkan *date*. Disini, pada setiap *window*, *decoder* akan mengambil data dari indeks awal hingga indeks akhir yang memiliki panjang sesuai dengan pertambahan *label length* dengan *prediction length*. Contoh dari *window* pertama, dimulai dari indeks 0 hingga sebelum indeks 25 dengan *range date* yaitu Tanggal 31 Juli 2019 pukul 00:00 hingga 1 Agustus 2019 pukul 00:00. Hal ini akan berlanjut hingga *window* terakhir.

Tabel 3.19 Window Decoder pada Data Testing FEDFormer

window	start index dec (end index enc - label_len)	end index dec (start index dec + label_len + pred_len)	Range start date - end date
window 1 for decoder	0	25	31/07/2019 00.00 hingga 01/08/2019 00.00
window 2 for decoder	1	26	31/07/2019 01.00 hingga 01/08/2019 01.00

window	start index dec (end index enc - label_len)	end index dec (start index dec + label_len + pred_len)	Range start date - end date
window 3 for decoder	2	27	31/07/2019 02.00 hingga 01/08/2019 02.00
window 4 for decoder	3	28	31/07/2019 03.00 hingga 01/08/2019 03.00
....			
window 744 for decoder	743	768	30/08/2019 23.00 hingga 31/08/2019 23.00

3.5 Training model

Dari pembagian dataset pada Subbab 3.4, selanjutnya akan dilakukan *training* untuk metode *timeseries* forecasting menggunakan berbagai model *training* yang akan digunakan pada penelitian ini yaitu LSTM, *Hybrid LSTM with Transformer*, FEDFormer FEB, FEDFormer WEB, *Hybrid FEDFormer FEB with LSTM*, dan *Hybrid FEDFormer WEB with LSTM*. Setelah didapatkan hasil prediksi berupa nilai riil pada Bulan Agustus 2019, maka akan masuk ke dalam metode *optimal sizing* untuk melakukan perhitungan dengan *Marine Predator Algorithm*.

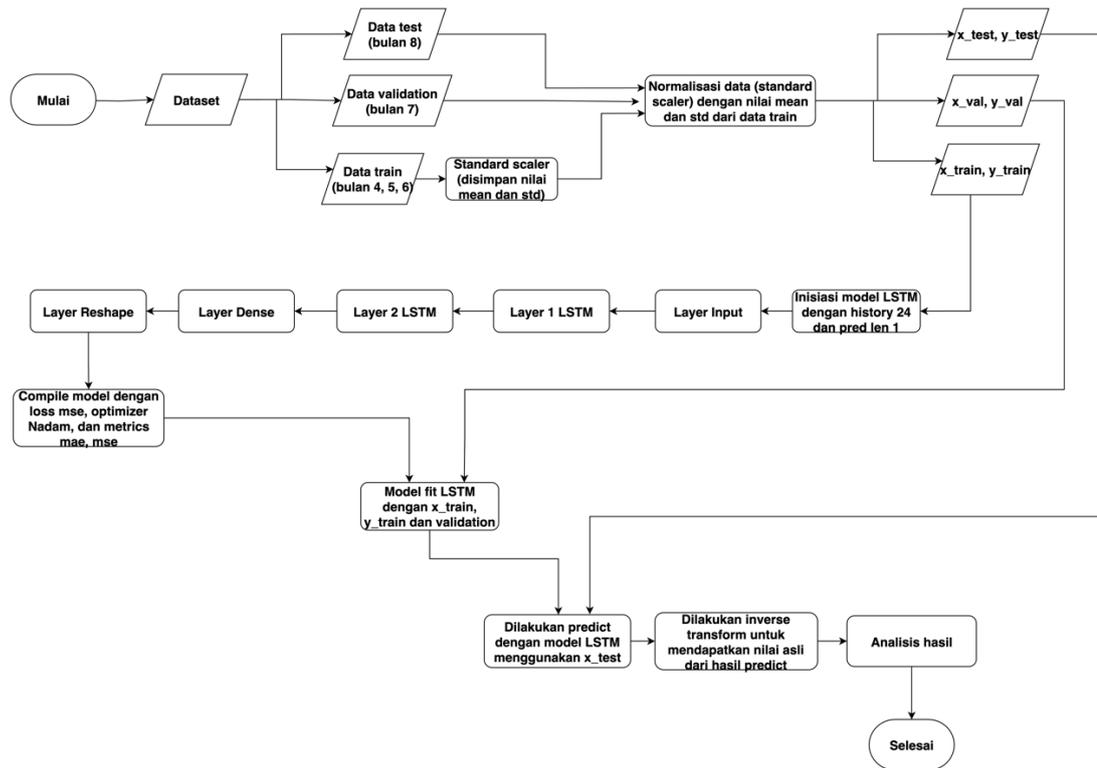
3.6 Pengembangan Model Penelitian

3.6.1 LSTM

Pada pengembangan model LSTM dari Gambar 3.5, akan dimulai dari pengumpulan dataset lalu dataset akan dipisah untuk digunakan sebagai data *train*, data *validation*, atau data *test*. Setelah itu, data juga akan diambil berdasarkan 24 jam sebelumnya untuk memprediksi 1 jam ke depan. Setelah pembagian dataset, selanjutnya untuk data *train* akan dihitung dengan *standard scaler* untuk mendapatkan nilai *mean* dan standar deviasinya. Nilai-nilai tersebut akan disimpan dan digunakan untuk melakukan normalisasi bagi keseluruhan dataset yaitu data *train*, data *validation*, dan data *test*. Hasil dari normalisasi tersebut, dapat dilakukan bangun model menggunakan LSTM untuk data *train*.

Pada LSTM, model dibangun dengan dibuat *layer input* terlebih dahulu. Setelah itu, dibuat layer pertama untuk LSTM yang akan mengembalikan *sequence* dari tiap *timestep* untuk dilakukan *training* pada layer berikutnya. Setelah selesai *layer* pertama LSTM, maka dilanjut membuat *layer* kedua LSTM. Kemudian, dilakukan proses *Dense* untuk memetakan output LSTM ke jumlah output yang diinginkan. Setelah itu dilakukan proses *Reshape* agar hasilnya memiliki dimensi yang sesuai dengan target prediksi yaitu *timestep* dan *output fitur*. Hasil dari LSTM ini akan di-*compile* dengan *loss MSE*, *optimizer Nadam* dengan *learning rate*, dan metricesnya adalah MAE dan MSE. Alasan penggunaan *Nadam* (Nesterov-accelerated Adaptive Moment Estimation) adalah adaptif terhadap gradien, cocok untuk *sequence* yang panjang, dan lebih cepat konvergen.

Setelah model berhasil dibangun, selanjutnya adalah melatih model dengan menggunakan data *train* dan data *validation*. Kemudian, akan dilakukan *predict model* yang sudah dilatih menggunakan data *testing*. Hasil dari *predict* ini masih berupa nilai yang telah dinormalisasi dengan *standard scaler*. Untuk itu, dibutuhkan *inverse transform* untuk mendapatkan nilai asli dari hasil *predict* karena nilai asli ini digunakan untuk metode *optimal sizing* nantinya.



Gambar 3.5 Flowchart LSTM

Keseluruhan Kode semu untuk model LSTM terdapat pada Kode semu 3.1.

Kode semu 3.1 Algoritma LSTM

```

Algoritma 1: LSTM
1:  DEFINE FUNCTION model_lstm( n_shape, n_feature_in, n_feature_out=1, n_output=1,
   n_hidden=512, s_activation='tanh', s_recurrent_activation='sigmoid', s_learning_rate=0.001):
2:  CREATE input layer with shape = [n_shape, n_feature_in]
3:  ADD LSTM layer WITH:
4:    units = n_hidden
5:    activation = s_activation
6:    recurrent_activation = s_recurrent_activation
7:    dropout = 0.0
8:    recurrent_dropout = 0.0
9:    return_sequences = True
10:   input = input layer
11:  ADD second LSTM layer with:
12:   units = n_hidden // 2
13:   default activations
14:   input = output of previous LSTM
15:  ADD Dense layer with:
16:   units = n_output × n_feature_out
17:   input = output of second LSTM
18:  RESHAPE output to shape = [n_output, n_feature_out]
19:  DEFINE model from input to reshaped output
20:  COMPILE model with:
21:   loss = mean squared error (MSE)
22:   optimizer = Nadam with learning rate = s_learning_rate
23:   metrics = [Mean Absolute Error (MAE), Mean Squared Error (MSE)]
24:  RETURN compiled model

```

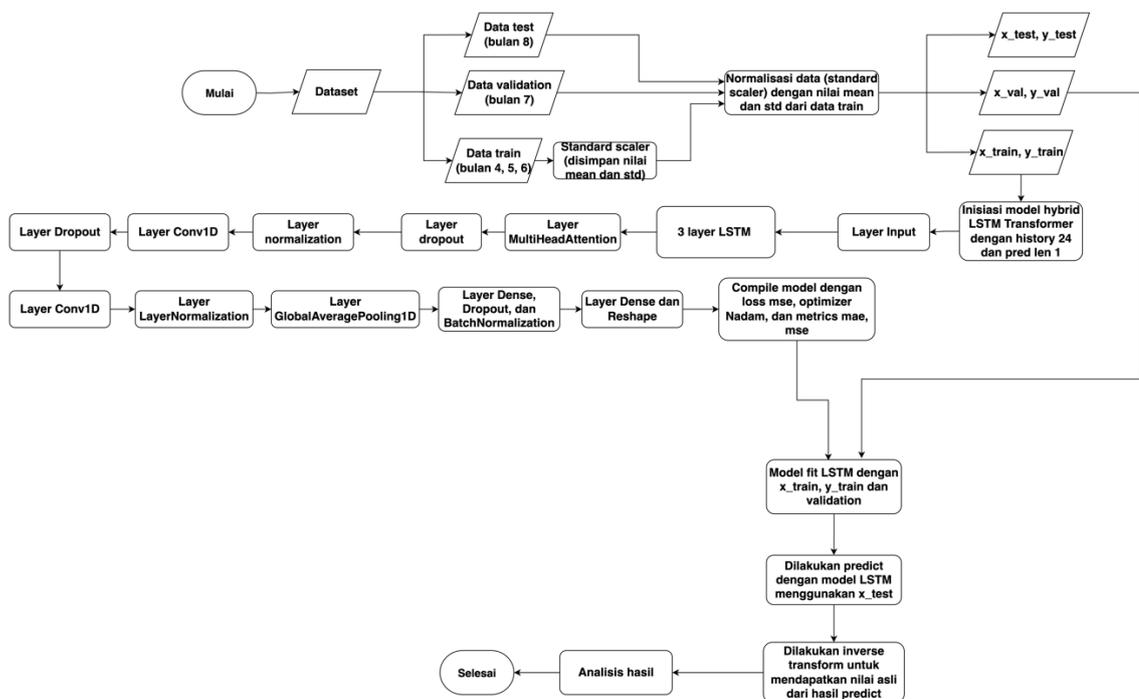
3.6.2 Hybrid LSTM with Transformer

Pada model Hybrid LSTM with Transformer, dataset juga akan dipisah menjadi *data train*, *data validation*, dan *data test*. Dari data *train*, akan dilakukan perhitungan *standard scaler* yang menyimpan nilai *mean* dan standar deviasi dari data *train*. Setelah itu, keseluruhan data akan dinormalisasi menggunakan *standard scaler* dengan *mean* dan standar deviasi dari data *train*.

Dari Gambar 3.6, pertama-tama akan dibangun model dengan inisiasi *model* Hybrid LSTM Transformer yang memiliki *history length* 24 (untuk mengambil data 24 jam sebelumnya) dan *prediction length*-nya 1 (untuk memprediksi 1 jam ke depan). Arsitektur model dimulai dari *layer input*, lalu masuk ke 3 *layer LSTM* bertingkat untuk menangkap informasi sekuensial dan dependensi jangka panjang dari data historis. Selanjutnya, hasil dari LSTM akan diproses oleh *layer MultiHeadAttention*, yang berfungsi untuk memungkinkan model fokus pada bagian-bagian penting dari urutan *input* dengan memperhatikan banyak konteks secara paralel. Hasil dari *layer MultiHeadAttention* ini akan dilanjutkan ke beberapa *layer* pengolahan yaitu *layer dropout* yang berguna untuk mencegah *overfitting*, *layer normalization*, *layer Conv1D* untuk mengekstraksi fitur lokal dari representasi sekuens, dan *layer GlobalAveragePooling1D* yang berguna untuk mereduksi dimensi *output* dan menghasilkan vektor representasi *global*.

Hasil vektor dari *layer GlobalAveragePooling1D* ini akan diproses melalui *layer dense*, *dropout*, *batch normalization*, dan *layer dense* terakhir yang akan diproyeksikan ke bentuk keluaran yang sesuai melalui *layer reshape*. Setelah itu, model akan dibangun dengan menghubungkan *layer input* dan *layer output* dari hasil *layer reshape*. Proses *compile* dilakukan dengan *loss* MSE, *optimize nadam*, dan *metrics* MAE serta MSE.

Setelah *model* berhasil dibangun, selanjutnya adalah melatih model dengan menggunakan data *train* dan data *validation*. Selanjutnya, akan dilakukan *predict model* yang sudah dilatih menggunakan data *testing*. Hasil dari *predict* ini masih berupa nilai yang telah dinormalisasi dengan *standard scaler*. Untuk itu, dibutuhkan *inverse transform* untuk mendapatkan nilai asli dari hasil *predict* karena nilai asli ini digunakan untuk *metode optimal sizing* nantinya.



Gambar 3.6 Flowchart Hybrid LSTM with Transformer

Dari arsitektur Hybrid LSTM *with* Transformer ini, memiliki *Kode semu* yang terdapat pada Kode semu 3.2.

Kode semu 3.2 Algoritma Hybrid LSTM *with* Transformer

```

Algoritma 2: Hybrid LSTM with Transformer
1:  DEFINE FUNCTION build_model(
      n_input, n_output,
      num_features_in, n_feature_out,
      head_size, num_heads, ff_dim,
      num_transformer_blocks, num_lstm_layers, lstm_units,
      mlp_units, dropout, mlp_dropout
  ):
2:  inp = Input tensor with shape [n_input, num_features_in]
3:  x = inp
4:  FOR i IN range(num_lstm_layers - 1):
5:    x = APPLY LSTM(units = lstm_units, return_sequences = True) on x
6:    x = APPLY Dropout(dropout) on x
7:  x = APPLY final LSTM layer (with return_sequences=True)
8:  FOR i IN range(num_transformer_blocks):
9:    x = CALL transformer_encoder(x, head_size, num_heads, ff_dim, dropout) # CALL Algorithm 3
10: x = APPLY GlobalAveragePooling1D to x
11: FOR dim IN mlp_units:
12:   x = APPLY Dense(units = dim) on x
13:   x = APPLY Dropout(mlp_dropout)
14:   x = APPLY BatchNormalization
15: x = Dense(units = n_output × n_feature_out)(x)
16: out = Reshape output to [n_output, n_feature_out]
17: CREATE Model with input = inp and output = out
18: COMPILE model with:
19:   loss = 'mse'
20:   optimizer = Nadam(learning_rate=0.001)
21:   metrics = [MAE, MSE]
22: RETURN model

```

Kode semu 3.3 Algoritma Transformer Encoder

```

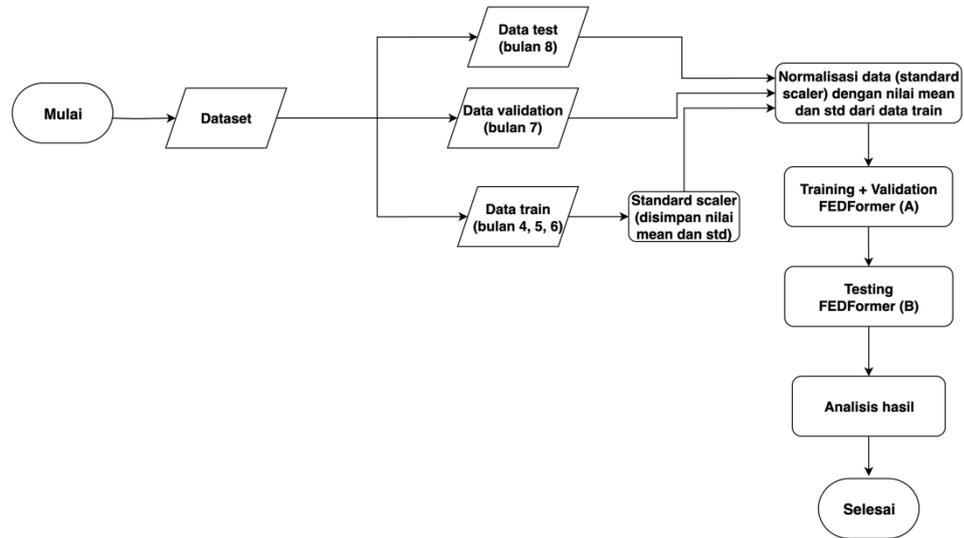
Algoritma 3: Transformer Encoder
1:  DEFINE FUNCTION transformer_encoder(inputs, head_size, num_heads, ff_dim, dropout=0):
2:    # Multi-Head Self-Attention
3:    x = APPLY MultiHeadAttention with:
4:      key_dim = head_size
5:      num_heads = num_heads
6:      dropout = dropout
7:    using inputs as both query and key
8:    x = APPLY Dropout(dropout) to x
9:    x = APPLY LayerNormalization to x
10:   res = x + inputs
11:
12:   x = APPLY Conv1D(filters=ff_dim, kernel_size=1, activation='relu') on res
13:   x = APPLY Dropout(dropout) to x
14:   x = APPLY Conv1D(filters=last_dim_of_inputs, kernel_size=1) on x
15:   x = APPLY LayerNormalization to x
16:   RETURN x + res

```

3.6.3 Hybrid FEDFormer *with* LSTM

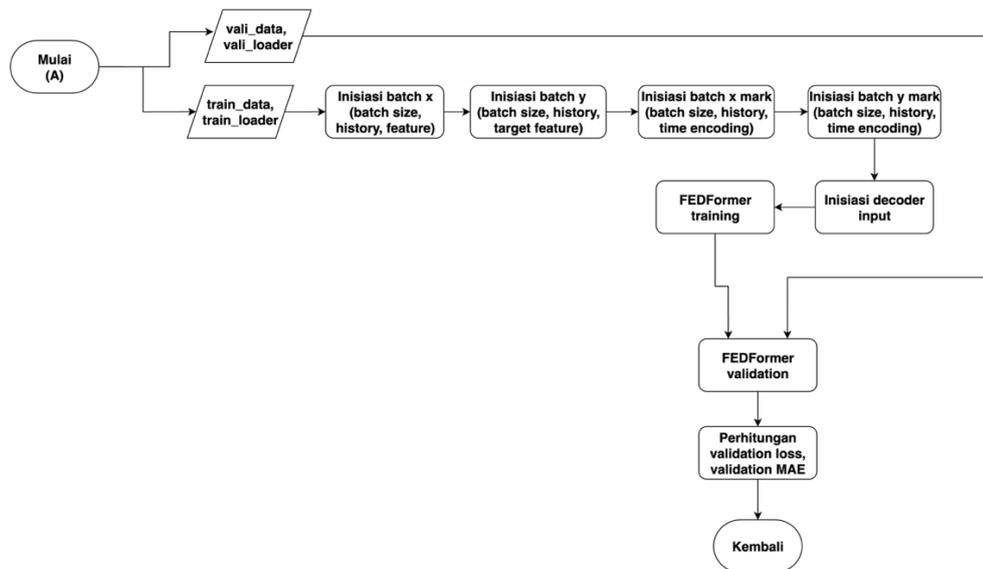
Pada model FEDFormer yang terlihat di Gambar 3.7, sama seperti sebelumnya, data akan dipisah menjadi data *train*, data *validation*, dan data *test*. Kemudian, data *train* juga akan dilakukan *standard scaler* untuk mendapatkan nilai *mean* dan standar deviasi lalu keseluruhan

data akan dinormalisasi menggunakan nilai *mean* dan standar deviasi dari data *train* tersebut. Setelah itu, akan dilakukan pemrosesan untuk *training* dan *validation* menggunakan model FEDFormer. Keseluruhan proses tersebut dilakukan untuk melatih model FEDFormer menggunakan data *train* dan divalidasi menggunakan data *validation*. Setelahnya, akan dilakukan *testing* menggunakan model yang sama dengan data *testing*. Setelah selesai, akan didapatkan hasil prediksi dan dilakukan analisis.



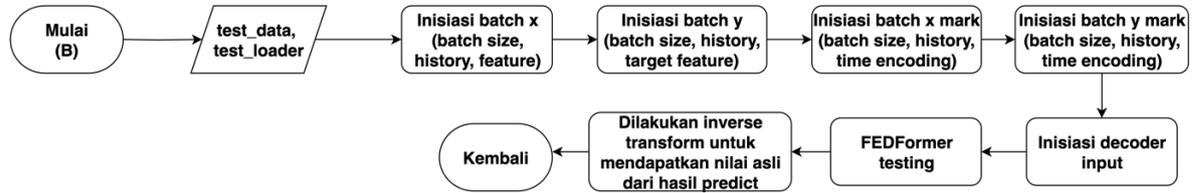
Gambar 3.7 Flowchart Awal FEDFormer

Pada proses *training* untuk FEDFormer pada Gambar 3.8, difokuskan dalam menggunakan data *train* dan data *validation*. Pertama-tama hasil dari pembagian dataset, kita akan menginisiasi *batch x* yang akan menjadi *input* untuk *encoder*. Kita juga menginisiasi *batch y* yang akan menjadi *input* untuk *decoder*. Selain itu, juga perlu menginisiasi *batch* untuk *timestep-nya* yaitu *batch x mark* dan *batch y mark*, Selanjutnya, kita inisiasi *decoder input* lalu melakukan *build model FEDFormer* untuk *training*. Setelah melakukan *training*, kita *build model FEDFormer* lagi untuk melakukan *validation*. Hasilnya, kita dapat melakukan perhitungan *validation loss* dan *validation MAE*.



Gambar 3.8 Flowchart *Training* dan *Validation* pada FEDFormer

Setelah melakukan proses *training* dan *validation*, maka proses selanjutnya yaitu *testing* yang terlihat pada Gambar 3.9. Disini kita gunakan data *testing* lalu kita inisiasi dulu untuk *batch x*, *batch x mark*, *batch y*, dan *batch y mark*. Selanjutnya, kita inisiasi *decoder input* lalu *build model* untuk FEDFormer menggunakan data *testing*. Hasilnya akan kita lakukan *inverse transform* untuk mendapatkan nilai asli dari hasil *predict* karena hasil ini digunakan untuk melakukan metode *optimal sizing*.



Gambar 3.9 Flowchart *Testing* pada FEDFormer

Dalam penelitian ini, dataset diuji dengan menggunakan 2 metode dekomposisi yang berbeda yaitu dengan Fourier dan Wavelet. Penjelasan terkait kedua metode tersebut adalah sebagai berikut:

1. Fourier

Pada gambar Gambar 3.10, keseluruhan proses pada FEDFormer untuk Fourier adalah dimulai dari *decomposition block* yang memisahkan dataset *time series* berdasarkan *trend* dan *seasonal*. Selanjutnya, dilakukan *encoder embedding* dimana input berupa *batch x* sebagai input *encoder* dan *x mark* untuk *timestep encoder*. Operasi yang dilakukan dalam proses ini terdapat pada Tabel 3.20. Proses lengkap pada *TokenEmbedding* terdapat pada Kode semu 5.4. Proses lengkap pada *TimeFeatureEmbedding* terdapat pada Kode semu 5.7. Proses lengkap pada *embedding* terdapat pada Kode semu 5.3.

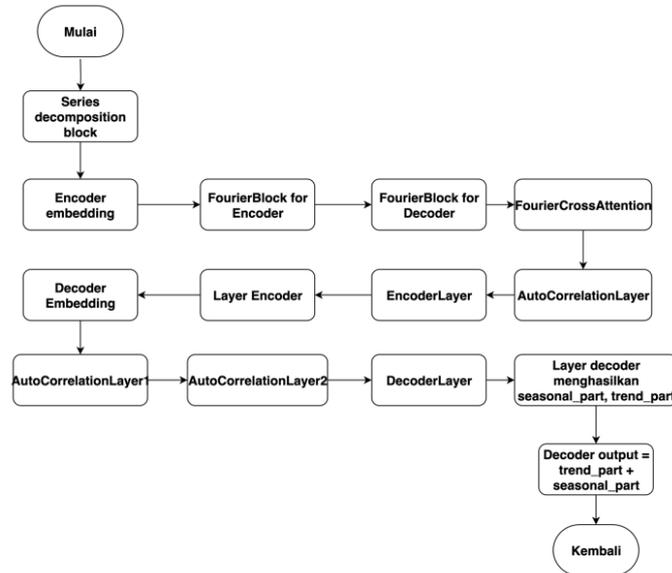
Tabel 3.20 Encoder Embedding

Operasi	Shape	Penjelasan
Input	$x = [32, 24, 9]$ $x_mark = [32, 24, 4]$	
Token Embedding	$x = [32, 24, 512]$	menggabungkan semua fitur di tiap jam menjadi vektor dimensi 512 dan hasilnya <i>embedding</i>
TimeFeature Embedding	$x_mark = [32, 24, 512]$	menjadi vektor dimensi 512
Token Embedding + TimeFeature Embedding	$[32, 24, 512]$	Menggabungkan value dengan waktu untuk tiap <i>timestep</i>

Proses selanjutnya adalah *Fourier Block* pada *encoder* dan *decoder* untuk mendapatkan *weight*. Pada *encoder*, variabel *encoder self attention* yang melakukan proses *fourier block*. Untuk *decoder*, variabel *decoder self attention* yang melakukan proses *fourier block*. Pada Tabel 3.21, untuk *encoder* memiliki *sequence length* sebesar 24. Sedangkan, untuk *decoder* hanya diambil 13. Hal tersebut disebabkan karena *decoder* hanya mengambil separuh data dari *sequence length* lalu ditambah oleh *prediction length* sehingga hasilnya adalah 13. Kemudian, jumlah frekuensi yang dihasilkan pada *encoder* adalah 13 dan pada *decoder* adalah 7. Disini, jumlah mode frekuensi yang dipilih dilakukan secara *random*. Untuk indeks frekuensi yang tidak dipilih, maka indeks tersebut akan bernilai 0. Proses terakhir adalah mengalikan nilai-nilai tersebut dengan *weight*. Proses lengkap untuk mendapatkan jumlah mode frekuensi yang dipilih terdapat pada Kode semu 5.9. Proses lengkap mengenai *FourierBlock* terdapat pada Kode semu 5.8.

Tabel 3.21 Proses Fourier Block

Komponen	Seq_len	Jumlah frekuensi yang dihasilkan (seq_len // 2 + 1)	Contoh modes yang dipilih	Indeks frekuensi diambil	Indeks frekuensi yang bernilai 0
Encoder	24	13	12	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]	[12]
Decoder	13	7	6	[0, 1, 2, 3, 4, 5]	[6]



Gambar 3.10 Flowchart FEDFormer Fourier

Selanjutnya akan dilakukan *FourierCrossAttention* yang diinisiasi oleh *decoder cross attention*. Disini akan dilakukan perkalian *dot product* antara *encoder* dengan *decoder*. *Query* berisi data *decoder* dan *key/value* berisi data *encoder*. Pada Tabel 3.22, sama seperti *FourierBlock*, dimana didapatkan jumlah mode frekuensi yang dipilih. Untuk *query* memiliki jumlah mode frekuensi 6 dan untuk *key/value* memiliki jumlah sebanyak 12. Untuk indeks-indeks yang tidak terpilih, akan memiliki nilai 0. Hasil akhir dari *FourierCrossAttention* adalah mengalikan nilai-nilai tersebut dengan *weight*. Proses lengkap mengenai *FourierCrossAttention* terdapat pada Kode semu 5.10.

Proses selanjutnya adalah pembentukan *AutoCorrelationLayer* untuk *encoder*. Disini akan mengolah data dari *encoder embedding*. Selanjutnya, data akan dipisah menjadi *query*, *key*, dan *value*. Kemudian, memasukkan *query*, *key*, *value*, dan *attention mask* ke proses *fourier block* yang diinisiasi oleh *encoder self attention*. Nilai *attention mask* ditetapkan *None* karena FEDFormer tidak terdapat posisi yang perlu disembunyikan dari proses *attention* karena semua *timestep* digunakan untuk memahami pola dan memprediksi masa depan. Pada *encoder self attention*, dijalankan proses *Fourier Block* yang menggunakan *query*, *key*, dan *value*. Hasilnya akan dikalikan dengan *weight* dan dikembalikan ke dimensi waktu. Proses lengkap mengenai *auto correlation layer* terdapat pada Kode semu 5.13.

Tabel 3.22 Proses Fourier Cross Attention

Komponen	Seq_len	Jumlah frekuensi yang dihasilkan (seq_len // 2 + 1)	Contoh modes yang dipilih	Indeks frekuensi diambil	Indeks frekuensi yang bernilai 0
Query (Decoder)	13	7	6	[0, 1, 2, 3, 4, 5]	[6]
Key/value (Encoder)	24	13	12	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]	[12]

Selanjutnya, akan dibentuk *EncoderLayer* yang memiliki *input* berupa hasil dari *encoder embedding*. Disini dilakukan proses *AutoCorrelationLayer*. Proses selanjutnya adalah mengalami *decomposition* untuk memisahkan antara *tren* dan *residual* atau *seasonal*. Hasil *residual* akan diproses melalui 2 lapis *convolution1D* dan mengalami aktivasi dengan ReLu. Lapisan *conv1D* digunakan untuk menangkap pola lokal dalam data sekuensial. Disini digunakan aktivasi ReLu mengikuti *default*. Selanjutnya, dilakukan proses *decomposition* kedua untuk mendapatkan *tren* gabungan antara *residual* hasil *decomposition* pertama dengan hasil *residual* dari *conv1D*. Hasil akhirnya mengembalikan *residual* dari *decomposition* kedua. Proses lengkap mengenai *EncoderLayer* terdapat pada Kode semu 5.14.

Setelah didapatkan *Encoder Layer*, selanjutnya akan masuk ke dalam *modul Encoder* yang akan digunakan pada *FEDFormer*. *Input* dari *encoder* berupa hasil dari *encoder embedding*. Pada penelitian ini ditetapkan bahwa jumlah *layer* untuk *encoder* adalah 2 sehingga *encoder* akan miliki 2 *encoder layer*. Hasil dari *encoder layer* akan dilakukan normalisasi. Hasil akhir dari *encoder* berupa output akhir dari proses *encoder*. Proses lengkap dalam *modul Encoder* terdapat dalam Kode semu 5.15.

Setelah membuat *encoder*, maka proses selanjutnya adalah menginisiasi *decoder embedding*. Proses dalam *decoder embedding* mirip dengan *encoder embedding*. Bedanya adalah dari *parameter input*. Untuk *decoder embedding* memiliki *input* berupa *timestep* dari *decoder* dan *seasonal* hasil dari *decomposition* dari data *encoder*. Proses pada *decoder embedding* terdapat pada Tabel 3.23. Disini juga dilakukan *TokenEmbedding* dan *TimeFeatureEmbedding* Proses lengkap pada *TokenEmbedding* terdapat pada Kode semu 5.4. Proses lengkap *TimeFeatureEmbedding* terdapat pada Kode semu 5.7. Proses lengkap pada *embedding* terdapat pada Kode semu 5.3.

Tabel 3.23 Proses Decoder Embedding

Operasi	Shape	Penjelasan
Input	seasonal_init = [32, 25, 9] x_mark = [32, 24, 4]	
Token Embedding	x = [32, 25, 512]	menggabungkan semua fitur di tiap jam menjadi vektor dimensi 512 dan hasilnya embedding
TimeFeature Embedding	x_mark = [32, 25, 512]	menjadi vektor dimensi 512
Token Embedding + TimeFeature Embedding	[32, 25, 512]	Menggabungkan value dengan waktu untuk tiap timestep

Selanjutnya, dibentuk *decoder layer* yang memiliki *input* berupa hasil *fourier block* menggunakan data *decoder*. Di *decoder layer* akan terjadi 2 kali *AutoCorrelationLayer*. *AutoCorrelationLayer* pertama menggunakan *decoder self attention* dari *fourier block*. Sementara itu, *AutoCorrelationLayer* kedua menggunakan *decoder cross attention* dari *fourier cross attention*. Pada *AutoCorrelationLayer* pertama akan menggunakan *input* dari *decoder*

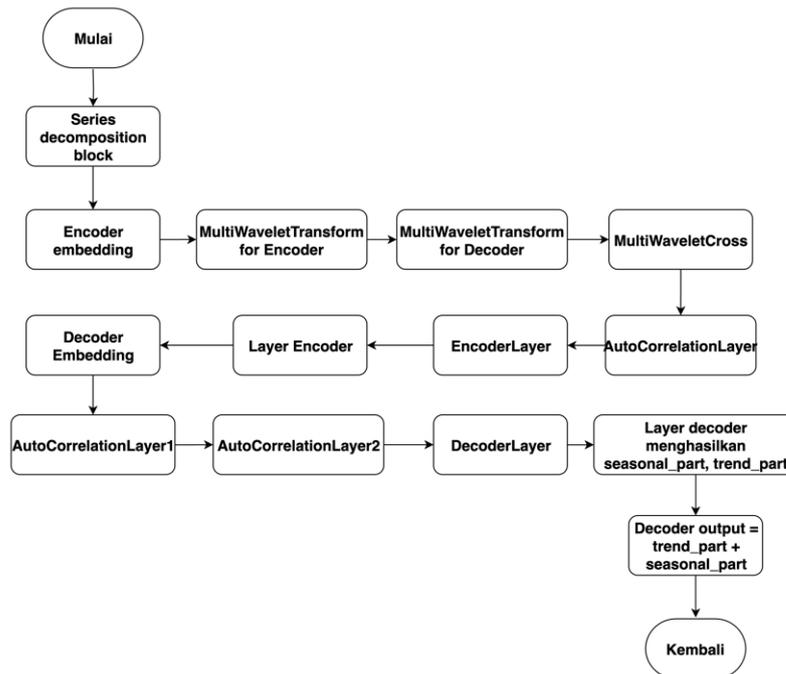
embedding (*query/key/value* = *decoder embedding*). Kemudian, dilakukan *decomposition* pertama untuk hasil *AutoCorrelationLayer* pertama. Hasilnya didapatkan *seasonal1* dan *tren1*. Pada *AutoCorrelationLayer* kedua akan menggunakan *input* dari *seasonal1* dan *encoder embedding* (*query* = *seasonal1*, *key/value* = *encoder embedding*). Hasil dari *AutoCorrelationLayer* kedua akan dilakukan *decomposition2* yang hasilnya adalah *seasonal2* dan *tren2*. Kemudian, hasil dari *seasonal2* ini diproses menggunakan *convolution1D* dan mengalami *activation ReLu*. Selanjutnya, akan dilakukan *decomposition3* yang hasilnya adalah *seasonal3* dan *tren3*. Didapatkan *residual tren* berupa pertambahan antara *tren1*, *tren2*, dan *tren3*. *Residual* akhir ini berupa *residual tren* yang diproyeksikan ke bentuk akhir *output* dengan *convolution*. Hasil dari *decoder layer* adalah *seasonal3* dan *residual* akhir. Proses lengkap dalam *decoder layer* terdapat pada Kode semu 5.16.

Pada *decoder*, dilakukan proses dalam *decoder layer* sebanyak *layer* yang ditetapkan. Pada penelitian ini, jumlah *layer decoder* yang ditetapkan adalah 1. Hasil *seasonal* dari *decoder layer* akan dilakukan normalisasi dan proyeksi. Selanjutnya, dilakukan proses LSTM dengan 1 *layer* LSTM. Kemudian, dilakukan *projection* dari hasil LSTM tersebut dan dilakukan normalisasi lagi. Hasil akhir dari *decoder* adalah *seasonal* dan *tren*. Proses lengkap mengenai *Decoder* terdapat pada Kode semu 5.17. Kembali dalam *class FEDFormer*, setelah mendapatkan *encoder* dan *decoder*, didapatkan hasil akhir berupa *decoder output* yang terdiri dari *tren part* dan *seasonal part*.

2. Wavelet

Pada wavelet, alur terkait proses *FEDFormer* sama seperti *fourier* (pada penjelasan sebelumnya) perbedaannya terletak pada *encoder_self_att*, *decoder_self_att*, dan *decoder_cross_att*. Pada *Fourier*, *encoder_self_att* dan *decoder_self_att* menggunakan modul *FourierBlock* dan *encoder_cross_att* menggunakan *FourierCrossAttention*. Pada *wavelet*, untuk *encoder_self_att* dan *decoder_self_att* menggunakan modul *MultiWaveletTransform* dan untuk *decoder_cross_att* menggunakan modul *MultiWaveletCross*. Flowchart pada wavelet terlihat pada *Gambar 3.11*.

Pada *MultiWaveletTransform*, proses di dalamnya terlihat pada *Tabel 3.24*. Disini, *input* untuk *encoder* berupa *encoder embedding*. Untuk *decoder*, memiliki *input* berupa *decoder embedding* dan *encoder embedding*. Pada *encoder*, untuk *query*, *key*, dan *value* memiliki shape yang sama. Dari awalnya 4 dimensi akan diubah *shape*-nya menjadi 3 dimensi agar dapat diproses dalam *linear transform*. *Linear transform 1* mengubah dimensi dari 512 menjadi 1024 karena model mengonversinya menjadi 128 *channel* dengan 8 *basis wavelet* per *channel*. Hasil dari *linear transform 1* ini diubah menjadi 4 dimensi lagi yang memecah sinyal menjadi 128 *channel* dan 8 *basis wavelet*. Kemudian, dilakukan *MultiWaveletTrasnformCoarsetoZoom1D* yang memecah sinyal ke banyak skala *wavelet* dan memberi *weight* pada komponen detail di tiap skala. Hasilnya berupa representasi yang menangkap komponen *seasonal* dan *tren* dalam sinyal. Setelah proses ini, data diproyeksikan kembali menjadi 3 dimensi melalui *linear transform 2*. Pada *decoder*, terdapat perbedaan dari *sequence length* pada *queries* dengan *key/value*. Hal ini karena *query* mengambil data dari *decoder embedding*. Sedangkan, untuk *key/value* mengambil data dari *encoder embedding*. Proses selanjutnya sama seperti *encoder* yang telah dijelaskan sebelumnya. Proses lengkap mengenai *MultiWaveletTransform* terdapat pada Kode semu 5.11.



Gambar 3.11 Flowchart FEDFormer Wavelet

Tabel 3.24 Proses MultiWaveletTransform

Komponen	Input shape [Batch, Seq_Len, Head, Embed_per_head]	Reshape value (H * Embed_per_head)	Linear transform 1 (jumlah channel * jumlah basis wavelet)	Reshape [Batch, Seq_Len, jumlah channel, jumlah basis wavelet]	Linear transform 2
Encoder (q, k, v)	[32, 24, 8, 64]	[32, 24, 512]	[32, 24, 1024]	[32, 24, 128, 8]	[32, 24, 512]
Decoder (q)	[32, 25, 8, 64]	[32, 24, 512]	[32, 25, 1024]	[32, 25, 128, 8]	[32, 25, 512]
Decoder (k/v)	[32, 24, 8, 64]	[32, 24, 512]	[32, 24, 1024]	[32, 24, 128, 8]	[32, 24, 512]

Pada *MultiWaveletCrossAttention*, prosesnya terlihat pada Tabel 3.25. Disini dilakukan pemrosesan terhadap 3 komponen yaitu *query*, *key*, dan *value* yang memiliki bentuk $[batch, sequence\ length, head, embed_per_head]$. Untuk *query* menggunakan data dari *decoder embedding*. Untuk *key/value* menggunakan data dari *encoder embedding*. *Sequence length* pada *decoder* adalah 13 karena *decoder* hanya mengambil separuh data dari *sequence length* awal lalu ditambah dengan *prediction length*. Kemudian, dilakukan *linear transform* yang hasilnya menjadi 3 dimensi yaitu $[batch, sequence\ length, 512]$ karena perkalian antara *head* dengan *embed per head*. Kemudian, data di-*reshape* lagi menjadi 4 dimensi dan proses ini memungkinkan model untuk mengurai sinyal ke dalam 64 *channel* dan 8 *basis wavelet* per *channel*, sehingga informasi *low frequency* dan *high frequency* dapat ditangkap secara lokal per *timestep*.

Tabel 3.25 Proses MultiWaveletCross

Komponen	Sequence length	Jumlah level wavelet	Linear transform 1 (jumlah channel * jumlah basis wavelet)	Reshape [Batch, Seq_Len, jumlah channel, jumlah basis wavelet]	Linear transform 2
Query (Decoder)	13	3	$(64 * 8) = 512$	[B, L, 64, 8]	[B, 13, 512]
Key/value (Encoder)	24	4	$(64 * 8) = 512$	[B, L, 64, 8]	[B, 13, 512]

Dalam *MultiWaveletCross*, setiap komponen diproses melalui beberapa tahapan dekomposisi *wavelet*. *Query*, *key*, dan *value* didekomposisi menjadi *high frequency* dan *low frequency* di tiap level. Setelah itu, dilakukan proses *attention* pada berbagai level. *Attention1* menangani antar komponen detail (*high frequency*). *Attention2* menangani antar komponen *smooth* (*low frequency*). *Attention3* fokus pada komponen *smooth* dan *Attention4* fokus pada *full attention* setelah semua level selesai. Kemudian, dilakukan *reshape* dan diproyeksikan kembali ke 3 dimensi pada *linear transform 2*. Hasil akhir dari *wavelet cross* ini adalah mengembalikan representasi akhir *input* yang telah diproses melalui *cross attention* berbasis *wavelet multi-level* dan nilai *attention map* (*None* karena tidak digunakan). Proses lengkap mengenai *MultiWaveletCross* terdapat pada Kode semu 5.12. Keseluruhan proses mengenai model FEDFormer dengan LSTM terdapat pada Kode semu 3.4.

Kode semu 3.4 Algoritma FEDFormer with LSTM

```

Algoritma 4: Main Code untuk FEDFormer
1:  DEFINE CLASS Model:
2:    INHERIT FROM nn.Module
3:    INITIALIZE(configs):
4:      SET model_version, model_selected, modes, seq_len, label_len, pred_len, output_attention
5:      IF configs.moving_avg is a list:
6:        INITIALIZE decomp = series_decomp_multi(kernel_size)
7:      ELSE:
8:        INITIALIZE decomp = series_decomp(kernel_size)
9:        INITIALIZE enc_embedding = DataEmbedding_wo_pos(enc_in, d_model, embed, freq, dropout)
10:       INITIALIZE dec_embedding = DataEmbedding_wo_pos(dec_in, d_model, embed, freq, dropout)
11:       IF version FROM configs IS 'Wavelet':
12:         INITIALIZE encoder_self_att = MultiWaveletTransform(d_model, L, base)
13:         INITIALIZE decoder_self_att = MultiWaveletTransform(d_model, L, base)
14:         INITIALIZE decoder_cross_att = MultiWaveletCross()
15:       ELSE:
16:         INITIALIZE encoder_self_att = FourierBlock()
17:         INITIALIZE decoder_self_att = FourierBlock()
18:         INITIALIZE decoder_cross_att = FourierCrossAttention()
19:       CALCULATE enc_modes = int(min(configs.modes, configs.seq_len // 2))
20:       CALCULATE dec_modes = int(min(configs.modes, (configs.seq_len // 2 + configs.pred_len) // 2))
21:       INITIALIZE encoder FROM Encoder module:
22:         FOR EACH EncoderLayer module:
23:           ADD EncoderLayer WITH:
24:             AutoCorrelationLayer(encoder_self_att)
25:       INITIALIZE decoder FROM Decoder module:
26:         FOR EACH DecoderLayer module:
27:           ADD DecoderLayer module WITH:
28:             AutoCorrelationLayer(decoder_self_att)
29:             AutoCorrelationLayer(decoder_cross_att)
30:         LSTM integration
31:
32:       FUNCTION forward(x_enc, x_mark_enc, x_dec, x_mark_dec, enc_self_mask=None,
33:                        dec_self_mask=None, dec_enc_mask=None):
        # Step 1: Compute mean and zeros for trend initialization

```

```

34: COMPUTE mean across time FROM x_enc
35: REPLICATE mean to match pred_len
36: INITIALIZE zeros = torch.zeros(x_dec.shape[0], pred_len, x_dec.shape[2])
37: #Step 2: Decomposition
38: GET seasonal_init AND trend_init = decomp(x_enc)
39: #Step 3: Prepare decoder input
40: CONCATENATE label_len + mean TO GET trend_init
41: PAD seasonal_init WITH zeros TO EXTEND pred_len
42: #Step 4: Encoder forward pass
43: APPLY enc_embedding ON x_enc AND x_mark_enc
44: PASS THROUGH encoder TO GET enc_out AND attns
45: #Step 5: Decoder forward pass
46: APPLY dec_embedding ON seasonal_init AND x_mark_dec
47: PASS THROUGH decoder WITH dec_out, enc_out, dec_self_mask, dec_enc_mask, trend_init TO
    GET seasonal_part AND trend_part
48: #Step 6: Final output
49: GET dec_out = trend_part + seasonal_part
50: IF output_attention is enabled:
51:     RETURN dec_out AND attns
52: ELSE:
53:     RETURN dec_out

```

3.6.4 FEDFormer

Proses dalam FEDFormer ini sama seperti subbab sebelumnya yaitu Subbab 3.6.3 terkait Hybrid FEDFormer *with* LSTM. Bedanya adalah pada proses *Decoder*, tidak terdapat LSTM. Dalam *Kode semu 3.5*, untuk membuat *Decoder module* tidak dibutuhkan integrasi LSTM. *Module decoder* ini terdiri dari beberapa *stack DecoderLayer* yang bertugas untuk memproses representasi *temporal* dari *input decoder*, memanfaatkan *cross-attention* terhadap *output encoder*, serta menggabungkan dan memproyeksikan komponen *seasonal* dan tren untuk menghasilkan *forecasting output*. Pertama-tama, modul *decoder* akan berisi berbagai *decoder layer*, *norm layer*, dan *layer projection*. Setelah itu, terjadi proses *forward* yang dimulai dari proses *stack* dalam *decoder layer*. Dalam *decoder layer*, *input* awal *x* (representasi temporal dari *decoder input*) diproses secara berurutan melalui setiap *DecoderLayer* yang ditetapkan dalam penelitian ini yaitu 1. Kemudian, akan diproses di dalam *DecoderLayer* (sama seperti subbab sebelumnya) dan *Kode semunya* terlihat dalam *Kode semu 5.16*. *Output* dari setiap *layer* berupa representasi *seasonal* yang diperbarui dan komponen *residual* tren yaitu akumulasi tren pada masing-masing *layer*. Komponen *residual* tren dari tiap *decoder layer* ditambahkan secara kumulatif ke variabel tren. Setelah itu, dilakukan proses dalam *layer* normalisasi terhadap *output seasonal* setelah melewati seluruh *stack DecoderLayer*. Kemudian, dilakukan *layer projection* dari *output seasonal* ke dimensi *target output*. Hasilnya, *decoder* mengembalikan 2 komponen yaitu representasi akhir *seasonal* komponen dan *trend*. Proses lengkap terkait proses *Decoder* tanpa LSTM ini terdapat pada *Kode semu 3.6*.

Kode semu 3.5 Algoritma FEDFormer *without* LSTM

```

Algoritma 22: Main Code untuk FEDFormer tanpa LSTM
1: DEFINE CLASS Model:
2:     INHERIT FROM nn.Module
3:     INITIALIZE(configs):
4:         SET model_version, model_selected, modes, seq_len, label_len, pred_len, output_attention
5:         IF configs.moving_avg is a list:
6:             INITIALIZE decomp = series_decomp_multi(kernel_size)
7:         ELSE:
8:             INITIALIZE decomp = series_decomp(kernel_size)
9:         INITIALIZE enc_embedding = DataEmbedding_wo_pos(enc_in, d_model, embed, freq, dropout)
10:        INITIALIZE dec_embedding = DataEmbedding_wo_pos(dec_in, d_model, embed, freq, dropout)

```

```

11: IF version FROM configs IS 'Wavelet':
12:     INITIALIZE encoder_self_att = MultiWaveletTransform(d_model, L, base)
13:     INITIALIZE decoder_self_att = MultiWaveletTransform(d_model, L, base)
14:     INITIALIZE decoder_cross_att = MultiWaveletCross()
15: ELSE:
16:     INITIALIZE encoder_self_att = FourierBlock()
17:     INITIALIZE decoder_self_att = FourierBlock()
18:     INITIALIZE decoder_cross_att = FourierCrossAttention()
19: CALCULATE enc_modes = int(min(configs.modes, configs.seq_len // 2))
20: CALCULATE dec_modes = int(min(configs.modes, (configs.seq_len // 2 + configs.pred_len) // 2))
21: INITIALIZE encoder FROM Encoder module:
22:     FOR EACH EncoderLayer module:
23:         ADD EncoderLayer WITH:
24:             AutoCorrelationLayer(encoder_self_att)
25: INITIALIZE decoder FROM Decoder module:
26:     FOR EACH DecoderLayer module:
27:         ADD DecoderLayer module WITH:
28:             AutoCorrelationLayer(decoder_self_att)
29:             AutoCorrelationLayer(decoder_cross_att)
30:
31:
32: FUNCTION forward(x_enc, x_mark_enc, x_dec, x_mark_dec, enc_self_mask=None,
33: dec_self_mask=None, dec_enc_mask=None):
34: # Step 1: Compute mean and zeros for trend initialization
35: COMPUTE mean across time FROM x_enc
36: REPLICATE mean to match pred_len
37: INITIALIZE zeros = torch.zeros(x_dec.shape[0], pred_len, x_dec.shape[2])
38: #Step 2: Decomposition
39: GET seasonal_init AND trend_init = decomp(x_enc)
40: #Step 3: Prepare decoder input
41: CONCATENATE label_len + mean TO GET trend_init
42: PAD seasonal_init WITH zeros TO EXTEND pred_len
43: #Step 4: Encoder forward pass
44: APPLY enc_embedding ON x_enc AND x_mark_enc
45: PASS THROUGH encoder TO GET enc_out AND attns
46: #Step 5: Decoder forward pass
47: APPLY dec_embedding ON seasonal_init AND x_mark_dec
48: PASS THROUGH decoder WITH dec_out, enc_out, dec_self_mask, dec_enc_mask, trend_init TO
49: GET seasonal_part AND trend_part
50: #Step 6: Final output
51: GET dec_out = trend_part + seasonal_part
52: IF output_attention is enabled:
53:     RETURN dec_out AND attns
54: ELSE:
55:     RETURN dec_out

```

Kode semu 3.6 Algoritma Decoder *without* LSTM

```

Algoritma 23: Decoder tanpa LSTM
1: DEFINE CLASS Decoder:
2:     INHERIT FROM nn.Module
3:     INITIALIZE(layers, d_model, norm_layer, projection):
4:         STORE layers as a list of DecoderLayer modules
5:         STORE optional normalization layer (norm)
6:         STORE optional output projection layer (projection)
7:
8:     FUNCTION forward(x, cross, x_mask=None, cross_mask=None, trend=None):
9:         FOR each DecoderLayer in self.layers:
10:             APPLY DecoderLayer on x with cross input and masks
11:             ADD returned residual_trend to trend
12:         IF norm is defined:
13:             APPLY norm to x
14:         IF projection is defined:
15:             APPLY projection to x

```

16:	RETURN:
17:	x: final decoder output
18:	trend: accumulated trend component

3.6.5 Marine Predator Algorithm

Hasil dari metode *timeseries forecasting* sebelumnya, akan diolah untuk mendapatkan *fitness function* terbaik yang telah dijelaskan pada Subbab 2.2.5. Pada implementasi MPA, algoritma ini diawali dengan inisiasi parameter berupa *lower bound*, *upper bound*, *dimension*, *fitness function*, *population size*, *top predator*, *top predator history*, *prey history*, *fitness history*, *top fitness*, *top fitness history*, *top fitness iterations*, *best fitness*, *mean fitness*, *ndex*, dan *best chromosome per-iteration*. Setelah itu, menetapkan populasi awal yang terdiri dari 2 variabel utama yaitu (cappv dan eboss) dan diambil secara acak dalam rentang nilai dalam penelitian ini adalah 100 hingga 10700. Hasil dari penetapan populasi akan dihitung nilai *fitness function*-nya terlebih dahulu yang terdapat pada Kode semu 5.18. Hasil dari *fitness function* ini akan dilakukan pengecekan per-*chromosome* dalam populasi apakah layak atau tidak. Apabila tidak layak, akan mengembalikan populasi baru yang hanya berisi solusi valid. Jika jumlah solusi valid kurang dari 2, maka fungsi akan mengembalikan nilai *False*, sehingga populasi akan diinisiasi ulang.

Setelah itu, dilakukan optimasi. Proses pencarian optimasi optimal dilakukan setelah *fitness* dihitung. Selanjutnya, populasi diperiksa untuk memastikan hanya solusi *valid* yang digunakan dalam proses selanjutnya. Jika seluruh *chromosome* tidak *valid*, maka populasi akan diinisiasi ulang hingga mendapatkan solusi yang *valid*. *Chromosome* dengan nilai *fitness* terbaik dipilih sebagai *top predator* dan dijadikan dasar untuk *elite matrix* yaitu representasi elit yang disalin sebanyak jumlah populasi. Seluruh riwayat *fitness*, populasi, dan solusi terbaik pada iterasi awal disimpan untuk keperluan analisis dan evaluasi. Kemudian, proses optimasi utama dijalankan dalam *loop* sebanyak jumlah iterasi maksimum yang ditentukan. Pada setiap iterasi, jika ditemukan *chromosome* baru dengan *fitness* lebih baik daripada sebelumnya, maka *chromosome* tersebut diperbarui sebagai *predator* terbaik dan *elite matrix* diubah mengikuti solusi tersebut. Proses yang terjadi dalam *marine memory saving* adalah menggabungkan populasi saat ini dengan populasi sebelumnya, lalu memilih *chromosome* terbaik untuk menjadi populasi baru. Selanjutnya, pergerakan populasi diperbarui berdasarkan fase saat ini, apakah itu *high velocity* (eksplorasi), *transition*, atau *low velocity* (eksploitasi) dengan masing-masing menggunakan metode aksi berbeda.

Pada *high velocity action*, akan diperbarui posisi *chromosome* dalam populasi selama fase *high velocity* (eksplorasi) pada *Marine Predator Algorithm*. Algoritma ini dimulai dengan membuat salinan dari *prey matrix* saat ini agar pembaruan posisi tidak memengaruhi data asli secara langsung. Selanjutnya, dihitung langkah pergerakan acak berbasis *Brownian Distribution* untuk tiap *chromosome* dalam populasi. Kemudian, dihitung *step size* dari hasil perkalian selisih antara *elite matrix* dan hasil perkalian *b motion* dengan posisi saat ini (*prey matrix*). Formula ini merepresentasikan arah dan besaran perpindahan tiap *chromosome* menuju solusi elit dengan pengaruh randomisasi *Brownian*. Vektor U dengan distribusi *uniform* dihasilkan dengan ukuran yang sama seperti *elite matrix* untuk memastikan keragaman dalam pembaruan posisi. Posisi baru tiap *chromosome* dihitung dengan menambahkan hasil perkalian P, U, *step size* ke posisi awal (*prey matrix* yang telah di clone). Parameter P mengontrol besarnya pengaruh langkah terhadap perpindahan *chromosome*. Proses lengkap dalam *HighVelocity Action* dan *Brownian Motion* terdapat pada Kode semu 3.8 dan Kode semu 3.12.

Pada *Transition action*, ini digunakan untuk memperbarui posisi *chromosome* dalam populasi selama fase transisi pada MPA. Fase transisi ini memadukan 2 pendekatan sekaligus yaitu eksplorasi dan eksploitasi, dengan membagi populasi menjadi 2 kelompok. Algoritma ini dimulai dengan menentukan jumlah *total chromosome* dalam populasi (*population_size*) dan membaginya menjadi 2 bagian sama besar (*n_half*). Kemudian, *elite matrix* dan *prey matrix* dipecah menjadi 2 bagian:

1. Bagian pertama (*pop1* dan *sliced elite*) akan digunakan untuk eksplorasi
2. Bagian kedua (*pop2*) digunakan untuk eksploitasi

Pada bagian eksplorasi, langkah acak berbasis *levy distribution* (*l_motion*) dihitung untuk seluruh *chromosome* pada *pop1*. *Step size* 1 dihitung dengan mengalikan *l_motion* dengan selisih antara *sliced elite* dan hasil perkalian *l_motion* dengan *pop1*. Hal ini menciptakan efek pergerakan acak namun tetap terarah menuju *elite solution*. Matriks acak (*U*) dengan distribusi *uniform* juga dibuat lalu posisi baru untuk *chromosome* pada *pop1* dihitung dengan rumus $pop1 + P \times U \times step_size1$. Disini, *P* adalah parameter pengontrol pengaruh langkah. Pada bagian eksploitasi, langkah acak menggunakan *Brownian Motion* (*b_motion*) dihitung untuk seluruh *chromosome* pada *pop2*. Ukuran langkah (*step_size2*) dihitung dari hasil perkalian *b_motion* dengan selisih antara hasil perkalian *b_motion* dengan *sliced elite* dan posisi saat ini *pop2*. Posisi baru untuk *chromosome* pada *pop2* diperbarui menggunakan rumus $sliced_elite + P \times CF \times step_size2$ dimana *CF* adalah *Control Factor* yang berperan sebagai pengurang agresivitas eksploitasi seiring iterasi. Terakhir, kedua bagian populasi yang sudah diperbarui (*pop1* dan *pop2*) digabungkan kembali (*concatenate*) sehingga membentuk populasi utuh yang siap digunakan untuk proses selanjutnya dalam algoritma. Proses lengkap terkait *levy flight* dan *Transition action* terdapat pada Kode semu 3.11 dan Kode semu 3.9.

Pada *Low Velocity Action*, ini akan memperbarui posisi *chromosome* dalam populasi pada fase *low velocity* (eksploitasi insentif) dalam MPA. Algoritma diawali dengan membuat salinan dari *prey matrix* dan *elite matrix* (*p_matrix_clone* dan *elite_matrix_clone*). Hal ini bertujuan untuk menjaga data asli tetap utuh selama proses pembaruan. Kemudian, dihasilkan langkah acak berbasis *Levy Flight* (*l_motion*) untuk setiap *chromosome* dalam populasi. Ukuran langkah (*step_size*) dihitung dengan mengalikan *l_motion* dengan selisih antara hasil perkalian (*l_motion*) dan posisi *elite* (*elite_matrix_clone*) dengan posisi populasi saat ini (*prey_matrix_clone*). Perhitungan ini memungkinkan *chromosome* tetap bergerak menuju solusi elit dengan pola acak yang lebih halus. Posisi baru tiap *chromosome* dalam populasi dihitung dengan rumus $elite_matrix_clone + P \times CF \times step_size$ dimana *P* adalah parameter pengontrol besarnya pengaruh langkah dan *CF* adalah *Control Factor* yang nilainya biasanya menurun seiring berjalannya iterasi untuk mengurangi tingkat eksplorasi. Hasil dari *Low Velocity Action* akan mengembalikan posisi baru populasi yang telah diperbarui yang akan digunakan dalam proses seleksi pada iterasi berikutnya. Proses lengkap terkait *low velocity action* terdapat pada Kode semu 3.10.

Setelah pergerakan utama (*high velocity action*, *transition action*, dan *low velocity action*), diterapkan mekanisme *Fish Aggregating Device* (*FAD*) untuk meningkatkan keragaman solusi. Seluruh informasi *fitness* dan perpindahan fase disimpan untuk analisis. Akhirnya, algoritma mengembalikan solusi terbaik (*top predator*), nilai *fitness* terbaik, serta riwayat *chromosome* terbaik pada tiap iterasi. Proses lengkap dari *MarinePredatorAlgorithm* terdapat pada Kode semu 3.7.

Kode semu 3.7 Algoritma MarinePredator Algorithm

Algoritma 25: MarinePredatorsAlgorithm
--

```

1: DEFINE CLASS MarinePredatorsAlgorithm:
2:   INITIALIZE(device=None):
3:     IF device IS None:
4:       SET device = 'cuda' IF available ELSE 'cpu'
5:     INITIALIZE:
6:       lower_bound, upper_bound, dimension, fitness_function, repair_function, population_size,
       top_predator, top_predator_history, prey_history, fitness_history, top_fitness, top_fitness_history,
       top_fitness_itations, best_fitness, mean_fitness, idx_idx, best_chromosome_per_iter, iter = 0, p
       = 0.5, FAD = 0.2, CF = 0

7:
8:   FUNCTION __initialize_population():
9:     CREATE empty population list
10:    FOR i IN range(population_size):
11:      GENERATE random cappv in [100, 20000]
12:      GENERATE random ebess in [100, 20000]
13:      APPEND [cappv, ebess] to population
14:    RETURN population as torch tensor on self.device
15:
16:   FUNCTION __calculate_fitness(population_tensor):
17:     CONVERT each individual to dict format:
18:       {cappv_desimal': int, 'ebess_desimal': int}
19:     RETURN fitness_function(population_list)
20:
21:   FUNCTION __repair_func(population):
22:     RETURN repaired population using repair_function and reshape
23:
24:   FUNCTION check(population):
25:     INITIALIZE count_keep = 0, new_population = []
26:     FOR chromosome IN population:
27:       IF chromosome['erase'] == False:
28:         APPEND to new_population
29:         count_keep += 1
30:     IF count_keep <= 1:
31:       RETURN False
32:     ELSE:
33:       RETURN new_population
34:
35:   FUNCTION __visualize():
36:     CREATE plot with:
37:       Line plot: top_fitness_history
38:       Line plot: mean_fitness
39:       Line plot: best_fitness
40:       Marker: phase transitions at idx_idx
41:     LABEL axes, title, legend, grid
42:     SHOW plot
43:
44:   FUNCTION optimizing(
       fitness_function, repair_function,
       dimension, lower_bound, upper_bound,
       population_size, max_iteration,
       early_stop=False, tolerance=1e-10,
       n_no_improvement=10, probability_reflective=0.5
   ):
45:     SET class parameters: fitness_function, repair_function, dimension, bounds, population_size
46:     # INITIALIZATION
47:     prey_matrix ← INITIALIZE population
48:     IF repair_function EXISTS:
49:       APPLY repair_function to prey_matrix
50:     fitness ← CALCULATE fitness of prey_matrix
51:     checked_population ← FILTER invalid individuals via check(fitness)
52:     WHILE checked_population IS False:
53:       REGENERATE prey_matrix
54:       APPLY repair_function (if any)
55:       RECALCULATE fitness
56:       checked_population ← check(fitness)
57:     fitness_values ← EXTRACT fitness values

```

```

58:  best_idx ← INDEX of best individual
59:  top_predator ← corresponding chromosome
60:  elite_matrix ← REPLICATE top_predator for whole population
61:  # INITIAL LOGGING
62:  RECORD fitness stats and histories
63:  SET iter = 0, p = 0.5
64:  stage_no_improvement_counter = 0
65:  best_seen_fitness = top_fitness
66:  current_phase = 1
67:  # MAIN LOOP
68:  WHILE iter < max_iteration:
69:    IF early_stop ENABLED:
70:      IF improvement > tolerance:
71:        RESET stage_no_improvement_counter
72:        UPDATE best_seen_fitness
73:      ELSE:
74:        INCREMENT stage_no_improvement_counter
75:      IF early_stopping triggered OR phase boundary reached:
76:        SWITCH to next phase (1→2→3) and LOG idx_idx
77:      ELSE:
78:        SWITCH phase based on iteration progress
79:        prey_matrix ← APPLY reflective_boundaries()
80:        IF repair_function EXISTS:
81:          APPLY to prey_matrix
82:        fitness ← CALCULATE fitness
83:        checked_population ← check(fitness)
84:        WHILE checked_population IS False:
85:          REGENERATE prey_matrix and repeat fitness + check
86:        # UPDATE TOP PREDATOR
87:        fitness_values ← GET current fitness values
88:        best_idx ← INDEX of current best
89:        IF new_fitness < top_fitness:
90:          UPDATE top_predator and top_fitness
91:          UPDATE elite_matrix from new top_predator
92:        LOG fitness history and progress
93:        # MARINE MEMORY SAVING
94:        COMBINE previous + current fitness and prey
95:        SORT and SELECT best N individuals as new prey_matrix
96:        # MOTION BASED ON PHASE
97:        IF elite_matrix and prey_matrix size MISMATCH:
98:          TRUNCATE to smallest size
99:        IF current_phase == 1:
100:         APPLY high_velocity_action()
101:        ELSE IF current_phase == 2:
101:         COMPUTE CF decay
102:         APPLY transition_action()
103:        ELSE IF current_phase == 3:
104:         COMPUTE CF decay
105:         APPLY low_velocity_action()
106:        INCREMENT iteration
107:        # FISH AGGREGATING DEVICE (FAD)
108:        COMPUTE CF decay
109:        APPLY fish_aggregating_device()
110:        CALL visualize()
111:        RETURN top_predator, top_fitness, best chromosome per iter

```

Kode semu 3.8 Algoritma High Velocity Action

Algoritma 26: High Velocity Action

```

1:  FUNCTION high_velocity_action(preymatrix, elitematrix, P, device):
2:    preymatrix_clone = copy preymatrix
3:    b_motion = brownian_motion(preymatrix)
4:    Step_size = b_motion x (elitematrix - b_motion x preymatrix)
5:    U = RANDOM UNIFORM matrix with same shape as elitematrix

```

```
6: RETURN prey_matrix_clone + P x U x step_size
```

Kode semu 3.9 Algoritma Transition Action

Algoritma 27: Transition Action

```
1: FUNCTION transition_action(preymatrix, elitematrix, P, CF, device):
2:   population_size = NUMBER OF ROWS in preymatrix
3:   n_half = INTEGER(0.5 x population_size)
4:   sliced_elite = elitematrix[0 : n_half, :]
5:   pop1 = preymatrix[0 : n_half, :] #untuk exploration
6:   pop2 = preymatrix[n_half:, :] #untuk exploitation
7:   #exploration part
8:   l_motion = levy_flight(pop1)
9:   step_size1 = l_motion x (sliced_elite - l_motion x pop1)
10:  U = RANDOM_UNIFORM_matrix with same shape as pop1
11:  pop1 = pop1 + P x U x step_size1
12:  #exploitation part
13:  b_motion = brownian_motion(pop2)
14:  step_size2 = b_motion x (b_motion x sliced_elite - pop2)
15:  pop2 = sliced_elite + P x CF x step_size2
16:  RETURN CONCATENATE(pop1, pop2) along row axis
```

Kode semu 3.10 Algoritma Low Velocity Action

Algoritma 28: Low Velocity Action

```
1: FUNCTION low_velocity_action(preymatrix, elitematrix, P, CF, device):
2:   preymatrix_clone = COPY preymatrix
3:   elitematrix_clone = COPY elitematrix
4:   l_motion = levy_flight(preymatrix_clone)
5:   step_size = l_motion x (l_motion x elitematrix_clone - preymatrix_clone)
6:   RETURN elitematrix_clone + P x CF x step_size
```

Kode semu 3.11 Algoritma Levy Flight

Algoritma 29: Levy Flight

```
1: FUNCTION levy_flight(current_position):
2:   sigma_x = 0.69657 #standar deviasi untuk Gaussian X
3:   sigma_y = 1.0 #standar deviasi untuk Gaussian Y
4:   gaussian_x = RANDOM_NORMAL(mean=0, std=sigma_x, shape=current_position.shape)
5:   gaussian_y = RANDOM_NORMAL(mean=0, std=sigma_y, shape=current_position.shape)
6:   step_size = 0.05 x gaussian_x / (ABS(gaussian_y) ^ (1 / 1.5))
7:   RETURN step_size
```

Kode semu 3.12 Algoritma Brownian Motion

Algoritma 30: Brownian Motion

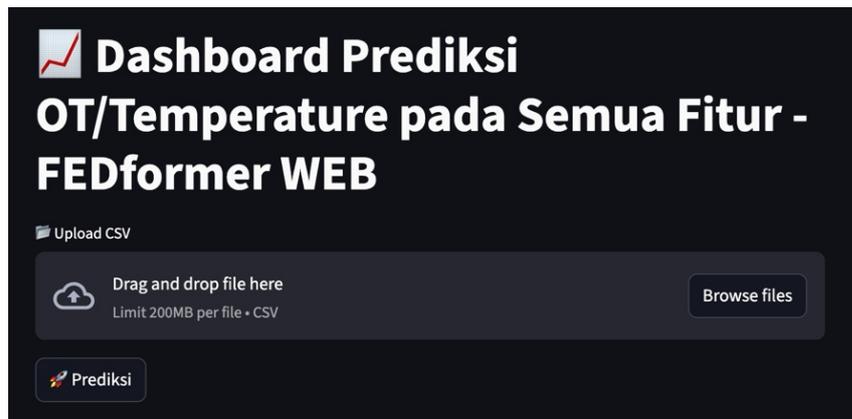
```
1: FUNCTION brownian_motion(current_position):
2:   step_size = RANDOM_NORMAL(mean=0, std=1.0, shape=current_position.shape)
3:   RETURN step_size
```

3.7 Pengembangan Antarmuka Pengguna

Implementasi pembuatan antar muka pengguna untuk target prediksi *temperature*, GHI, dan *demand* listrik dibuat menggunakan *library streamlit* menggunakan bahasa *python*. Selain itu, model yang digunakan untuk memprediksi masing-masing target dipilih berdasarkan salah satu model terbaik dari keenam model yang diuji yaitu LSTM, *hybrid LSTM with Transformer*,

FEDFormer FEB, FEDFormer WEB, *hybrid* FEDFormer FEB *with* LSTM, dan *hybrid* FEDFormer WEB *with* LSTM.

Pada *Gambar 3.12*, terdapat tampilan antarmuka untuk prediksi target *temperature*. Di dalam tampilan ini, pengguna akan memasukkan *file* yang memiliki format *csv* yang merupakan dataset *timeseries* selama tahun 2019 dan memiliki kolom berupa fitur-fitur yang menjadi acuan. Untuk target *temperature* menggunakan dataset dengan semua fitur yang artinya adalah isi *file csv* akan berupa data dari fitur *temperature*, *dew point*, GHI, *pressure*, *wind speed*, *relative humidity*, *precipitable water*, *cloud type*, dan *demand* listrik. Untuk fitur *temperature* akan diubah nama menjadi OT karena dia menjadi target prediksi. Setelah pengguna memasukkan *file*, maka pengguna dapat menekan tombol prediksi untuk mendapatkan hasil prediksi pada bulan Agustus 2019.



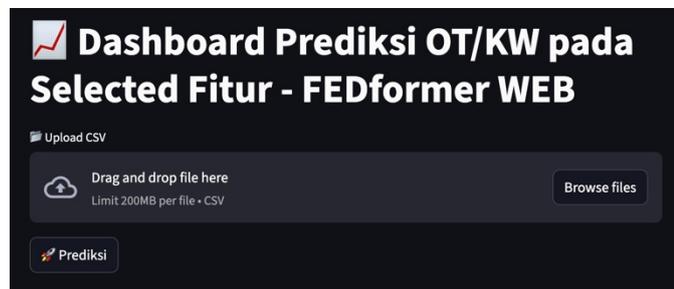
Gambar 3.12 Rancangan Antarmuka Prediksi Temperature

Pada *Gambar 3.13*, terdapat juga tampilan antarmuka untuk prediksi target GHI. Dalam tampilan ini, pengguna juga memasukkan *file* yang memiliki *format csv* yang merupakan dataset *timeseries* selama tahun 2019 dan memiliki kolom berupa fitur-fitur yang menjadi acuan. Untuk target GHI, akan menggunakan fitur-fitur yang diseleksi berdasarkan *correlation matrix*. Fitur-fitur yang digunakan adalah *temperature*, *dew point*, *relative humidity*, GHI, dan *demand* listrik. Untuk fitur GHI akan diubah juga namanya menjadi OT karena dia merupakan target yang akan diprediksi. Setelah mengunggah *file*, pengguna dapat menekan tombol prediksi untuk melakukan prediksi pada bulan Agustus 2019.

Pada *Gambar 3.14*, terdapat juga tampilan antarmuka untuk prediksi *demand* listrik. Disini, pengguna juga memasukkan *file* berformat *csv* yang merupakan data *timeseries* selama tahun 2019. Untuk fitur *demand* listrik, akan menggunakan fitur-fitur yang diseleksi berdasarkan *correlation matrix*. Fitur-fitur yang digunakan adalah *temperature*, *dew point*, *precipitable water*, GHI, *pressure*, dan *demand* listrik (KW). Untuk fitur *demand* listrik (KW) akan diubah namanya menjadi OT karena fitur tersebut yang akan menjadi target prediksi. Setelah mengunggah *file*, pengguna dapat menekan tombol prediksi untuk melakukan prediksi pada bulan Agustus 2019.



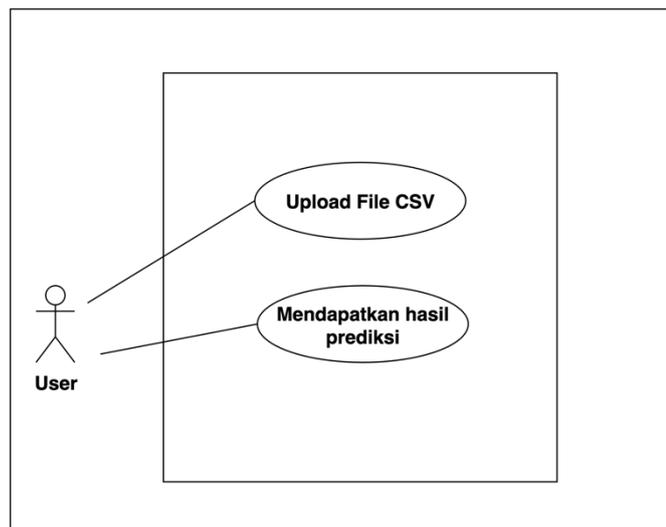
Gambar 3.13 Rancangan Antarmuka Prediksi GHI



Gambar 3.14 Rancangan Antarmuka Prediksi Demand Listrik

Pada Gambar 3.15 terlihat *usecase diagram* untuk antarmuka pengguna. Hanya terdapat satu aktor yaitu *user* yang berinteraksi dengan antarmuka pengguna. *User* dapat melakukan dua aktivitas yaitu *upload* file CSV dan mendapatkan hasil prediksi. File CSV yang harus diunggah *user* berbeda tergantung target prediksi yang diinginkan. Apabila *user* ingin melakukan prediksi untuk target *temperature*, maka *user* mengunggah file CSV yang berisi dataset menggunakan semua fitur yaitu *temperature* (yang akan diubah nama fiturnya menjadi OT karena sebagai target prediksi), *dew point*, *GHI*, *pressure*, *wind speed*, *relative humidity*, *precipitable water*, *cloud type*, dan *demand* listrik (KW). Selain itu, apabila *user* ingin melakukan prediksi untuk target GHI, maka *user* mengunggah file CSV yang berisi dataset menggunakan *selected* fitur yaitu *temperature*, *dew point*, *relative humidity*, GHI (yang akan diubah nama fiturnya menjadi OT karena sebagai target prediksi), dan *demand* listrik (KW). Terakhir, apabila *user* ingin melakukan prediksi untuk target *demand* listrik (KW), maka *user* mengunggah file CSV yang berisi dataset menggunakan *selected* fitur yaitu *temperature*, *dew point*, *precipitable water*, *GHI*, *pressure*, dan *demand* listrik (KW) (yang akan diubah nama menjadi OT karena sebagai target prediksi).

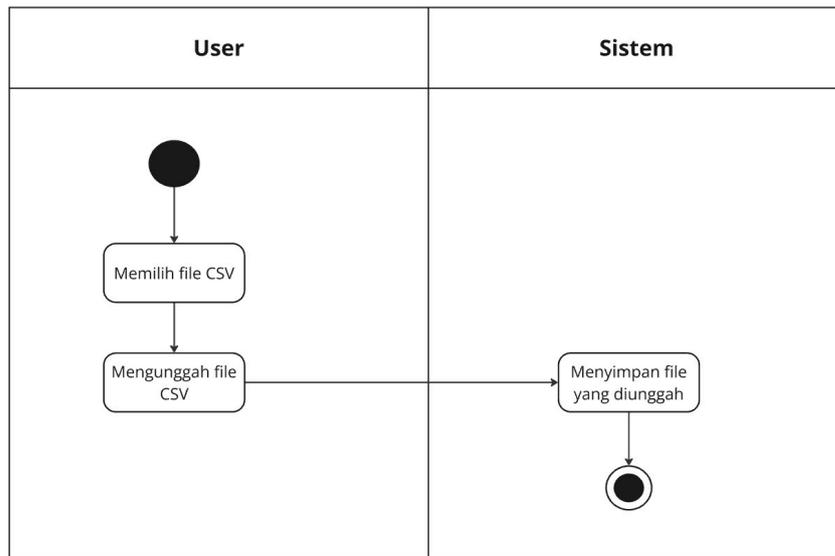
Setelah mengunggah *file* CSV, *user* mendapatkan hasil prediksi yang dilakukan oleh sistem. Disini berupa hasil prediksi selama Bulan Agustus 2019. Hasil prediksi tergantung dari target prediksi yang diinginkan oleh *user*. Apabila *user* menginginkan target prediksi berupa *temperature*, maka hasil prediksi merupakan data *temperature* selama Bulan Agustus 2019. Apabila *user* menginginkan target prediksi berupa GHI, maka hasil prediksi merupakan data GHI selama Bulan Agustus 2019. Terakhir, apabila *user* menginginkan target prediksi berupa *demand* listrik (KW), maka hasil prediksi merupakan data *demand* listrik (KW) selama Bulan Agustus 2019.



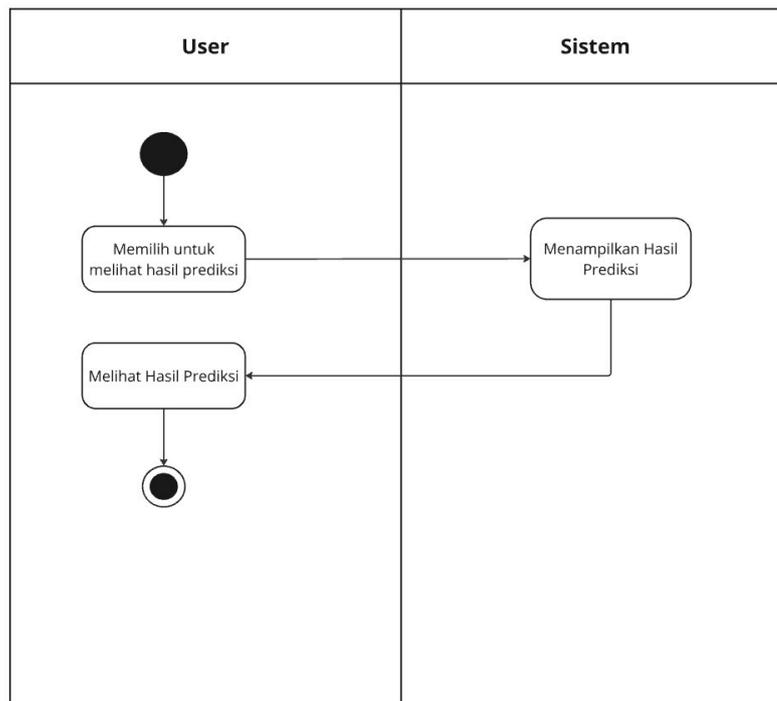
Gambar 3.15 Usecase Diagram Antarmuka Pengguna

Pada *Gambar 3.16*, terdapat *activity diagram* untuk *usecase upload file*. Disini, pertama-tama *user* akan memilih *file csv* sesuai target prediksi yang diinginkan. Apabila *user* ingin melakukan prediksi untuk target *temperature*, maka *user* mengunggah file CSV yang berisi dataset menggunakan semua fitur yaitu *temperature* (yang akan diubah nama fiturnya menjadi OT karena sebagai target prediksi), *dew point*, *GHI*, *pressure*, *wind speed*, *relative humidity*, *precipitable water*, *cloud type*, dan *demand listrik (KW)*. Selain itu, apabila *user* ingin melakukan prediksi untuk target *GHI*, maka *user* mengunggah file CSV yang berisi dataset menggunakan *selected* fitur yaitu *temperature*, *dew point*, *relative humidity*, *GHI* (yang akan diubah nama fiturnya menjadi OT karena sebagai target prediksi), dan *demand listrik (KW)*. Terakhir, apabila *user* ingin melakukan prediksi untuk target *demand listrik (KW)*, maka *user* mengunggah file CSV yang berisi dataset menggunakan *selected* fitur yaitu *temperature*, *dew point*, *precipitable water*, *GHI*, *pressure*, dan *demand listrik (KW)* (yang akan diubah nama menjadi OT karena sebagai target prediksi). Setelah memilih *file* yang diinginkan, selanjutnya *user* akan mengunggah *file CSV* tersebut. Kemudian, sistem akan menyimpan *file* yang diunggah.

Pada *Gambar 3.17*, terdapat *activity diagram* untuk *usecase* mendapatkan hasil prediksi. Disini, *user* memilih untuk melihat hasil prediksi. Kemudian, sistem akan menampilkan hasil prediksi yang diinginkan oleh *user* tergantung dari target prediksi. Apabila *user* menginginkan target prediksi berupa *temperature*, maka hasil prediksi merupakan data *temperature* selama Bulan Agustus 2019. Apabila *user* menginginkan target prediksi berupa *GHI*, maka hasil prediksi merupakan data *GHI* selama Bulan Agustus 2019. Terakhir, apabila *user* menginginkan target prediksi berupa *demand listrik (KW)*, maka hasil prediksi merupakan data *demand listrik (KW)* selama Bulan Agustus 2019. Setelah ditampilkan hasilnya oleh sistem, maka *user* dapat melihat hasil prediksi. Kode semu lengkap mengenai antarmuka pengguna dalam penelitian ini terdapat pada *Kode semu 3.13*.



Gambar 3.16 Activity Diagram Upload File



Gambar 3.17 Activity Diagram Mendapatkan Hasil Prediksi

Kode semu 3.13 Antarmuka Pengguna

```

Antarmuka Pengguna
1:  DEFINE FUNCTION load_model(path):
2:    LOAD TorchScript model from path
3:    SET model → eval mode
4:    RETURN model
5:  DEFINE FUNCTION time_features(datetime_index):
  
```

```

6:     FOR EACH timestamp IN datetime_index:
7:         COMPUTE month/12
8:         COMPUTE day/31
9:         COMPUTE dayofweek/7
10:        COMPUTE hour/24
11:    END FOR
12:    RETURN time feature array
13:  DEFINE FUNCTION build_tensors(win_df, target_ts):
14:    EXTRACT feature columns from win_df
15:    NORMALIZE features using X_MEAN and X_STD → z
16:    SET x_enc = z
17:    CREATE dummy array with zeros of shape (1, number of features)
18:    CONCATENATE x_enc and dummy → x_dec
19:    EXTRACT time features from win_df.index → x_mark_enc
20:    EXTRACT time features from target_ts → x_mark_dec
21:    CONCATENATE x_mark_enc and target_ts time features → x_mark_dec
22:    RETURN x_enc, x_mark_enc, x_dec, x_mark_dec as tensors
23:  LOAD scaler metadata from scaler_meta_semua_temp.npz #Depends on model that you want
24:  EXTRACT X_MEAN, X_STD, OT_MEAN, OT_STD, TRAIN_COLS
25:  ASSERT that TRAIN_COLS equals FEATURE_COLS
26:  DEFINE FUNCTION predict_one(model, win_df, target_ts):
27:    BUILD tensors → xe, xme, xd, xmd
28:    CALL model(xe, xme, xd, xmd) → z
29:    IF output z has more than one element:
30:      SELECT the last element of z
31:    DENORMALIZE z using OT_MEAN and OT_STD
32:    RETURN prediction as float
33:  START Streamlit app
34:  DISPLAY dashboard title
35:  UPLOAD CSV file from user
36:  IF file is not uploaded:
37:    SHOW warning and STOP
38:  READ CSV into dataframe
39:  CONVERT “date” and “time” columns into datetime index
40:  DROP “date” and “time” columns
41:  SORT dataframe by datetime index
42:  CHECK for missing columns
43:  IF any FEATURE_COLS are missing in the CSV:
44:    SHOW error and STOP
45:  LOAD model using load_model()
46:  DEFINE prediction period:
47:    SET start = August 1st, 00:00
48:    SET end = August 31st, 23:00
49:  INITIALIZE list for predictions and timestamps:
50:  FOR each hour t from start to end:
51:    SELECT window of data from t-24h to t-1h
52:    IF window has SEQ_LEN samples:
53:      CALL predict_one(model, window, t) → prediction
54:      APPEND prediction to preds
55:      APPEND t to times
56:      INCREMENT t by 1 hour
57:  END FOR
58:  IF predictions exist:
59:    CREATE dataframe of results with times and preds
60:    SHOW success message with number of predictions
61:    DISPLAY result table
62:    DISPLAY line chart
63:  ELSE:
64:    SHOW error: historical data before August 1st is incomplete

```

3.8 Evaluasi dan Perbandingan

Metode penelitian yang dikembangkan untuk *timeseries forecasting* pada penelitian ini akan dilakukan evaluasi menggunakan metrik MAE dan MSE berdasarkan hasil berupa data

yang telah dinormalisasi menggunakan *Standar Scaler* maupun yang telah dikembalikan ke dimensi semula dengan menggunakan *inverse transform*. Tujuan dari perbandingan ke data asli (yang didapat dari *inverse transform*) adalah untuk mendapatkan nilai MAE dan MSE terendah dari data asli setelah prediksi sebelum masuk ke *Marine Predator Algorithm* menggunakan data asli tersebut. Pada penelitian ini, akan dilakukan evaluasi terhadap enam model yaitu LSTM, Hybrid LSTM *with* Transformer, FEDFormer FEB, FEDFormer WEB, Hybrid FEDFormer FEB *with* LSTM, dan Hybrid FEDFormer WEB *with* LSTM. Evaluasi akan dilakukan pada dua skenario percobaan yaitu menggunakan semua fitur atau menggunakan *selected fitur* berdasarkan hasil dari *correlation matrix*. Hasil dari evaluasi ini akan mendapatkan untuk setiap target prediksi, model mana yang memiliki nilai evaluasi terbaik.

3.8.1 Perangkat Keras

1. 12th Gen Intel® Core™ i9-12900K (24CPUs), ~3.2GHz
2. Random Access Memory (RAM) 64GB
3. GeForce® RTX 3080 Ti *with* 8GB
4. Solid State Drive (SSD) 1TB
5. Apple m3 pro, RAM 18GB, dan SSD 1TB

3.9 Spesifikasi Environment

Untuk mendukung uji coba performa model penelitian, digunakan beberapa peralatan yang dapat dilihat sebagai berikut:

3.9.1 Perangkat Lunak

1. Sistem Operasi Windows 11
2. Sistem Operasi macOS Sequoia version 15.1
3. Visual Studio Code
4. Python beserta library pendukung penelitian

BAB 4 HASIL DAN PEMBAHASAN

4.1 Parameter Penelitian

Pada penelitian ini, digunakan berbagai *parameter* untuk masing-masing metodenya yaitu:

1. Metode *timeseries forecasting*

Pada metode *timeseries forecasting*, digunakan *sequence length* sebesar 24 dan *prediction length* sebesar 1 yang artinya melakukan *one-step ahead forecasting* dengan melihat 24 jam ke belakang untuk prediksi 1 jam ke depan. Target yang akan diprediksi adalah *temperature*, GHI, dan *demand* listrik (KW). Selain menetapkan *sequence length* dan *prediction length*, *label len* juga ditetapkan sebesar 24 (khusus untuk model berbasis FEDFormer). Hal ini disebabkan karena dalam *metode timeseries forecasting* yang dilakukan pada penelitian ini, digunakan model *encoder-decoder* pada masing-masing model yang akan diuji, sehingga dibutuhkan *length* pada *decoder* untuk membantu *decoder* belajar bentuk dari *sequence length*nya. Untuk konfigurasi *input* dan *output* pada *enc_in* dan *dec_in*, disesuaikan dengan percobaan yang dilakukan. Saat diuji menggunakan semua fitur yang ada dalam dataset, maka *enc_in* dan *dec_in* adalah 9 (fitur *demand* listrik (KW), *dew point*, GHI, *pressure*, *wind speed*, *relative humidity*, *precipitable water*, *temperature*, dan *cloud type*). Saat diuji menggunakan fitur yang difilter menggunakan *correlation matrix*, maka *enc_in* dan *dec_in* untuk *target temperature* adalah 7 (*demand* listrik (KW), *dew point*, *precipitable water*, GHI, *pressure*, *wind speed*, dan *temperature*), *enc_in* dan *dec_in* untuk *target GHI* adalah 5 (*demand* listrik (KW), *temperature*, *dew point*, *relative humidity*, dan GHI), serta *enc_in* dan *dec_in* untuk *target demand* listrik (KW) adalah 6 (*temperature*, *dew point*, *precipitable water*, GHI, *pressure*, *demand* listrik (KW)).

Struktur *layer* dari masing-masing model yang digunakan dalam penelitian ini sebagian besar mengikuti konfigurasi *default* dari *source code* resmi. Penelitian ini tidak berfokus pada eksplorasi arsitektur dan *hyperparameter tuning*, melainkan fokusnya adalah eksplorasi untuk mencari model terbaik antara model-model yang telah lama seperti (LSTM, hybrid LSTM with Transformer) dengan FEDFormer dan Hybrid FEDFormer with LSTM. Selain itu, pengembangan dari penelitian ini adalah menggabungkan LSTM dengan FEDFormer pada bagian *Decoder*. Hal ini bertujuan untuk membantu menjaga kontinuitas pola *sequence* yang dihasilkan, mengingat FEDFormer unggul dalam menangkap *global properties*, sementara LSTM memperkuat kemampuan model dalam mempertahankan *local temporal patterns* pada *output sequence*. Dalam pengujian pada FEDFormer akan dilakukan pengujian untuk masing-masing blok transformer yaitu Fourier Enhanced Blocks (FEB) dan Wavelet Enhanced Blocks (WEB).

Pelatihan model akan dilakukan dengan 10 *epoch* dan menggunakan strategi *early stopping* dengan *patience* 3 yang berarti proses pelatihan akan dihentikan lebih awal apabila tidak terjadi perbaikan performa pada data validasi selama 3 *epoch* berturut-turut. Strategi ini diterapkan untuk mencegah *overfitting* dan memastikan bahwa model yang dihasilkan memiliki generalisasi yang baik terhadap data yang belum pernah dilihat. Parameter lainnya mengikuti konfigurasi *default* dari *source code* aslinya, seperti untuk *d_model*, *d_ff*, *kernel_size*, dan lain sebagainya

2. Metode *optimal sizing*

Pada metode *optimal sizing*, dalam melakukan perhitungan untuk mendapatkan nilai *fitness function* terbaik, ditetapkan berbagai parameter yang sesuai dengan penelitian (Hernando C. S, 2023) yang terdapat dalam Tabel 4.1.

Tabel 4.1 Parameter Optimal Sizing

Variable	Value
<i>System loss</i>	0,13
<i>Beta 0</i>	3832
<i>Beta 1</i>	0,68
<i>BESS CH EFF</i>	1,64
<i>BESS DCH EFF</i>	0,95
<i>COST CAP BESS</i>	0,9
<i>ALPHA I</i>	50000 NTD per kW
<i>EFF</i>	0,03
<i>POWER RATING</i>	0,95
<i>ENERGY RATING</i>	320 NTD per kW
<i>BESS INSTALLATION COST</i>	360 NTD per kW
<i>BESS OPERATION MAINTENANCE</i>	15
<i>CCAP</i>	50000
<i>PV INSTALLATION COST</i>	1200
<i>PV OPERATION MAINTENANCE</i>	0,04

Dimana *system loss* berarti rugi dalam sistem PV dan konversinya yang terdapat pada Persamaan 2.7, lalu *beta_0*, *beta_1*, dan *beta_2* dipanggil dalam Persamaan 2.15, *BESS_CH EFF* dan *BESS_DCH EFF* sebagai nilai efisiensi untuk daya *charge* dan *discharge* pada BESS, *COST_CAP_BESS* merupakan harga BESS per kW, *ALPHA_I* dan *EFF* merupakan nilai konstanta untuk Persamaan 2.17, *POWER_RATING*, *BESS_OPERATION_MAINTENANCE*, *ENERGY_RATING*, dan *BESS_INSTALLATION_COST* digunakan pada Persamaan 2.11, *CCAP* digunakan untuk mendapatkan biaya *degeneration* setiap satu siklus baterai pada Persamaan 2.14, serta *PV_INSTALLATION_COST* dan *PV_OPERATION_MAINTENANCE* digunakan untuk rumus untuk mendapatkan *cost* dari solar PV pada Persamaan 2.9.

Untuk isi dari *chromosome* dalam populasi dalam *Marine Predator Algorithm*, diberikan angka dengan *range* 100 hingga 10.700. Selain itu, untuk inisiasi MPA, *lower bound* yang digunakan adalah 100, *upper bound* 10.700, *population size* 100, dan *max iteration* yang dilakukan dalam MPA adalah 100. Struktur dari *Marine Predator Algorithm* adalah *default* dari *source code*. Hanya diganti dalam pengambilan populasi dan juga *fitness function*.

4.2 Skenario Eksperimen

Penelitian ini menyusun berbagai skenario dari dua metode penelitian yang dilakukan yaitu metode *timeseries forecasting* dan metode *optimal sizing*. Dalam metode *timeseries forecasting*, akan dilakukan berbagai skenario. Skenario-skenario yang akan dilakukan terdapat pada Tabel 4.2. Setelah mendapatkan nilai prediksi terbaik dari masing-masing target yaitu *temperature*, GHI, dan *demand* listrik (KW), maka akan dilakukan skenario perhitungan untuk mendapatkan nilai kapasitas PV dan kapasitas BESS terbaik menggunakan model matematika yang telah dijelaskan pada Subbab 2.2.5 dan menggunakan algoritma optimasi berupa *Marine Predator Algorithm*. Hasil dari MPA ini, dapat menentukan kapan baterai melakukan *discharging* dan *charging* pada waktu yang terbaik pada Bulan Agustus 2019.

4.3 Hasil Penelitian

Pada bagian ini, akan menampilkan hasil pengujian menggunakan berbagai model untuk *timeseries forecasting* dan juga pengujian dalam metode *optimal sizing*. Skenario yang akan ditampilkan pada subbab ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Skenario *Timeseries Forecasting*

Jenis skenario	Hasil Penelitian
Skenario 1	Hasil implementasi antara model FEDFormer FEB dan WEB tanpa LSTM pada Semua Fitur vs <i>Selected</i> Fitur
Skenario 2	Hasil implementasi antara model FEDFormer FEB dan WEB dengan LSTM pada Semua Fitur vs <i>Selected</i> Fitur
Skenario 3	Hasil implementasi antara model FEDFormer FEB tanpa LSTM dan dengan LSTM pada Semua Fitur vs <i>Selected</i> Fitur
Skenario 4	Hasil implementasi antara model FEDFormer WEB tanpa LSTM dan dengan LSTM pada Semua Fitur vs <i>Selected</i> Fitur
Skenario 5	Hasil Implementasi antara model LSTM, Hybrid LSTM with Transformer, dan FEDFormer (FEB dan WEB) with or without LSTM pada Semua Fitur vs <i>Selected</i> Fitur

Pada model yang menggunakan semua fitur, fitur-fitur yang digunakan adalah *temperature*, *dew point*, *GHI*, *pressure*, *wind speed*, *relative humidity*, *precipitable water*, *cloud type*, dan *demand listrik (KW)*. Pada model yang menggunakan fitur *selected* berdasarkan *correlation matrix*, *target temperature* akan menggunakan *fitur temperature*, *dew point*, *precipitable water*, *GHI*, *pressure*, *wind speed*, dan *demand listrik (KW)*. Untuk target *GHI* menggunakan fitur *temperature*, *dew point*, *relative humidity*, *GHI*, dan *demand listrik (KW)*. Terakhir, untuk target *demand listrik (KW)* menggunakan fitur *temperature*, *dew point*, *precipitable water*, *GHI*, *pressure*, dan *demand listrik (KW)*.

4.3.1 Hasil Implementasi Skenario 1

Pada implementasi ini, dilakukan pengujian hasil evaluasi percobaan dataset dengan model FEDFormer FEB dan WEB tanpa LSTM pada semua fitur atau berdasarkan fitur *selected* dengan *Correlation Matrix*. Percobaan dilakukan dengan melakukan prediksi untuk target *temperature*, *GHI*, dan *demand listrik (KW)*.

a. Prediksi untuk *temperature*

Pada prediksi *temperature*, hasil evaluasi didapatkan dengan evaluasi nilai MAE dan MSE untuk masing-masing percobaan yang dilakukan. Pada *Tabel 4.3*, dilakukan percobaan untuk mendapatkan nilai MAE yang menjadi perbandingan antara kedua model (FEDFormer FEB dan FEDFormer WEB) dari percobaan dengan semua fitur atau percobaan dengan fitur *selected* menggunakan *correlation matrix* dalam standar *scaler*, sedangkan pada *Tabel 4.4* merupakan hasil percobaan dalam skala asli. Dari *Tabel 4.3*, dalam standar *scaler*, terlihat bahwa nilai *average* MAE terendah berdasarkan percobaan dengan semua fitur ataupun dengan fitur *selected*, terdapat pada model FEDFormer WEB yang memiliki nilai sebesar 0,05655 dan 0,06033 yang terlihat pada warna kuning. Diantara kedua nilai *average* tersebut, model FEDFormer WEB dengan semua fitur merupakan percobaan yang memiliki nilai *average* MAE terkecil.

Dalam standar deviasi, pada percobaan dengan semua fitur, model yang memiliki nilai standar deviasi MAE terendah dimiliki oleh model FEDFormer WEB dengan nilai sebesar 0,00105 yang terlihat pada warna kuning. Sementara itu, untuk percobaan dengan *selected* fitur, model yang memiliki nilai standar deviasi MAE terendah dimiliki oleh model FEDFormer FEB dengan nilai sebesar 0,00309 yang terlihat pada warna kuning. Dari nilai-nilai standar deviasi tersebut, model FEDFormer WEB dengan percobaan semua fitur memiliki nilai standar deviasi MAE terkecil.

Tabel 4.3 Hasil MAE Skenario 1 *Temperature Standar Scaler*

Temperature	MAE			
	Semua fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,07035	0,05693	0,06766	0,05571
2	0,06485	0,05716	0,07131	0,06346
3	0,07859	0,05498	0,06920	0,05978
4	0,07394	0,05605	0,06786	0,05695
5	0,06984	0,05764	0,07511	0,06573
average	0,07152	0,05655	0,07023	0,06033
stdev	0,00511	0,00105	0,00309	0,00424

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.4*, dalam skala asli, model FEDFormer WEB memiliki nilai *average* MAE terendah untuk percobaan dengan semua fitur dan *selected* fitur. Hasil *average* MAE pada percobaan dengan semua fitur untuk model FEDFormer WEB adalah 0,15635. Pada percobaan dengan *selected* fitur, model FEDFormer WEB memiliki *average* MAE sebesar 0,16678 yang terlihat pada warna kuning. Dari kedua nilai *average* MAE tersebut, yang memiliki nilai *average* MAE terkecil adalah model FEDFormer WEB dengan menggunakan semua fitur.

Selain itu, untuk standar deviasi, model FEDFormer WEB juga memiliki nilai standar deviasi MAE terendah daripada model lainnya dengan percobaan menggunakan semua fitur yang memiliki nilai sebesar 0,00291 yang terlihat pada warna kuning. Untuk percobaan dengan *selected* fitur, model FEDFormer FEB memiliki nilai standar deviasi MAE terendah dengan nilai sebesar 0,00855 yang terlihat pada warna kuning. Dari kedua nilai standar deviasi MAE tersebut, yang memiliki nilai standar deviasi MAE terendah adalah model FEDFormer WEB pada percobaan menggunakan semua fitur. Warna hijau pada *Tabel 4.4* merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.4 Hasil MAE Skenario 1 *Temperature Skala Asli*

Temperature	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,19451	0,15740	0,18705	0,15402
2	0,17930	0,15802	0,19716	0,17544
3	0,21729	0,15200	0,19130	0,16527
4	0,20443	0,15495	0,18760	0,15745
5	0,19309	0,15937	0,20765	0,18173
average	0,19772	0,15635	0,19415	0,16678
stdev	0,01413	0,00291	0,00855	0,01173

Nilai terendah dari average/standar deviasi

Nilai sampel terendah dari lima percobaan

Pada *Tabel 4.5*, dalam standar *scaler*, ditampilkan hasil evaluasi percobaan yang bertujuan untuk mendapatkan nilai MSE dari dua model yaitu FEDFormer FEB dan FEDFormer WEB. Evaluasi dilakukan baik pada model yang menggunakan seluruh fitur yang tersedia maupun pada model yang hanya menggunakan fitur-fitur terpilih berdasarkan hasil seleksi menggunakan *correlation matrix* dan data melalui proses standar *scaler*. Sementara itu, pada *Tabel 4.6*, juga disajikan hasil evaluasi MSE yang dihitung pada skala asli pada kedua model

baik menggunakan seluruh fitur yang tersedia maupun menggunakan *selected* fitur berdasarkan hasil seleksi dengan *correlation* fitur.

Tabel 4.5 Hasil MSE Skenario 1 *Temperature* Standar *Scaler*

Temperature	MSE			
	Semua fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,00810	0,00538	0,00743	0,00513
2	0,00680	0,00522	0,00838	0,00628
3	0,00976	0,00500	0,00799	0,00575
4	0,00932	0,00517	0,00758	0,00535
5	0,00778	0,00530	0,00898	0,00699
average	0,00835	0,00521	0,00807	0,00590
stdev	0,00120	0,00015	0,00063	0,00075

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.5*, dalam standar *scaler*, model FEDFormer WEB memiliki nilai *average* MSE terendah baik dari percobaan dengan semua fitur maupun dengan *selected fitur* yang memiliki nilai sebesar 0,00521 dan 0,00590 yang terlihat pada warna kuning. Dari kedua nilai *average* MSE tersebut, yang memiliki nilai *average* MSE terendah adalah model FEDFormer WEB dengan percobaan semua fitur.

Untuk standar deviasi, pada percobaan dengan semua fitur, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi terendah daripada model lainnya yaitu sebesar 0,00015 yang terlihat pada warna kuning. Pada percobaan dengan *selected* fitur, nilai standar deviasi MSE terendah dimiliki oleh model FEDFormer FEB yang memiliki nilai sebesar 0,00063 yang terlihat pada warna kuning. Dari kedua nilai standar deviasi tersebut, yang memiliki nilai standar deviasi MSE terendah adalah model FEDFormer WEB dalam percobaan semua fitur.

Tabel 4.6 Hasil MSE Skenario 1 *Temperature* Skala Asli

Temperature	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,06188	0,04113	0,05683	0,03925
2	0,05198	0,03989	0,06406	0,04803
3	0,07461	0,03821	0,06110	0,04395
4	0,07123	0,03948	0,05797	0,04089
5	0,05948	0,04052	0,06862	0,05340
average	0,06383	0,03985	0,06172	0,04510
stdev	0,00914	0,00111	0,00478	0,00572

Nilai terendah dari average/standar deviasi

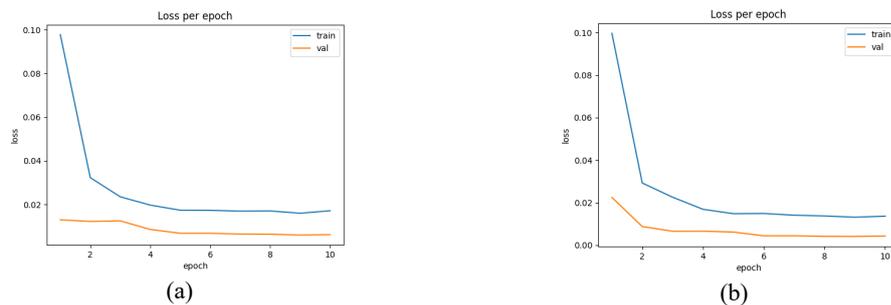
Pada *Tabel 4.6*, dalam skala asli, model FEDFormer WEB memiliki nilai *average* MSE terendah baik pada percobaan menggunakan semua fitur maupun dengan *selected fitur* yang memiliki nilai sebesar 0,03985 dan 0,04510 yang terlihat pada warna kuning. Dari kedua nilai tersebut, yang memiliki nilai *average* MSE terendah adalah model FEDFormer WEB dalam percobaan semua fitur.

Untuk standar deviasi, pada percobaan dengan semua fitur, model FEDFormer WEB memiliki nilai standar deviasi terendah daripada model lainnya yaitu sebesar 0,00111 yang terlihat pada warna kuning. Pada percobaan dengan *selected* fitur, model FEDFormer FEB yang

memiliki nilai standar deviasi MSE terendah yaitu sebesar 0,00478 yang terlihat pada warna kuning. Dari kedua nilai tersebut, yang memiliki nilai standar deviasi terendah adalah model FEDFormer WEB dalam percobaan semua fitur.

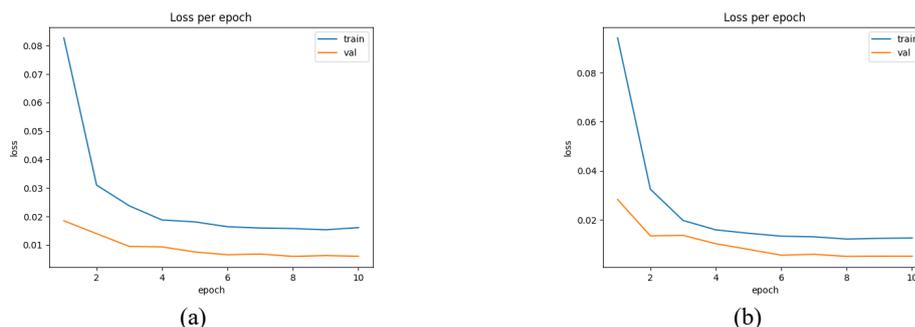
Berdasarkan *Tabel 4.4*, diambil sampel yang memiliki nilai terendah dari seluruh percobaan yang dilakukan. Hal tersebut ditandai dengan warna hijau. Untuk FEDFormer FEB pada semua fitur mengambil sampel percobaan kedua. Untuk FEDFormer WEB pada semua fitur mengambil sampel percobaan ketiga. Untuk FEDFormer FEB pada *selected* fitur mengambil sampel pertama dan untuk FEDFormer WEB mengambil sampel pertama.

Pada *Gambar 4.1*, terlihat grafik *validation loss* pada percobaan dengan *selected* fitur untuk model FEDFormer FEB dan FEDFormer WEB. Hasilnya adalah untuk model FEDFormer FEB menunjukkan pergerakan *loss* selama proses *training model*. *Train loss* mengalami penurunan yang signifikan dari *epoch* ke-1 hingga ke-3. Kemudian pada *epoch* ke-5, *train loss* mencapai kestabilan. Di sisi lain, *validation loss* berada pada nilai yang lebih rendah dibanding *train loss* dan mengalami sedikit penurunan sebelum akhirnya stabil dibawah sekitar 0,015. Untuk model FEDFormer WEB, terlihat bahwa *train loss* mengalami penurunan signifikan dari *epoch* ke-1 hingga ke-4 kemudian stabil dibawah 0,02. *Validation loss* menunjukkan nilai yang lebih rendah daripada *train loss* selama *training* dan memiliki nilai yang awalnya menurun lalu stabil mendekati 0,005.



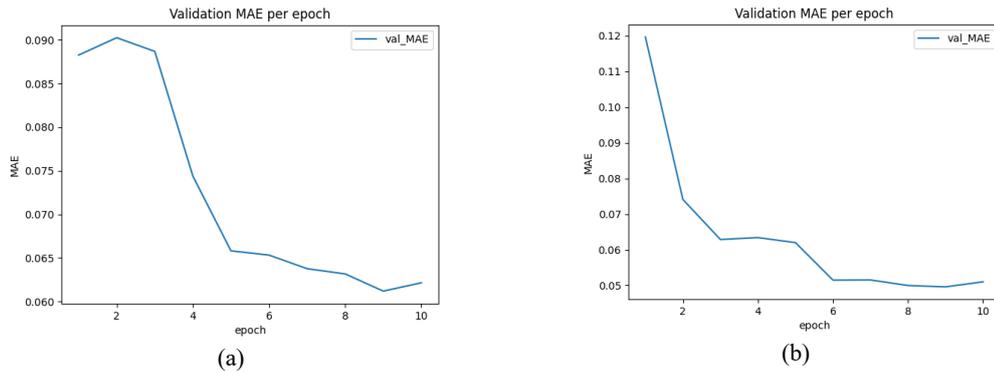
Gambar 4.1 Grafik *Validation loss Temperature Selected* Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada *Gambar 4.2*, untuk percobaan menggunakan semua fitur, pada model FEDFormer FEB terlihat bahwa *train loss* turun signifikan dari *epoch* ke-1 hingga ke-3 lalu melandai mulai *epoch* ke-5. Selain itu, *validation loss* juga mengalami penurunan dan stabil pada nilai mendekati 0,005. Pada model FEDFormer WEB, terlihat bahwa *train loss* menurun dari *epoch* ke-1 hingga ke-3 kemudian mengalami penurunan secara gradual. Untuk *validation loss* juga menunjukkan pola menurun yang stabil dan mencapai titik mendekati 0,015.



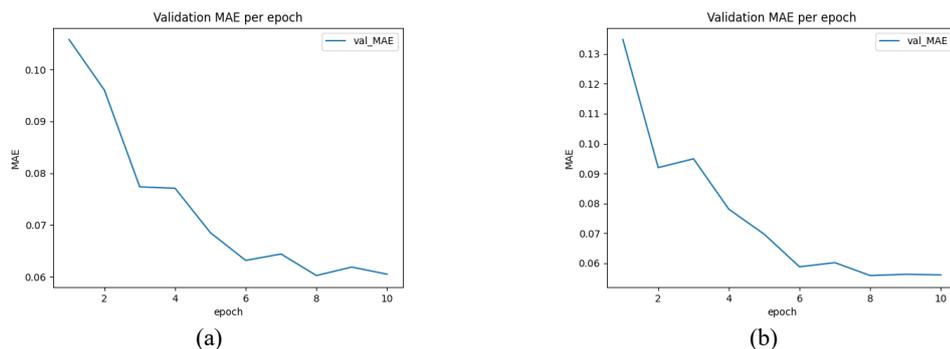
Gambar 4.2 Grafik *Validation loss Temperature Semua* Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.3, terdapat grafik *validation* MAE berdasarkan *selected* fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, di *epoch* pertama hingga ketiga mengalami sedikit kenaikan namun setelahnya menurun secara signifikan dan mulai melandai pada *epoch* ke-5. Untuk model FEDFormer WEB, terlihat penurunan MAE yang cukup tajam dari *epoch* pertama hingga *epoch* ketiga. Setelah itu, nilai MAE cenderung stagnan lalu mulai *epoch* ke 5 menurun lagi hingga mulai stabil di *epoch* ke-8 hingga akhir.



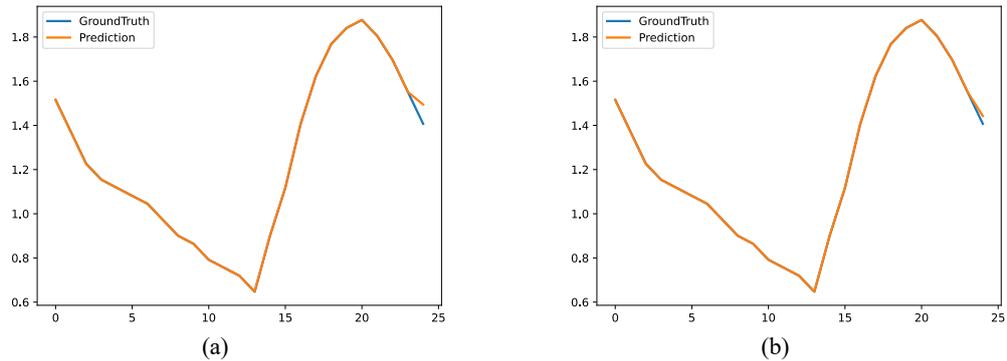
Gambar 4.3 Grafik Validation MAE Temperature Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.4, terdapat juga grafik *validation* MAE yang berdasarkan percobaan menggunakan semua fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, Pola penurunan MAE terlihat cukup konsisten sepanjang 10 *epoch* meski sempat mengalami sedikit kenaikan pada pola dari *epoch* ke-6 hingga 7 sebelum menurun dan sedikit naik lagi pada *epoch* ke-8 lalu menurun lagi. Secara umum, grafik menunjukkan tren menurun yang stabil dari nilai MAE di atas 0,10 hingga di bawah 0,065. Pada model FEDFormer WEB, MAE menurun tajam dari *epoch* ke-1 hingga 2 lalu mengalami sedikit kenaikan pada *epoch* ke-2 hingga ke-3 kemudian menurun hingga *epoch* ke-6. Kemudian nilai MAE cenderung menurun dan stabil hingga *epoch* ke-10.



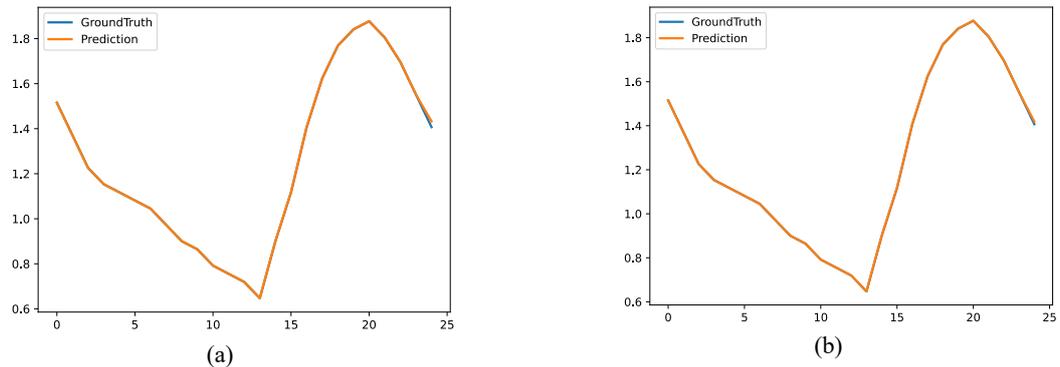
Gambar 4.4 Grafik Validation MAE Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.5, terdapat grafik prediksi untuk percobaan menggunakan *selected* fitur pada model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, kurva prediksi mengikuti pola *ground truth* secara umum namun terdapat deviasi di akhir *timestep* dan prediksi cenderung terlambat mengikuti penurunan nilai *ground truth*. Untuk model FEDFormer WEB, kurva prediksi hampir menyatu sepenuhnya dengan *ground truth* dari awal *timestep* hingga akhir. Tidak terlihat deviasi yang berlebihan sepanjang kurva.



Gambar 4.5 Grafik Prediksi Temperature Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.6, terdapat juga grafik prediksi untuk percobaan menggunakan semua fitur pada model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terlihat performa prediksi yang dekat dengan *ground truth* sepanjang *timestep* meski pada *timestep* akhir terlihat sedikit sekali *gap* antara *ground truth* dengan nilai *prediction*. Untuk model FEDFormer WEB, terlihat bahwa hasil prediksi sangat akurat dan halus dengan kurva prediksi hampir identik dengan *ground truth*. Kurva *predict* juga mampu menangkap *timestep* akhir dengan baik.



Gambar 4.6 Grafik Prediksi Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

b. Prediksi untuk GHI

Pada prediksi untuk target GHI, dilakukan percobaan untuk mendapatkan hasil evaluasi MAE pada model FEDFormer FEB dan FEDFormer WEB dengan percobaan semua fitur maupun dengan fitur *selected*. Pada Tabel 4.7, dalam standar *scaler*, model FEDFormer WEB dalam percobaan dengan semua fitur dan *selected* fitur memiliki nilai *average* MAE terendah daripada model lainnya yaitu sebesar 0,24819 dan 0,23211 yang terlihat pada warna kuning. Dari kedua nilai *average* MAE tersebut, model FEDFormer WEB dengan percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai *average* MAE terendah daripada model lainnya baik dari percobaan dengan semua fitur maupun *selected* fitur yaitu sebesar 0,00677 dan 0,00141 yang terlihat pada warna kuning. Dari kedua nilai standar deviasi tersebut, model FEDFormer WEB dengan percobaan menggunakan *selected* fitur memiliki nilai terendah.

Tabel 4.7 Hasil MAE Skenario 1 GHI Standar Scaler

GHI	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,26480	0,24845	0,25306	0,23410
2	0,25393	0,25374	0,24699	0,23299
3	0,25453	0,23660	0,24568	0,23161
4	0,25290	0,25187	0,24042	0,23062
5	0,26810	0,25031	0,24775	0,23126
average	0,25885	0,24819	0,24678	0,23211
stdev	0,00706	0,00677	0,00453	0,00141

Nilai terendah dari average/standar deviasi

Dalam skala asli pada Tabel 4.8, model FEDFormer WEB memiliki nilai *average* MAE terendah daripada model lainnya di kedua percobaan baik menggunakan semua fitur maupun *selected* fitur dengan nilai sebesar 63,74603 dan 59,61641 yang terlihat pada warna kuning. Dari kedua nilai *average* MAE tersebut, model FEDFormer WEB dengan percobaan *selected* fitur memiliki nilai *average* terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi terendah daripada model lainnya di kedua percobaan baik menggunakan semua fitur maupun *selected* fitur dengan nilai sebesar 1,73812 dan 0,36198 yang terlihat pada warna kuning. Dari kedua nilai standar deviasi tersebut, model FEDFormer WEB dengan percobaan *selected* fitur memiliki nilai standar deviasi terendah. Warna hijau pada Tabel 4.8 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.8 Hasil MAE Skenario 1 GHI Skala Asli

GHI	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	68,01223	63,81179	64,99726	60,12680
2	65,21964	65,17021	63,43626	59,84094
3	65,37519	60,76867	63,10144	59,48606
4	64,95467	64,69069	61,74897	59,23209
5	68,85921	64,28880	63,63170	59,39618
average	66,48419	63,74603	63,38313	59,61641
stdev	1,81274	1,73812	1,16403	0,36198

Nilai terendah dari average/standar deviasi

Nilai sampel terendah dari lima percobaan

Pada Tabel 4.9, terkait nilai MSE, model FEDFormer WEB memiliki nilai *average* MSE terendah daripada model lainnya baik di percobaan dengan semua fitur maupun *selected* fitur dengan nilai sebesar 0,16407 dan 0,15946 yang terlihat pada warna kuning. Dari kedua nilai *average* MSE tersebut, model FEDFormer WEB dengan percobaan *selected* fitur memiliki nilai terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MSE terendah daripada model lainnya baik di percobaan dengan semua fitur maupun dengan *selected* fitur dengan nilai sebesar 0,00237 dan 0,00090 yang terlihat pada warna kuning. Dari kedua nilai standar deviasi MSE tersebut, model FEDFormer WEB dengan percobaan menggunakan *selected* fitur memiliki nilai terendah.

Tabel 4.9 Hasil MSE Skenario 1 GHI Standar Scaler

GHI	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,17132	0,16315	0,16738	0,16026
2	0,16820	0,16675	0,16551	0,16052
3	0,16545	0,16055	0,16476	0,15902
4	0,16630	0,16444	0,16188	0,15914
5	0,17008	0,16547	0,16536	0,15836
average	0,16827	0,16407	0,16498	0,15946
stdev	0,00247	0,00237	0,00199	0,00090

Nilai terendah dari average/standar deviasi

Pada Tabel 4.10, dalam skala asli, model FEDFormer WEB memiliki nilai *average* MSE terendah daripada model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur dengan nilai sebesar 10823,3788 dan 10519,37532 yang terlihat pada warna kuning. Dari kedua nilai *average* MSE tersebut, model FEDFormer WEB dengan percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model FEDFormer WEB baik pada percobaan semua fitur maupun dengan *selected* fitur memiliki nilai terendah daripada model lainnya yaitu sebesar 156,53530 dan 59,61348 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB menggunakan percobaan *selected* fitur memiliki nilai terendah.

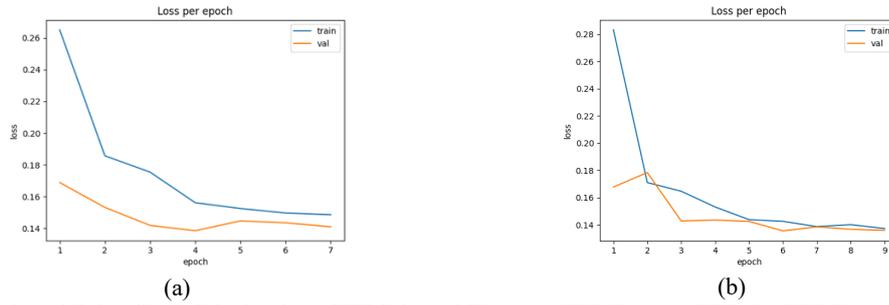
Tabel 4.10 Hasil MSE Skenario 1 GHI Skala Asli

GHI	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	11301,41953	10762,41626	11041,50543	10572,18207
2	11095,67983	11000,16986	10918,49476	10589,15938
3	10914,48306	10591,07054	10868,81746	10490,42824
4	10970,39344	10847,77496	10678,96242	10498,38167
5	11220,00711	10915,46240	10908,72200	10446,72523
average	11100,39659	10823,37880	10883,30041	10519,37532
stdev	162,94463	156,53530	131,22261	59,61348

Nilai terendah dari average/standar deviasi

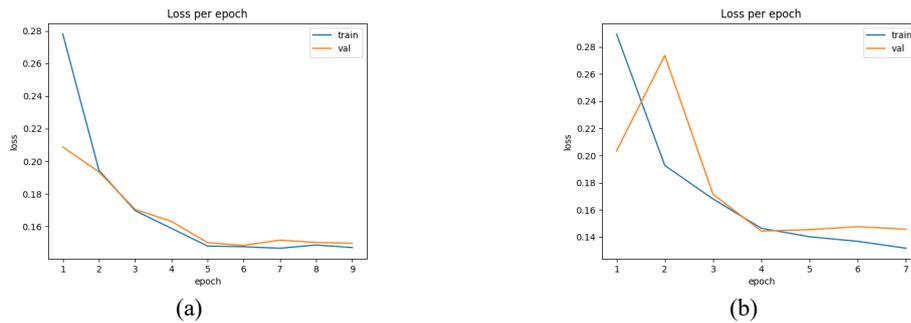
Berdasarkan Tabel 4.8, diambil sampel yang memiliki nilai terendah dari seluruh percobaan yang dilakukan yang terdapat pada warna hijau. Untuk FEDFormer FEB pada semua fitur mengambil sampel percobaan keempat. Untuk FEDFormer WEB pada semua fitur mengambil sampel percobaan ketiga. Untuk FEDFormer FEB pada *selected* fitur mengambil sampel percobaan keempat dan untuk FEDFormer WEB pada percobaan *selected* fitur mengambil percobaan keempat.

Pada Gambar 4.7, terdapat kurva *validation loss* untuk percobaan menggunakan *selected* fitur pada model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terlihat penurunan *training loss* yang signifikan pada 4 *epoch* awal dari sekitar 0,26 hingga sekitar 0,16 kemudian secara perlahan menurun hingga mendekati 0,15. Untuk *validation loss* juga menurun secara konsisten meskipun sempat menaik dari *epoch* ke-4 hingga ke-5 kemudian menurun lagi. Untuk model FEDFormer WEB, terlihat penurunan tajam pada *training loss* dari 0,28 hingga sekitar 0,17 dalam 3 *epoch* pertama. Namun *validation loss* mengalami fluktuasi yaitu naik dari *epoch* ke-1 hingga 2 kemudian turun drastis pada *epoch* ke-3. Selanjutnya nilai mulai stabil dan konvergen.



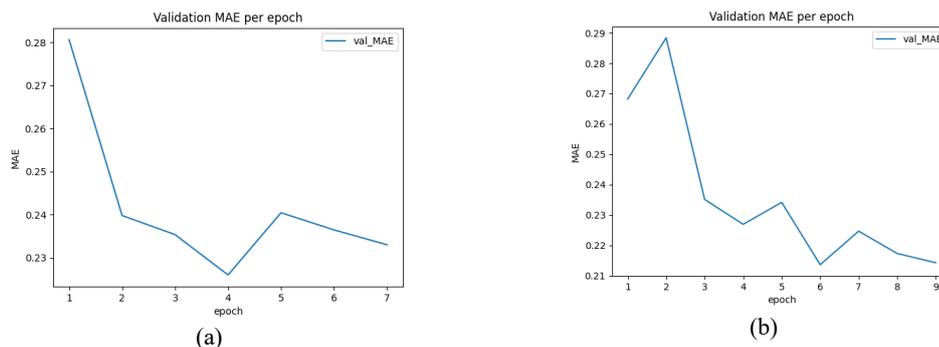
Gambar 4.7 Grafik Validation loss GHI Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.8, terdapat grafik *validation loss* untuk percobaan menggunakan semua fitur pada model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terlihat penurunan *train* dan *validation loss* yang hampir identik sepanjang proses *training*. Setelah *epoch* ke-2, kedua kurva saling berdekatan dan mengikuti yang menandakan *generalization gap* sangat kecil. Untuk model FEDFormer WEB, *validation loss* sempat meningkat tajam dari *epoch* ke-1 hingga 2 dan melebihi *training loss* yang bisa menandakan bahwa terjadi *overfitting* dini namun setelahnya kedua kurva menurun dan mulai konvergen di sekitar *epoch* ke-4 hingga akhir *epoch* ke-7 karena mengalami *early stopping*.



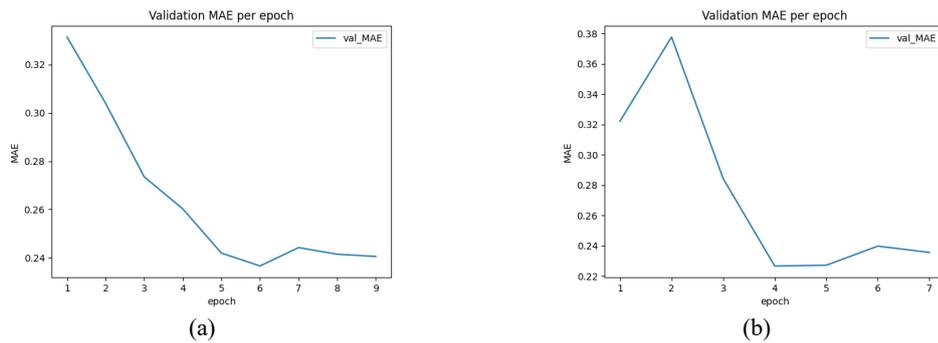
Gambar 4.8 Grafik Validation loss GHI Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.9, terdapat grafik *validation MAE* pada percobaan menggunakan *selected* fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terlihat penurunan *validation MAE* cukup stabil dari *epoch* ke-1 hingga ke-4 dengan nilai awal sekitar 0,28 hingga akhir mendekati nilai 0,21 pada *epoch* ke-4. Namun setelah itu, terjadi sedikit kenaikan antara *epoch* ke-4 hingga ke-5 kemudian mengalami penurunan hingga *epoch* ke-7 dan disini diterapkan *early stopping*. Untuk model FEDFormer WEB, pola MAE terlihat cukup fluktuatif. Setelah penurunan tajam dari *epoch* ke-2 hingga ke-3, grafik menunjukkan zig-zag dari *epoch* ke-4 hingga *epoch* ke-9.



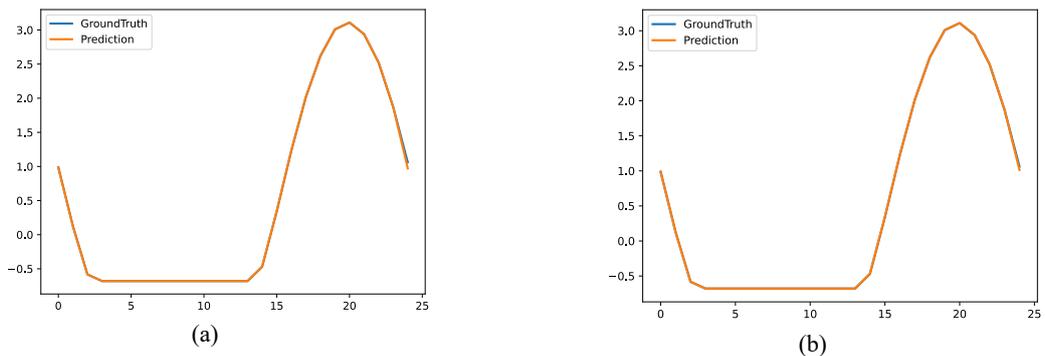
Gambar 4.9 Grafik Validation MAE GHI Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.10, terdapat grafik *validation* MAE untuk percobaan menggunakan semua fitur pada model FEDFormer FEB dan model FEDFormer WEB. Pada model FEDFormer FEB, *validation* MAE terlihat menurun dan konsisten dari *epoch* ke-1 hingga ke-6 dari sekitar 0,33 hingga sekitar 0,23. Setelah itu, nilai MAE sedikit menaik dan mendatar stabil hingga *epoch* ke-9. Pada model FEDFormer WEB, terlihat bahwa terdapat peningkatan *validation* MAE yang cukup tinggi pada awal *training* yaitu pada *epoch* ke-1 hingga ke-2. Setelah itu, *validation* MAE menurun hingga *epoch* ke-4 dari nilai sekitar 0,37 menjadi sekitar 0,23. Kemudian, *validation* MAE mengalami kenaikan lagi dari *epoch* ke-5 hingga ke-6 sebelum kemudian cenderung bergerak stabil hingga *epoch* terakhir. Disini terjadi *early stopping*.



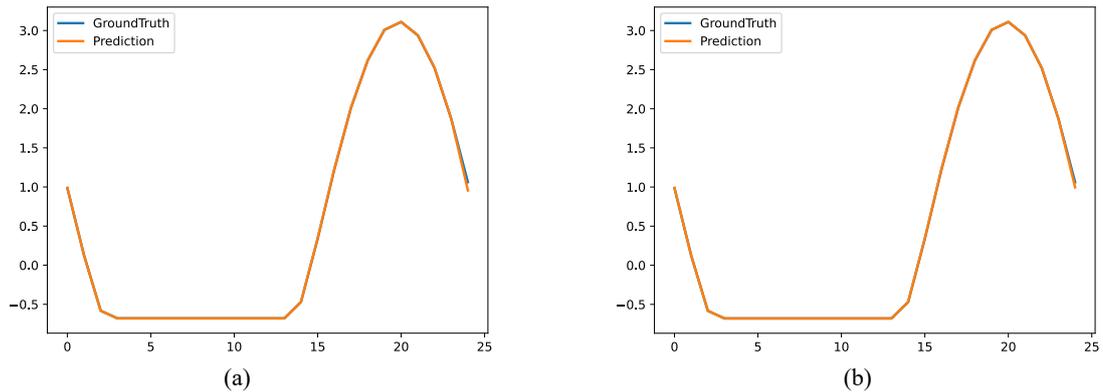
Gambar 4.10 Grafik Validation MAE GHI Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.11, terdapat grafik prediksi untuk percobaan menggunakan *selected* fitur pada model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB menunjukkan kurva prediksi yang mengikuti pola *ground truth* secara umum dan pada bagian akhir *timestep* hanya terlihat sedikit perbedaan antara kurva *predict* dengan kurva *ground truth*. Untuk model FEDFormer WEB, terlihat bahwa grafik *predict* memiliki hasil yang sangat akurat dan nyaris tidak ada *gap* antara kurva *predict* dengan kurva *ground truth*.



Gambar 4.11 Grafik Prediksi GHI Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.12, terdapat percobaan menggunakan semua fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terlihat bahwa kurva *predict* mampu mengikuti pola *ground truth* namun pada *timestep* terakhir terlihat perbedaan diantaranya. Untuk model FEDFormer WEB, kondisi kurva nyaris sempurna, akan tetapi pada *timestep* terakhir masih terlihat sedikit *gap* antara nilai *predict* dengan nilai *ground truth*.



Gambar 4.12 Grafik Prediksi GHI Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

c. Prediksi untuk *demand* listrik (KW)

Pada prediksi *demand* listrik (KW), hasil evaluasi didapatkan dengan menghitung nilai MAE dan MSE untuk masing-masing percobaan yang dilakukan. Pada Tabel 4.11, dilakukan percobaan untuk mendapatkan nilai MAE pada standar *scaler* dengan percobaan menggunakan semua fitur atau *selected* fitur dengan *correlation matrix*. Disini, terlihat bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah daripada model lainnya baik pada percobaan menggunakan semua fitur maupun percobaan dengan *selected* fitur. Hasil *average* MAE pada percobaan semua fitur dan *selected* fitur adalah 0,16465 dan 0,14752 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Selain itu, untuk standar deviasi, didapatkan bahwa model FEDFormer WEB memiliki nilai terendah daripada model lainnya baik dari percobaan menggunakan semua fitur ataupun dengan *selected* fitur. Hasil standar deviasi MAE untuk percobaan menggunakan semua fitur dan dengan *selected* fitur adalah 0,00273 dan 0,00317 yang terlihat pada warna kuning. Dari kedua percobaan tersebut, dapat diketahui bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.11 Hasil MAE Skenario 1 *Demand* Listrik Standar *Scaler*

KW	MAE				<div style="background-color: yellow; border: 1px solid black; padding: 2px; display: inline-block;"> Nilai terendah dari average/standar deviasi </div>
	Semua Fitur		Selected Fitur		
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB	
1	0,18190	0,16310	0,16368	0,14684	
2	0,19047	0,16787	0,17887	0,14521	
3	0,16578	0,16459	0,17197	0,14637	
4	0,17196	0,16664	0,16550	0,14609	
5	0,18882	0,16104	0,18054	0,15308	
average	0,17979	0,16465	0,17211	0,14752	
stdev	0,01070	0,00273	0,00761	0,00317	

Pada Tabel 4.12, dalam skala asli, model FEDFormer WEB memiliki nilai *average* MAE terendah daripada model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil nilai *average* MAE untuk percobaan dengan semua fitur dan *selected* fitur adalah 20,42159 dan 18,29667 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB menggunakan percobaan *selected* fitur memiliki nilai *average* MAE terendah.

Tabel 4.12 Hasil MAE Skenario 1 *Demand Listrik Skala Asli*

KW	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	22,56146	20,22939	20,30089	18,21313
2	23,62449	20,82153	22,18556	18,01008
3	20,56164	20,41450	21,32935	18,15417
4	21,32801	20,66883	20,52657	18,11908
5	23,41891	19,97368	22,39243	18,98689
average	22,29890	20,42159	21,34696	18,29667
stdev	1,32677	0,33863	0,94391	0,39286

Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Selain itu, pada standar deviasi, model FEDFormer WEB juga memiliki nilai terendah daripada model lainnya pada percobaan semua fitur dan *selected* fitur dengan nilai sebesar 0,33863 dan 0,39286 yang terlihat pada warna kuning. Dari kedua nilai tersebut, percobaan model FEDFormer WEB dengan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.12 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.13 Hasil MSE Skenario 1 *Demand Listrik Standar Scaler*

KW	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	0,05623	0,04676	0,04742	0,03869
2	0,05783	0,04845	0,05552	0,03844
3	0,04685	0,04734	0,05056	0,03868
4	0,05111	0,04810	0,04860	0,03864
5	0,06044	0,04481	0,05796	0,04209
average	0,05449	0,04709	0,05201	0,03931
stdev	0,00546	0,00143	0,00454	0,00156

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.13*, untuk hasil evaluasi MAE pada standar *scaler*, diketahui bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur dan *selected* fitur memiliki nilai *average* MSE terendah daripada model lainnya. Hasil *average* MSE pada percobaan semua fitur dan *selected* fitur untuk model FEDFormer WEB adalah 0,04709 dan 0,03931 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah.

Selain itu, untuk standar deviasi, model FEDFormer WEB juga memiliki standar deviasi MSE terendah daripada model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil standar deviasi MSE-nya adalah 0,00143 dan 0,00156 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah.

Pada *Tabel 4.14*, dalam skala asli, didapatkan hasil evaluasi MSE dari semua percobaan yang dilakukan. Terlihat bahwa model FEDFormer WEB memiliki nilai *average* MSE terkecil daripada model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil *average* MSE pada percobaan menggunakan semua fitur maupun *selected* fitur adalah 724,44574 dan 604,71874 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan dengan *selected* fitur memiliki nilai *average* MSE terkecil.

Selain itu, pada standar deviasi, model FEDFormer WEB juga memiliki nilai standar deviasi MSE terendah daripada model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil nilai standar deviasi MSE pada semua fitur dan *selected* fitur adalah 22,06033 dan 23,940145 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEBFormer WEB pada percobaan menggunakan semua fitur memiliki standar deviasi MSE terkecil.

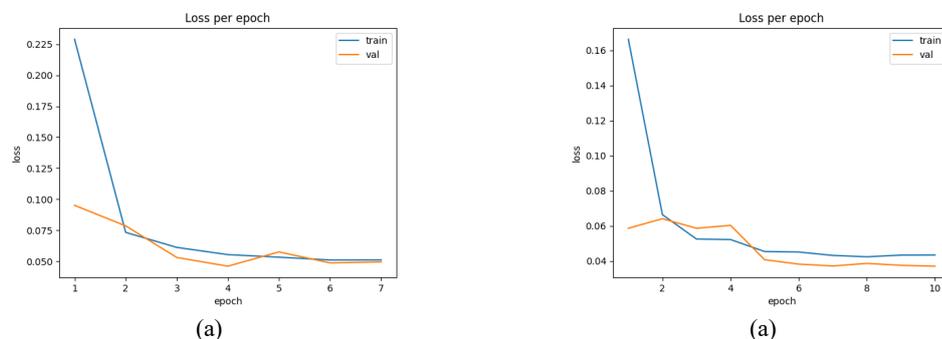
Tabel 4.14 Hasil MSE Skenario 1 Demand Listrik Skala Asli

KW	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	FEDFormer WEB	FEDFormer FEB	FEDFormer WEB
1	864,95705	719,25861	729,48152	595,25789
2	889,70471	745,30741	854,09893	591,41976
3	720,71770	728,29094	777,80976	595,09895
4	786,29849	739,97640	747,59435	594,36244
5	929,70566	689,39535	891,58373	647,45466
average	838,27672	724,44574	800,11366	604,71874
stdev	84,02052	22,06033	69,86742	23,94015

Nilai terendah dari average/standar deviasi

Berdasarkan Tabel 4.12, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terdapat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk FEDFormer FEB pada semua fitur mengambil sampel percobaan ketiga. Untuk FEDFormer WEB pada semua fitur mengambil sampel percobaan kelima. Untuk FEDFormer FEB pada *selected* fitur mengambil sampel percobaan pertama, dan untuk FEDFormer WEB pada percobaan *selected* fitur mengambil percobaan kedua.

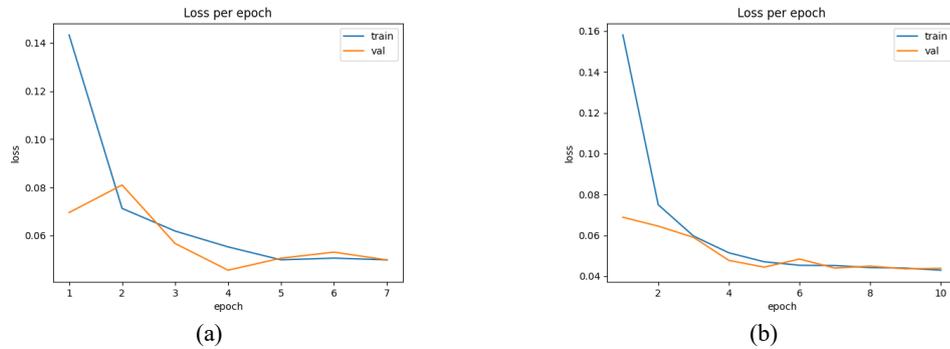
Pada Gambar 4.13, terdapat kurva *validation loss* pada percobaan menggunakan *selected* fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, untuk *train loss* memiliki penurunan dari *epoch* ke-1 hingga ke-3 dari nilai diatas 0,225 hingga sekitar 0,075. Kemudian nilai *train loss* mengalami penurunan dan stabil hingga *epoch* ke-7 karena terjadi *early stopping*. Untuk *validation loss*, juga mengalami penurunan dan stabil mulai *epoch* ke-4 mengikuti *train loss*. Pada model FEDFormer WEB, terlihat bahwa penurunan *train loss* terjadi secara cepat hingga mencapai nilai mendekati 0,04 pada akhir *epoch* ke-10. Namun *validation loss* mengalami sedikit kenaikan dari *epoch* ke-3 hingga ke-4 sebelum akhirnya turun dan stabil.



Gambar 4.13 Grafik Validation loss Demand Listrik Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

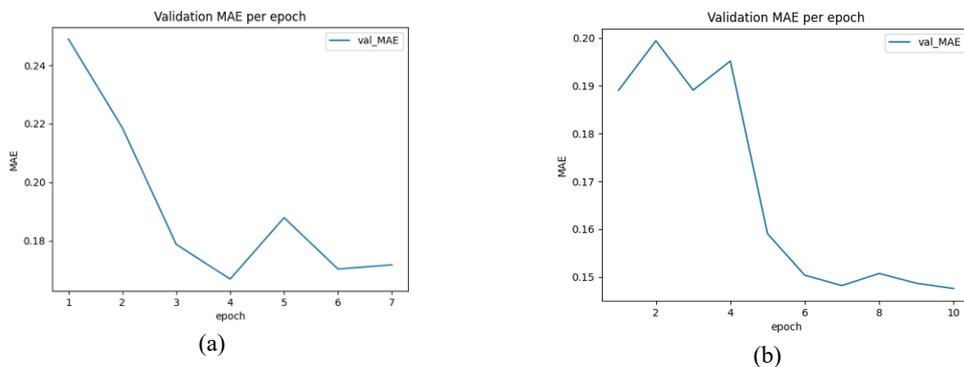
Pada Gambar 4.14, terdapat kurva *validation loss* pada percobaan menggunakan semua fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, *train loss* dan *validation loss* terlihat semakin menyatu mulai dari *epoch* 5 hingga akhir. Untuk *train loss*, mengalami penurunan dari 0,14 hingga sekitar 0,05. Pada *validation loss* kenaikan dari

epoch pertama hingga kedua. Kemudian menurun hingga *epoch ke-4*. Selanjutnya, mengalami kenaikan dari *epoch ke-4* hingga ke-6 lalu menurun hingga *epoch ke-7*. Disini terjadi *early stopping*. Untuk model FEDFormer WEB, *train loss* turun dari awal *epoch* sekitar 0,16 hingga sekitar 0,045 pada *epoch ke-10*. Untuk *validation loss* juga mengikuti pola serupa dan tidak ada fluktuasi berarti.



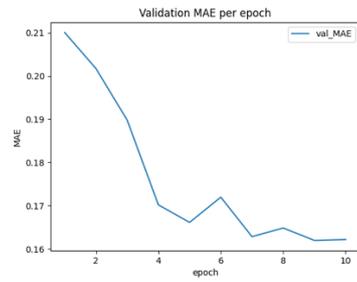
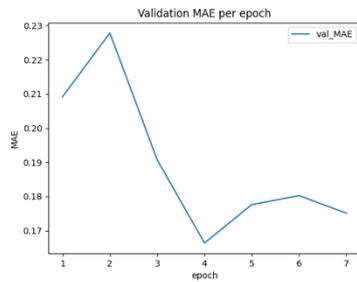
Gambar 4.14 Grafik Validation loss Demand Listrik Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.15, terkait *validation MAE*, percobaan dilakukan menggunakan *selected* fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, grafik menunjukkan penurunan yang signifikan pada 4 *epoch* pertama dari nilai sekitar 0,25 hingga sekitar 0,15. Setelah itu, kurva mengalami sedikit fluktuasi di *epoch ke-4* hingga ke-6 sebelum kembali stabil. Untuk model FEDFormer WEB, nilai *validation MAE* mengalami fluktuasi pada awal *training* dari *epoch ke-1* hingga ke-4 kemudian turun hingga *epoch ke-7* sebelum mengalami sedikit kenaikan lagi dan berakhir stabil di *epoch* terakhir yaitu 10.



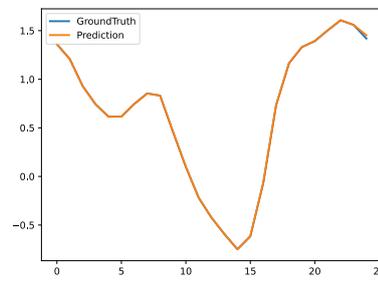
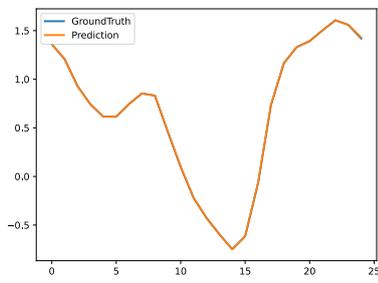
Gambar 4.15 Grafik Validation MAE Demand Listrik Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.16, terkait grafik *validation MAE* untuk model FEDFormer FEB dan model FEDFormer WEB, terlihat di awal *training* bahwa *validation MAE* mengalami penurunan dari *epoch* kedua hingga keempat dari sekitar 0,23 hingga 0,16. Kemudian nilai *validation* mengalami kenaikan dari *epoch ke-4* hingga ke-6 kemudian turun menuju *epoch ke-7*. Untuk model FEDFormer WEB, penurunan terlihat dari *epoch ke-1* hingga 5 (dari nilai sekitar 0,21 hingga sekitar 0,165). Kemudian sempat mengalami sedikit kenaikan dari *epoch ke-5* hingga ke-6 sebelum akhirnya menurun dan stabil.



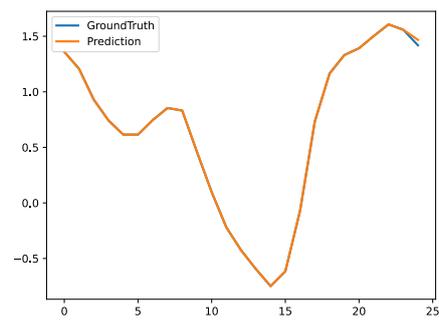
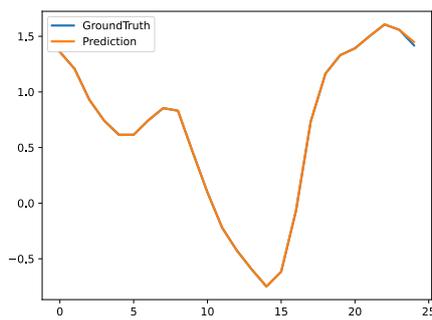
Gambar 4.16 Grafik Validation MAE Demand Listrik Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.17, terdapat grafik prediksi pada percobaan menggunakan *selected* fitur untuk model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terlihat pola prediksi yang mengikuti *ground truth* secara keseluruhan hingga akhir *timestep* yang terlihat sangat sempurna dan tidak terlihat adanya perbedaan. Untuk model FEDFormer WEB, pada akhir *timestep* terlihat sedikit perbedaan antara nilai *ground truth* dan *prediction*.



Gambar 4.17 Grafik Prediksi Demand Listrik Selected Fitur (a) FEDFormer FEB (b) FEDFormer WEB

Pada Gambar 4.18, terdapat grafik prediksi untuk percobaan menggunakan semua fitur pada model FEDFormer FEB dan FEDFormer WEB. Pada model FEDFormer FEB, terdapat sedikit sekali perbedaan antara kurva *predict* dengan nilai *ground truth* pada *timestep* akhir. Hal tersebut juga terjadi pada model FEDFormer WEB dimana terdapat sedikit perbedaan antara kurva *predict* dengan nilai *ground truth*.



Gambar 4.18 Grafik Prediksi Demand Listrik Semua Fitur (a) FEDFormer FEB (b) FEDFormer WEB

4.3.2 Hasil Implementasi Skenario 2

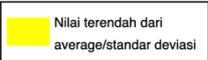
Pada implementasi ini, dilakukan pengujian hasil evaluasi percobaan dataset dengan model Hybrid FEDFormer FEB *with* LSTM dan Hybrid FEDFormer WEB *with* LSTM pada semua fitur atau berdasarkan *selected* fitur dengan *correlation matrix*. Percobaan dilakukan dengan melakukan prediksi untuk target *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada prediksi *temperature*, hasil evaluasi didapatkan dengan mendapatkan nilai MAE dan MSE untuk masing-masing percobaan yang dilakukan. Pada *Tabel 4.15*, dilakukan percobaan untuk mendapatkan nilai MAE dari model Hybrid FEDFormer FEB *with* LSTM dan model Hybrid FEDFormer WEB *with* LSTM dalam standar *scaler*. Dari tabel, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur memiliki nilai *average* MAE terendah daripada model lainnya yaitu 0,08848 dan 0,08079 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur merupakan nilai *average* MAE terendah.

Dalam standar deviasi, model Hybrid FEDFormer WEB *with* LSTM menggunakan percobaan dengan semua fitur memiliki nilai standar deviasi MAE terendah yaitu 0,00828 yang terlihat pada warna kuning. Sementara itu, model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MAE terendah daripada model Hybrid FEDFormer WEB *with* LSTM dengan nilai 0,00833 yang terlihat pada warna kuning. Dari kedua percobaan, model Hybrid FEDFormer FEB *with* LSTM menggunakan *selected* fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.15 Hasil MAE Skenario 2 *Temperature* Standar *Scaler*

Temperature	MAE				
	Semua fitur		Selected Fitur		
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	
1	0,09928	0,08514	0,08689	0,08038	
2	0,11161	0,09967	0,09218	0,09301	
3	0,10020	0,08937	0,10822	0,06522	
4	0,09782	0,09114	0,08946	0,08431	
5	0,11788	0,07708	0,09549	0,08102	
average	0,10536	0,08848	0,09445	0,08079	
stdev	0,00889	0,00828	0,00833	0,01006	

Pada *Tabel 4.16*, dalam skala asli, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terkecil daripada model lainnya baik dengan percobaan menggunakan semua fitur maupun dengan *selected* fitur. Nilai *average* MAE untuk Hybrid FEDFormer WEB *with* LSTM baik pada percobaan semua fitur maupun *selected* fitur adalah 0,24462 dan 0,22336 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Untuk standar deviasi, pada percobaan semua fitur, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi terendah dibanding model lainnya yaitu sebesar 0,02289 yang terlihat pada warna kuning. Pada percobaan dengan *selected* fitur, model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi terendah dibanding model lainnya yaitu sebesar 0,02304 yang terlihat pada warna kuning. Dari kedua percobaan tersebut, model Hybrid FEDFormer WEB *with* LSTM yang menggunakan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada *Tabel 4.16* merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.16 Hasil MAE Skenario 2 *Temperature* Skala Asli

Temperatu re	MAE			
	Semua Fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	0,27449	0,23539	0,24022	0,22222
2	0,30857	0,27555	0,25484	0,25716
3	0,27703	0,24708	0,29918	0,18031
4	0,27045	0,25197	0,24734	0,23310
5	0,32589	0,21309	0,26401	0,22400
average	0,29129	0,24462	0,26112	0,22336
stdev	0,02457	0,02289	0,02304	0,02780

Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Pada *Tabel 4.17*, dalam standar *scaler*, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah dari model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil *average* MSE-nya adalah 0,01288 dan 0,01156 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM dengan percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terkecil. Untuk standar deviasi, model Hybrid FEDFormer WEB *with* LSTM juga memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil standar deviasi MSE-nya adalah 0,00228 dan 0,00241 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM dengan percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.17 Hasil MSE Skenario 2 *Temperature* Standar *Scaler*

Temperatur e	MSE			
	Semua fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	0,01575	0,01185	0,01268	0,01215
2	0,01945	0,01603	0,01380	0,01422
3	0,01598	0,01316	0,02005	0,00776
4	0,01578	0,01354	0,01301	0,01260
5	0,02152	0,00983	0,01591	0,01105
average	0,01770	0,01288	0,01509	0,01156
stdev	0,00265	0,00228	0,00304	0,00241

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.18*, dalam skala asli, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah daripada model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,09846 dan 0,08833 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM dengan percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terkecil. Selain itu, untuk standar deviasi, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur. Hasilnya adalah 0,01742 dan 0,01839 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah.

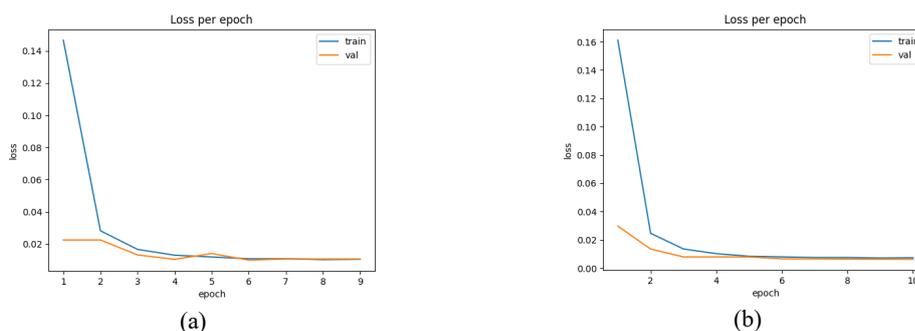
Tabel 4.18 Hasil MSE Skenario 2 *Temperature* Skala Asli

Temperatur e	MSE			
	Semua Fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	0,12039	0,09058	0,09695	0,09286
2	0,14864	0,12250	0,10546	0,10867
3	0,12213	0,10060	0,15322	0,05935
4	0,12058	0,10349	0,09946	0,09631
5	0,16452	0,07514	0,12161	0,08444
average	0,13525	0,09846	0,11534	0,08833
stdev	0,02027	0,01742	0,02325	0,01839

Nilai terendah dari average/standar deviasi

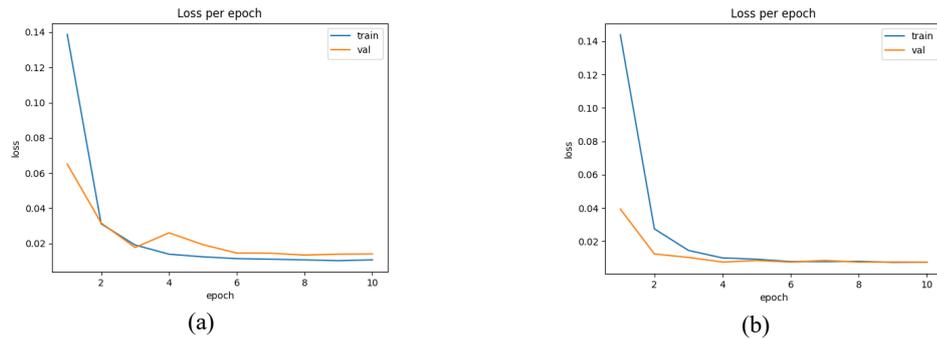
Berdasarkan *Tabel 4.16*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk Hybrid FEDFormer FEB *with LSTM* pada semua fitur mengambil percobaan keempat. Untuk Hybrid FEDFormer WEB *with LSTM* pada semua fitur mengambil percobaan kelima. Untuk Hybrid FEDFormer FEB *with LSTM* pada *selected* fitur mengambil percobaan pertama dan untuk Hybrid FEDFormer WEB *with LSTM* pada *selected* fitur mengambil percobaan ketiga.

Pada *Gambar 4.19*, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk model Hybrid FEDFormer FEB *with LSTM* dan model Hybrid FEDFormer WEB *with LSTM*. Pada model Hybrid FEDFormer FEB *with LSTM*, *train loss* terlihat menurun dari 0,14 hingga sekitar 0,015. *Validation loss* juga menurun pada *epoch* ke-1 hingga ketiga lalu menjadi stabil di nilai mendekati *train loss*. Untuk model Hybrid FEDFormer WEB *with LSTM*, *train loss* terjadi penurunan dan menjadi stabil dari 0,16 hingga sekitar 0,01. *Validation loss* juga mengikuti *train loss*.



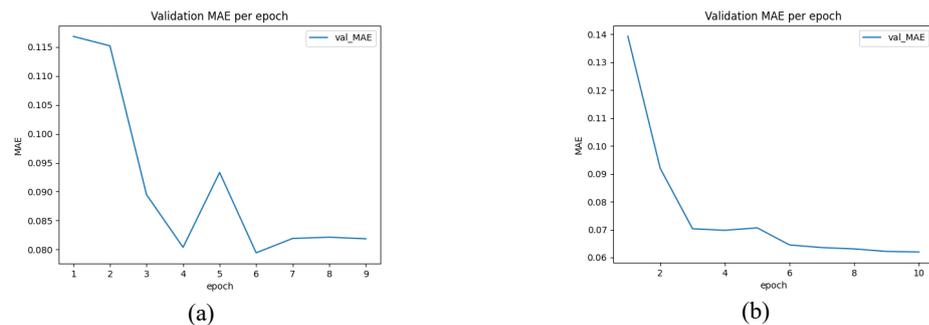
Gambar 4.19 Grafik *Validation loss* *Temperature Selected Fitur* (a) Hybrid FEDFormer FEB *with LSTM* (b) Hybrid FEDFormer WEB *with LSTM*

Pada *Gambar 4.20*, terdapat grafik *validation loss* pada percobaan menggunakan semua fitur. Pada model Hybrid FEDFormer FEB *with LSTM*, *train loss* menurun dari sekitar 0,14 hingga melandai sekitar 0,01. Untuk *validation loss* sempat menaik dari *epoch* ke-3 hingga ke-4 kemudian turun dan stabil kembali. Pada model Hybrid FEDFormer WEB *with LSTM*, terlihat adanya pola konvergensi yang konsisten. *Train loss* dan *validation loss* mengalami penurunan drastis hingga *epoch* ke-4 kemudian stabil dibawah 0,02.



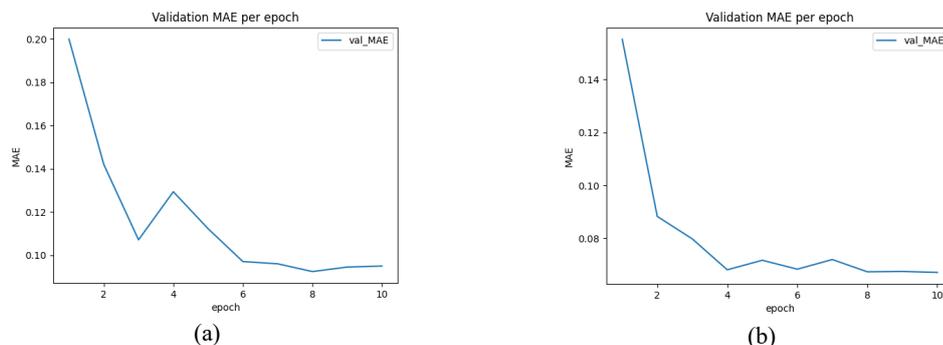
Gambar 4.20 Grafik Validation loss Temperature Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.21, terdapat grafik *validation* MAE pada percobaan menggunakan *selected* fitur untuk model Hybrid FEDFormer FEB *with* LSTM dan Hybrid FEDFormer WEB *with* LSTM. Pada model Hybrid FEDFormer FEB *with* LSTM terlihat penurunan MAE pada *epoch* ke-1 hingga ke-4 dari 0,115 hingga sekitar 0,080. Kemudian, terjadi fluktuasi pada *epoch* ke-4 hingga ke-6 sebelum kembali stabil di bawah 0,085 dari *epoch* ke 7 hingga *epoch* terakhir yaitu *epoch* ke-8. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat penurunan MAE dari awal sebesar 0,14 hingga mencapai sekitar 0,06 pada *epoch* ke-10.



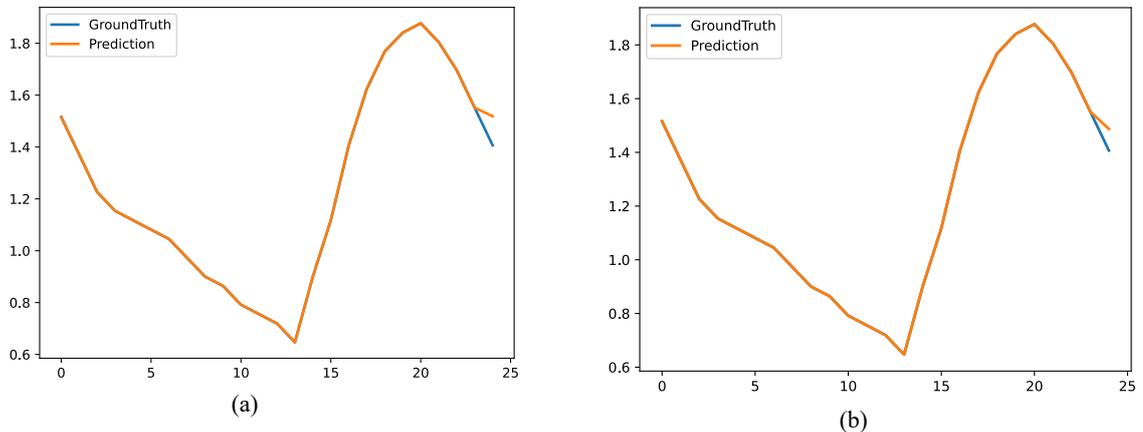
Gambar 4.21 Grafik Validation MAE Temperature Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.22, percobaan semua fitur dilakukan untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa *epoch* ke-1 hingga 3 mengalami penurunan dari 0,20 hingga 0,11. Akan tetapi terjadi sedikit kenaikan dari *epoch* ke-3 hingga *epoch* ke-4 kemudian kembali turun dan stabil hingga *epoch* ke-10 dengan nilai sekitar 0,09. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat penurunan MAE secara konsisten dari *epoch* ke-1 hingga 10 dari sekitar 0,15 hingga sekitar 0,075. Terjadi sedikit fluktuasi antara *epoch* ke-4 hingga 8 dan kemudian menurun di *epoch* ke-8 dan stabil hingga akhir.



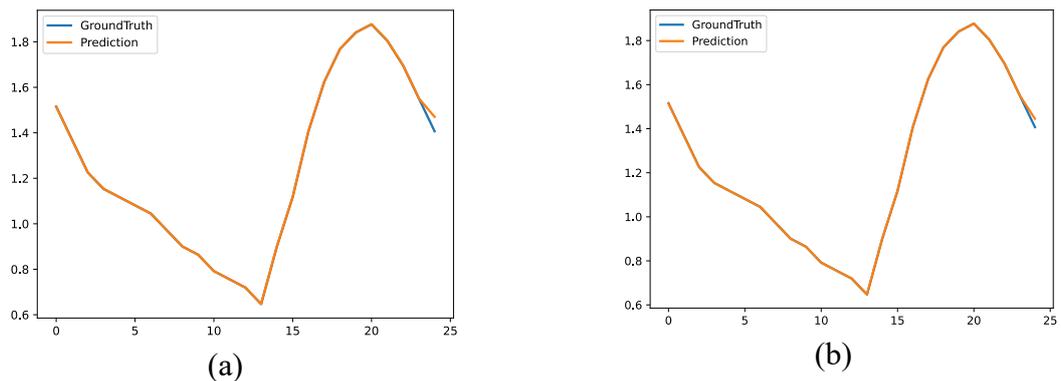
Gambar 4.22 Grafik Validation MAE Temperature Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.23, terdapat kurva prediksi pada percobaan menggunakan *selected* fitur untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, kurva *prediction* salah melakukan *predict* di akhir *timestep* sehingga terlihat perbedaan yang cukup besar pada *timestep* akhir antara kurva *predict* dengan kurva *ground truth*. Pada model Hybrid FEDFormer WEB *with* LSTM juga terlihat perbedaan pada akhir *timestep* antara kurva *predict* dengan kurva *ground truth*.



Gambar 4.23 Grafik Prediksi Temperature Selected Fitur (a) Hybrid FEDFormer FEB *with* LSTM (b) Hybrid FEDFormer WEB *with* LSTM

Pada Gambar 4.24, terdapat kurva prediksi pada percobaan menggunakan semua fitur untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, terlihat sedikit perbedaan antara kurva *predict* dan kurva *ground truth* pada *timestep* akhir. Hal itu juga terlihat pada model Hybrid FEDFormer WEB *with* LSTM, dimana kurva *predict* memiliki sedikit perbedaan dengan kurva *ground truth*.



Gambar 4.24 Grafik Prediksi Temperatur Semua Fitur (a) Hybrid FEDFormer FEB *with* LSTM (b) Hybrid FEDFormer WEB *with* LSTM

b. Prediksi untuk GHI

Pada prediksi untuk target GHI, dilakukan percobaan untuk mendapatkan hasil evaluasi MAE pada model Hybrid FEDFormer FEB *with* LSTM dan Hybrid FEDFormer WEB *with* LSTM. Pada Tabel 4.19, dalam standar *scaler*, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah daripada model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,24249 dan 0,21851 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid

FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Selain itu, untuk standar deviasi, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00343 dan 0,00456 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.19 Hasil MAE Skenario 2 GHI Standar Scaler

GHI	MAE			
	Semua Fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	0,24679	0,24526	0,23045	0,21749
2	0,24133	0,23737	0,21979	0,21863
3	0,25287	0,24270	0,22896	0,22095
4	0,25244	0,24125	0,21612	0,22388
5	0,25488	0,24589	0,24142	0,21163
average	0,24966	0,24249	0,22735	0,21851
stdev	0,00554	0,00343	0,00992	0,00456

Nilai terendah dari average/standar deviasi

Pada Tabel 4.20, dalam skala asli, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 62,28274 dan 56,12330 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM dengan percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,88086 dan 1,17108 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM menggunakan percobaan dengan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.20 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.20 Hasil MAE Skenario 2 GHI Skala Asli

GHI	MAE			
	Semua Fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	63,38679	62,99392	59,18806	55,85972
2	61,98481	60,96682	56,45244	56,15254
3	64,94755	62,33643	58,80717	56,74877
4	64,83642	61,96182	55,50826	57,50058
5	65,46290	63,15471	62,00682	54,35489
average	64,12370	62,28274	58,39255	56,12330
stdev	1,42285	0,88086	2,54710	1,17108

Nilai terendah dari average/standar deviasi

Nilai sampel terendah dari lima percobaan

Pada Tabel 4.21, dalam standar *scaler*, didapatkan hasil evaluasi menggunakan MSE. Dapat terlihat bahwa pada percobaan menggunakan semua fitur, model Hybrid FEDFormer FEB *with* LSTM memiliki nilai *average* MSE terendah yaitu sebesar 0,16751 yang terlihat pada

warna kuning, sedangkan pada percobaan menggunakan *selected* fitur, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah yaitu sebesar 0,15240 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah.

Selain itu, untuk standar deviasi, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik dalam percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00319 dan 0,00211 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi terendah.

Tabel 4.21 Hasil MSE Skenario 2 GHI Standar Scaler

GHI	MSE			
	Semua Fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	0,16917	0,17113	0,15273	0,15468
2	0,16206	0,16322	0,15338	0,15362
3	0,16799	0,16671	0,15734	0,15326
4	0,17063	0,16732	0,15086	0,15086
5	0,16769	0,17050	0,16156	0,14957
average	0,16751	0,16777	0,15517	0,15240
stdev	0,00326	0,00319	0,00428	0,00211

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.22*, dalam skala asli, untuk percobaan menggunakan semua fitur maka model Hybrid FEDFormer FEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 11050,13683 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 10053,324 yang terlihat pada warna kuning. Dari kedua nilai tersebut, percobaan menggunakan *selected* fitur pada model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah.

Untuk standar deviasi, dilakukan percobaan juga untuk semua fitur dan juga dengan *selected* fitur. Untuk percobaan menggunakan semua fitur, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MSE terendah dibanding model lainnya yaitu sebesar 210,70287 yang terlihat pada warna kuning. Hal itu juga sama untuk percobaan dengan *selected* fitur karena model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MSE terendah yaitu sebesar 139,22278 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM menggunakan percobaan *selected* fitur memiliki nilai standar deviasi MSE terendah.

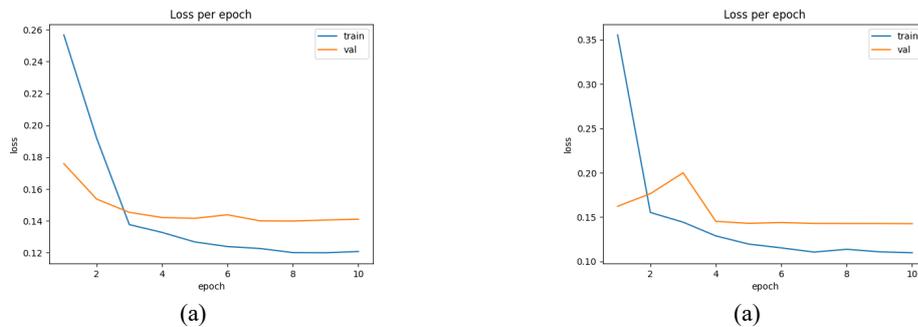
Tabel 4.22 Hasil MSE Skenario 2 GHI Skala Asli

GHI	MSE			
	Semua Fitur		Selected Fitur	
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM
1	11159,70864	11288,90380	10075,45874	10203,61859
2	10690,60529	10767,05891	10118,02911	10134,03266
3	11081,84626	10997,14356	10379,48074	10110,43554
4	11256,36144	11037,60226	9951,70152	9952,06136
5	11062,16251	11247,65145	10657,75824	9866,47185
average	11050,13683	11067,67200	10236,48567	10053,32400
stdev	215,02767	210,70287	282,39237	139,22278

Nilai terendah dari average/standar deviasi

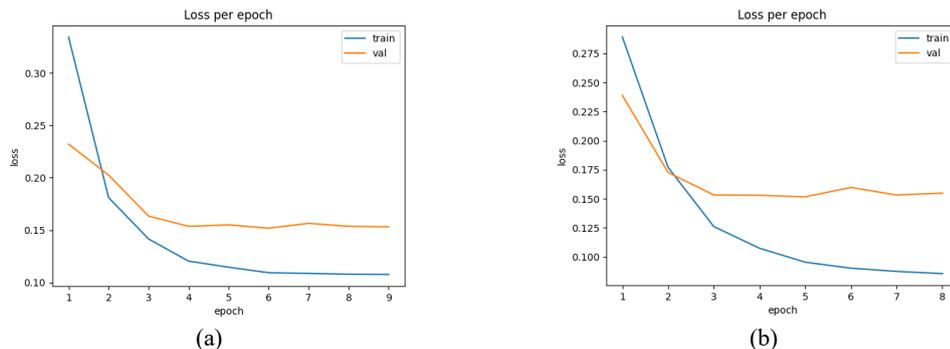
Berdasarkan Tabel 4.20, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk Hybrid FEDFormer FEB *with* LSTM pada semua fitur mengambil percobaan kedua. Untuk Hybrid FEDFormer WEB *with* LSTM pada semua fitur mengambil percobaan kedua. Untuk Hybrid FEDFormer FEB *with* LSTM pada *selected* fitur mengambil percobaan keempat dan untuk Hybrid FEDFormer WEB *with* LSTM pada *selected* fitur mengambil percobaan kelima.

Pada Gambar 4.25, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, untuk *train loss* terlihat mengalami penurunan dari *epoch* ke-1 hingga *epoch* ke-3 sebelum kemudian melandai menurun hingga *epoch* ke-10. Perubahan nilai MAE nya dari sekitar 0,26 hingga 0,13. Untuk *validation loss*, terlihat bahwa terjadi penurunan dari *epoch* pertama hingga ke-5, kemudian terjadi cenderung hingga akhir. Pada model Hybrid FEDFormer WEB *with* LSTM, untuk *train loss* terlihat selalu mengalami penurunan di sepanjang *epoch* dan nilainya dari sekitar 0,35 hingga hampir mendekati 0,10. Untuk *validation loss*, mengalami lonjakan naik pada *epoch* ke-2 hingga ke-3 kemudian mengalami penurunan dan stabil hingga akhir *epoch*.



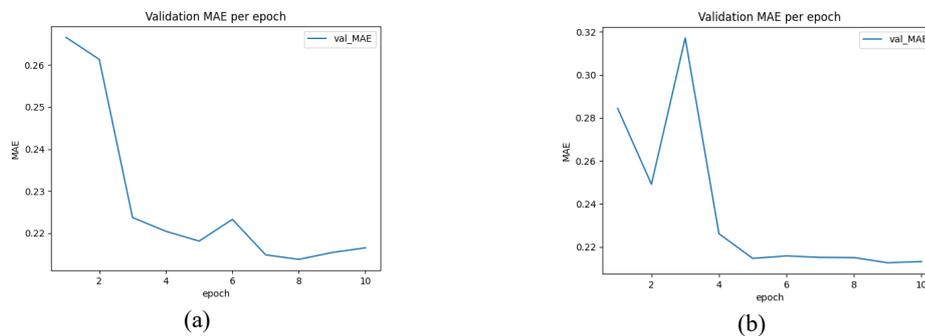
Gambar 4.25 Grafik Validation loss GHI Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.26, terdapat kurva grafik *validation loss* pada percobaan dengan semua fitur untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, *train loss* terlihat mengalami penurunan dari sekitar 0,31 hingga sekitar 0,12. Selain itu, pada *validation loss*, juga mengalami penurunan dari sekitar 0,23 hingga sekitar 0,17 dari *epoch* ke-1 hingga 5 kemudian mengalami kestabilan hingga *epoch* ke-9. Pada model Hybrid FEDFormer WEB *with* LSTM, *train loss* mengalami penurunan dari *epoch* ke-1 hingga akhir dan penurunan dari nilai sekitar 0,280 hingga sekitar 0,050. Untuk *validation loss* juga mengalami penurunan dari *epoch* ke-1 hingga *epoch* ke-3 yang memiliki nilai dari sekitar 0,230 hingga sekitar 0,160. Kemudian, *validation loss* mengalami kestabilan meski pada *epoch* ke-5 dan 6 mengalami sedikit kenaikan sebelum menurun dan stabil lagi hingga akhir *epoch*.



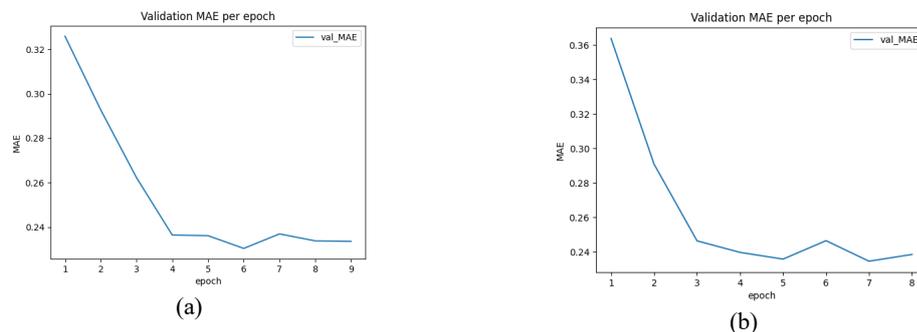
Gambar 4.26 Grafik Validation loss GHI Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.27, terdapat grafik *validation* MAE pada percobaan menggunakan *selected* fitur pada kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa kurva mengalami penurunan dari *epoch* ke-1 hingga *epoch* ke-5 kemudian mengalami sedikit kenaikan pada *epoch* ke-4 hingga 6 lalu turun lagi dan mengalami sedikit kenaikan pada *epoch* ke-8 hingga akhir dengan nilai sekitar 0,21. Pada model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa kurva mengalami fluktuasi dari *epoch* pertama hingga *epoch* ke-4. Kemudian, pada *epoch* ke-4 mulai mengalami penurunan dan stabil hingga *epoch* terakhir dengan nilai sekitar 0,21.



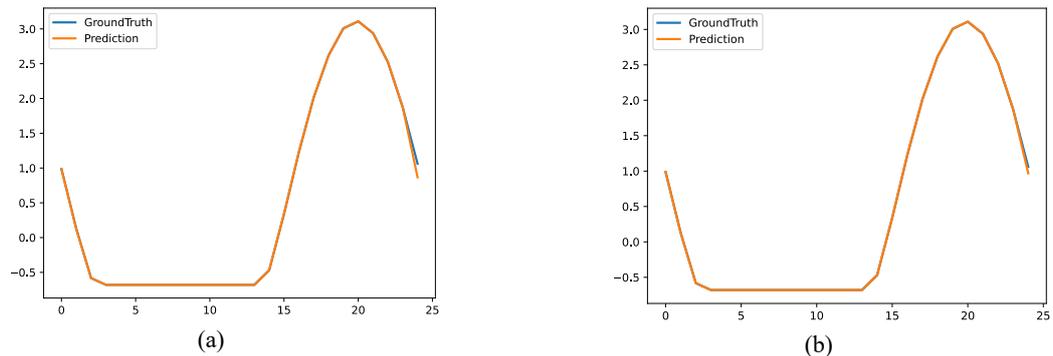
Gambar 4.27 Grafik Validation MAE GHI Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.28, terdapat grafik *validation* MAE pada percobaan dengan semua fitur untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa kurva menurun dari *epoch* ke-1 hingga *epoch* ke-6 lalu mengalami sedikit kenaikan dari *epoch* ke-6 hingga 7 sebelum mengalami penurunan hingga *epoch* terakhir (yaitu dari nilai sekitar 0,33 menjadi nilai dibawah 0,24). Pada model Hybrid FEDFormer WEB *with* LSTM, terlihat kurva menurun dari *epoch* pertama hingga *epoch* ke- 5. Lalu mengalami kenaikan dari *epoch* ke-5 hingga ke-6 sebelum akhirnya menurun dan stabil hingga *epoch* ke-8. Perubahan nilainya dari 0,36 menjadi nilai sekitar 0,24.



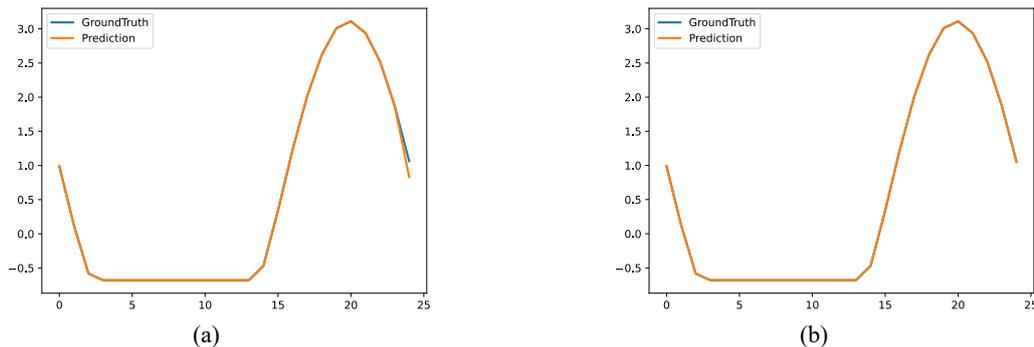
Gambar 4.28 Grafik Validation MAE GHI Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.29, terdapat grafik prediksi dengan percobaan menggunakan *selected* fitur pada kedua model. Untuk model Hybrid FEDFormer FEB *with* LSTM, pada *timestep* akhir, terlihat bahwa kurva *predict* dan kurva *ground truth* memiliki jarak. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat perbedaan kurva yang sangat tipis antara kurva *predict* dan *ground truth* pada akhir *timestep*.



Gambar 4.29 Grafik Prediksi GHI Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.30, terdapat grafik prediksi untuk percobaan menggunakan semua fitur. Pada model Hybrid FEDFormer FEB *with* LSTM, di *timestep* akhir juga memiliki perbedaan yang terlihat antara kurva *prediction* dengan kurva *ground truth*. Sementara itu, untuk model Hybrid FEDFormer WEB *with* LSTM, kurva terlihat tampak sempurna. Diantara kurva *predict* dan kurva *ground truth* tidak terlihat adanya perbedaan yang menunjukkan percobaan menggunakan semua fitur untuk model Hybrid FEDFormer WEB *with* LSTM memiliki hasil yang cukup akurat.



Gambar 4.30 Grafik Prediksi GHI Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

c. Prediksi untuk *demand* listrik

Pada percobaan untuk prediksi nilai *demand* listrik (KW), didapatkan hasil evaluasi MAE dan MSE. Pada Tabel 4.23, dilakukan percobaan untuk mendapatkan nilai MAE dari model Hybrid FEDFormer FEB *with* LSTM dan Hybrid FEDFormer WEB *with* LSTM dalam standar *scaler*. Dari tabel, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,17769 dan 0,16423 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Untuk standar deviasi, pada percobaan menggunakan semua fitur didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00611 yang terlihat pada warna kuning. Selain itu, pada percobaan dengan *selected* fitur didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00418 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB

with LSTM pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.23 Hasil MAE Skenario 2 Demand Listrik Standar Scaler

KW	MAE				<div style="border: 1px solid black; padding: 2px; width: fit-content;"> Nilai terendah dari average/standar deviasi </div>
	Semua Fitur		Selected Fitur		
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	
1	0,19123	0,18647	0,18830	0,16434	
2	0,20751	0,17906	0,17939	0,17036	
3	0,19670	0,18366	0,17780	0,16355	
4	0,19754	0,17615	0,16589	0,15858	
5	0,19445	0,16310	0,17734	0,16432	
average	0,19749	0,17769	0,17774	0,16423	
stdev	0,00611	0,00908	0,00798	0,00418	

Pada Tabel 4.24, dalam skala asli, didapatkan hasil evaluasi MAE pada skala asli. Model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 22,03871 dan 20,36934 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi terendah dibanding model lainnya yaitu sebesar 0,75827 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,51885 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan dengan *selected* fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.24 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

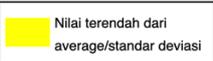
Tabel 4.24 Hasil MAE Skenario 2 Demand Listrik Skala Asli

KW	MAE				<div style="border: 1px solid black; padding: 2px; width: fit-content;"> Nilai terendah dari average/standar deviasi Nilai sampel terendah dari lima percobaan </div>
	Semua Fitur		Selected Fitur		
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	
1	23,71824	23,12741	23,35452	20,38278	
2	25,73774	22,20846	22,24920	21,12946	
3	24,39640	22,77954	22,05321	20,28550	
4	24,50124	21,84851	20,57496	19,66812	
5	24,11765	20,22963	21,99516	20,38084	
average	24,49425	22,03871	22,04541	20,36934	
stdev	0,75827	1,12604	0,98996	0,51885	

Pada Tabel 4.25, dalam standar *scaler*, didapatkan nilai MSE untuk masing-masing percobaan. Dari tabel, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya baik dalam percobaan menggunakan

semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,05486 dan 0,04795 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi MSE yang lebih rendah dibanding model lain yaitu sebesar 0,00359. Pada percobaan menggunakan *selected* fitur, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MSE yang lebih rendah dibanding model lainnya yaitu sebesar 0,00193. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MSE yang paling rendah.

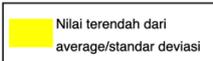
Tabel 4.25 Hasil MSE Skenario 2 Demand Listrik Standar Scaler

KW	MSE				
	Semua Fitur		Selected Fitur		
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	
1	0,06425	0,06083	0,06159	0,04856	
2	0,07289	0,05625	0,05614	0,05031	
3	0,06667	0,05718	0,05603	0,04725	
4	0,06991	0,05329	0,04963	0,04509	
5	0,06513	0,04676	0,05604	0,04855	
average	0,06777	0,05486	0,05588	0,04795	
stdev	0,00359	0,00527	0,00423	0,00193	

Pada *Tabel 4.26*, dalam skala asli, didapatkan nilai MSE untuk masing-masing percobaan. Dari tabel, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 843,97654 dan 737,68716 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah.

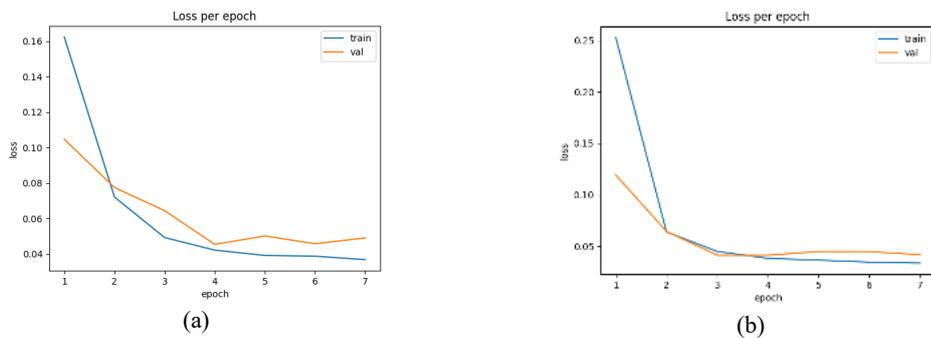
Untuk standar deviasi, pada percobaan dengan semua fitur didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi terendah dibanding model lainnya yaitu sebesar 55,16068 yang terlihat pada warna kuning. Sementara itu, pada percobaan dengan *selected* fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MSE terendah yaitu sebesar 29,76191 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM dalam percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.26 Hasil MSE Skenario 2 Demand Listrik Skala Asli

KW	MSE				
	Semua Fitur		Selected Fitur		
	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	Hybrid FEDFormer FEB with LSTM	Hybrid FEDFormer WEB with LSTM	
1	988,33411	935,83662	947,40966	747,09837	
2	1121,37878	865,30788	863,58404	773,90543	
3	1025,58801	879,63107	861,90876	726,89564	
4	1075,39954	819,78214	763,55557	693,62748	
5	1001,92161	719,32500	862,01981	746,90888	
average	1042,52441	843,97654	859,69557	737,68716	
stdev	55,16068	81,06775	65,11928	29,76191	

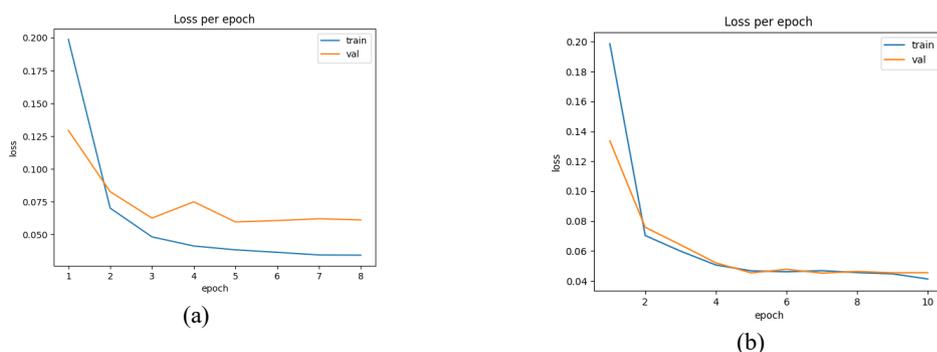
Berdasarkan *Tabel 4.24*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk Hybrid FEDFormer FEB *with* LSTM pada semua fitur mengambil percobaan pertama. Untuk Hybrid FEDFormer WEB *with* LSTM pada semua fitur mengambil percobaan kelima. Untuk Hybrid FEDFormer FEB *with* LSTM pada *selected* fitur mengambil percobaan keempat dan untuk Hybrid FEDFormer WEB *with* LSTM pada *selected* fitur mengambil percobaan keempat.

Pada *Gambar 4.31*, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk kedua model. Pada model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa *training loss* mengalami penurunan secara konsisten dari *epoch* awal hingga akhir yaitu sekitar 0,16 menjadi sekitar 0,04 dan *validation loss* juga mengalami penurunan hingga akhir *epoch* dengan nilai sekitar 0,06. Pada model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa *training loss* selalu menurun dari awal *epoch* hingga akhir *epoch* dimana nilai MAE awalnya sekitar 0,25 menjadi sekitar 0,04 di akhir *epoch*. Untuk *validation loss* sempat menaik pada *epoch* ke-3 hingga ke-5 kemudian menurun dengan nilai sekitar 0,05.



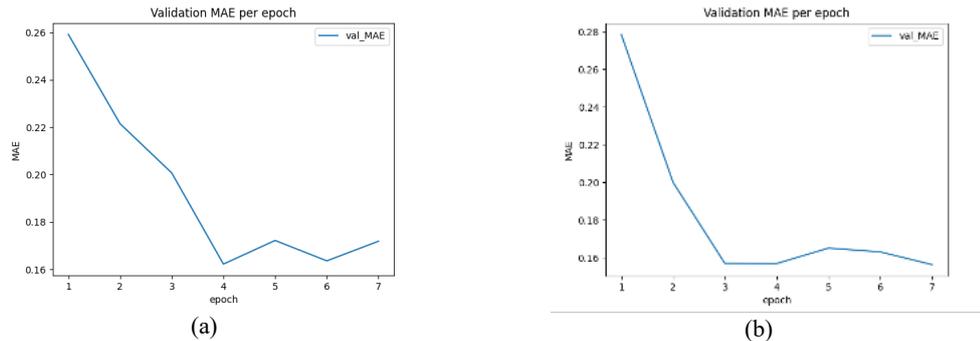
Gambar 4.31 Grafik Validation loss Demand Listrik Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada *Gambar 4.32*, terdapat grafik *validation loss* untuk percobaan menggunakan semua fitur pada kedua model. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa *train loss* mengalami penurunan selama *training* dari *epoch* awal hingga *epoch* akhir hingga mencapai nilai sekitar 0,040. Akan tetapi, untuk *validation loss*, mengalami fluktuasi dari *epoch* ke-3 hingga ke-5 kemudian stabil dengan nilai sekitar 0,075. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa *training loss* dan *validation loss* mengalami penurunan dari awal hingga akhir *training* dan *validation loss* cenderung mengikuti pergerakan *training loss* mulai dari *epoch* ke-2 hingga *epoch* terakhir.



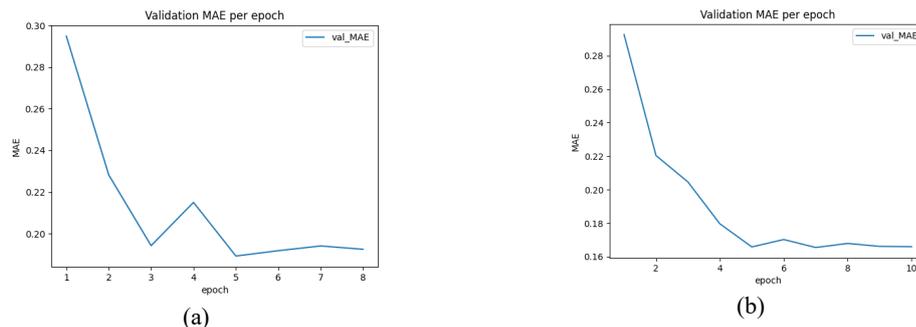
Gambar 4.32 Grafik Validation loss Demand Listrik Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.33, terdapat grafik *validation* MAE pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa terjadi penurunan MAE yang signifikan dari *epoch* ke-1 hingga ke-4 dengan nilai dari sekitar 0,26 menjadi sekitar 0,17. Setelah itu, grafik mengalami sedikit fluktuasi dari *epoch* ke-4 hingga ke-6 kemudian mengalami kenaikan dari *epoch* ke-6 hingga ke-7 dengan nilai sekitar 0,17. Untuk model Hybrid FEDFormer WEB *with* LSTM, dari *epoch* pertama hingga ketiga mengalami penurunan (dari nilai sekitar 0,28 hingga 0,16). Kemudian terjadi sedikit kenaikan pada *epoch* ke-4 hingga ke-5 lalu kembali menurun hingga *epoch* ke-7 dengan nilai sekitar 0,16.



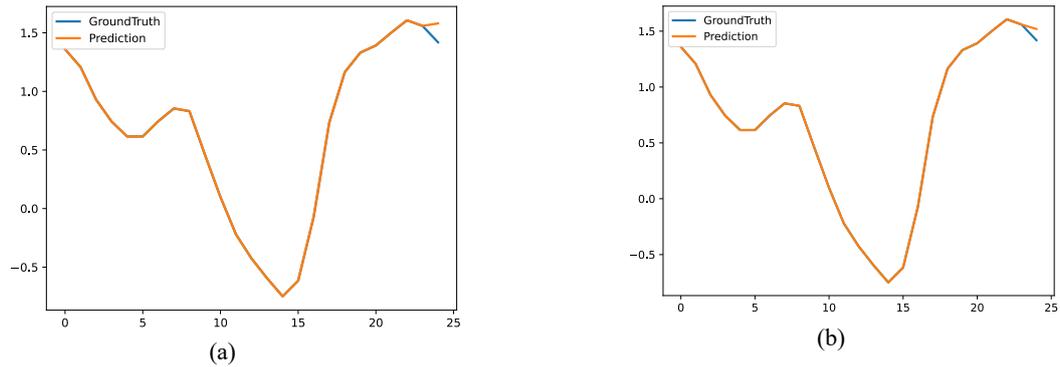
Gambar 4.33 Grafik Validation MAE Demand Listrik Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.34, terdapat grafik *validation* MAE pada percobaan semua fitur untuk kedua model. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa terjadi penurunan nilai MAE yang signifikan dari *epoch* ke-1 hingga ke-3 yaitu dari sekitar nilai 0,29 hingga sekitar nilai 0,19. Kemudian terjadi fluktuasi ringan dari *epoch* ke-3 hingga ke-5 lalu mengalami sedikit kenaikan hingga *epoch* ke-7 dan kembali menurun hingga *epoch* terakhir yaitu *epoch* ke-8 dengan nilai sekitar 0,19. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa kurva menurun dari *epoch* ke-1 hingga ke-5 dari nilai sekitar 0,30 hingga hampir mendekati nilai 0,16. Pada *epoch* ke-5 hingga ke-8 mengalami fluktuasi ringan lalu menjadi stabil hingga *epoch* terakhir yaitu *epoch* ke-10 dengan nilai sekitar 0,16.



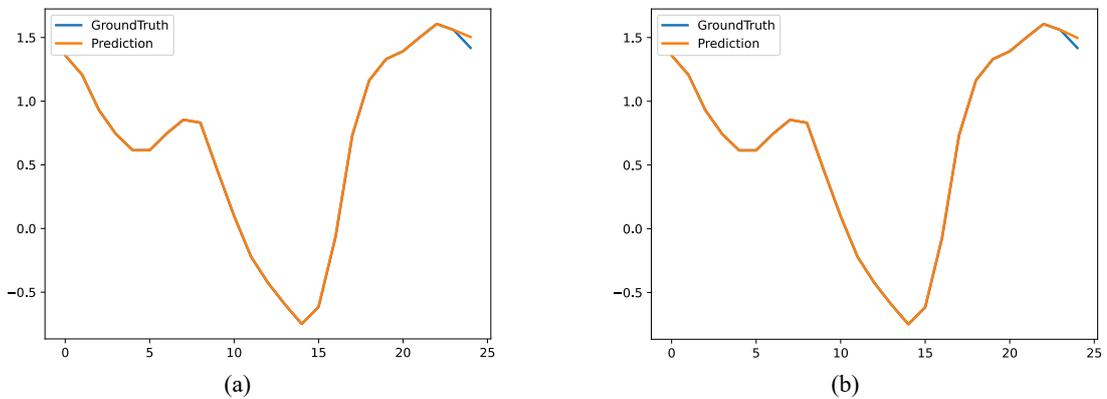
Gambar 4.34 Grafik Validation MAE Demand Listrik Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.35, terdapat kurva prediksi untuk percobaan menggunakan *selected* fitur. Untuk kedua model, terlihat bahwa di *timestep* akhir memiliki perbedaan antara kurva *predict* dengan kurva *ground truth* meski *timestep* sebelumnya menunjukkan pola yang sesuai.



Gambar 4.35 Grafik Prediksi Demand Listrik Selected Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.36, terdapat grafik prediksi untuk percobaan menggunakan semua fitur pada kedua model. Hasilnya adalah kurva prediksi mampu mengikuti kurva *ground truth* dengan baik sebelum akhirnya terdapat perbedaan pada *timestep* akhir.



Gambar 4.36 Grafik Prediksi Demand Listrik Semua Fitur (a) Hybrid FEDFormer FEB with LSTM (b) Hybrid FEDFormer WEB with LSTM

4.3.3 Hasil Implementasi Skenario 3

Pada implementasi ini, dilakukan pengujian hasil evaluasi percobaan dataset dengan model FEDFormer FEB dan model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan semua fitur maupun dengan *selected* fitur menggunakan *correlation matrix*. Percobaan ini dilakukan dengan melakukan prediksi untuk target *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada prediksi *temperature*, hasil evaluasi MAE dan MSE didapatkan untuk masing-masing percobaan yang dilakukan. Pada Tabel 4.27, dilakukan percobaan untuk mendapatkan nilai MAE dalam standar *scaler*. Dari tabel tersebut, terlihat bahwa model FEDFormer FEB memiliki nilai *average* MAE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,07152 dan 0,07023 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai yang lebih rendah.

Untuk standar deviasi, model FEDFormer FEB memiliki nilai standar deviasi MAE terkecil baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00511 dan 0,00309 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.27 Hasil MAE Skenario 3 *Temperature* Standar Scaler

Temperature	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,07035	0,09928	0,06766	0,08689
2	0,06485	0,11161	0,07131	0,09218
3	0,07859	0,10020	0,06920	0,10822
4	0,07394	0,09782	0,06786	0,08946
5	0,06984	0,11788	0,07511	0,09549
average	0,07152	0,10536	0,07023	0,09445
stdev	0,00511	0,00889	0,00309	0,00833

Nilai terendah dari average/standar deviasi

Pada Tabel 4.28, dalam skala asli, model FEDFormer FEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun menggunakan *selected* fitur. Hasilnya adalah 0,19772 dan 0,19415 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, model FEDFormer FEB memiliki standar deviasi terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,01413 dan 0,00855 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.28 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.28 Hasil MAE Skenario 3 *Temperature* Skala Asli

Temperature	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,19451	0,27449	0,18705	0,24022
2	0,17930	0,30857	0,19716	0,25484
3	0,21729	0,27703	0,19130	0,29918
4	0,20443	0,27045	0,18760	0,24734
5	0,19309	0,32589	0,20765	0,26401
average	0,19772	0,29129	0,19415	0,26112
stdev	0,01413	0,02457	0,00855	0,02304

Nilai terendah dari average/standar deviasi

Nilai sampel terendah dari lima percobaan

Pada Tabel 4.29, dalam standar *scaler*, didapatkan bahwa model FEDFormer FEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00835 dan 0,00807 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model FEDFormer FEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00120 dan 0,00063 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada *selected* fitur memiliki nilai standar MSE terendah.

Tabel 4.29 Hasil MSE Skenario 3 *Temperature Standar Scaler*

Temperature	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,00810	0,01575	0,00743	0,01268
2	0,00680	0,01945	0,00838	0,01380
3	0,00976	0,01598	0,00799	0,02005
4	0,00932	0,01578	0,00758	0,01301
5	0,00778	0,02152	0,00898	0,01591
average	0,00835	0,01770	0,00807	0,01509
stdev	0,00120	0,00265	0,00063	0,00304

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.30*, dalam skala asli, model FEDFormer FEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,06383 dan 0,06172 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai terendah. Untuk standar deviasi, model FEDFormer FEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00914 dan 0,00478 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.30 Hasil MSE Skenario 3 *Temperature Skala Asli*

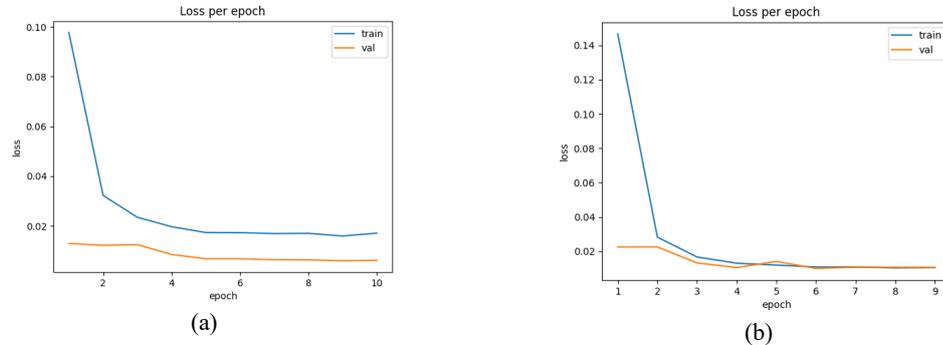
Temperatur e	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,06188	0,12039	0,05683	0,09695
2	0,05198	0,14864	0,06406	0,10546
3	0,07461	0,12213	0,06110	0,15322
4	0,07123	0,12058	0,05797	0,09946
5	0,05948	0,16452	0,06862	0,12161
average	0,06383	0,13525	0,06172	0,11534
stdev	0,00914	0,02027	0,00478	0,02325

Nilai terendah dari average/standar deviasi

Berdasarkan *Tabel 4.28*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk FEDFormer FEB pada percobaan semua fitur menggunakan sampel kedua. Untuk Hybrid FEDFormer FEB *with* LSTM pada percobaan semua fitur menggunakan sampel keempat. Untuk FEDFormer FEB pada *selected* fitur menggunakan percobaan pertama dan untuk Hybrid FEDFormer FEB *with* LSTM pada *selected* fitur menggunakan sampel pertama.

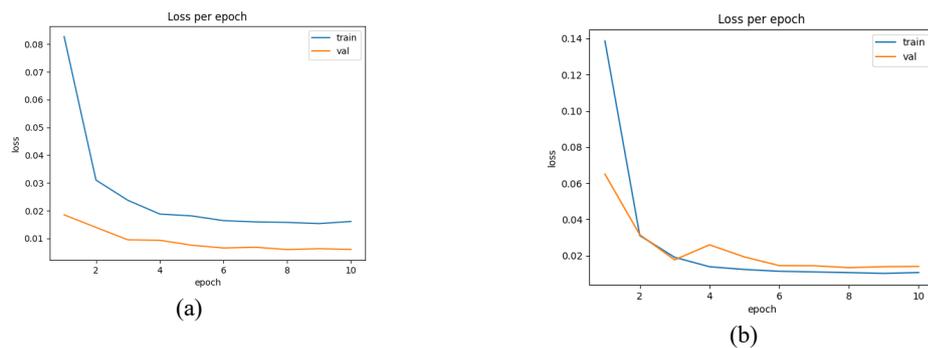
Pada *Gambar 4.37*, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer FEB, terlihat bahwa *train loss* mengalami penurunan pada *epoch* pertama hingga ke-3 kemudian mengalami kestabilan hingga *epoch* terakhir yaitu *epoch* ke-10 dengan nilai sekitar 0,03. Untuk *validation loss* juga awalnya mengalami stabil dari *epoch* ke-1 hingga ke-3, kemudian menurun dan cenderung stabil lagi dari *epoch* ke-4 hingga *epoch* ke-10 dengan nilai sekitar 0,01. Untuk model Hybrid FEDFormer FEB *with* LSTM, *train loss* juga menurun dari *epoch* pertama hingga *epoch* ke-4 kemudian

stabil hingga *epoch* terakhir yaitu *epoch* ke-9 dengan nilai sekitar 0,01. Untuk *validation loss*, mengalami penurunan dari *epoch* ke-2 hingga ke-4 kemudian terjadi sedikit fluktuasi dari *epoch* ke-4 hingga ke-6 lalu stabil hingga *epoch* ke-9 dengan nilai sekitar 0,01. Nilai *validation loss* mengikuti nilai *training loss* pada *epoch* ke-6 hingga *epoch* terakhir.



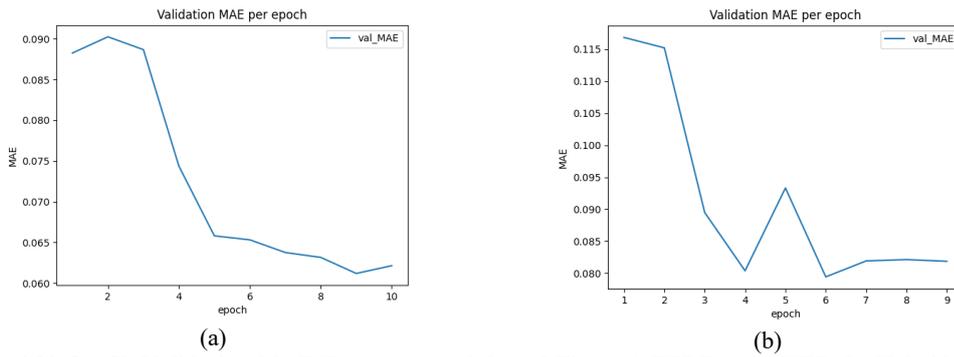
Gambar 4.37 Grafik Validation loss Temperature Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.38, terdapat grafik *validation loss* untuk kedua model. Untuk model FEDFormer FEB, *train loss* terlihat menurun dan cenderung stabil dari *epoch* ke-4 hingga *epoch* ke-10 dengan nilai sekitar 0,025. Untuk *validation loss* juga menurun dari *epoch* ke-1 hingga ke-3 kemudian mengalami sedikit kenaikan lalu menurun lagi dan cenderung stabil dari *epoch* ke-4 hingga *epoch* ke-10 dengan nilai sekitar 0,005. Untuk model Hybrid FEDFormer FEB with LSTM, terlihat bahwa *train loss* mengalami penurunan dari *epoch* pertama hingga *epoch* ke-4 kemudian cenderung stabil hingga *epoch* terakhir yaitu *epoch* ke-10 dengan nilai sekitar 0,02. Untuk *validation loss*, awalnya mengalami penurunan dari *epoch* pertama hingga *epoch* ke-3 dan nilainya sangat mendekati *train loss*. Kemudian mengalami fluktuasi dari *epoch* ke-3 hingga ke-6 lalu cenderung stabil dari *epoch* ke-6 hingga ke-10 dan memiliki sedikit *gap* dengan *train loss*.



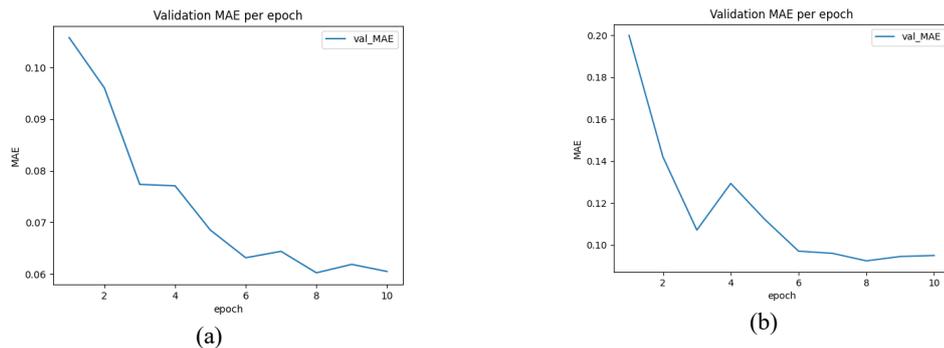
Gambar 4.38 Grafik Validation loss Temperature Semua Fitur (a) FEDFormer FEB (b) FEDFormer with LSTM

Pada Gambar 4.39, terdapat grafik *validation MAE* untuk *selected* fitur pada kedua model. Untuk model FEDFormer FEB, terlihat bahwa tren penurunan MAE cukup konsisten dari awal hingga akhir penelitian. Tidak ada fluktuasi tajam sejak *epoch* ke-1 hingga ke-10. Nilai MAE turun dari sekitar 0,09 menuju sekitar 0,06. Untuk model Hybrid FEDFormer FEB with LSTM, terlihat bahwa MAE awal sangat tinggi yaitu sekitar 0,115 lalu menurun tajam hingga *epoch* ke-4. Kemudian, terdapat fluktuasi antara *epoch* ke-4 hingga ke-6 yang menunjukkan ketidakstabilan saat proses validasi. Setelah *epoch* ke-6, kurva mulai stabil hingga *epoch* terakhir dengan nilai sekitar 0,083.



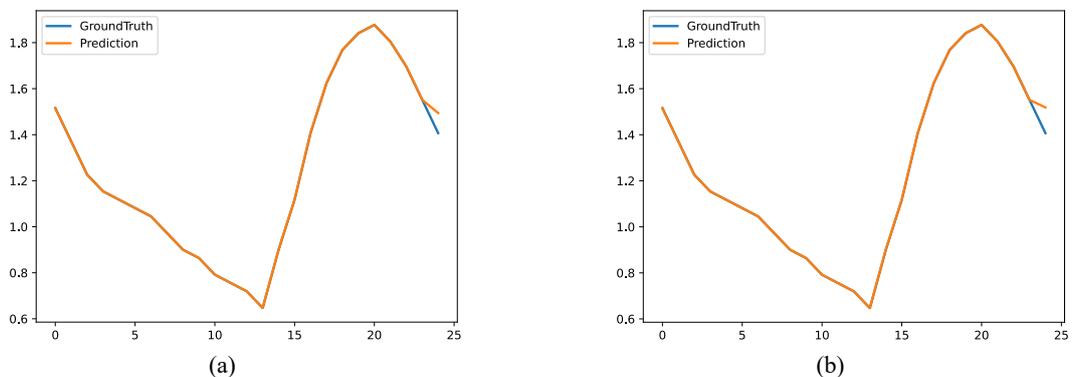
Gambar 4.39 Grafik Validation MAE Temperature Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.40, terdapat grafik *validation* MAE pada percobaan menggunakan semua fitur di kedua model. Untuk model FEDFormer FEB, terlihat bahwa kurva MAE mengalami penurunan cukup tajam hingga *epoch* ke-6. Kemudian kurva mulai mendatar dan mendekati nilai minimum yaitu sekitar nilai 0,06. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa kurva MAE menurun tajam dari nilai 0,20 ke nilai sekitar 0,11 pada *epoch* ke-3. Kemudian, terjadi fluktuasi dari *epoch* ke-3 hingga ke-5 kemudian menurun dan cenderung stabil hingga *epoch* terakhir dengan nilai sekitar 0,10.



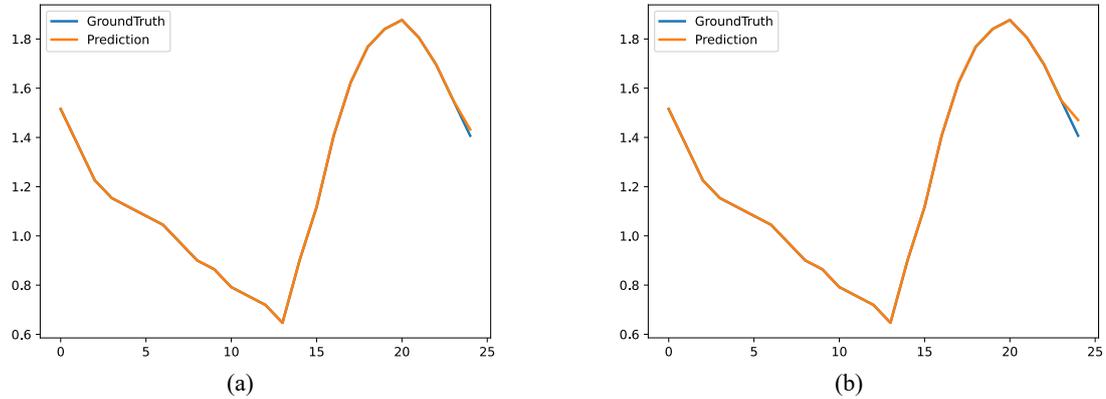
Gambar 4.40 Grafik Validation MAE Temperature Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.41, terdapat grafik prediksi untuk percobaan dengan *selected* fitur pada kedua model. Hasilnya terlihat bahwa kurva *prediction* pada kedua model mampu mengikuti *ground truth* dengan baik pada *timestep* awal namun gagal pada *timestep* terakhir karena terlihat adanya *gap* antara kurva prediksi dengan kurva *ground truth*.



Gambar 4.41 Grafik Prediksi Temperature Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.42, terdapat grafik prediksi untuk percobaan menggunakan semua fitur pada kedua model. Untuk kedua model, kurva *predict* mampu mengikuti kurva *ground truth* dengan baik pada *timestep* awal. Untuk *timestep* akhir, pada model FEDFormer FEB terlihat sedikit sekali *gap*-nya yang menunjukkan bahwa kurva cenderung memiliki hasil prediksi yang akurat dibanding dengan model Hybrid FEDFormer FEB with LSTM karena pada akhir *timestep*, terlihat *gap* yang sedikit besar antara kurva *prediction* dengan kurva *ground truth*.



Gambar 4.42 Gambar Prediksi Temperature Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

b. Prediksi untuk GHI

Pada prediksi untuk GHI, pada Tabel 4.31 yang berupa nilai MAE pada standar *scaler*, didapatkan bahwa model Hybrid FEDFormer FEB with LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,24966 dan 0,22735 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer FEB with LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan dengan semua fitur, didapatkan bahwa model Hybrid FEDFormer FEB with LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00554 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, didapatkan bahwa model FEDFormer FEB memiliki standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00453 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki standar deviasi MAE terendah.

Tabel 4.31 Hasil MAE Skenario 3 GHI Standar *Scaler*

GHI	MAE				<div style="background-color: yellow; border: 1px solid black; padding: 2px; width: fit-content;"> Nilai terendah dari average/standar deviasi </div>
	Semua Fitur		Selected Fitur		
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	
1	0,26480	0,24679	0,25306	0,23045	
2	0,25393	0,24133	0,24699	0,21979	
3	0,25453	0,25287	0,24568	0,22896	
4	0,25290	0,25244	0,24042	0,21612	
5	0,26810	0,25488	0,24775	0,24142	
average	0,25885	0,24966	0,24678	0,22735	
stdev	0,00706	0,00554	0,00453	0,00992	

Pada Tabel 4.32, dalam skala asli, model Hybrid FEDFormer FEB with LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan

semua fitur maupun dengan *selected* fitur. Hasilnya adalah 64,12370 dan 58,39255 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Selain itu, pada standar deviasi, dalam percobaan menggunakan semua fitur didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 1,42285 yang terlihat pada warna kuning. Selain itu, untuk percobaan menggunakan *selected* fitur, model FEDFormer FEB memiliki standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 1,16403 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM pada percobaan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.32 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.32 Hasil MAE Skenario 3 GHI Skala Asli

GHI	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	68,01223	63,38679	64,99726	59,18806
2	65,21964	61,98481	63,43626	56,45244
3	65,37519	64,94755	63,10144	58,80717
4	64,95467	64,83642	61,74897	55,50826
5	68,85921	65,46290	63,63170	62,00682
average	66,48419	64,12370	63,38313	58,39255
stdev	1,81274	1,42285	1,16403	2,54710

Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Pada Tabel 4.33, dalam standar *scaler*, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,16751 dan 0,15517 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model FEDFormer FEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik dalam percobaan menggunakan semua fitur maupun *selected* fitur. Hasilnya adalah 0,00247 dan 0,00199 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.33 Hasil MSE Skenario 3 GHI Standar Scaler

GHI	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,17132	0,16917	0,16738	0,15273
2	0,16820	0,16206	0,16551	0,15338
3	0,16545	0,16799	0,16476	0,15734
4	0,16630	0,17063	0,16188	0,15086
5	0,17008	0,16769	0,16536	0,16156
average	0,16827	0,16751	0,16498	0,15517
stdev	0,00247	0,00326	0,00199	0,00428

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.34*, dalam skala asli, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 11050,13683 dan 10236,48567 yang terlihat pada warna kuning. Dari kedua nilai tersebut, dapat diketahui bahwa model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terkecil. Untuk standar deviasi, didapatkan bahwa model FEDFormer FEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 162,94463 dan 131,22261 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MSE yang paling rendah.

Tabel 4.34 Hasil MSE Skenario 3 GHI Skala Asli

GHI	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	11301,41953	11159,70864	11041,50543	10075,45874
2	11095,67983	10690,60529	10918,49476	10118,02911
3	10914,48306	11081,84626	10868,81746	10379,48074
4	10970,39344	11256,36144	10678,96242	9951,70152
5	11220,00711	11062,16251	10908,72200	10657,75824
average	11100,39659	11050,13683	10883,30041	10236,48567
stdev	162,94463	215,02767	131,22261	282,39237

Nilai terendah dari average/standar deviasi

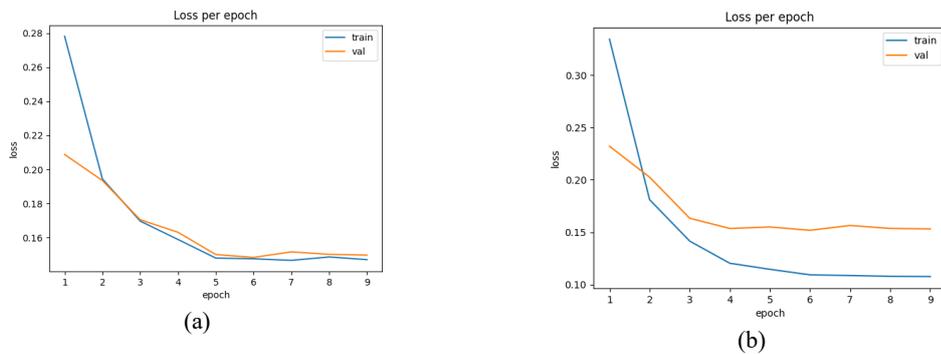
Berdasarkan *Tabel 4.32*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk FEDFormer FEB pada percobaan semua fitur menggunakan sampel keempat. Untuk Hybrid FEDFormer FEB *with* LSTM pada percobaan semua fitur menggunakan sampel kedua. Untuk FEDFormer FEB pada *selected* fitur menggunakan percobaan keempat dan untuk Hybrid FEDFormer FEB *with* LSTM pada *selected* fitur menggunakan sampel keempat.

Pada *Gambar 4.43*, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer FEB, terlihat *training loss* mengalami penurunan selama *training* dari *epoch* pertama hingga terakhir dengan nilai sekitar 0,16. Untuk *validation loss* juga mengalami penurunan dari *epoch* pertama hingga *epoch* ke-4 kemudian mengalami kenaikan dan cenderung stabil setelah *epoch* ke-5 hingga *epoch* terakhir yaitu 7 dengan nilai sekitar 0,15. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa *train loss* mengalami penurunan dari awal *epoch* hingga akhir dengan nilai sekitar 0,13. Untuk *validation loss*, mengalami penurunan dari *epoch* pertama hingga *epoch* ke-3 lalu cenderung stabil hingga *epoch* terakhir yaitu *epoch* ke-10.



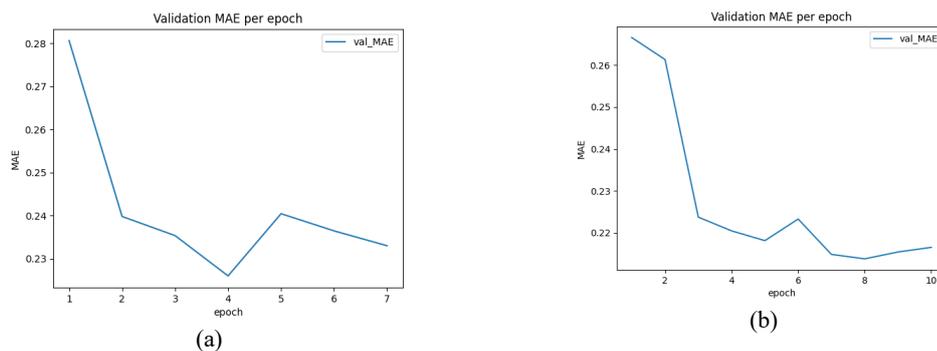
Gambar 4.43 Grafik Validation loss GHI Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.44, terdapat grafik *validation loss* pada percobaan menggunakan semua fitur untuk kedua model. Untuk FEDFormer FEB, terlihat bahwa *training loss* dan *validation loss* cukup dekat dan sejajar mulai dari *epoch* ke-4 hingga ke-9 dengan nilai sekitar 0,15. Kemudian, tidak terdapat fluktuasi dan kurva tampak stabil. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa *training loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch* dengan nilai sekitar 0,11. Untuk *validation loss*, mengalami penurunan dari *epoch* ke-1 hingga ke-4 kemudian mengalami kestabilan hingga akhir *epoch* dengan nilai sekitar 0,14.



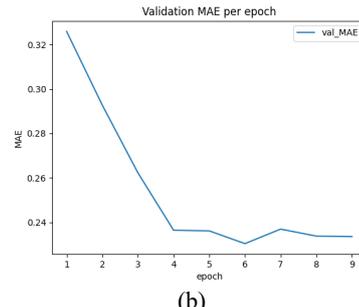
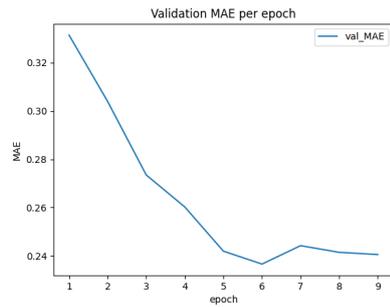
Gambar 4.44 Grafik Validation loss GHI Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.45, terdapat grafik *validation MAE* pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer FEB, terdapat penurunan *validation MAE* dari *epoch* pertama ke *epoch* ke-4. Kemudian terdapat sedikit fluktuasi dari *epoch* ke-4 hingga ke-5 lalu *epoch* selanjutnya mengalami penurunan dengan nilai sekitar 0,23. Untuk model Hybrid FEDFormer FEB *with* LSTM, terlihat bahwa *validation MAE* mengalami penurunan dari *epoch* pertama ke *epoch* ke-5. Lalu terdapat sedikit fluktuasi dari *epoch* ke-5 hingga ke-7. Selanjutnya mengalami sedikit kenaikan dari *epoch* ke-8 hingga akhir dengan nilai sekitar 0,21.



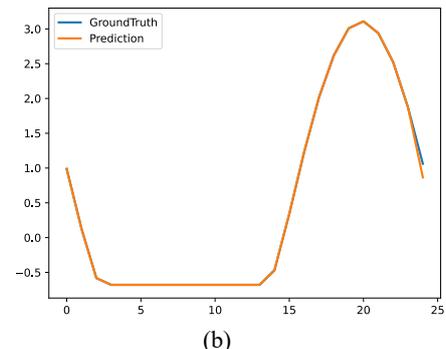
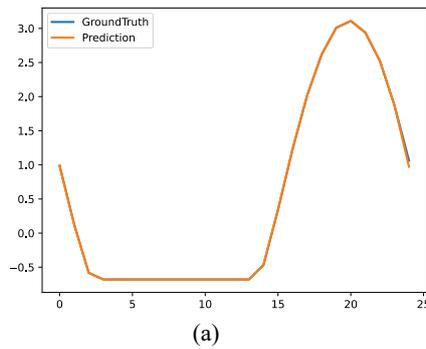
Gambar 4.45 Grafik Validation MAE GHI Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.46, terdapat grafik *validation MAE* pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer FEB, terlihat bahwa nilai *validation MAE* mengalami penurunan dari *epoch* pertama hingga *epoch* ke-6 kemudian mengalami sedikit fluktuasi dari *epoch* ke-6 hingga ke-7 sebelum akhirnya kembali menurun dan stabil dengan nilai sekitar 0,24. Untuk model Hybrid FEDFormer FEB *with* LSTM, terjadi penurunan *validation MAE* dari *epoch* pertama hingga *epoch* ke-6. Kemudian terjadi sedikit fluktuasi dari *epoch* ke-6 hingga *epoch* ke-8 lalu kembali menurun dan stabil hingga *epoch* terakhir dengan nilai sekitar 0,23.



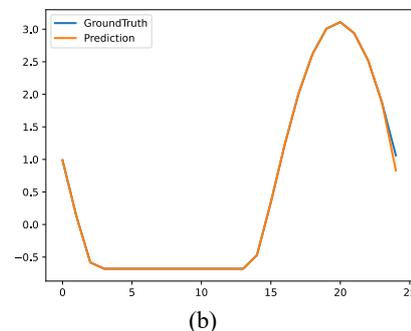
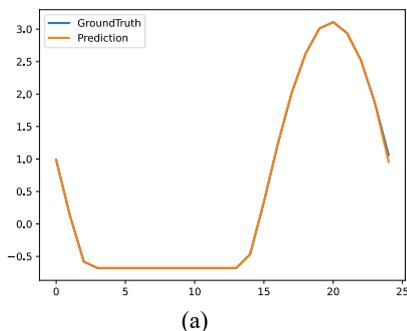
(a) (b)
Gambar 4.46 Grafik Validation MAE GHI Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.47, terdapat grafik prediksi pada percobaan *selected* fitur untuk kedua model. Dari kedua model tersebut, kurva prediksi mampu mengikuti pola *ground truth* pada *timestep* awal. Pada *timestep* akhir, terjadi perbedaan antara kedua kurva *predict* dan *ground truth* namun yang terlihat lebih sedikit *gap*nya terdapat pada model FEDFormer FEB.



(a) (b)
Gambar 4.47 Grafik Prediksi GHI Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada Gambar 4.48, terdapat grafik prediksi pada percobaan dengan semua fitur untuk kedua model. Dari kedua model tersebut, kurva *predict* mampu mengikuti nilai *ground truth* dengan baik hingga sebelum *timestep* akhir. Kemudian, terdapat *gap* antara kurva *predict* dan kurva *ground truth* pada *timestep* akhir. Meski demikian, model FEDFormer FEB memiliki *gap* yang lebih kecil daripada Hybrid FEDFormer FEB *with* LSTM.



(a) (b)
Gambar 4.48 Grafik Prediksi GHI Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

c. Prediksi untuk *demand* listrik (KW)

Pada percobaan untuk prediksi *demand* listrik (KW), pada Tabel 4.35 yang merupakan hasil percobaan dengan standar *scaler*, didapatkan hasil bahwa model FEDFormer FEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan

semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,17979 dan 0,17211 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada *selected* fitur memiliki nilai *average* MAE paling rendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00611 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, model FEDFormer FEB memiliki nilai standar deviasi terendah dibanding model lainnya yaitu sebesar 0,00761 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.35 Hasil MAE Skenario 3 Demand Listrik Standar Scaler

KW	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,18190	0,19123	0,16368	0,18830
2	0,19047	0,20751	0,17887	0,17939
3	0,16578	0,19670	0,17197	0,17780
4	0,17196	0,19754	0,16550	0,16589
5	0,18882	0,19445	0,18054	0,17734
average	0,17979	0,19749	0,17211	0,17774
stdev	0,01070	0,00611	0,00761	0,00798

Nilai terendah dari average/standar deviasi

Pada Tabel 4.36, dalam skala asli, didapatkan hasil bahwa model FEDFormer FEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 22,29890 dan 21,34696 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,75827 yang terlihat pada warna kuning. Sementara itu, pada percobaan *selected* fitur, didapatkan bahwa model FEDFormer FEB memiliki nilai standar deviasi terendah dibanding model lainnya yaitu sebesar 0,94391 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer FEB *with* LSTM pada percobaan dengan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.36 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.36 Hasil MAE Skenario 3 Demand Listrik Skala Asli

KW	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	22,56146	23,71824	20,30089	23,35452
2	23,62449	25,73774	22,18556	22,24920
3	20,56164	24,39640	21,32935	22,05321
4	21,32801	24,50124	20,52657	20,57496
5	23,41891	24,11765	22,39243	21,99516
average	22,29890	24,49425	21,34696	22,04541
stdev	1,32677	0,75827	0,94391	0,98996

Nilai terendah dari average/standar deviasi

Nilai sampel terendah dari lima percobaan

Pada *Tabel 4.37*, dalam standar *scaler*, didapatkan bahwa model FEDFormer FEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,05449 dan 0,05201 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer FEB pada percobaan dengan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model Hybrid FEDFormer FEB *with LSTM* memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00359 dan 0,00423 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer FEB *with LSTM* pada percobaan dengan semua fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.37 Hasil MSE Skenario 3 Demand Listrik Standar Scaler

KW	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	0,05623	0,06425	0,04742	0,06159
2	0,05783	0,07289	0,05552	0,05614
3	0,04685	0,06667	0,05056	0,05603
4	0,05111	0,06991	0,04860	0,04963
5	0,06044	0,06513	0,05796	0,05604
average	0,05449	0,06777	0,05201	0,05588
stdev	0,00546	0,00359	0,00454	0,00423

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.38*, dalam skala asli, didapatkan bahwa model FEDFormer FEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 838,27672 dan 800,11366 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer FEB pada *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model Hybrid FEDFormer FEB *with LSTM* memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 55,16068 dan 65,11928 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer FEB *with LSTM* pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.38 Hasil MSE Skenario 3 Demand Listrik Skala Asli

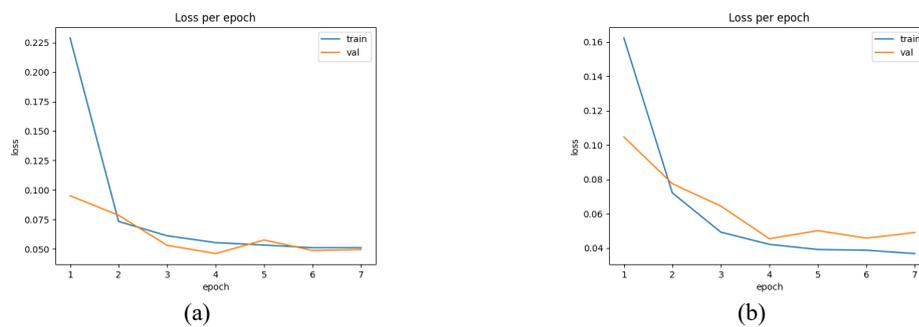
KW	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM
1	864,95705	988,33411	729,48152	947,40966
2	889,70471	1121,37878	854,09893	863,58404
3	720,71770	1025,58801	777,80976	861,90876
4	786,29849	1075,39954	747,59435	763,55557
5	929,70566	1001,92161	891,58373	862,01981
average	838,27672	1042,52441	800,11366	859,69557
stdev	84,02052	55,16068	69,86742	65,11928

Nilai terendah dari average/standar deviasi

Berdasarkan *Tabel 4.36*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang

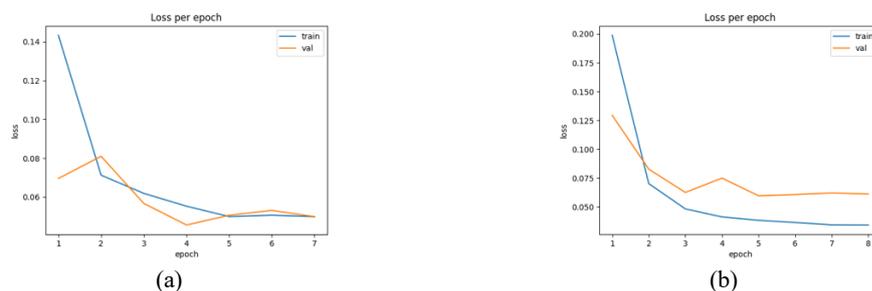
memiliki nilai terendah. Untuk FEDFormer FEB pada percobaan semua fitur menggunakan sampel ketiga. Untuk Hybrid FEDFormer FEB *with* LSTM pada percobaan semua fitur menggunakan sampel pertama. Untuk FEDFormer FEB pada *selected* fitur menggunakan percobaan pertama dan untuk Hybrid FEDFormer FEB *with* LSTM pada *selected* fitur menggunakan sampel keempat.

Pada *Gambar 4.49*, terdapat grafik *validation loss* untuk percobaan menggunakan *selected* fitur pada kedua model. Untuk model FEDFormer FEB, terlihat bahwa kurva *training* dan *validation loss* memiliki penurunan yang cukup sejalan dengan nilai akhir sekitar 0,060. Pada *validation loss*, mengalami sedikit fluktuasi antara *epoch* ke-4 hingga ke-5 kemudian menurun hingga *epoch* ke-7 dengan nilai yang hampir mendekati *training loss*. Untuk model Hybrid FEDFormer FEB *with* LSTM, *train loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch* dengan nilai akhir sekitar 0,04. Untuk *validation loss* memiliki nilai akhir sekitar 0,05.



Gambar 4.49 Grafik *Validation loss* Demand Listrik *Selected* Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB *with* LSTM

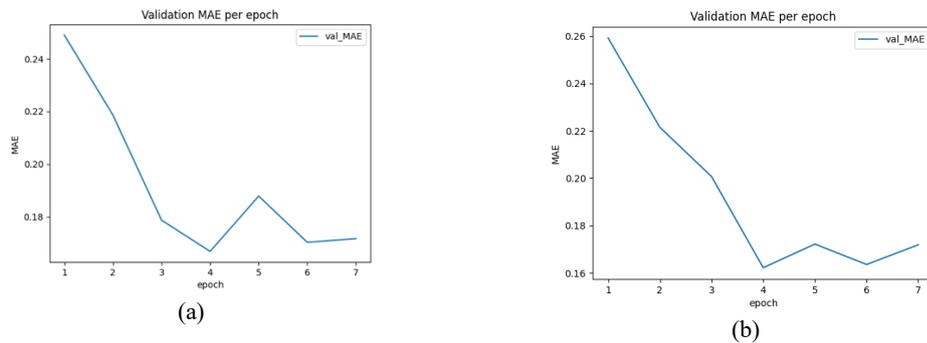
Pada *Gambar 4.50*, terdapat grafik *validation loss* untuk percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer FEB, terlihat bahwa *train loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch* dengan nilai akhir sekitar 0,05. Pada *validation loss*, mengalami sedikit fluktuasi dari *epoch* ke-4 hingga ke-6 kemudian menurun hingga memiliki nilai hampir mendekati nilai *train loss*. Untuk model Hybrid FEDFormer FEB *with* LSTM, kurva *training loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch* dengan nilai akhir sekitar 0,030. Pada *validation loss*, terjadi sedikit fluktuasi antara *epoch* ke-3 hingga *epoch* ke-5 kemudian cenderung stabil hingga *epoch* terakhir dengan nilai sekitar 0,074.



Gambar 4.50 Grafik *Validation loss* Demand Listrik Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB *with* LSTM

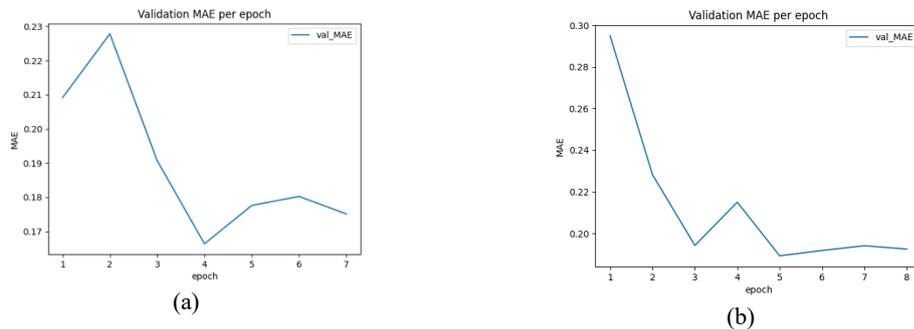
Pada *Gambar 4.51*, terdapat grafik *validation MAE* pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer FEB, terlihat bahwa penurunan *validation MAE* terjadi dari *epoch* ke-1 hingga ke-4 dari nilai sekitar 0,25 ke sekitar 0,15. Kemudian terjadi sedikit fluktuasi antara *epoch* ke-4 hingga ke-6 lalu cenderung stabil hingga *epoch* terakhir dengan nilai sekitar 0,17. Untuk model Hybrid FEDFormer FEB *with* LSTM, nilai *validation MAE* menurun dari *epoch* 1 hingga *epoch* ke-4 dari sekitar 0,26 hingga sekitar 0,16.

Kemudian mengalami sedikit fluktuasi dari *epoch* ke-4 hingga ke-6 lalu kembali menaik menuju *epoch* terakhir dengan nilai sekitar 0,17.



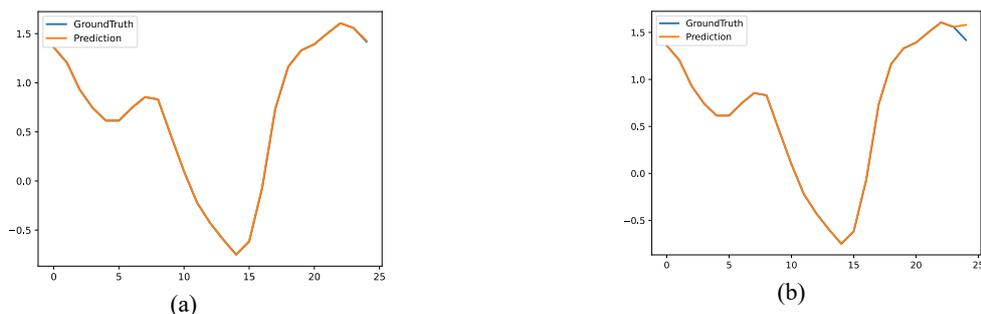
Gambar 4.51 Grafik Validation MAE Demand Listrik Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada *Gambar 4.52*, terdapat grafik *validation* MAE pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer FEB, terlihat bahwa terjadi penurunan dari *epoch* kedua hingga keempat lalu terjadi kenaikan dari *epoch* ke-4 hingga ke-6 lalu mengalami penurunan menuju *epoch* terakhir yaitu *epoch* ke-7 dengan nilai akhir sekitar 0,175. Untuk model Hybrid FEDFormer FEB with LSTM, terlihat bahwa nilai *validation* menurun dari *epoch* pertama hingga ke-3. Kemudian mengalami sedikit fluktuasi dari *epoch* ke-3 hingga ke-5 lalu cenderung stabil hingga *epoch* terakhir dengan nilai sekitar 0,19.



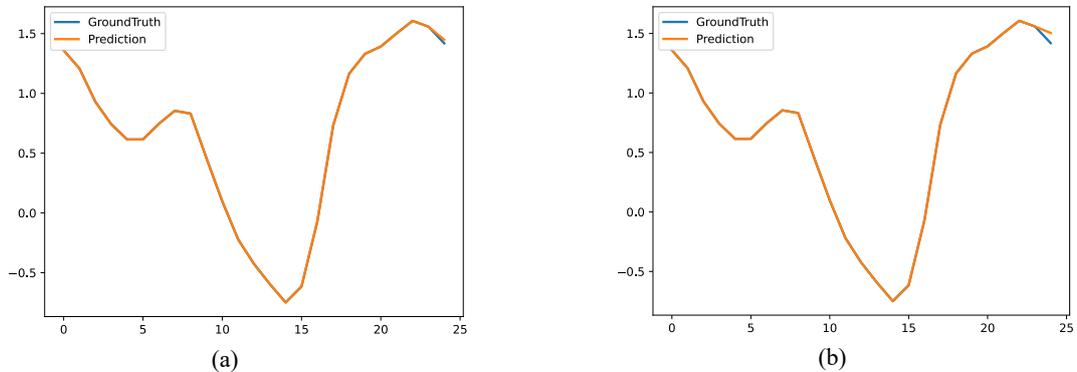
Gambar 4.52 Grafik Validation MAE Demand Listrik Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada *Gambar 4.53*, terdapat kurva prediksi pada percobaan menggunakan *selected* fitur untuk kedua model. Pada kedua model, terlihat jelas bahwa kurva *predict* mampu mengikuti nilai *ground truth* dengan baik. Apalagi pada model FEDFormer FEB, kurva *predict* nyaris mendekati sempurna pada *timestep* akhir. Untuk model Hybrid FEDFormer FEB with LSTM, masih terlihat *gap* antara kurva *predict* dan kurva *ground truth* pada akhir *timestep*.



Gambar 4.53 Grafik Prediksi Demand Listrik Selected Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

Pada *Gambar 4.54*, terdapat grafik prediksi untuk semua fitur pada kedua model. Hasilnya adalah keseluruhan kurva *predict* mampu mengikuti pola *ground truth* kecuali di akhir *timestep*. Untuk model FEDFormer FEB hanya terlihat sedikit sekali *gap* antara kurva *predict* dengan kurva *ground truth*, sedangkan model lainnya terlihat *gap* yang jelas.



Gambar 4.54 Grafik Prediksi Demand Listrik Semua Fitur (a) FEDFormer FEB (b) Hybrid FEDFormer FEB with LSTM

4.3.4 Hasil Implementasi Skenario 4

Pada implementasi ini, dilakukan pengujian hasil evaluasi percobaan dengan model FEDFormer WEB dan Hybrid FEDFormer WEB *with* LSTM pada semua fitur maupun pada fitur *selected* berdasarkan *correlation matrix*. Percobaan ini dilakukan dengan melakukan prediksi untuk target *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada prediksi *temperature*, hasil evaluasi didapatkan dengan memperoleh nilai MAE dan MSE. Pada *Tabel 4.39*, didapatkan nilai MAE pada standar *scaler* dan terlihat bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,05655 dan 0,06033 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00105 dan 0,00424 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan dengan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.39 Hasil MAE Skenario 4 *Temperature* Standar *Scaler*

Temperature	MAE				<div style="background-color: yellow; border: 1px solid black; padding: 2px; display: inline-block;"> Nilai terendah dari average/standar deviasi </div>
	Semua Fitur		Selected Fitur		
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	
1	0,05693	0,08514	0,05571	0,08038	
2	0,05716	0,09967	0,06346	0,09301	
3	0,05498	0,08937	0,05978	0,06522	
4	0,05605	0,09114	0,05695	0,08431	
5	0,05764	0,07708	0,06573	0,08102	
average	0,05655	0,08848	0,06033	0,08079	
stdev	0,00105	0,00828	0,00424	0,01006	

Pada *Tabel 4.40*, dalam skala asli, model FEDFormer WEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun

dengan *selected* fitur. Hasilnya adalah 0,15635 dan 0,16678 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00291 dan 0,01173 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.40 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.40 Hasil MAE Skenario 4 *Temperature* Skala Asli

Temperature	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,15740	0,23539	0,15402	0,22222
2	0,15802	0,27555	0,17544	0,25716
3	0,15200	0,24708	0,16527	0,18031
4	0,15495	0,25197	0,15745	0,23310
5	0,15937	0,21309	0,18173	0,22400
average	0,15635	0,24462	0,16678	0,22336
stdev	0,00291	0,02289	0,01173	0,02780

Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Pada Tabel 4.41, dalam standar *scaler*, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00521 dan 0,00590 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan model FEDFormer WEB pada percobaan semua fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, didapatkan bahwa model FEDFormer WEB memiliki standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00015 dan 0,00075 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi terendah.

Tabel 4.41 Hasil MSE Skenario 4 *Temperature* Standar *Scaler*

Temperature	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,00538	0,01185	0,00513	0,01215
2	0,00522	0,01603	0,00628	0,01422
3	0,00500	0,01316	0,00575	0,00776
4	0,00517	0,01354	0,00535	0,01260
5	0,00530	0,00983	0,00699	0,01105
average	0,00521	0,01288	0,00590	0,01156
stdev	0,00015	0,00228	0,00075	0,00241

Nilai terendah dari average/standar deviasi

Pada Tabel 4.42, dalam skala asli, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,03985 dan 0,04510 yang terlihat

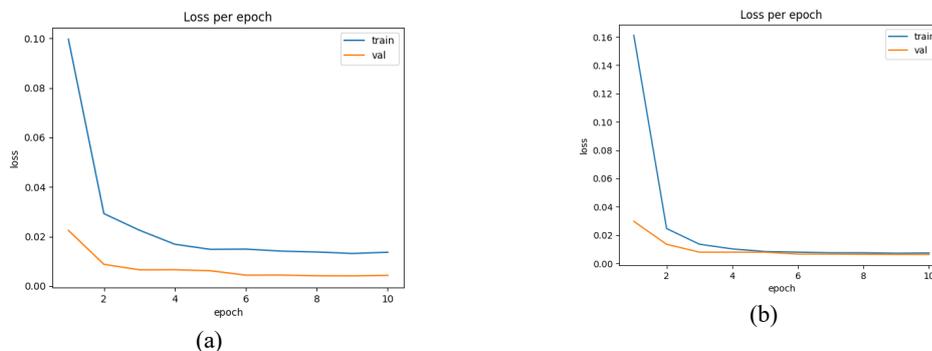
pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MSE terendah dibanding lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,0011 dan 0,00572 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah.

Tabel 4.42 Hasil MSE Skenario 4 Temperature Skala Asli

Temperature	MSE				<div style="background-color: yellow; width: 15px; height: 10px; display: inline-block; margin-right: 5px;"></div> Nilai terendah dari average/standar deviasi
	Semua Fitur		Selected Fitur		
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	
1	0,04113	0,09058	0,03925	0,09286	
2	0,03989	0,12250	0,04803	0,10867	
3	0,03821	0,10060	0,04395	0,05935	
4	0,03948	0,10349	0,04089	0,09631	
5	0,04052	0,07514	0,05340	0,08444	
average	0,03985	0,09846	0,04510	0,08833	
stdev	0,00111	0,01742	0,00572	0,01839	

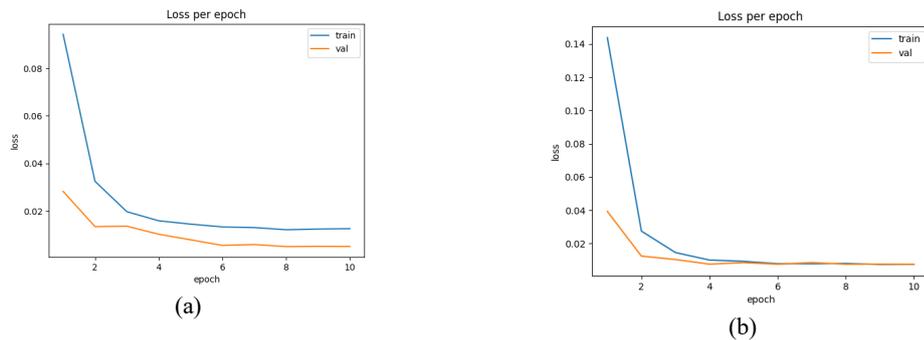
Berdasarkan *Tabel 4.40*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk model FEDFormer WEB pada percobaan semua fitur menggunakan sampel ketiga. Untuk model Hybrid FEDFormer WEB *with* LSTM pada percobaan semua fitur menggunakan sampel kelima. Untuk model FEDFormer WEB pada percobaan *selected* fitur menggunakan percobaan pertama dan untuk model Hybrid FEDFormer WEB *with* LSTM pada percobaan *selected* fitur menggunakan percobaan ketiga

Pada *Gambar 4.55*, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa *training loss* menurun dari *epoch* pertama hingga *epoch* ke-4 kemudian mengalami stabil hingga akhir *epoch* dengan nilai sekitar 0,02. Untuk *validation loss*, mengalami penurunan dari *epoch* pertama hingga kedua lalu mengalami kestabilan hingga akhir *epoch* dengan nilai sekitar 0,01. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa kurva *training loss* dan *validation loss* saling mendekati dengan nilai akhir sekitar 0,01. Hal tersebut dimulai dari *epoch* ke-4 hingga *epoch* terakhir.



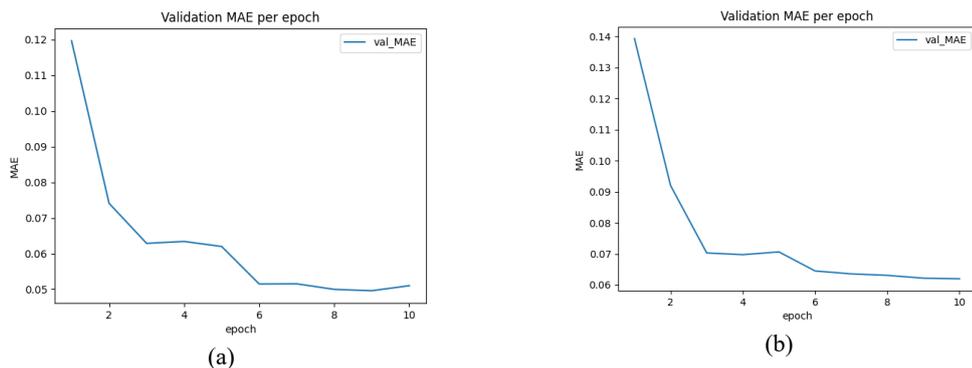
Gambar 4.55 Grafik Validation loss Temperature Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.56, terdapat grafik *validation loss* pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa kurva *training loss* mengalami penurunan dari awal *epoch* hingga *epoch* ke-3 lalu mengalami kestabilan hingga akhir *epoch* dengan nilai akhir adalah 0,02. Untuk *validation loss* juga cenderung menurun dari awal *epoch* hingga *epoch* ke-6 lalu menjadi stabil hingga *epoch* terakhir yaitu *epoch* ke-10 dengan nilai akhir adalah 0,01. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa *training loss* mengalami penurunan dari awal *epoch* hingga *epoch* ke-4 lalu menjadi stabil hingga akhir. Untuk *validation loss*, mengalami penurunan dari awal *epoch* hingga *epoch* ke-4 kemudian menjadi stabil dengan nilai yang hampir menyerupai *training loss* hingga akhir *epoch*. Keduanya memiliki nilai akhir sekitar 0,01.



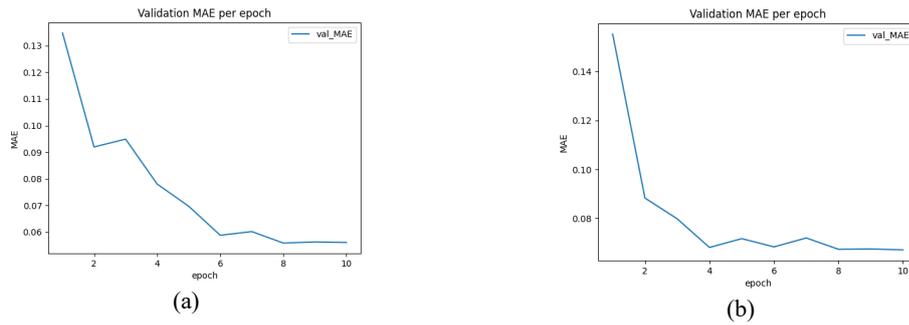
Gambar 4.56 Grafik Validation loss Temperature Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.57, terdapat grafik *validation MAE* pada percobaan *selected* fitur untuk kedua model. Untuk model FEDFormer WEB, nilai *validation MAE* menurun dari *epoch* ke-1 hingga *epoch* ke-3 kemudian mengalami kestabilan sebelum turun lagi dari *epoch* ke-5 dan mencapai kestabilan hingga *epoch* terakhir dengan nilai akhir sekitar 0,05. Untuk model Hybrid FEDFormer WEB *with* LSTM, nilai *validation MAE* menurun dari *epoch* pertama hingga *epoch* ke-3. Kemudian, nilai stabil hingga *epoch* ke-5 lalu menurun hingga *epoch* ke-10 dengan nilai akhir sekitar 0,06.



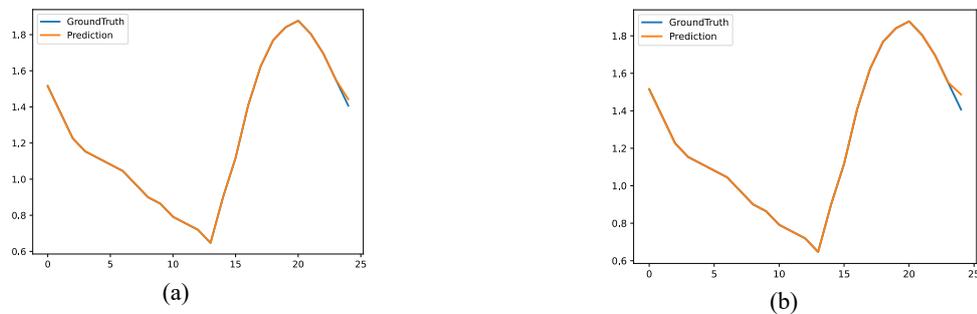
Gambar 4.57 Grafik Validation MAE Temperature Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.58, terdapat grafik *validation MAE* pada percobaan semua fitur untuk kedua model. Untuk model FEDFormer WEB, kurva cenderung menurun dari *epoch* pertama hingga *epoch* ke-6 lalu mulai mencapai kestabilan hingga *epoch* ke-10 dengan nilai akhir sekitar 0,06. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat nilai *validation* menurun dari *epoch* pertama hingga ke-4 lalu mengalami kestabilan hingga *epoch* terakhir dengan nilai akhir sekitar 0,06.



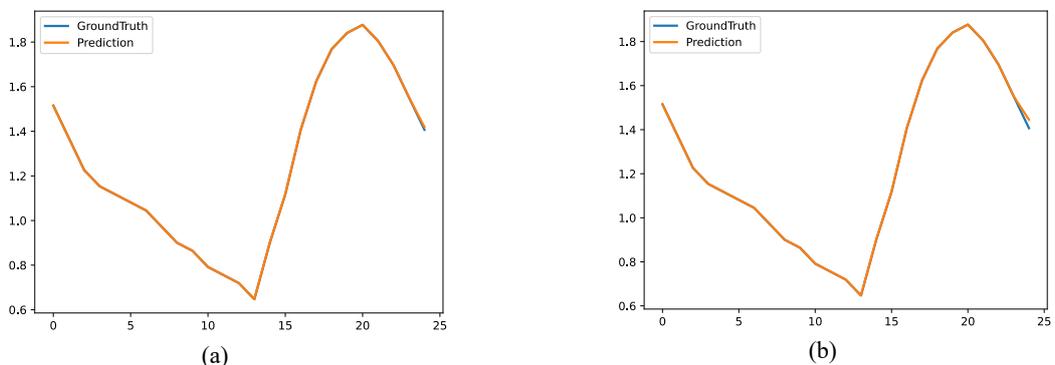
Gambar 4.58 Grafik Validation MAE Temperature Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.59, terdapat grafik prediksi pada percobaan *selected* fitur untuk kedua model. Hasilnya adalah kedua model memiliki kurva *predict* yang mengikuti kurva *ground truth* dengan baik kecuali pada akhir *timestep*. Pada model FEDFormer WEB, terlihat sedikit *gap* antara kurva *predict* dengan kurva *ground truth*. Untuk model Hybrid FEDFormer WEB with LSTM, terdapat lumayan besar *gap* antara kurva *predict* dengan kurva *ground truth*.



Gambar 4.59 Grafik Prediksi Temperature Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.60, terdapat grafik prediksi untuk percobaan menggunakan semua fitur pada kedua model. Untuk model FEDFormer WEB, terlihat bahwa kurva *predict* mengikuti dengan baik pada kurva *ground truth*. Bahkan, terlihat sangat sedikit *gap* antara kurva *predict* dengan kurva *ground truth* pada akhir *timestep*. Untuk model Hybrid FEDFormer WEB with LSTM, terlihat perbedaan antara kurva *predict* dengan kurva *ground truth* pada akhir *timestep* meski pada *timestep* sebelumnya menunjukkan keakuratan.



Gambar 4.60 Grafik Prediksi Temperature Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

b. Prediksi untuk GHI

Pada percobaan untuk prediksi target GHI, didapatkan hasil MAE dan MSE pada masing-masing percobaan yang dilakukan. Pada Tabel 4.43, dalam standar *scaler*, didapatkan bahwa

model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,24249 dan 0,21851 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE sebesar 0,00343 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi MAE sebesar 0,00141 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan *selected* fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.43 Hasil MAE Skenario 4 GHI Standar Scaler

GHI	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,24845	0,24526	0,23410	0,21749
2	0,25374	0,23737	0,23299	0,21863
3	0,23660	0,24270	0,23161	0,22095
4	0,25187	0,24125	0,23062	0,22388
5	0,25031	0,24589	0,23126	0,21163
average	0,24819	0,24249	0,23211	0,21851
stdev	0,00677	0,00343	0,00141	0,00456

Nilai terendah dari average/standar deviasi

Pada Tabel 4.44, dalam skala asli, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 62,28274 dan 56,12330 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,88086 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,36198 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi terendah. Warna hijau pada Tabel 4.44 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.44 Hasil MAE Skenario 4 GHI Skala Asli

GHI	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	63,81179	62,99392	60,12680	55,85972
2	65,17021	60,96682	59,84094	56,15254
3	60,76867	62,33643	59,48606	56,74877
4	64,69069	61,96182	59,23209	57,50058
5	64,28880	63,15471	59,39618	54,35489
average	63,74603	62,28274	59,61641	56,12330
stdev	1,73812	0,88086	0,36198	1,17108

Nilai terendah dari average/standar deviasi
Nilai sampel terendah dari lima percobaan

Pada *Tabel 4.45*, dalam standar *scaler*, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya pada percobaan menggunakan semua fitur yaitu sebesar 0,16407 yang terlihat pada warna kuning. Pada percobaan menggunakan *selected* fitur, model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 0,15240 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00237 dan 0,00090 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan *selected* fitur memiliki nilai standar deviasi MSE paling rendah.

Tabel 4.45 Hasil MSE Skenario 4 GHI Standar Scaler

GHI	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,16315	0,17113	0,16026	0,15468
2	0,16675	0,16322	0,16052	0,15362
3	0,16055	0,16671	0,15902	0,15326
4	0,16444	0,16732	0,15914	0,15086
5	0,16547	0,17050	0,15836	0,14957
average	0,16407	0,16777	0,15946	0,15240
stdev	0,00237	0,00319	0,00090	0,00211

Nilai terendah dari average/standar deviasi

Pada *Tabel 4.46*, dalam skala asli, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya pada percobaan semua fitur yaitu sebesar 10823,3788 yang terlihat pada warna kuning. Sementara itu, pada percobaan dengan *selected* fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 10053,324 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 156,53530 dan 59,61348 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MSE terendah.

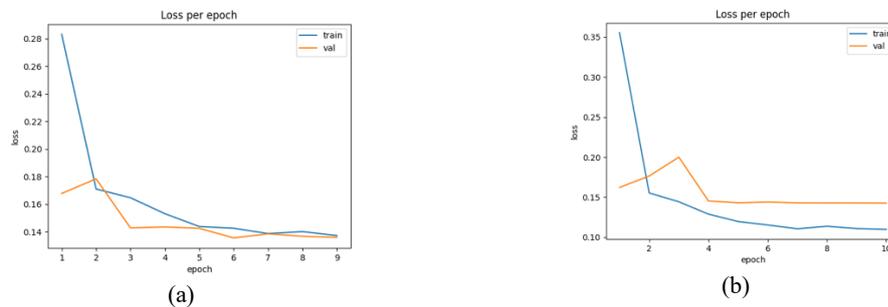
Tabel 4.46 Hasil MSE Skenario 4 GHI Skala Asli

GHI	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	10762,41626	11288,90380	10572,18207	10203,61859
2	11000,16986	10767,05891	10589,15938	10134,03266
3	10591,07054	10997,14356	10490,42824	10110,43554
4	10847,77496	11037,60226	10498,38167	9952,06136
5	10915,46240	11247,65145	10446,72523	9866,47185
average	10823,37880	11067,67200	10519,37532	10053,32400
stdev	156,53530	210,70287	59,61348	139,22278

Nilai terendah dari average/standar deviasi

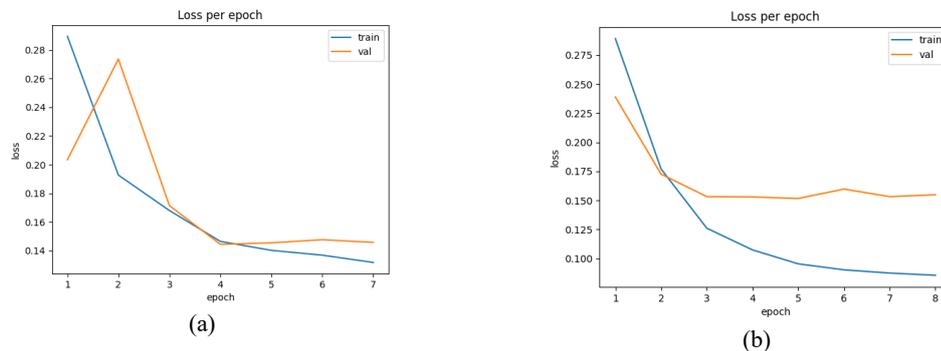
Berdasarkan *Tabel 4.44*, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk model FEDFormer WEB pada percobaan semua fitur menggunakan sampel ketiga. Untuk model Hybrid FEDFormer WEB *with* LSTM pada percobaan semua fitur menggunakan sampel kedua. Untuk model FEDFormer WEB pada percobaan *selected* fitur menggunakan percobaan keempat dan untuk model Hybrid FEDFormer WEB *with* LSTM pada percobaan *selected* fitur menggunakan percobaan kelima.

Pada *Gambar 4.61*, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer WEB, *training loss* terlihat menurun hingga akhir *epoch*. Untuk *validation loss*, terlihat stabil dan cenderung menurun mendekati nilai *training loss*. Untuk model Hybrid FEDFormer WEB *with* LSTM, pada *training loss* terlihat penurunan dari awal *epoch* hingga akhir *epoch*. Untuk *validation loss*, mengalami penurunan dari *epoch* ke-3 hingga *epoch* ke-4 kemudian mengalami kestabilan hingga akhir *epoch*.



Gambar 4.61 Grafik Validation loss GHI Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

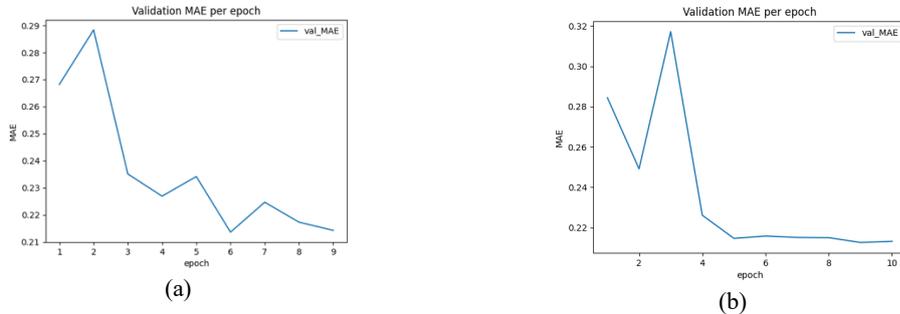
Pada *Gambar 4.62*, terdapat grafik *validation loss* pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer WEB, *training loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. *Validation loss* mengalami kenaikan dari *epoch* pertama ke *epoch* ke-2 sebelum kemudian menurun dan mendekati nilai *training loss* hingga akhir *epoch*. Untuk model Hybrid FEDFormer WEB *with* LSTM, *training loss* menunjukkan penurunan dari awal *epoch* hingga akhir *epoch*. *Validation loss* mengalami penurunan dari *epoch* pertama hingga *epoch* ke-3. Kemudian mengalami kestabilan hingga *epoch* terakhir.



Gambar 4.62 Grafik Validation loss GHI Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

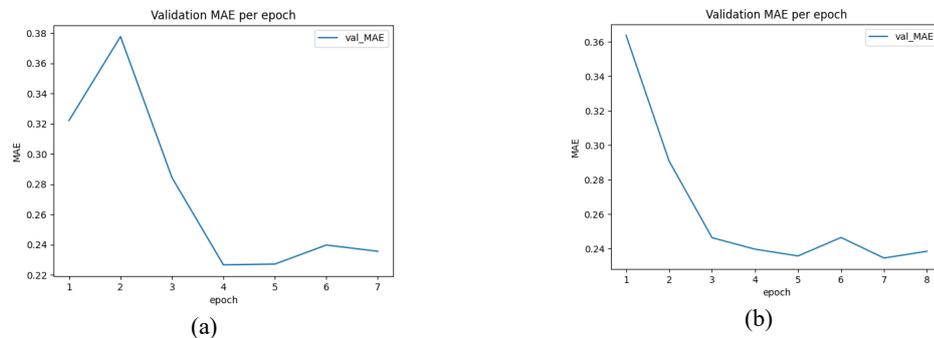
Pada *Gambar 4.63*, terdapat grafik *validation MAE* pada percobaan *selected* fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa *validation MAE* menurun dari *epoch* ke-2 hingga *epoch* ke-4. Kemudian menaik lalu turun lagi dari *epoch* ke-5 ke *epoch* ke-6. Lalu menaik lagi dan turun lagi dari *epoch* ke-7 menuju *epoch* terakhir. Dengan nilai sekitar

0,21 Pada model Hybrid FEDFormer WEB *with* LSTM, nilai *validation* MAE menurun dari *epoch* pertama menuju *epoch* kedua. Lalu terjadi kenaikan dari *epoch* kedua menuju *epoch* ketiga kemudian menurun hingga *epoch* ke-5. Kemudian, nilai menjadi stabil hingga akhir *epoch* dengan nilai dibawah 0,22.



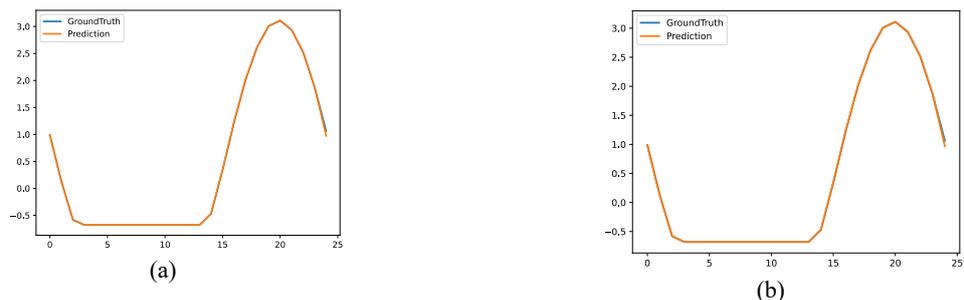
Gambar 4.63 Grafik Validation MAE GHI Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.64, terdapat grafik *validation* MAE pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa nilai MAE meningkat dari *epoch* pertama menuju *epoch* kedua sebelum akhirnya menurun hingga *epoch* ke-4 dan stabil di *epoch* ke-5. Kemudian mengalami sedikit kenaikan hingga *epoch* ke-6 dan menurun hingga *epoch* terakhir pada nilai sekitar 0,23. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa nilai *validation* MAE menurun dari *epoch* pertama hingga *epoch* ke-5. Kemudian mengalami sedikit kenaikan sebelum menurun dan mulai stabil di *epoch* ke-7 dengan nilai yang awalnya mencapai kurang lebih 0,37 menjadi 0,23.



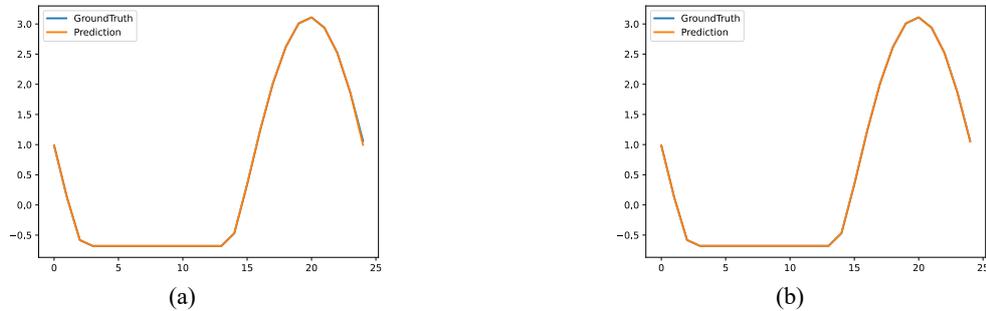
Gambar 4.64 Grafik Validation MAE GHI Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.65, terdapat grafik prediksi pada percobaan menggunakan *selected* fitur untuk kedua model. Dari kedua model, terlihat bahwa kurva *prediction* mampu mengikuti kurva *ground truth* dengan memiliki sedikit *gap* pada akhir *timestep* meski *timestep* sebelumnya menunjukkan tidak ada perbedaan.



Gambar 4.65 Grafik Prediksi GHI Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.66, terdapat grafik prediksi pada percobaan menggunakan semua fitur untuk kedua model. Dari kedua model, terlihat bahwa kurva *predict* mampu mengikuti kurva *ground truth* dengan baik. Perbedaannya adalah pada model FEDFormer WEB terlihat sedikit perbedaan sedangkan pada model Hybrid FEDFormer WEB with LSTM tidak terlihat perbedaan yang menandakan bahwa kurva *predict* memiliki akurasi yang nyaris sempurna.



Gambar 4.66 Grafik Prediksi GHI Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

c. Prediksi untuk *demand* listrik (KW)

Pada prediksi untuk target *demand* listrik (KW), didapatkan nilai hasil evaluasi MAE dan MSE. Pada Tabel 4.47, dalam standar *scaler*, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,16465 dan 0,14752 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan dengan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun *selected* fitur. Hasilnya adalah 0,00273 dan 0,00317 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.47 Hasil MAE Skenario 4 Demand Listrik Standar Scaler

KW	MAE				<div style="background-color: yellow; border: 1px solid black; padding: 2px; display: inline-block;"> Nilai terendah dari average/standar deviasi </div>
	Semua Fitur		Selected Fitur		
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	
1	0,16310	0,18647	0,14684	0,16434	
2	0,16787	0,17906	0,14521	0,17036	
3	0,16459	0,18366	0,14637	0,16355	
4	0,16664	0,17615	0,14609	0,15858	
5	0,16104	0,16310	0,15308	0,16432	
average	0,16465	0,17769	0,14752	0,16423	
stdev	0,00273	0,00908	0,00317	0,00418	

Pada Tabel 4.48, dalam skala asli, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 20,42159 dan 18,29667 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan dengan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MAE terendah baik pada percobaan menggunakan semua fitur maupun menggunakan *selected* fitur. Hasilnya adalah 0,33863 dan 0,39286 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model

FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terkecil. Warna hijau pada Tabel 4.48 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.48 Hasil MAE Skenario 4 Demand Listrik Skala Asli

KW	MAE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	20,22939	23,12741	18,21313	20,38278
2	20,82153	22,20846	18,01008	21,12946
3	20,41450	22,77954	18,15417	20,28550
4	20,66883	21,84851	18,11908	19,66812
5	19,97368	20,22963	18,98689	20,38084
average	20,42159	22,03871	18,29667	20,36934
stdev	0,33863	1,12604	0,39286	0,51885

Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Pada Tabel 4.49, dalam standar *scaler*, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan semua fitur maupun percobaan menggunakan *selected* fitur. Hasilnya adalah 0,04709 dan 0,03931 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, didapatkan model FEDFormer WEB yang memiliki nilai standar deviasi MSE terendah baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00143 dan 0,00156 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terkecil.

Tabel 4.49 Hasil MSE Skenario 4 Demand Listrik Standar Scaler

KW	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,04676	0,06083	0,03869	0,04856
2	0,04845	0,05625	0,03844	0,05031
3	0,04734	0,05718	0,03868	0,04725
4	0,04810	0,05329	0,03864	0,04509
5	0,04481	0,04676	0,04209	0,04855
average	0,04709	0,05486	0,03931	0,04795
stdev	0,00143	0,00527	0,00156	0,00193

Nilai terendah dari average/standar deviasi

Pada Tabel 4.50, dalam skala asli, didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 724,44574 dan 604,71874 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, model FEDFormer WEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 22,06033 dan 23,94015 yang terlihat pada warna kuning. Dari kedua nilai

tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai terendah.

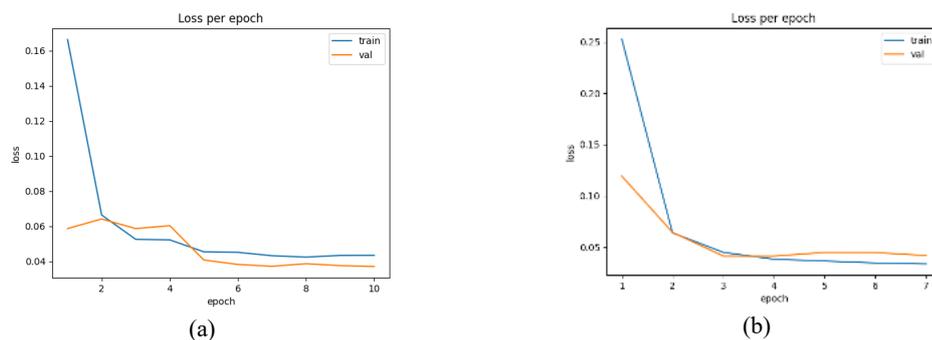
Tabel 4.50 Hasil MSE Skenario 4 Demand Listrik Skala Asli

KW	MSE			
	Semua Fitur		Selected Fitur	
	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	719,25861	935,83662	595,25789	747,09837
2	745,30741	865,30788	591,41976	773,90543
3	728,29094	879,63107	595,09895	726,89564
4	739,97640	819,78214	594,36244	693,62748
5	689,39535	719,32500	647,45466	746,90888
average	724,44574	843,97654	604,71874	737,68716
stdev	22,06033	81,06775	23,94015	29,76191

Nilai terendah dari average/standar deviasi

Berdasarkan Tabel 4.48, dilakukan analisis dengan mengambil satu sampel pada masing-masing model yang terlihat pada warna hijau. Dari kelima percobaan, diambil sampel yang memiliki nilai terendah. Untuk model FEDFormer WEB pada percobaan semua fitur menggunakan sampel kelima. Untuk model Hybrid FEDFormer WEB with LSTM pada percobaan semua fitur menggunakan sampel kelima. Untuk model FEDFormer WEB pada percobaan *selected* fitur menggunakan percobaan kedua dan untuk model Hybrid FEDFormer WEB with LSTM pada percobaan *selected* fitur menggunakan percobaan keempat

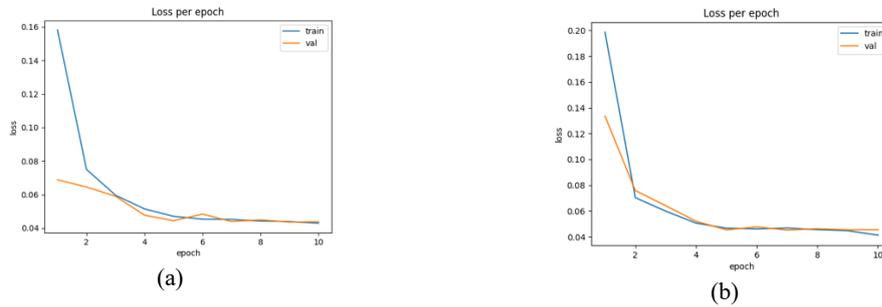
Pada Gambar 4.67, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur pada kedua model. Untuk model FEDFormer WEB, *training loss* menurun dari *epoch* pertama hingga *epoch* terakhir yaitu *epoch* ke-10 dengan nilai akhir sekitar 0,05. *Validation loss* mengalami penurunan juga dari *epoch* ke-4 menuju *epoch* ke-5 lalu mulai stabil hingga akhir *epoch* dengan nilai akhir sekitar 0,04. Pada model Hybrid FEDFormer WEB with LSTM, *training loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. Untuk *validation loss* mengalami kenaikan dari *epoch* ketiga hingga kelima. Kemudian, mengalami penurunan hingga akhir *epoch*. Keduanya memiliki nilai akhir sekitar 0,05.



Gambar 4.67 Grafik Validation loss Demand Listrik Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

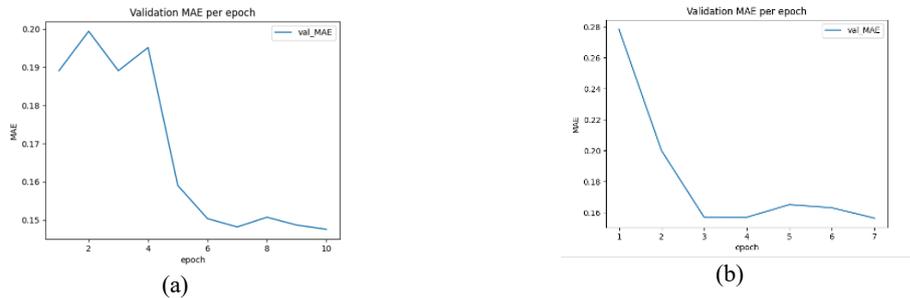
Pada Gambar 4.68, terdapat grafik *validation loss* pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer WEB, *training loss* mengalami penurunan dari *epoch* pertama hingga *epoch* ke-6, kemudian mengalami kestabilan hingga *epoch* akhir yaitu *epoch* ke-10. Untuk *validation loss* mengalami penurunan dari awal *epoch* hingga *epoch* ke-5. Lalu mengalami sedikit fluktuasi dari *epoch* ke-5 hingga *epoch* ke-6 lalu menurun dan cenderung stabil serta mengikuti nilai *train loss* hingga akhir *epoch*. Untuk model Hybrid

FEDFormer WEB *with* LSTM, kedua model menurun dari *epoch* pertama menuju *epoch* ke-4. Lalu kedua model saling mendekati dari *epoch* ke-4 hingga *epoch* terakhir yaitu *epoch* ke-10.



Gambar 4.68 Grafik Validation loss Demand Listrik Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.69, terdapat grafik *validation* MAE pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa nilai *validation* mengalami fluktuasi dari *epoch* pertama hingga *epoch* ke-4 lalu mengalami penurunan hingga *epoch* ke-7. Kemudian terjadi sedikit fluktuasi hingga akhirnya menurun pada *epoch* ke-8 hingga *epoch* terakhir dengan nilai sekitar 0,15. Pada model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa nilai *validation* menurun dari awal *epoch* hingga *epoch* ke-4. Sempat terjadi kenaikan dari *epoch* ke-4 hingga ke-5 lalu menurun hingga akhir *epoch* dengan nilai sekitar 0,16.



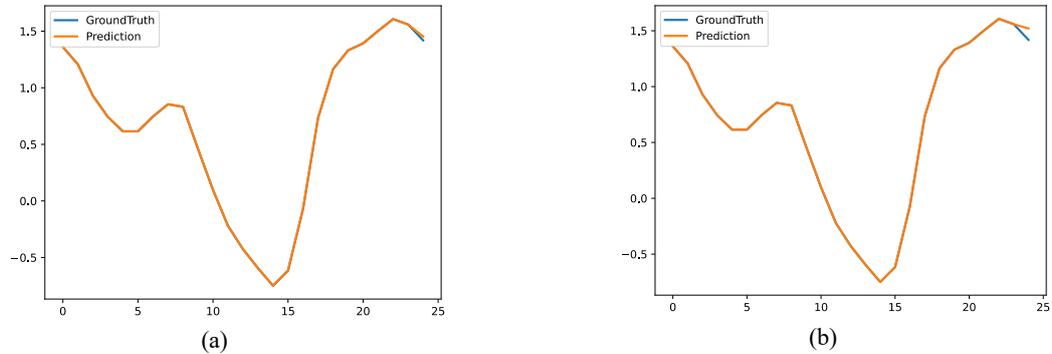
Gambar 4.69 Grafik Validation MAE Demand Listrik Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.70, terdapat grafik *validation* MAE pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa nilai *validation* MAE menurun dari *epoch* pertama menuju *epoch* ke-5. Kemudian mengalami sedikit fluktuasi dari *epoch* ke-5 hingga ke-7 kemudian cenderung mengalami kestabilan hingga *epoch* ke-10 dengan nilai akhir sebesar 0,16. Pada model Hybrid FEDFormer WEB *with* LSTM, terlihat penurunan dari *epoch* pertama hingga *epoch* ke-5. Kemudian mengalami sedikit kenaikan dan menurun lagi serta cenderung stabil hingga *epoch* ke-10 dengan nilai akhir sebesar 0,16.



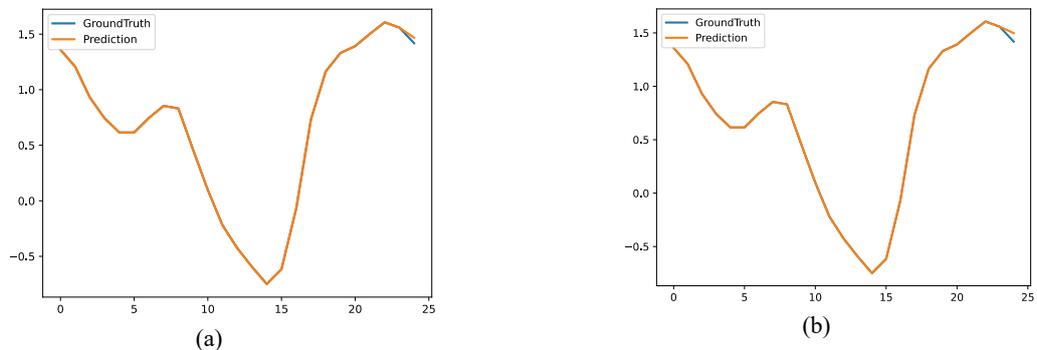
Gambar 4.70 Grafik Validation MAE Demand Listrik Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.71, terdapat grafik prediksi pada percobaan menggunakan *selected* fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa kurva *predict* mampu mengikuti kurva *ground truth* dengan baik. Pada akhir *timestep*, hanya terlihat sedikit sekali perbedaan antara kurva *predict* dengan *ground truth*. Untuk model Hybrid FEDFormer WEB with LSTM, kurva *predict* mengikuti dengan baik nilai dari kurva *ground truth*. Akan tetapi, pada akhir *timestep*, terlihat perbedaan yang jelas antara kurva *predict* dan kurva *ground truth*.



Gambar 4.71 Grafik Prediksi Demand Listrik Selected Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.72, terdapat grafik prediksi pada percobaan menggunakan semua fitur untuk kedua model. Untuk model FEDFormer WEB, terlihat bahwa kurva *predict* mampu mengikuti kurva *ground truth* dengan baik meski di akhir *timestep* terlihat sedikit perbedaan. Hal tersebut berlaku juga pada model Hybrid FEDFormer WEB with LSTM, dimana pada akhir *timestep* terlihat perbedaan antara kurva *predict* dengan kurva *ground truth*.



Gambar 4.72 Grafik Prediksi Demand Listrik Semua Fitur (a) FEDFormer WEB (b) Hybrid FEDFormer WEB with LSTM

4.3.5 Hasil Implementasi Skenario 5

Pada implementasi ini dilakukan implementasi terhadap model LSTM, hybrid LSTM with Transformer, FEDFormer FEB, Hybrid FEDFormer FEB with LSTM, FEDFormer WEB, dan Hybrid FEDFormer WEB with LSTM. Perbandingan dilakukan juga pada percobaan menggunakan semua fitur maupun dengan *selected* fitur dalam memprediksi target *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada prediksi *temperature*, dilakukan evaluasi berupa MAE untuk percobaan menggunakan semua fitur dan menggunakan *selected* fitur maupun evaluasi berupa MSE untuk percobaan menggunakan semua fitur maupun *selected* fitur. Pada Tabel 4.51, dilakukan percobaan untuk mendapatkan nilai MAE yang menjadi perbandingan antara keenam model dari percobaan dengan semua fitur maupun dengan *selected* fitur menggunakan *correlation*

matrix dalam standar *scaler*. Diantara keenam model, model FEDFormer WEB memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua model maupun *selected* model dengan nilai sebesar 0,05655 dan 0,06033 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MAE terendah.

Untuk standar deviasi, pada percobaan menggunakan semua fitur didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00105 yang terlihat pada warna kuning. Pada percobaan menggunakan *selected* fitur, didapatkan bahwa model FEDFormer FEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00309 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan dengan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.51 Hasil MAE Skenario 5 *Temperature* Standar *Scaler*

MAE						
Temperature	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,06989	0,20125	0,07035	0,09928	0,05693	0,08514
2	0,05779	0,30846	0,06485	0,11161	0,05716	0,09967
3	0,05902	0,41317	0,07859	0,10020	0,05498	0,08937
4	0,04779	0,14392	0,07394	0,09782	0,05605	0,09114
5	0,05869	0,30038	0,06984	0,11788	0,05764	0,07708
<i>average</i>	0,05863	0,27344	0,07152	0,10536	0,05655	0,08848
<i>std</i>	0,00783	0,10424	0,00511	0,00889	0,00105	0,00828
Temperature	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,06826	0,15062	0,06766	0,08689	0,05571	0,08038
2	0,08077	0,17862	0,07131	0,09218	0,06346	0,09301
3	0,07086	0,27969	0,06920	0,10822	0,05978	0,06522
4	0,07263	0,14100	0,06786	0,08946	0,05695	0,08431
5	0,06529	0,20957	0,07511	0,09549	0,06573	0,08102
<i>average</i>	0,07156	0,19190	0,07023	0,09445	0,06033	0,08079
<i>std</i>	0,00584	0,05588	0,00309	0,00833	0,00424	0,01006

Nilai terendah dari average/standar deviasi

Pada Tabel 4.52, dalam skala asli, model FEDFormer WEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,15635 dan 0,16678 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi terendah dibanding model lainnya dengan nilai sebesar 0,00291 yang terlihat pada warna kuning. Pada percobaan menggunakan *selected* fitur didapatkan bahwa model FEDFormer FEB memiliki nilai standar deviasi MAE terkecil dibanding model lainnya yaitu sebesar 0,00855 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan

menggunakan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.52 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Pada Tabel 4.53, dalam standar *scaler*, didapatkan nilai MSE untuk masing-masing model yang diimplementasi. Model FEDFormer WEB memiliki nilai *average* MSE terendah baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur dengan nilai sebesar 0,00521 dan 0,00590 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MSE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur didapatkan bahwa model FEDFormer WEB memiliki nilai terendah dibanding model lainnya yaitu sebesar 0,00015 yang terlihat pada warna kuning. Pada percobaan menggunakan *selected* fitur, didapatkan bahwa model FEDFormer FEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya yaitu sebesar 0,00063 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi MSE terkecil.

Tabel 4.52 Hasil MAE Skenario 5 *Temperature* Skala Asli

MAE						
Temperature	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,19185	0,55248	0,19451	0,27449	0,15740	0,23539
2	0,15864	0,84679	0,17930	0,30857	0,15802	0,27555
3	0,16203	1,13423	0,21729	0,27703	0,15200	0,24708
4	0,13119	0,39509	0,20443	0,27045	0,15495	0,25197
5	0,16111	0,82459	0,19309	0,32589	0,15937	0,21309
average	0,16096	0,75064	0,19772	0,29129	0,15635	0,24462
std	0,02149	0,28615	0,01413	0,02457	0,00291	0,02289
Temperature	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,18739	0,41349	0,18705	0,24022	0,15402	0,22222
2	0,22173	0,49035	0,19716	0,25484	0,17544	0,25716
3	0,19453	0,76779	0,19130	0,29918	0,16527	0,18031
4	0,19937	0,38708	0,18760	0,24734	0,15745	0,23310
5	0,17924	0,57532	0,20765	0,26401	0,18173	0,22400
average	0,19645	0,52681	0,19415	0,26112	0,16678	0,22336
std	0,01605	0,15340	0,00855	0,02304	0,01173	0,02780

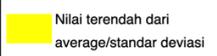
Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Pada Tabel 4.54, dalam skala asli, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dari model lainnya baik pada percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur. Hasilnya adalah 0,03985 dan 0,04510 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MSE terendah. Untuk standar

deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model FEDFormer WEB memiliki standar deviasi MSE terendah dibanding model lainnya yaitu sebesar 0,00111 yang terlihat pada warna kuning. Pada percobaan menggunakan *selected* fitur, model FEDFormer FEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya. Hasilnya adalah 0,00478 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah dibanding model FEDFormer FEB pada percobaan menggunakan *selected* fitur.

Tabel 4.53 Hasil MSE Skenario 5 *Temperature* Standar Scaler

MSE						
Temperature	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,01000	0,08406	0,00810	0,01575	0,00538	0,01185
2	0,00607	0,16183	0,00680	0,01945	0,00522	0,01603
3	0,00574	0,39236	0,00976	0,01598	0,00500	0,01316
4	0,00375	0,03284	0,00932	0,01578	0,00517	0,01354
5	0,00566	0,23677	0,00778	0,02152	0,00530	0,00983
average	0,00624	0,18157	0,00835	0,01770	0,00521	0,01288
std	0,00229	0,14098	0,00120	0,00265	0,00015	0,00228
Temperature	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,00936	0,04587	0,00743	0,01268	0,00513	0,01215
2	0,01210	0,08623	0,00838	0,01380	0,00628	0,01422
3	0,01186	0,14378	0,00799	0,02005	0,00575	0,00776
4	0,01190	0,03298	0,00758	0,01301	0,00535	0,01260
5	0,00778	0,07998	0,00898	0,01591	0,00699	0,01105
average	0,01060	0,07777	0,00807	0,01509	0,00590	0,01156
std	0,00194	0,04318	0,00063	0,00304	0,00075	0,00241



Nilai terendah dari average/standar deviasi

Pada Tabel 4.52, berdasarkan nilai MAE, diambil satu sample untuk model LSTM, Hybrid LSTM *with* Transformer, dan FEDFormer WEB agar dilakukan analisis yang terlihat pada warna hijau. Untuk percobaan menggunakan semua fitur, model LSTM akan mengambil percobaan keempat, model Hybrid LSTM *with* Transformer akan mengambil percobaan keempat, dan model FEDFormer WEB akan mengambil percobaan ketiga. Sementara itu, untuk percobaan menggunakan *selected* fitur, model LSTM akan mengambil percobaan kelima, model Hybrid LSTM *with* Transformer akan mengambil percobaan keempat, dan model FEDFormer WEB akan mengambil percobaan pertama. Pengambilan sample percobaan berdasarkan nilai terendah dari kelima percobaan yang telah dilakukan.

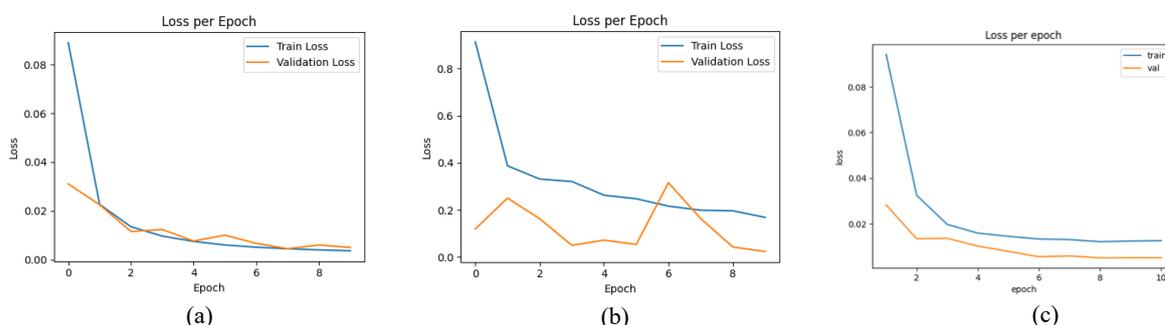
Tabel 4.54 Hasil MSE Skenario 5 *Temperature* Skala Asli

MSE						
Temperature	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,07533	0,63349	0,06188	0,12039	0,04113	0,09058
2	0,04573	1,21957	0,05198	0,14864	0,03989	0,12250
3	0,04329	2,95687	0,07461	0,12213	0,03821	0,10060
4	0,02827	0,24745	0,07123	0,12058	0,03948	0,10349
5	0,04268	1,78430	0,05948	0,16452	0,04052	0,07514
average	0,04706	1,36834	0,06383	0,13525	0,03985	0,09846
std	0,01723	1,06242	0,00914	0,02027	0,00111	0,01742
Temperature	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,07055	0,34568	0,05683	0,09695	0,03925	0,09286
2	0,09118	0,64981	0,06406	0,10546	0,04803	0,10867
3	0,08937	1,08352	0,06110	0,15322	0,04395	0,05935
4	0,08965	0,24853	0,05797	0,09946	0,04089	0,09631
5	0,05862	0,60274	0,06862	0,12161	0,05340	0,08444
average	0,07987	0,58606	0,06172	0,11534	0,04510	0,08833
std	0,01460	0,32539	0,00478	0,02325	0,00572	0,01839



Nilai terendah dari average/standar deviasi

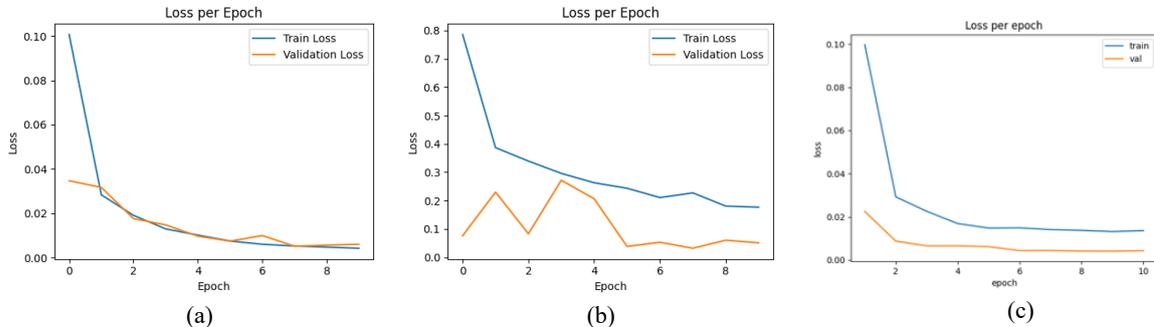
Pada Gambar 4.73, terdapat grafik *validation* pada ketiga model untuk percobaan menggunakan semua fitur. Pada model LSTM, terlihat bahwa *train loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. Untuk *validation loss*, mengalami penurunan lalu sedikit naik turun mengikuti pola *validation loss*. Pada model hybrid LSTM with Transformer, terdapat fluktuasi pada *validation loss* terutama pada *epoch* ke-5 hingga 7. Pada model FEDFormer WEB, baik *train loss* dan *validation loss* mengalami penurunan yang konsisten dan stabil tanpa ada fluktuasi.



Gambar 4.73 Grafik Validation loss Temperature Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

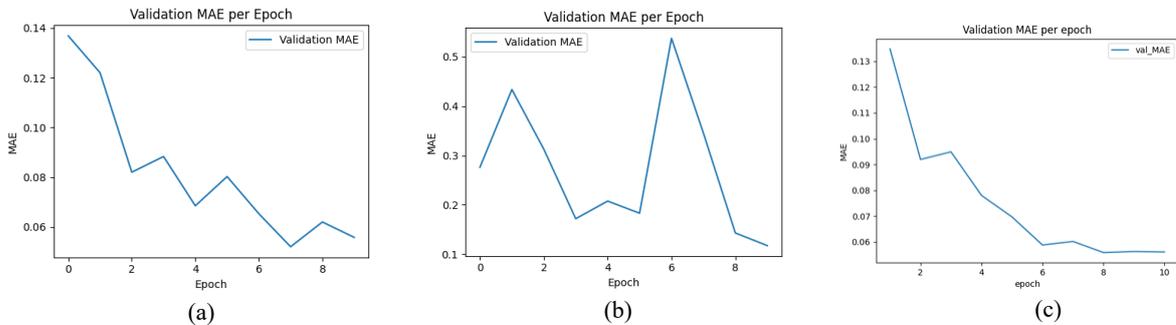
Pada Gambar 4.74, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur untuk ketiga model. Pada model LSTM, terlihat penurunan *train loss* dan *validation loss* secara signifikan. Nilai *validation loss* juga cenderung mengikuti nilai dari *train loss*. Pada model hybrid LSTM with Transformer, kurva *train loss* menurun secara bertahap lalu

validation loss terlihat mengalami fluktuasi dari *epoch* ke-0 hingga ke-5 sebelum akhirnya mulai stabil dari *epoch* ke-6 hingga *epoch* terakhir. Pada model FEDFormer WEB, terlihat bahwa nilai *train loss* dan *validation loss* mengalami penurunan dengan tren yang stabil.



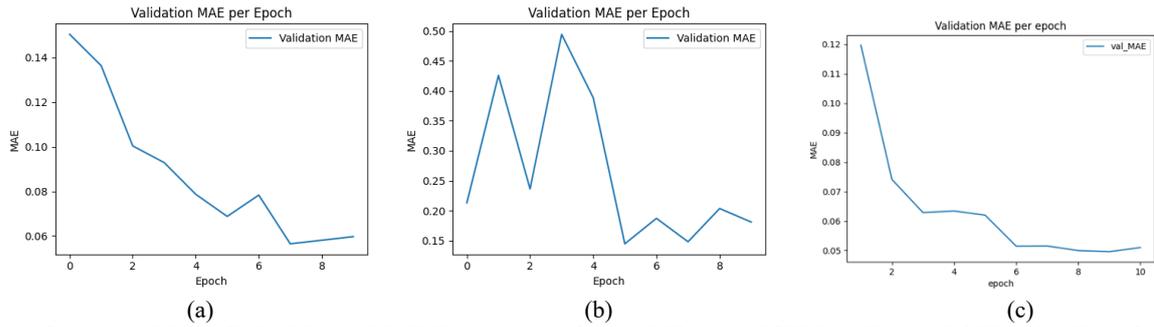
Gambar 4.74 Grafik Validation loss Temperature Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

Pada Gambar 4.75, terdapat *validation MAE* pada percobaan menggunakan semua fitur untuk ketiga model. Pada model LSTM, terlihat pola penurunan pada *validation MAE* yang cukup stabil dari *epoch* ke-0 hingga *epoch* ke-7 meski terdapat sedikit fluktuasi seperti pada *epoch* ke-2 hingga ke-4 dan *epoch* ke-4 hingga ke-7. Pada model hybrid LSTM with Transformer, terlihat fluktuasi pada *validation MAE* yang sangat tinggi. Hal itu terlihat dari *epoch* ke-0 hingga ke-7 dengan lonjakan signifikan pada *epoch* ke-6. Hal ini mengindikasikan bahwa model mengalami ketidakstabilan saat proses validasi MAE. Pada model FEDFormer WEB, terlihat bahwa nilai penurunan *validation MAE* cukup konsisten dari *epoch* pertama hingga *epoch* terakhir meski ada sedikit kenaikan dari *epoch* ke-2 hingga *epoch* ke-3.



Gambar 4.75 Grafik Validation MAE Temperature Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

Pada Gambar 4.76, terdapat *validation MAE* pada percobaan menggunakan *selected* fitur untuk ketiga model. Pada model LSTM, terlihat pola penurunan MAE yang konsisten dari awal hingga akhir *epoch* meski terdapat sedikit fluktuasi pada *epoch* ke-5 hingga ke-7. Pada model hybrid LSTM with Transformer, terdapat fluktuasi yang tajam dengan MAE yang melonjak hingga hampir menyentuh 0,5 pada *epoch* ke-3. Kemudian mengalami penurunan drastis. Pada model FEDFormer WEB, *validation MAE* mengalami penurunan yang konsisten dan cenderung stabil pada *epoch* ke-5 hingga akhir *epoch*.



Gambar 4.76 Grafik Validation MAE Temperature Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

b. Prediksi untuk GHI

Pada Tabel 4.55, dalam standar *scaler*, didapatkan nilai MAE dari keenam model yang dilakukan. Model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,24249 dan 0,21851 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan menggunakan semua fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00343 yang terlihat pada warna kuning. Pada percobaan menggunakan *selected* fitur, didapatkan bahwa model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,00141 yang terlihat pada warna kuning. Diantara kedua nilai tersebut, model FEDFormer WEB memiliki nilai standar deviasi MAE terendah.

Pada Tabel 4.56, dalam skala asli, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah dibanding model lainnya. Hal ini berlaku baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasil *average* MAE pada percobaan menggunakan semua fitur adalah 62,28274, sedangkan pada percobaan menggunakan *selected* fitur didapatkan nilai sebesar 56,12330. Kedua nilai tersebut ditandai dengan warna kuning sebagai indikator nilai terbaik dalam aspek *average* MAE untuk masing-masing skenario. Berdasarkan hasil tersebut, dapat disimpulkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, pada percobaan semua fitur didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai standar deviasi MAE terendah dibanding model lainnya yaitu sebesar 0,88086 yang terlihat pada warna kuning. Pada percobaan dengan *selected* fitur, didapatkan bahwa model FEDFormer WEB memiliki standar deviasi terendah dibanding model lainnya yaitu sebesar 0,36198 yang terlihat pada warna kuning. Dari kedua nilai tersebut, model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.56 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Tabel 4.55 Hasil MAE Skenario 5 GHI Standar Scaler

MAE						
GHI	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,27375	0,30616	0,26480	0,24679	0,24845	0,24526
2	0,26338	0,45307	0,25393	0,24133	0,25374	0,23737
3	0,25268	0,39656	0,25453	0,25287	0,23660	0,24270
4	0,26573	0,38187	0,25290	0,25244	0,25187	0,24125
5	0,25603	0,36549	0,26810	0,25488	0,25031	0,24589
average	0,26231	0,38063	0,25885	0,24966	0,24819	0,24249
std	0,00831	0,05310	0,00706	0,00554	0,00677	0,00343
GHI	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,23313	0,41733	0,25306	0,23045	0,23410	0,21749
2	0,25230	0,46160	0,24699	0,21979	0,23299	0,21863
3	0,24258	0,37209	0,24568	0,22896	0,23161	0,22095
4	0,24441	0,31555	0,24042	0,21612	0,23062	0,22388
5	0,24262	0,43001	0,24775	0,24142	0,23126	0,21163
average	0,24301	0,39932	0,24678	0,22735	0,23211	0,21851
std	0,00682	0,05680	0,00453	0,00992	0,00141	0,00456

Nilai terendah dari average/standar deviasi

Tabel 4.56 Hasil MAE Skenario 5 GHI Skala Asli

MAE						
GHI	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	70,37196	78,70296	68,01223	63,38679	63,81179	62,99392
2	67,70453	116,46769	65,21964	61,98481	65,17021	60,96682
3	64,95377	101,94228	65,37519	64,94755	60,76867	62,33643
4	68,30931	98,16536	64,95467	64,83642	64,69069	61,96182
5	65,81634	93,95328	68,85921	65,46290	64,28880	63,15471
average	67,43118	97,84631	66,48419	64,12370	63,74603	62,28274
std	2,13537	13,64971	1,81274	1,42285	1,73812	0,88086
GHI	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	59,92958	107,28122	64,99726	59,18806	60,12680	55,85972
2	64,85692	118,66073	63,43626	56,45244	59,84094	56,15254
3	62,35952	95,65132	63,10144	58,80717	59,48606	56,74877
4	62,82871	81,11690	61,74897	55,50826	59,23209	57,50058
5	62,36792	110,54008	63,63170	62,00682	59,39618	54,35489
average	62,46853	102,65005	63,38313	58,39255	59,61641	56,12330
std	1,75374	14,60052	1,16403	2,54710	0,36198	1,17108

MAE	
	Nilai terendah dari average/standar deviasi
	Nilai sampel terendah dari lima percobaan

Pada *Tabel 4.57*, dalam standar *scaler*, pada percobaan menggunakan semua fitur didapatkan bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 0,164070 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MSE terendah yaitu sebesar 0,15240 yang terlihat pada warna kuning. Jika dibandingkan antara kedua percobaan tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah.

Selain itu, dari segi standar deviasi MSE, didapatkan bahwa model FEDFormer WEB memiliki standar deviasi MSE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada percobaan menggunakan semua fitur, model ini memiliki standar deviasi MSE sebesar 0,00237. Sementara itu, pada percobaan menggunakan *selected* fitur, model ini memiliki standar deviasi MSE sebesar 0,00090 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai terendah dibanding model FEDFormer WEB pada percobaan menggunakan semua fitur.

Tabel 4.57 Hasil MSE Skenario 5 GHI Standar Scaler

MSE						
GHI	Semua Fitur					
	LSTM	Hybrid LSTM <i>with</i> Transformer	FEDFormer FEB	Hybrid FEDFormer FEB <i>with</i> LSTM	FEDFormer WEB	Hybrid FEDFormer WEB <i>with</i> LSTM
1	0,20127	0,23071	0,17132	0,16917	0,16315	0,17113
2	0,17297	0,35580	0,16820	0,16206	0,16675	0,16322
3	0,17220	0,43221	0,16545	0,16799	0,16055	0,16671
4	0,19018	0,34877	0,16630	0,17063	0,16444	0,16732
5	0,17899	0,27145	0,17008	0,16769	0,16547	0,17050
<i>average</i>	0,18312	0,32779	0,16827	0,16751	0,16407	0,16777
<i>std</i>	0,01244	0,07862	0,00247	0,00326	0,00237	0,00319
GHI	Selected Fitur					
	LSTM	Hybrid LSTM <i>with</i> Transformer	FEDFormer FEB	Hybrid FEDFormer FEB <i>with</i> LSTM	FEDFormer WEB	Hybrid FEDFormer WEB <i>with</i> LSTM
1	0,15852	0,30291	0,16738	0,15273	0,16026	0,15468
2	0,16929	0,33613	0,16551	0,15338	0,16052	0,15362
3	0,16523	0,22509	0,16476	0,15734	0,15902	0,15326
4	0,15976	0,23257	0,16188	0,15086	0,15914	0,15086
5	0,15923	0,29637	0,16536	0,16156	0,15836	0,14957
<i>average</i>	0,16241	0,27861	0,16498	0,15517	0,15946	0,15240
<i>std</i>	0,00468	0,04795	0,00199	0,00428	0,00090	0,00211

 Nilai terendah dari average/standar deviasi

Pada

Tabel 4.58, dalam skala asli, didapatkan hasil bahwa pada percobaan menggunakan semua fitur maka model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 10823,3788 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, model Hybrid FEDFormer WEB with LSTM memiliki nilai memiliki nilai *average* MSE terendah dibanding model lainnya yaitu sebesar 10053,324 yang terlihat pada warna kuning. Jika dibandingkan antara kedua percobaan tersebut, dapat disimpulkan bahwa model Hybrid FEDFormer WEB with LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah.

Selain itu, dalam aspek standar deviasi MSE, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada percobaan menggunakan semua fitur, standar deviasi MSE diperoleh sebesar 156,53530 yang terlihat pada warna kuning. Sementara itu, pada percobaan menggunakan *selected* fitur, standar deviasi MSE diperoleh sebesar 59,61348 yang terlihat pada warna kuning. Hal ini menunjukkan bahwa model FEDFormer WEB memiliki tingkat konsistensi yang lebih baik dalam memprediksi data, baik ketika menggunakan semua fitur maupun ketika dilakukan seleksi fitur. Secara khusus, pada percobaan dengan *selected* fitur, model FEDFormer WEB menunjukkan nilai standar deviasi MSE yang paling rendah dibandingkan seluruh model lainnya.

Tabel 4.58 Hasil MSE Skenario 5 GHI Skala Asli

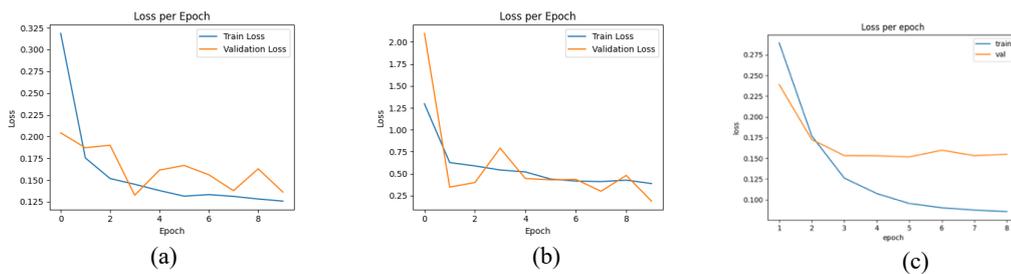
MSE						
GHI	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	13300,34309	15245,89690	11301,41953	11159,70860	10762,41626	11288,90380
2	11430,02781	23511,76410	11095,67983	10690,60530	11000,16986	10767,05891
3	11379,32495	28561,28640	10914,48306	11081,84630	10591,07054	10997,14356
4	12567,15604	23047,47250	10970,39344	11256,36140	10847,77496	11037,60226
5	11828,18903	17938,25590	11220,00711	11062,16250	10915,46240	11247,65145
<i>average</i>	12101,00818	21660,93510	11100,39659	11050,13680	10823,37880	11067,67200
<i>std</i>	821,76178	5195,52589	162,94463	215,02767	156,53530	210,70287
GHI	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	10475,44757	20017,05727	11041,50543	10075,45874	10572,18207	10203,61859
2	11187,10203	22211,79283	10918,49476	10118,02911	10589,15938	10134,03266
3	10918,40298	14874,06375	10868,81746	10379,48074	10490,42824	10110,43554
4	10557,42514	15368,96113	10678,96242	9951,70152	10498,38167	9952,06136
5	10522,19977	19584,70554	10908,72200	10657,75824	10446,72523	9866,47185
<i>average</i>	10732,11550	18411,31610	10883,30041	10236,48567	10519,37532	10053,32400
<i>std</i>	309,10173	3168,87002	131,22261	282,39237	59,61348	139,22278

Nilai terendah dari average/standar deviasi

Pada tabel Tabel 4.56, berdasarkan nilai MAE, diambil satu sample untuk model LSTM, Hybrid LSTM with Transformer, dan Hybrid FEDFormer WEB with LSTM agar dilakukan

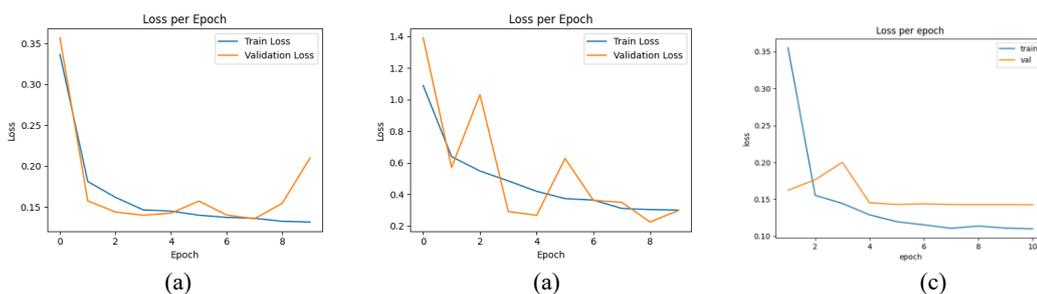
analisis yang terlihat pada warna hijau. Untuk percobaan menggunakan semua fitur, model LSTM akan mengambil percobaan ketiga, model Hybrid LSTM *with* Transformer akan mengambil percobaan pertama, dan model Hybrid FEDFormer WEB *with* LSTM akan mengambil percobaan kedua. Sementara itu, untuk percobaan menggunakan *selected* fitur, model LSTM akan mengambil percobaan pertama, model Hybrid LSTM *with* Transformer akan mengambil percobaan keempat, dan model Hybrid FEDFormer WEB *with* LSTM akan mengambil percobaan kelima. Pengambilan sample percobaan berdasarkan nilai terendah dari kelima percobaan yang telah dilakukan.

Pada Gambar 4.77, terdapat grafik *validation loss* dari percobaan menggunakan semua fitur untuk ketiga model. Pada model LSTM, *train loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. *Validation loss* menunjukkan pola fluktuatif dan mengalami kenaikan nilai *loss* pada *epoch* ke-1 hingga ke-2, *epoch* ke-3 hingga ke-4, *epoch* ke-4 hingga ke-5, dan *epoch* ke-7 hingga ke-8. Pada model hybrid LSTM *with* Transformer, *train loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. *Validation loss* terlihat tidak stabil dan memiliki puncak ekstrem pada *epoch* awal lalu memiliki beberapa lonjakan pada *epoch-epoch* selanjutnya. Pada model Hybrid FEDFormer WEB *with* LSTM, *train loss* mengalami penurunan dengan baik dari awal *epoch* hingga akhir *epoch*. *Validation loss* sempat memiliki nilai yang sama seperti *train loss* namun setelahnya mengalami sedikit fluktuasi dan cenderung stabil dari *epoch* ke-4 hingga akhir.



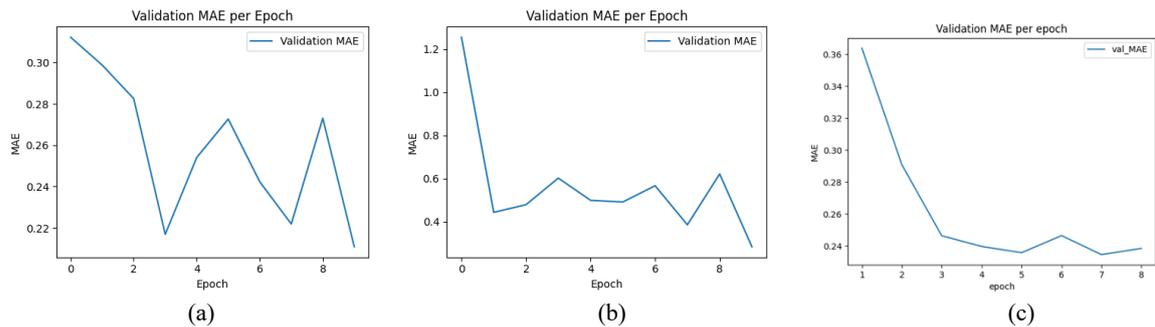
Gambar 4.77 Grafik *Validation loss* GHI Semua Fitur (a) LSTM (b) Hybrid LSTM *with* Transformer (c) Hybrid FEDFormer WEB *with* LSTM

Pada Gambar 4.78, terdapat grafik *validation loss* dari percobaan menggunakan *selected* fitur untuk ketiga model. Pada model LSTM, terlihat penurunan nilai *training loss* dan *validation loss* yang signifikan pada awal *training*. *Training loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. Untuk *validation loss*, terdapat sedikit fluktuasi dari *epoch* ke-4 hingga ke-5 kemudian menurun hingga *epoch* ke-7. Lalu mengalami kenaikan dari *epoch* ke-7 hingga *epoch* terakhir. Pada model hybrid LSTM *with* Transformer, terlihat bahwa *train loss* mengalami penurunan yang stabil dari awal *epoch* hingga akhir *epoch*. Untuk *validation loss*, menunjukkan fluktuasi besar dan tidak konsisten dari awal *training* hingga akhir *training*. Pada model Hybrid FEDFormer WEB *with* LSTM, *train loss* terus mengalami penurunan dan stabil. Untuk *validation loss* sempat mengalami kenaikan dari *epoch* ke-2 hingga ke-3. Lalu *epoch* berikutnya menurun dan mengalami kestabilan hingga akhir *epoch*.



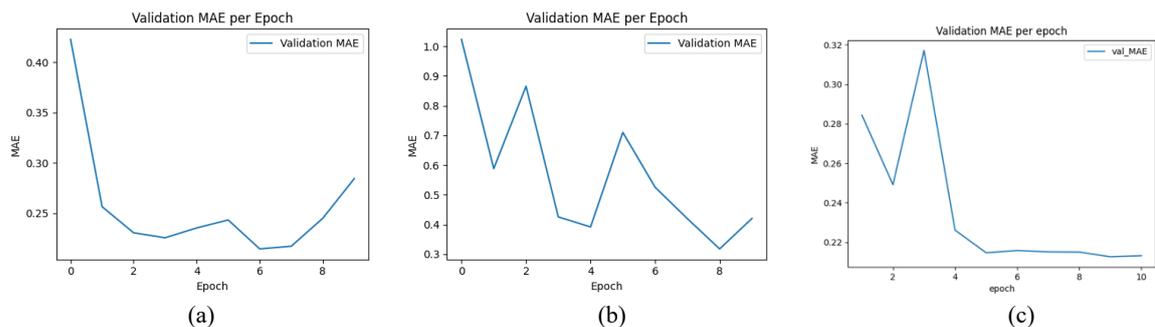
Gambar 4.78 Grafik Validation loss GHI Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.79, terdapat grafik *validation* MAE pada percobaan menggunakan semua fitur. Pada model LSTM, terlihat bahwa grafik menunjukkan penurunan dari *epoch* pertama hingga *epoch* ke-3. Kemudian, grafik mengalami berbagai fluktuasi dan menurun hingga akhir *training*. Pada model hybrid LSTM *with* transformer, terlihat bahwa pada *epoch* ke-2 hingga ke-8 terjadi fluktuasi yang sering terjadi sebelum akhirnya menurun dari *epoch* ke-8 hingga akhir *training*. Pada model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa kurva *validation* menurun dari awal *epoch* hingga *epoch* ke-5. Kemudian mengalami sedikit kenaikan dari *epoch* ke-5 hingga ke-6 lalu menurun dan cenderung stabil hingga akhir *training*.



Gambar 4.79 Grafik Validation MAE GHI Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM

Pada Gambar 4.80, terdapat grafik *validation* MAE untuk percobaan menggunakan *selected* fitur. Pada model LSTM, terlihat bahwa kurva *validation* menurun dari *epoch* awal hingga *epoch* ke-3, kemudian terjadi sedikit fluktuasi dari *epoch* ke-3 hingga ke-6 lalu mengalami kenaikan dari *epoch* ke-7 hingga akhir *training*. Pada model hybrid LSTM *with* Transformer terlihat bahwa kurva *validation* tidak menunjukkan kestabilan. Untuk model Hybrid FEDFormer WEB *with* LSTM, terlihat bahwa terjadi fluktuasi dari *epoch* ke-2 hingga ke-4 lalu stabil hingga *epoch* terakhir.



Gambar 4.80 Grafik Validation MAE GHI Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) Hybrid FEDFormer WEB with LSTM

c. Prediksi untuk *demand* listrik (KW)

Pada Tabel 4.59, dalam standar *scaler*, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun *selected* fitur. Hasilnya adalah 0,16465 dan 0,14752 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur

maupun percobaan menggunakan *selected* fitur. Hasilnya adalah 0,00273 dan 0,00317 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah.

Tabel 4.59 Hasil MAE Skenario 5 Demand Listrik Standar Scaler

MAE						
KW	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,16601	0,45566	0,18190	0,19123	0,16310	0,18647
2	0,19102	0,43852	0,19047	0,20751	0,16787	0,17906
3	0,19238	0,52321	0,16578	0,19670	0,16459	0,18366
4	0,19176	0,55125	0,17196	0,19754	0,16664	0,17615
5	0,17215	0,52846	0,18882	0,19445	0,16104	0,16310
average	0,18266	0,49942	0,17979	0,19749	0,16465	0,17769
std	0,01260	0,04929	0,01070	0,00611	0,00273	0,00908
KW	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,15010	0,39243	0,16368	0,18830	0,14684	0,16434
2	0,14565	0,43806	0,17887	0,17939	0,14521	0,17036
3	0,14065	0,39916	0,17197	0,17780	0,14637	0,16355
4	0,17136	0,47124	0,16550	0,16589	0,14609	0,15858
5	0,15139	0,29211	0,18054	0,17734	0,15308	0,16432
average	0,15183	0,39860	0,17211	0,17774	0,14752	0,16423
std	0,01170	0,06747	0,00761	0,00798	0,00317	0,00418

Nilai terendah dari average/standar deviasi

Pada Tabel 4.60, dalam skala asli, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah baik pada percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur. Hasilnya adalah 20,42159 dan 18,29667 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah. Untuk standar deviasi, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai standar deviasi MAE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur. Hasilnya adalah 0,33863 dan 0,39286 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah. Warna hijau pada Tabel 4.60 merupakan nilai MAE terendah dari kelima percobaan yang dilakukan pada masing-masing model. Nilai terendah tersebut akan menjadi sampel untuk dilakukan analisis.

Pada Tabel 4.61, dalam standar *scaler*, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,04709 dan 0,03931 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE

terendah. Untuk standar deviasi, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai standar deviasi terendah dibanding model lainnya baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 0,00143 dan 0,00156 yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MSE terendah dibanding model FEDFormer WEB pada percobaan menggunakan *selected* fitur.

Tabel 4.60 Hasil MAE Skenario 5 Demand Listrik Skala Asli

MAE						
KW	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	20,66601	56,72296	22,56146	23,71824	20,22939	23,12741
2	23,77984	54,58962	23,62450	25,73774	20,82153	22,20846
3	23,94852	65,13274	20,56164	24,39640	20,41450	22,77954
4	23,87136	68,62246	21,32801	24,50124	20,66883	21,84851
5	21,43026	65,78636	23,41891	24,11765	19,97368	20,22963
average	22,73920	62,17083	22,29890	24,49425	20,42159	22,03871
std	1,56833	6,13642	1,32677	0,75827	0,33863	1,12604
KW	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	18,68594	48,85243	20,30089	23,35452	18,21313	20,38278
2	18,13130	54,53193	22,18556	22,24920	18,01008	21,12946
3	17,50905	49,69014	21,32935	22,05321	18,15417	20,28550
4	21,33225	58,66247	20,52657	20,57496	18,11908	19,66812
5	18,84573	36,36425	22,39243	21,99516	18,98689	20,38084
average	18,90085	49,62024	21,34696	22,04541	18,29667	20,36934
std	1,45690	8,39944	0,94391	0,98996	0,39286	0,51885

Nilai terendah dari average/standar deviasi
 Nilai sampel terendah dari lima percobaan

Pada Tabel 4.62, dalam skala asli, didapatkan hasil bahwa model FEDFormer WEB memiliki nilai *average* MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada percobaan menggunakan semua fitur, model ini memiliki nilai *average* MSE sebesar 724,44574. Sementara itu, pada percobaan menggunakan *selected* fitur, model ini memiliki nilai *average* MSE sebesar 604,71874 yang terlihat pada warna kuning. Berdasarkan perbandingan antara kedua percobaan tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MSE terendah. Dalam aspek standar deviasi MSE, model FEDFormer WEB memiliki nilai standar deviasi MSE terendah dibanding model lainnya baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Hasilnya adalah 22,06033 pada percobaan menggunakan semua fitur dan 23,94015 pada percobaan menggunakan *selected* fitur yang terlihat pada warna kuning. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi terendah.

Tabel 4.61 Hasil MSE Skenario 5 Demand Listrik Standar Scaler

MSE						
KW	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,04981	0,31276	0,05623	0,06425	0,04676	0,06083
2	0,06290	0,35912	0,05783	0,07289	0,04845	0,05625
3	0,06783	0,42120	0,04685	0,06667	0,04734	0,05718
4	0,06361	0,45868	0,05111	0,06991	0,04810	0,05329
5	0,05304	0,45639	0,06044	0,06513	0,04481	0,04676
average	0,05944	0,40163	0,05449	0,06777	0,04709	0,05486
std	0,00764	0,06392	0,00546	0,00359	0,00143	0,00527
KW	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	0,03991	0,27450	0,04742	0,06159	0,03869	0,04856
2	0,03854	0,32277	0,05552	0,05614	0,03844	0,05031
3	0,03639	0,24690	0,05056	0,05603	0,03868	0,04725
4	0,04903	0,33177	0,04860	0,04963	0,03864	0,04509
5	0,04023	0,14393	0,05796	0,05604	0,04209	0,04855
average	0,04082	0,26398	0,05201	0,05588	0,03931	0,04795
std	0,00483	0,07561	0,00454	0,00423	0,00156	0,00193

Nilai terendah dari
average/standar deviasi

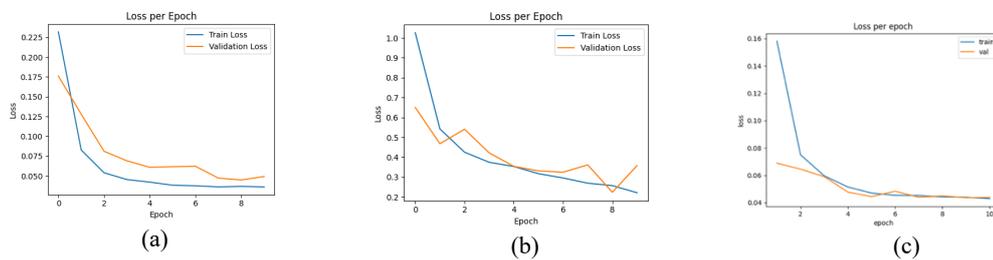
Tabel 4.62 Hasil MSE Skenario 5 Demand Listrik Skala Asli

MSE						
KW	Semua Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	771,95528	4846,75555	864,95705	988,33411	719,25861	935,83662
2	974,79568	5565,27402	889,70471	1121,37878	745,30741	865,30788
3	1051,20460	6527,31322	720,71770	1025,58801	728,29094	879,63107
4	985,73972	7108,13115	786,29849	1075,39954	739,97640	819,78214
5	821,94724	7072,58667	929,70566	1001,92161	689,39535	719,32500
average	921,12850	6224,01212	838,27672	1042,52441	724,44574	843,97654
std	118,38934	990,51859	84,02052	55,16068	22,06033	81,06775
KW	Selected Fitur					
	LSTM	Hybrid LSTM with Transformer	FEDFormer FEB	Hybrid FEDFormer FEB with LSTM	FEDFormer WEB	Hybrid FEDFormer WEB with LSTM
1	618,41645	4253,89808	729,48152	947,40966	595,25789	747,09837
2	597,25282	5001,89551	854,09893	863,58404	591,41976	773,90543
3	563,97440	3826,19790	777,80976	861,90876	595,09895	726,89564
4	759,86394	5141,34931	747,59435	763,55557	594,36244	693,62748
5	623,38314	2230,51013	891,58373	862,01981	647,45466	746,90888
average	632,57815	4090,77019	800,11366	859,69557	604,71874	737,68716
std	74,90182	1171,64868	69,86742	65,11928	23,94015	29,76191

MSE	
	Nilai terendah dari average/standar deviasi

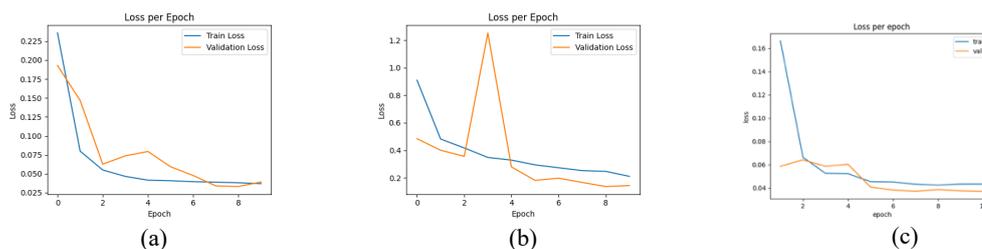
Pada Tabel 4.60, berdasarkan nilai MAE, diambil satu sample untuk model LSTM, Hybrid LSTM *with* Transformer, dan FEDFormer WEB agar dilakukan analisis yang terlihat pada warna hijau. Untuk percobaan menggunakan semua fitur, model LSTM akan mengambil percobaan pertama, model Hybrid LSTM *with* Transformer akan mengambil percobaan kedua, dan model FEDFormer WEB akan mengambil percobaan kelima. Sementara itu, untuk percobaan menggunakan *selected* fitur, model LSTM akan mengambil percobaan ketiga, model Hybrid LSTM *with* Transformer akan mengambil percobaan kelima, dan model FEDFormer WEB akan mengambil percobaan kedua. Pengambilan sampel percobaan berdasarkan nilai terendah dari kelima percobaan yang telah dilakukan.

Pada Gambar 4.81, terdapat grafik *validation loss* pada percobaan menggunakan semua fitur. Pada model LSTM, grafik menunjukkan penurunan baik pada *train loss* maupun pada *validation loss*. Pada model hybrid LSTM *with* Transformer, terlihat bahwa *train loss* mengalami penurunan dari awal *epoch* hingga akhir *epoch*. Untuk *validation loss*, terjadi fluktuasi dari *epoch* pertama hingga ke-4, dan *epoch* ke-6 hingga terakhir. Untuk model FEDFormer WEB, terlihat bahwa *train loss* menurun dari awal *epoch* hingga akhir lalu *validation loss* mengalami sedikit kenaikan dari *epoch* ke-5 hingga ke-6 sebelum akhirnya stabil hingga *epoch* terakhir dengan nilai mendekati *training loss*.



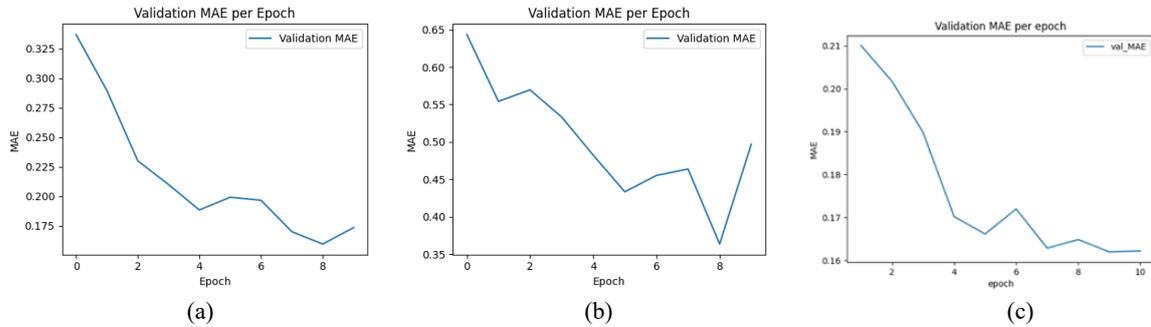
Gambar 4.81 Grafik Validation loss Demand Listrik Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

Pada Gambar 4.82, terdapat grafik *validation loss* pada percobaan menggunakan *selected* fitur. Untuk model LSTM, terlihat bahwa *train loss* mengalami penurunan dari awal dan mengalami kestabilan dari *epoch* ke-6 hingga akhir. *Validation loss* juga mengalami fluktuasi dari *epoch* ke-2 hingga ke-6 sebelum akhirnya cenderung stabil hingga *epoch* terakhir. Untuk model hybrid LSTM *with* Transformer, *train loss* mengalami penurunan dari awal hingga akhir *training* namun *validation loss* mengalami lonjakan kenaikan dari *epoch* ke-2 hingga ke-3 yang menunjukkan ketidakstabilan. Pada model FEDFormer WEB, kedua kurva mengalami penurunan secara konsisten tanpa adanya fluktuasi ekstrim.



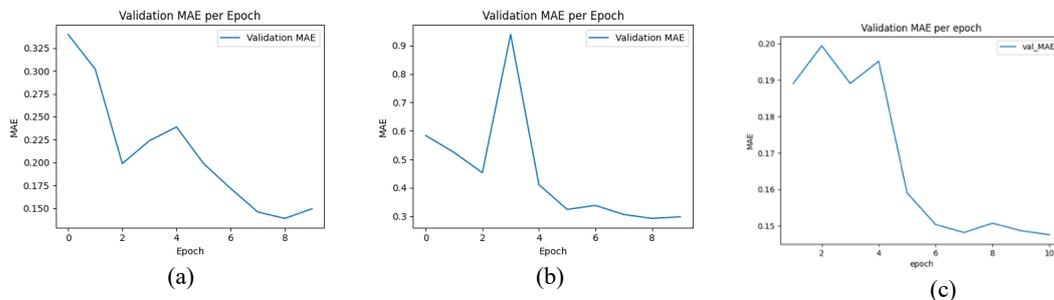
Gambar 4.82 Grafik Validation loss Demand Listrik Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

Pada Gambar 4.83, terdapat grafik *validation* MAE untuk percobaan menggunakan semua fitur. Pada model LSTM, terlihat bahwa kurva menurun dari *epoch* ke-0 hingga *epoch* ke-4 kemudian sedikit menaik lalu menurun lagi hingga *epoch* ke-8. Pada model hybrid LSTM *with* Transformer, terlihat bahwa kurva mengalami penurunan hingga *epoch* ke-5. Lalu mengalami sedikit kenaikan dan menurun lagi hingga *epoch* ke-8. Dari *epoch* ke-8 ke *epoch* terakhir mengalami kenaikan. Pada model FEDFormer WEB, terlihat fluktuasi pada *epoch* ke-5 dan ke-7 sebelum akhirnya stabil hingga *training* terakhir.



Gambar 4.83 Grafik Validation MAE Demand Listrik Semua Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

Pada Gambar 4.84, terdapat grafik *validation* MAE pada percobaan dengan *selected* fitur. Pada model LSTM, terlihat bahwa terjadi kenaikan dari *epoch* ke-2 hingga ke-4 kemudian mengalami penurunan hingga ke-8. Lalu dari *epoch* ke-8 menuju *epoch* terakhir, mengalami sedikit kenaikan. Pada model hybrid LSTM *with* Transformer, terdapat lonjakan tajam dari *epoch* ke-2 hingga *epoch* ke-3 yang terjadi karena anomali dan mengalami ketidakstabilan sebelum akhirnya mengalami penurunan hingga akhir *training*. Pada model FEDFormer WEB, terlihat bahwa terjadi fluktuasi dari awal *epoch* menuju *epoch* ke-4. Selanjutnya mengalami penurunan hingga akhir *training*.



Gambar 4.84 Grafik Validation MAE Demand Listrik Selected Fitur (a) LSTM (b) Hybrid LSTM with Transformer (c) FEDFormer WEB

4.3.6 Hasil Implementasi *Marine Predator Algorithm*

Pada *Marine Predator Algorithm* dilakukan pengujian menggunakan hasil prediksi berbagai target yang didapat dari *timeseries* forecasting berdasarkan nilai *average* MAE terendah dari keseluruhan model yang diuji. Pengambilan target *temperature*, diambil dari hasil prediksi menggunakan model FEDFormer WEB pada percobaan menggunakan semua fitur berdasarkan Tabel 4.52. Untuk pengambilan target GHI, diambil dari hasil prediksi menggunakan Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur berdasarkan Tabel 4.56. Untuk pengambilan target *demand* listrik (KW), diambil dari hasil prediksi menggunakan FEDFormer WEB pada percobaan menggunakan *selected* fitur berdasarkan Tabel 4.60.

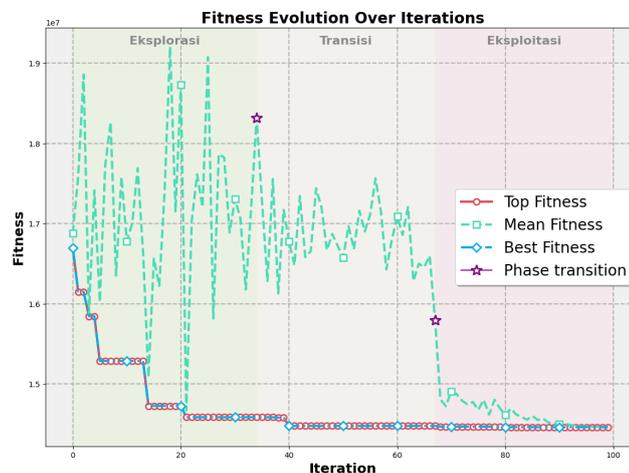
Pada *Tabel 4.63*, terlihat *total cost*, kapasitas pv, kapasitas BESS, harga PV, dan harga BESS yang didapat dari percobaan dengan MPA untuk mencari *total cost* terkecil. Didapatkan 11 chromosome terbaik dari 100 percobaan yang dilakukan. Terlihat hasil bahwa kapasitas PV 7303 dan kapasitas BESS 10689 memiliki *total cost* terkecil yaitu sebesar \$14.459.293,1528 NTD yang terlihat pada penanda warna kuning di nomor 11.

Tabel 4.63 Hasil Chromosome Terbaik

No	Total cost (NTD)	PV capacity (kW)	BESS capacity (kW)	Harga PV (NTD)	Harga BESS (NTD)
1	16.696.328,0413	8863	10577	10.687.987,4204	6.008.340,6209
2	16.149.455,2858	8492	10564	10.240.594,5136	5.908.860,7722
3	15.843.087,7296	8271	10608	9.974.088,2268	5.868.999,5028
4	15.285.766,8523	7889	10610	9.513.430,3012	5.772.336,5511
5	14.724.728,1045	7487	10680	9.028.654,1596	5.696.073,9449
6	14.583.271,9234	7422	10556	8.950.269,9576	5.633.001,9658
7	14.578.830,8380	7382	10700	8.902.033,5256	5.676.797,3124
8	14.475.106,4041	7311	10700	8.816.413,8588	5.658.692,5453
9	14.465.509,0396	7307	10690	8.811.590,2156	5.653.918,8240
10	14.462.965,2175	7305	10691	8.809.178,3940	5.653.786,8235
11	14.459.293,1528	7303	10689	8.806.766,5724	5.652.526,5804

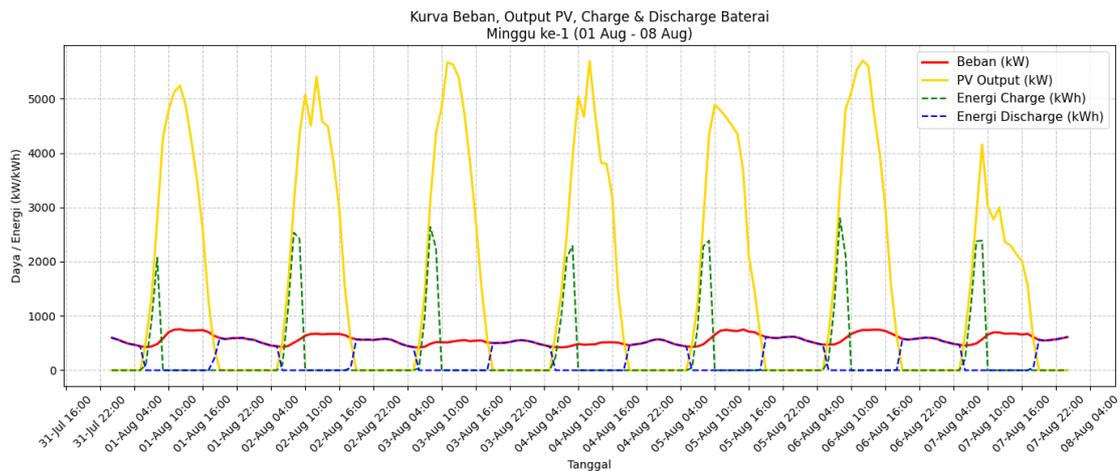
Pada *Gambar 4.85*, terkait hasil evolusi *fitness* selama iterasi dimana penelitian ini menggunakan 100 iterasi, didapatkan bahwa grafik ini menampilkan 3 metrik penting yaitu *top fitness*, *mean fitness*, dan *best fitness*. *Top fitness* adalah nilai *fitness* tertinggi dari elite chromosome. *Mean fitness* adalah rata-rata nilai *fitness* dari seluruh populasi. *Best fitness* adalah nilai *fitness* terbaik yang ditemukan selama ini dimana nilai *fitness* terbaik adalah yang memiliki value yang lebih rendah daripada *fitness* terbaik sebelumnya.

Pada fase eksplorasi, nilai *mean fitness* sangat fluktuatif karena mencerminkan pergerakan acak agen dalam pencarian *global*. Pada *top fitness* dan *best fitness* masih menunjukkan penurunan yang tajam yang menandakan bahwa ada *fitness* yang lebih rendah pada iterasi-iterasi berikutnya. Pada fase transisi, kurva *mean fitness* terlihat lebih stabil meski masih terdapat fluktuasi. Untuk *top fitness* dan *best fitness* mulai mengalami kestabilan. Pada fase eksploitasi, *mean fitness* menurun dan menjadi stabil mendekati *top fitness* dan *best fitness*. *Top fitness* dan *best fitness* juga mendekati konvergen dan mulai stagnan yang menandakan telah mencapai solusi optimal.



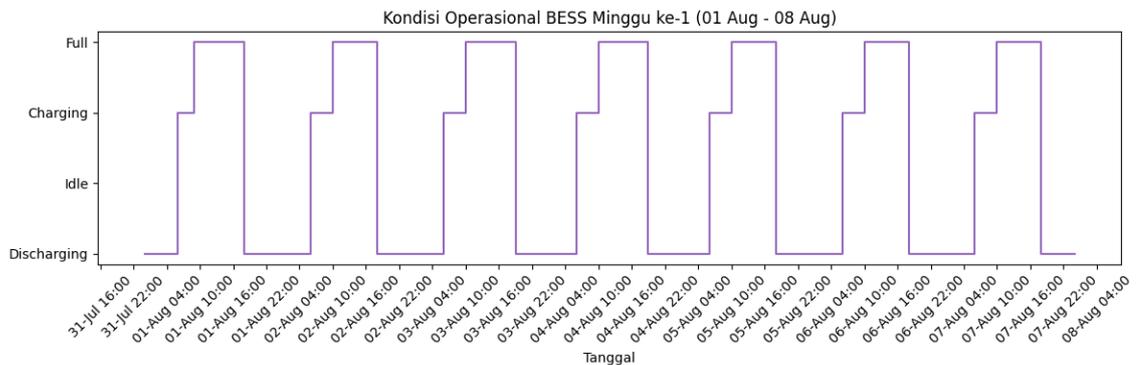
Gambar 4.85 Grafik Hasil MPA

Pada *Gambar 4.86*, merupakan grafik untuk menganalisis terkait beban listrik, *output* dari PV, dan kapan baterai *discharging* dan *charging* selama periode minggu pertama Bulan Agustus 2019. Pada sumbu x merupakan *datetime* dan pada sumbu y merupakan total daya per hari. Garis merah menunjukkan beban listrik yang dibutuhkan dan relatif stabil sepanjang hari. Sementara itu, garis kuning yang merupakan PV *output* menunjukkan pola harian untuk daya keluar PV dimana saat pagi hari hingga puncaknya yaitu tengah hari akan menaik lalu akan menurun hingga nol pada malam hari. Aktivitas *charging* baterai terlihat pada garis berwarna hijau yang terjadi saat *output* PV melebihi beban yang artinya ada kelebihan energi setelah memenuhi beban listrik yang dibutuhkan. Sebaliknya, untuk *discharging* baterai yang terlihat pada garis berwarna biru terjadi ketika *output* PV tidak tersedia atau *output* PV tidak mampu memenuhi kebutuhan listrik pada jam tersebut.



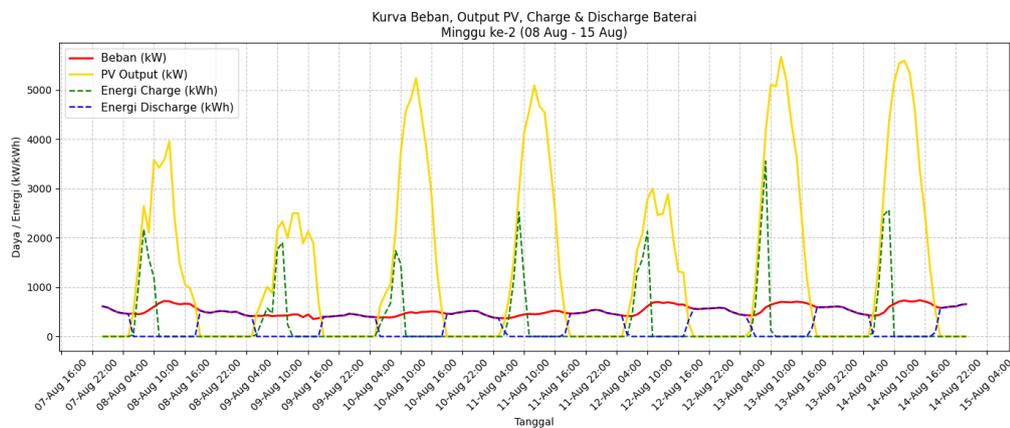
Gambar 4.86 Grafik Beban, *Output* PV, Charge dan Discharge Minggu Pertama

Pada *Gambar 4.87*, merupakan kondisi operasional BESS pada minggu pertama. Pada sumbu x merupakan *datetime* dan sumbu y merupakan kondisi apakah BESS dalam kondisi *discharging*, *idle*, *charging*, atau *full*. Pada minggu ini, baterai tidak ada yang mengalami *idle* yang artinya adalah kebutuhan listrik per hari terpenuhi. BESS dalam kondisi *discharging* artinya adalah BESS memasok daya listrik untuk memenuhi kebutuhan karena daya dari solar panel (PV) tidak mampu untuk memenuhinya. BESS dalam kondisi *charging* artinya adalah BESS mendapatkan daya listrik dari solar panel (PV) karena solar panel masih memiliki sisa daya setelah memenuhi kebutuhan listrik. BESS dalam kondisi *full* artinya adalah BESS berada dalam ambang batas yaitu 80% dan tidak boleh melakukan *charging* meski daya dari solar panel masih terdapat sisa karena untuk menjaga umur baterai.



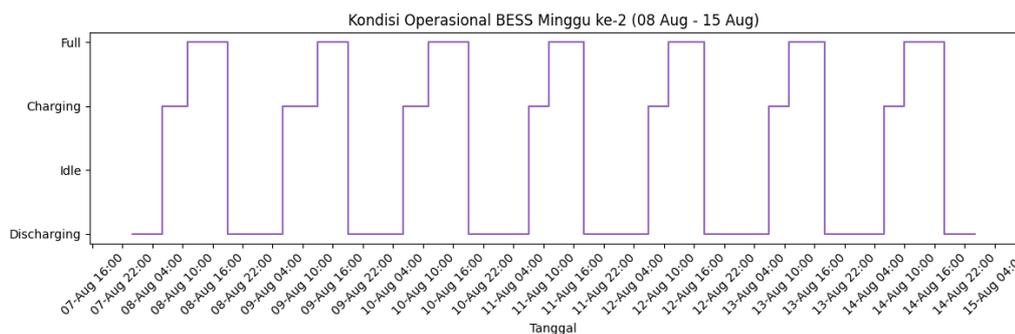
Gambar 4.87 Grafik Penjadwalan Charge dan Discharge BESS Minggu Pertama

Pada *Gambar 4.88*, merupakan grafik untuk menganalisis terkait beban listrik, *output* dari PV, dan kapan baterai *discharging* dan *charging* selama periode minggu kedua Bulan Agustus 2019. Pada sumbu x merupakan *datetime* dan pada sumbu y merupakan total daya per hari. Garis merah merupakan beban listrik yang dibutuhkan dan relatif stabil sepanjang hari. Garis kuning merupakan PV *output* yang menunjukkan pola harian untuk daya keluar PV dimana saat pagi hari hingga siang hari akan menaik lalu akan menurun hingga nol pada malam hari. Aktivitas *charging* baterai terlihat pada garis berwarna hijau yang terjadi saat *output* PV melebihi beban yang menunjukkan bahwa adanya sisa energi setelah memenuhi beban listrik yang dibutuhkan. Sebaliknya, untuk *discharging* baterai yang terlihat pada garis berwarna biru terjadi ketika *output* PV tidak tersedia atau *output* PV tidak mampu memenuhi kebutuhan listrik pada jam tersebut.



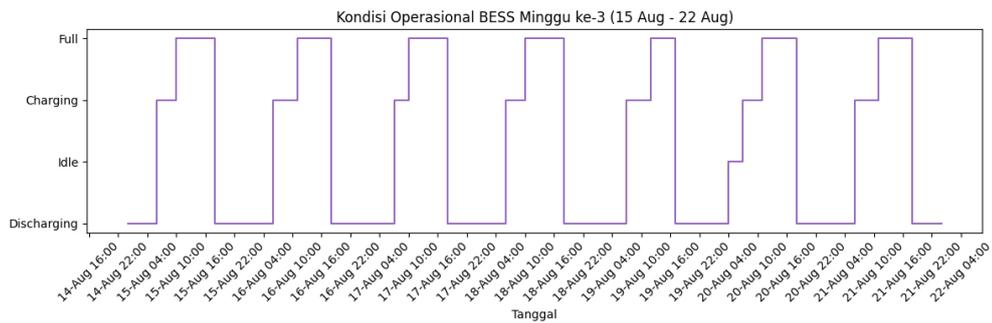
Gambar 4.88 Grafik Beban, Output PV, Charge dan Discharge Minggu Kedua

Pada *Gambar 4.89*, merupakan kondisi operasional BESS pada minggu kedua. Pada sumbu x merupakan *datetime* dan sumbu y merupakan kondisi apakah BESS dalam kondisi *discharging*, *idle*, *charging*, atau *full*. Pada minggu ini, baterai tidak ada yang mengalami kondisi *idle* yang artinya adalah kebutuhan listrik per hari terpenuhi. BESS dalam kondisi *discharging* artinya adalah BESS memasok daya listrik untuk memenuhi kebutuhan karena daya dari solar panel (PV) tidak mampu untuk memenuhinya. BESS dalam kondisi *charging* artinya adalah BESS mendapatkan daya listrik dari solar panel (PV) karena solar panel masih memiliki sisa daya setelah memenuhi kebutuhan listrik. BESS dalam kondisi *full* artinya adalah BESS berada dalam ambang batas yaitu 80% dan tidak boleh melakukan *charging* meski daya dari solar panel masih terdapat sisa karena untuk menjaga umur baterai.



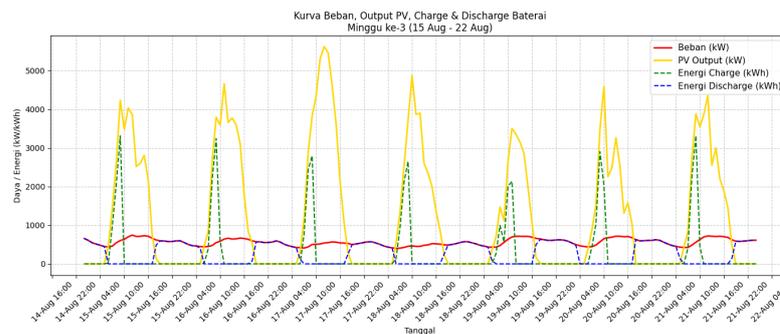
Gambar 4.89 Grafik Penjadwalan Charge dan Discharge BESS Minggu Kedua

Pada Gambar 4.90, merupakan kondisi operasional BESS pada minggu ketiga. Pada sumbu x merupakan *datetime* dan sumbu y merupakan kondisi apakah BESS dalam kondisi *discharging*, *idle*, *charging*, atau *full*. Pada minggu ini, baterai terdapat kondisi *idle*. Hal itu terlihat pada tanggal 20 Agustus sekitar pukul 04:00 hingga 06:00. Artinya adalah BESS tidak mampu memenuhi kebutuhan listrik pada jam tersebut karena BESS berada di ambang batas yaitu 20%. Oleh karena itu, bangunan akan mengalami *blackout* atau pemadaman listrik. BESS dalam kondisi *discharging* artinya adalah BESS memasok daya listrik untuk memenuhi kebutuhan karena daya dari solar panel (PV) tidak mampu untuk memenuhinya. BESS dalam kondisi *charging* artinya adalah BESS mendapatkan daya listrik dari solar panel (PV) karena solar panel masih memiliki sisa daya setelah memenuhi kebutuhan listrik. BESS dalam kondisi *full* artinya adalah BESS berada dalam ambang batas yaitu 80% dan tidak boleh melakukan *charging* meski daya dari solar panel masih terdapat sisa karena untuk menjaga umur baterai.



Gambar 4.90 Grafik Penjadwalan Charge dan Discharge BESS Minggu Ketiga

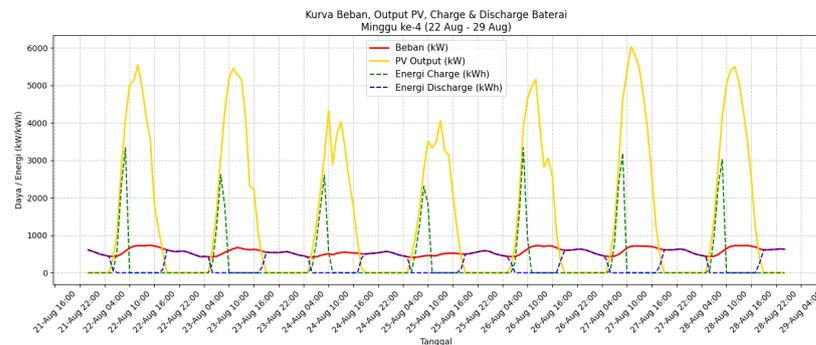
Pada Gambar 4.91, merupakan grafik untuk menganalisis terkait beban listrik, *output* dari PV, dan kapan baterai *discharging* dan *charging* selama periode minggu ketiga Bulan Agustus 2019. Pada sumbu x merupakan *datetime* dan pada sumbu y merupakan total daya per hari. Garis merah menunjukkan beban listrik yang dibutuhkan dan relatif stabil sepanjang hari. Sementara itu, garis kuning yang merupakan PV *output* menunjukkan pola harian untuk daya keluar PV dimana saat pagi hari hingga puncaknya yaitu tengah hari akan menaik lalu akan menurun hingga nol pada malam hari. Aktivitas *charging* baterai terlihat pada garis berwarna hijau yang terjadi saat *output* PV melebihi beban yang artinya ada kelebihan energi setelah memenuhi beban listrik yang dibutuhkan. Sebaliknya, untuk *discharging* baterai yang terlihat pada garis berwarna biru terjadi ketika *output* PV tidak tersedia atau *output* PV tidak mampu memenuhi kebutuhan listrik pada jam tersebut. Pada minggu ketiga ini, terlihat bahwa saat jam 4 hingga jam 6 pagi pada tanggal 20 Agustus, baterai tidak melakukan *discharging* dan tidak terdapat daya pada PV *output* sehingga tidak ada *supply* energi akibat baterai *idle*. Baterai tidak melakukan *discharging* karena baterai berada di ambang batas minimum daya yang harus dimiliki baterai yaitu sebesar 20%.



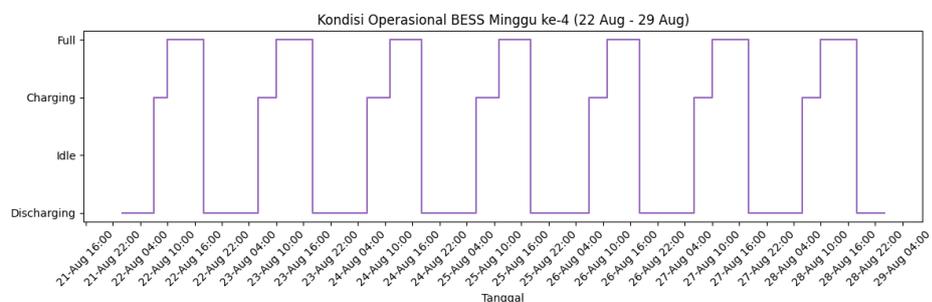
Gambar 4.91 Grafik Beban, Output PV, Charge dan Discharge Minggu Ketiga

Pada *Gambar 4.92*, merupakan grafik untuk menganalisis terkait beban listrik, *output* dari PV, dan kapan baterai *discharging* dan *charging* selama periode minggu keempat Bulan Agustus 2019. Pada sumbu x merupakan *datetime* dan pada sumbu y merupakan total daya per hari. Garis merah menunjukkan beban listrik yang dibutuhkan dan relatif stabil sepanjang hari. Sementara itu, garis kuning yang merupakan PV *output* menunjukkan pola harian untuk daya keluar PV dimana saat pagi hari hingga puncaknya yaitu tengah hari akan menaik lalu akan menurun hingga nol pada malam hari. Aktivitas *charging* baterai terlihat pada garis berwarna hijau yang terjadi saat *output* PV melebihi beban yang artinya ada kelebihan energi setelah memenuhi beban listrik yang dibutuhkan. Sebaliknya, untuk *discharging* baterai yang terlihat pada garis berwarna biru terjadi ketika *output* PV tidak tersedia atau *output* PV tidak mampu memenuhi kebutuhan listrik pada jam tersebut.

Pada *Gambar 4.93*, merupakan kondisi operasional BESS pada minggu keempat. Pada sumbu x merupakan *datetime* dan sumbu y merupakan kondisi apakah BESS dalam kondisi *discharging*, *idle*, *charging*, atau *full*. Pada minggu ini, baterai tidak ada yang mengalami *idle* yang artinya adalah kebutuhan listrik per hari terpenuhi. BESS dalam kondisi *discharging* artinya adalah BESS memasok daya listrik untuk memenuhi kebutuhan karena daya dari solar panel (PV) tidak mampu untuk memenuhinya. BESS dalam kondisi *charging* artinya adalah BESS mendapatkan daya listrik dari solar panel (PV) karena solar panel masih memiliki sisa daya setelah memenuhi kebutuhan listrik. BESS dalam kondisi *full* artinya adalah BESS berada dalam ambang batas yaitu 80% dan tidak boleh melakukan *charging* meski daya dari solar panel masih terdapat sisa karena untuk menjaga umur baterai.

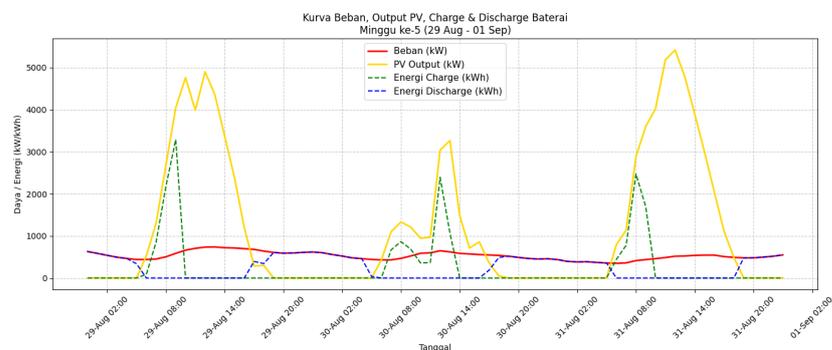


Gambar 4.92 Grafik Beban, Output PV, Charge dan Discharge Minggu Keempat



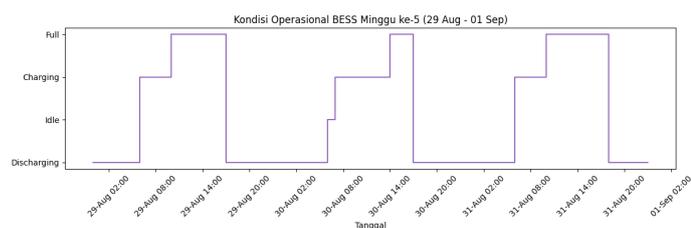
Gambar 4.93 Grafik Penjadwalan Charge dan Discharge BESS Minggu Keempat

Pada *Gambar 4.94*, merupakan grafik untuk menganalisis terkait beban listrik, *output* dari PV, dan kapan baterai *discharging* dan *charging* selama periode minggu kelima Bulan Agustus 2019. Pada sumbu x merupakan *datetime* dan pada sumbu y merupakan total daya per hari. Garis merah merupakan beban listrik yang dibutuhkan dan relatif stabil sepanjang hari. Garis kuning merupakan PV *output* yang menunjukkan pola harian untuk daya keluar PV dimana saat pagi hari hingga siang hari akan menaik lalu akan menurun hingga nol pada malam hari. Aktivitas *charging* baterai terlihat pada garis berwarna hijau yang terjadi saat *output* PV melebihi beban yang menunjukkan bahwa adanya sisa energi setelah memenuhi beban listrik yang dibutuhkan. Sebaliknya, untuk *discharging* baterai yang terlihat pada garis berwarna biru terjadi ketika *output* PV tidak tersedia atau *output* PV tidak mampu memenuhi kebutuhan listrik pada jam tersebut. Pada minggu kelima ini, terlihat bahwa saat jam 6 pagi pada tanggal 30 Agustus, baterai tidak melakukan *discharging* dan tidak terdapat daya pada PV *output* sehingga tidak ada *supply* energi akibat baterai *idle*. Baterai tidak melakukan *discharging* karena baterai berada di ambang batas minimum daya yang harus dimiliki baterai yaitu sebesar 20%



Gambar 4.94 Grafik Beban, Output PV, Charge dan Discharge Minggu Kelima

Pada *Gambar 4.95*, merupakan kondisi operasional BESS pada minggu ketiga. Pada sumbu x merupakan *datetime* dan sumbu y merupakan kondisi apakah BESS dalam kondisi *discharging*, *idle*, *charging*, atau *full*. Pada minggu ini, baterai terdapat kondisi *idle*. Hal itu terlihat pada tanggal 30 Agustus jam 06:00. Artinya adalah BESS tidak mampu memenuhi kebutuhan listrik pada jam tersebut karena BESS berada di ambang batas yaitu 20%. Oleh karena itu, bangunan akan mengalami *blackout* atau pemadaman listrik. BESS dalam kondisi *discharging* artinya adalah BESS memasok daya listrik untuk memenuhi kebutuhan karena daya dari solar panel (PV) tidak mampu untuk memenuhinya. BESS dalam kondisi *charging* artinya adalah BESS mendapatkan daya listrik dari solar panel (PV) karena solar panel masih memiliki sisa daya setelah memenuhi kebutuhan listrik. BESS dalam kondisi *full* artinya adalah BESS berada dalam ambang batas yaitu 80% dan tidak boleh melakukan *charging* meski daya dari solar panel masih terdapat sisa karena untuk menjaga umur baterai.



Gambar 4.95 Grafik Penjadwalan Charge dan Discharge BESS Minggu Kelima

4.4 Pembahasan

Pada bagian ini, akan dilakukan pembahasan terkait Subbab 4.3. Pembahasan juga dilengkapi dengan uji T-Test untuk validasi perubahan yang dilakukan pada model yang diuji dan pengujian menggunakan Pearson *Correlation*. Hasil Pearson *Correlation* pada dataset yang digunakan untuk penelitian ini adalah terdapat pada Tabel 4.64.

Tabel 4.64 Tabel pearson *correlation*

Fitur 1	Fitur 2	Nilai Pearson
Demand listrik (KW)	Temperature	0,666
Demand listrik (KW)	Dew point	0,6134
Demand listrik (KW)	GHI	0,3209
Demand listrik (KW)	Pressure	-0,5123
Demand listrik (KW)	Wind speed	-0,1493
Demand listrik (KW)	Relative Humidity	0,0467
Demand listrik (KW)	Precipitable water	0,5065
Demand listrik (KW)	Cloud type	0,075
Temperature	Dew point	0,9477
Temperature	GHI	0,4036
Temperature	Pressure	-0,8327
Temperature	Wind speed	-0,3118
Temperature	Relative Humidity	0,1562
Temperature	Precipitable water	0,7805
Temperature	Cloud type	0,041
Dew point	GHI	0,2086
Dew point	Pressure	-0,8579
Dew point	Wind speed	-0,3734
Dew point	Relative Humidity	0,4627
Dew point	Precipitable water	0,8642
Dew point	Cloud type	0,0963
GHI	Pressure	-0,1669
GHI	Wind speed	-0,0087
GHI	Relative Humidity	-0,4714
GHI	Precipitable water	0,0837
GHI	Cloud type	-0,1545
Pressure	Wind speed	0,2989
Pressure	Relative Humidity	-0,3441
Pressure	Precipitable water	-0,8074
Pressure	Cloud type	-0,1479
Wind speed	Relative Humidity	-0,2983
Wind speed	Precipitable water	-0,2575
Wind speed	Cloud type	0,0893
Relative Humidity	Precipitable water	0,5058
Relative Humidity	Cloud type	0,1812
Precipitable water	Cloud type	0,3474

4.4.1 Pembahasan Skenario 1

Pada pembahasan ini, dilakukan analisis terkait hasil skenario 1 yang telah dilakukan pada Subbab 4.3.1. Pembahasan akan terbagi menjadi 3 target prediksi yaitu *temperature*, GHI, dan *demand listrik (KW)*

a. Prediksi untuk *Temperature*

Pada Tabel 4.4, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average MAE* terendah baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur yaitu sebesar 0,15635 dan 0,16678. Dari kedua nilai tersebut, diketahui bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average MAE* terendah.

Akan dilakukan uji T-Test terkait model FEDFormer WEB pada percobaan menggunakan semua fitur dengan model FEDFormer WEB pada percobaan menggunakan *selected* fitur yang

terdapat pada *Tabel 4.65*. Pada kedua metrik, *t-stat* memiliki nilai negatif yaitu sebesar -1,92989 dan -2,01844 yang menandakan bahwa rata-rata model FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang buruk daripada model FEDFormer WEB menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,04487 pada metrik MAE dan 0,03912 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model juga signifikan berbeda secara statistik dan model FEDFormer WEB pada semua fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.65 Hasil T-Test Skenario 1 Temperature

Metrik	FEDFormer WEB Semua Fitur	FEDFormer WEB Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	0,15635	0,16678	-1,92989	0,04487
MSE	0,03985	0,04510	-2,01844	0,03912

Pada prediksi untuk *temperature* ini, dalam percobaan menggunakan *selected* fitur maka fitur yang digunakan hanya *dew point*, *precipitable water*, *demand listrik (KW)*, *GHI*, *wind speed*, dan *pressure* yang memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *relative humidity* dan *cloud type* tidak dipilih karena memiliki nilai *correlation* sebesar 0,156180 dan 0,041042.

Akan tetapi, pada *Tabel 4.64*, fitur *relative humidity* memiliki hubungan yang cukup kuat dengan fitur *dew point* dan *pressure* karena memiliki nilai *absolute* $\geq 0,2$ dengan nilai pearson masing-masing adalah 0,4627 dan -0,3441. Demikian pula untuk fitur *cloud type* yang memiliki hubungan cukup kuat dengan fitur *precipitable water* karena memiliki nilai *absolute* $\geq 0,2$ dengan nilai pearson *correlation*nya adalah 0,3474. Hasil temuan ini menunjukkan bahwa meski fitur *relative humidity* dan *cloud type* tidak memiliki korelasi tinggi terhadap variabel target yaitu *temperature*, akan tetapi keduanya berperan sebagai fitur pendukung pada fitur-fitur utama yang digunakan untuk prediksi target.

b. Prediksi untuk GHI

Pada *Tabel 4.8*, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur yaitu sebesar 63,74603 dan 59,61641. Dari kedua nilai tersebut, diketahui bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Pada *Tabel 4.66*, dilakukan uji coba *t-test* terkait model FEDFormer WEB baik pada percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai positif yaitu 5,20110 dan 4,05828 yang menandakan bahwa rata-rata FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang baik daripada model FEDFormer WEB menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,00041 pada metrik MAE dan 0,00182 pada metrik MSE. Nilai dari *P-value* ini menegaskan bahwa antara kedua model terdapat perbedaan yang signifikan secara statistik dan model FEDFormer WEB menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.66 Hasil T-Test Skenario 1 GHI

Metrik	FEDFormer WEB Semua Fitur	FEDFormer WEB Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	63,74603	59,61641	5,20110	0,00041
MSE	10823,37880	10519,37532	4,05828	0,00182

Pada prediksi untuk GHI ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *demand* listrik (KW), *dew point*, dan *relative humidity* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada Tabel 4.64. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand* listrik, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *demand* listrik (KW) memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Terakhir, untuk fitur *relative humidity* memiliki hubungan kuat dengan *precipitable water*, *dew point*, GHI, *wind speed*, dan *pressure*. Meskipun fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

c. Prediksi untuk *Demand* listrik (KW)

Pada Tabel 4.12, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 20,42159 dan 18,29667. Dari kedua nilai tersebut, diketahui bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terkecil.

Pada Tabel 4.67, dilakukan uji coba *t-test* terkait model FEDFormer WEB baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai positif yaitu 9,16090 dan 8,22370 yang menandakan bahwa rata-rata FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang baik daripada FEDFormer WEB menggunakan semua fitur. Selain itu, untuk *p-value* kedua metrik ini memiliki nilai dibawah 0,05 yaitu 0,00001 pada metrik MAE dan 0,00002 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model memiliki perbedaan yang signifikan secara statistik dan model FEDFormer WEB menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.67 Hasil T-Test Skenario 1 *Demand* Listrik

Metrik	FEDFormer WEB Semua Fitur	FEDFormer WEB Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	20,42159	18,29667	9,16090	0,00001
MSE	724,44574	604,71874	8,22370	0,00002

Pada prediksi untuk *demand* listrik (KW) ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *dew point*, *precipitable water*, GHI, dan *pressure* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *cloud type*, *relative humidity*, dan *wind speed* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada Tabel 4.64. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand* listrik, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Untuk fitur *precipitable*

water memiliki hubungan kuat dengan *demand* listrik (KW), *temperature*, *dew point*, *pressure*, *relative humidity*, *wind speed*, *precipitable water*, dan *cloud type*. Untuk fitur GHI memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, dan *relative humidity*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, *relative humidity*, *wind speed*, dan *precipitable water*. Meskipun fitur *cloud type*, *relative humidity*, dan *wind speed* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

4.4.2 Pembahasan Skenario 2

Pada pembahasan ini, dilakukan analisis terkait hasil skenario 2 yang telah dilakukan pada Subbab 4.3.2. Pembahasan akan terbagi menjadi 3 target prediksi yaitu *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada *Tabel 4.16*, dalam skala asli, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur yaitu sebesar 0,24462 dan 0,22336. Dari kedua nilai tersebut, didapatkan bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Pada *Tabel 4.68*, dilakukan uji T-Test terkait model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan semua fitur dengan model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai positif yaitu 1,31995 pada metrik MAE dan 0,89484 pada metrik MSE yang menunjukkan bahwa rata-rata Hybrid FEDFormer WEB *with* LSTM menggunakan *selected* fitur memberikan hasil yang baik daripada model Hybrid FEDFormer WEB *with* LSTM menggunakan semua fitur. Selain itu, untuk *p-value* memiliki nilai $\geq 0,05$ yaitu 0,11168 pada metrik MAE dan 0,19850 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model tidak memiliki perbedaan yang signifikan secara statistik meski model Hybrid FEDFormer WEB *with* LSTM dengan menggunakan *selected* fitur memiliki nilai *mean* MAE rendah.

Tabel 4.68 Hasil T-Test Skenario 2 *Temperature*

Metrik	Hybrid FEDFormer WEB <i>with</i> LSTM Semua Fitur	Hybrid FEDFormer WEB <i>with</i> LSTM <i>Selected</i> Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	0,24462	0,22336	1,31995	0,11168
MSE	0,09846	0,08833	0,89484	0,19850

Pada prediksi untuk *temperature* ini, dalam percobaan menggunakan *selected* fitur maka fitur yang digunakan hanya *dew point*, *precipitable water*, *demand* listrik (KW), GHI, *wind speed*, dan *pressure* yang memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *relative humidity* dan *cloud type* tidak dipilih karena memiliki nilai *correlation* yang lebih kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI,

dan *precipitable water*. Untuk fitur *precipitable water* memiliki hubungan kuat dengan *demand listrik (KW)*, *temperature*, *dew point*, *pressure*, *relative humidity*, *wind speed*, *precipitable water*, dan *cloud type*. Untuk fitur *demand listrik (KW)* memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur GHI memiliki hubungan kuat dengan *demand listrik*, *temperature*, *dew point*, dan *relative humidity*. Untuk fitur *wind speed* memiliki hubungan kuat dengan *temperature*, *dew point*, *pressure*, *relative humidity*, dan *precipitable water*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand listrik*, *temperature*, *dew point*, *relative humidity*, *wind speed*, dan *precipitable water*. Meskipun fitur *relative humidity* dan *cloud type* berkorelasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

b. Prediksi untuk GHI

Pada Tabel 4.20, dalam skala asli, terlihat bahwa model Hybrid FEDFormer WEB with LSTM memiliki nilai *average MAE* terendah baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur yaitu sebesar 62,28274 dan 56,12330. Dari kedua nilai tersebut, dapat terlihat bahwa model Hybrid FEDFormer WEB with LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average MAE* terendah.

Pada Tabel 4.69, dilakukan uji coba *t-test* terkait model Hybrid FEDFormer WEB with LSTM pada percobaan menggunakan semua fitur dan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai positif yaitu 9,39889 dan 8,98120 yang menandakan bahwa rata-rata model Hybrid FEDFormer WEB with LSTM menggunakan *selected* fitur memberikan hasil yang baik daripada model Hybrid FEDFormer WEB with LSTM menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,00001 pada metrik MAE dan 0,00001 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model terdapat perbedaan yang signifikan secara statistik dan model Hybrid FEDFormer WEB with LSTM menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.69 Hasil T-Test Skenario 2 GHI

Metrik	Hybrid FEDFormer WEB with LSTM Semua Fitur	Hybrid FEDFormer WEB with LSTM Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	62,28274	56,12330	9,39889	0,00001
MSE	11067,67200	10053,32400	8,98120	0,00001

Pada prediksi untuk GHI ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *demand listrik (KW)*, *dew point*, dan *relative humidity* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada Tabel 4.64. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand listrik*, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *demand listrik (KW)* memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand listrik (KW)*, *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Terakhir, untuk fitur *relative humidity* memiliki hubungan kuat dengan

precipitable water, dew point, GHI, wind speed, dan pressure. Meskipun fitur *precipitable water, wind speed, cloud type, dan pressure* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

c. Prediksi untuk *Demand* listrik (KW)

Pada *Tabel 4.24*, dalam skala asli, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 22,03871 dan 20,36934. Dari kedua nilai tersebut, diketahui bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* terkecil.

Pada *Tabel 4.70*, dilakukan uji coba *t-test* terkait model Hybrid FEDFormer WEB *with* LSTM baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada kedua metrik, *t-stat* menunjukkan nilai positif yaitu 3,01076 dan 2,75214 yang menunjukkan bahwa rata-rata Hybrid FEDFormer WEB *with* LSTM menggunakan *selected* fitur memberikan hasil yang baik daripada model Hybrid FEDFormer WEB *with* LSTM menggunakan semua fitur. Untuk *p-value* memiliki nilai $\leq 0,05$ yaitu sebesar 0,00840 pada metrik MAE dan 0,01249 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model memiliki perbedaan yang signifikan secara statistik dan model Hybrid FEDFormer WEB *with* LSTM menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.70 Hasil T-Test Skenario 2 Demand Listrik

Metrik	Hybrid FEDFormer WEB <i>with</i> LSTM Semua Fitur	Hybrid FEDFormer WEB <i>with</i> LSTM Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	22,03871	20,36934	3,01076	0,00840
MSE	843,97654	737,68716	2,75214	0,01249

Pada prediksi untuk *demand* listrik (KW) ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature, dew point, precipitable water, GHI, dan pressure* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *cloud type, relative humidity, dan wind speed* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand* listrik, *dew point, GHI, pressure, wind speed, dan precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature, demand* listrik (KW), *pressure, wind speed, relative humidity, GHI, dan precipitable water*. Untuk fitur *precipitable water* memiliki hubungan kuat dengan *demand* listrik (KW), *temperature, dew point, pressure, relative humidity, wind speed, precipitable waiter, dan cloud type*. Untuk fitur GHI memiliki hubungan kuat dengan *demand* listrik, *temperature, dew point, dan relative humidity*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand* listrik, *temperature, dew point, relative humidity, wind speed, dan precipitable water*. Meskipun fitur *cloud type, relative humidity, dan wind speed* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

4.4.3 Pembahasan Skenario 3

Pada pembahasan ini, dilakukan analisis terkait hasil skenario 3 yang telah dilakukan pada Subbab 4.3.3. Pembahasan ini akan terbagi menjadi 3 target prediksi yaitu *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada *Tabel 4.28*, dalam skala asli, terlihat bahwa model FEDFormer FEB memiliki nilai *average* MAE terendah baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 0,19772 dan 0,19415. Dari kedua nilai tersebut, diketahui bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Pada *Tabel 4.71*, dilakukan uji T-test terkait model FEDFormer FEB pada percobaan menggunakan semua fitur dengan model FEDFormer FEB pada percobaan menggunakan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai positif yaitu 0,48321 pada metrik MAE dan 0,45875 pada metrik MSE yang menunjukkan bahwa rata-rata FEDFormer FEB menggunakan *selected* fitur memberikan hasil yang baik daripada model FEDFormer FEB menggunakan semua fitur. Selain itu, untuk *p-value* memiliki nilai $\geq 0,05$ yaitu 0,32095 pada metrik MAE dan 0,32931 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model tidak memiliki perbedaan yang signifikan secara statistik meski model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai *mean* MAE rendah.

Tabel 4.71 Hasil T-Test Skenario 3 *Temperature*

Metrik	FEDFormer FEB Semua Fitur	FEDFormer FEB Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	0,19772	0,19415	0,48321	0,32095
MSE	0,06383	0,06172	0,45875	0,32931

Pada prediksi untuk *temperature* ini, dalam percobaan menggunakan *selected* fitur maka fitur yang digunakan hanya *dew point*, *precipitable water*, *demand* listrik (KW), GHI, *wind speed*, dan *pressure* yang memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *relative humidity* dan *cloud type* tidak dipilih karena memiliki nilai *correlation* yang lebih kecil.

Analisis dilakukan sama seperti sebelumnya yaitu melakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Untuk fitur *precipitable water* memiliki hubungan kuat dengan *demand* listrik (KW), *temperature*, *dew point*, *pressure*, *relative humidity*, *wind speed*, *precipitable water*, dan *cloud type*. Untuk fitur *demand* listrik (KW) memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur GHI memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, dan *relative humidity*. Untuk fitur *wind speed* memiliki hubungan kuat dengan *temperature*, *dew point*, *pressure*, *relative humidity*, dan *precipitable water*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, *relative humidity*, *wind speed*, dan *precipitable water*. Meskipun fitur *relative humidity* dan *cloud type* berkorelasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai

yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

b. Prediksi untuk GHI

Pada *Tabel 4.32*, dalam skala asli, terlihat bahwa model Hybrid FEDFormer FEB *with* LSTM memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua fitur maupun percobaan menggunakan *selected* fitur yaitu sebesar 64,12370 dan 58,39255. Dari kedua nilai tersebut, dapat diketahui bahwa model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Pada *Tabel 4.72*, dilakukan uji coba *t-test* terkait model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan semua fitur dan model Hybrid FEDFormer FEB *with* LSTM pada percobaan menggunakan *selected* fitur. Pada kedua metrik, *t-stat* menghasilkan nilai positif yaitu sebesar 4,39243 dan 5,12588 yang menunjukkan bahwa rata-rata model Hybrid FEDFormer FEB *with* LSTM menggunakan *selected* fitur memberikan hasil yang baik daripada model Hybrid FEDFormer FEB *with* LSTM menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai $\leq 0,05$ yaitu 0,00116 pada metrik MAE dan 0,00045 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model terdapat perbedaan yang signifikan secara statistik dan model Hybrid FEDFormer FEB *with* LSTM menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.72 Hasil T-Test Skenario 3 GHI

Metrik	Hybrid FEDFormer FEB <i>with</i> LSTM Semua Fitur	Hybrid FEDFormer FEB <i>with</i> LSTM <i>Selected</i> Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	64,12370	58,39255	4,39243	0,00116
MSE	11050,13682	10236,48567	5,12588	0,00045

Pada prediksi untuk GHI ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *demand* listrik (KW), *dew point*, dan *relative humidity* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand* listrik, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *demand* listrik (KW) memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Terakhir, untuk fitur *relative humidity* memiliki hubungan kuat dengan *precipitable water*, *dew point*, GHI, *wind speed*, dan *pressure*. Meskipun fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

c. Prediksi untuk *Demand* listrik (KW)

Pada *Tabel 4.36*, dalam skala asli, terlihat bahwa model FEDFormer FEB memiliki nilai *average MAE* yang lebih kecil baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 22,29890 dan 21,34696. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer FEB pada percobaan menggunakan *selected* fitur memiliki nilai *average MAE* yang lebih kecil.

Pada *Tabel 4.73*, dilakukan uji coba *t-test* terkait model FEDFormer FEB baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada kedua metrik, *t-test* menunjukkan nilai positif yaitu sebesar 1,30728 dan 0,78093 yang menunjukkan bahwa rata-rata FEDFormer FEB pada percobaan menggunakan *selected* fitur memberikan hasil yang baik daripada model FEDFormer FEB pada percobaan menggunakan semua fitur. Untuk *p-value* memiliki nilai $\geq 0,05$ yaitu 0,11372 pada metrik MAE dan 0,22867 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model tidak memiliki perbedaan yang signifikan secara statistik meski model FEDFormer FEB dengan menggunakan *selected* fitur memiliki nilai *mean MAE* rendah.

Tabel 4.73 Hasil T-Test Skenario 3 Demand Listrik

Metrik	FEDFormer FEB Semua Fitur	FEDFormer FEB Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	22,29890	21,34696	1,30728	0,11372
MSE	838,27672	800,11366	0,78093	0,22867

Pada prediksi untuk *demand* listrik (KW) ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *dew point*, *precipitable water*, GHI, dan *pressure* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *cloud type*, *relative humidity*, dan *wind speed* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand* listrik, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Untuk fitur *precipitable water* memiliki hubungan kuat dengan *demand* listrik (KW), *temperature*, *dew point*, *pressure*, *relative humidity*, *wind speed*, *precipitable waiter*, dan *cloud type*. Untuk fitur GHI memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, dan *relative humidity*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, *relative humidity*, *wind speed*, dan *precipitable water*. Meskipun fitur *cloud type*, *relative humidity*, dan *wind speed* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

4.4.4 Pembahasan Skenario 4

Pada pembahasan ini, dilakukan analisis terkait hasil skenario 4, yang telah dilakukan pada Subbab 4.3.4. Pembahasan ini akan terbagi menjadi 3 target prediksi yaitu *temperature*, GHI, dan *demand* listrik (KW).

a. Prediksi untuk *Temperature*

Pada *Tabel 4.40*, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average MAE* terendah baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 0,15635 dan 0,16678. Dari kedua nilai tersebut, di ketahui bahwa

model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai *average* MAE terendah.

Pada *Tabel 4.74*, dilakukan uji T-Test terkait model FEDFormer WEB pada percobaan menggunakan semua fitur dengan model FEDFormer WEB pada percobaan menggunakan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai negatif yaitu -1,92989 dan -2,01844 yang menandakan bahwa rata-rata model FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang buruk daripada model FEDFormer WEB menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,04487 pada metrik MAE dan 0,03912 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model juga signifikan berbeda secara statistik dan model FEDFormer WEB pada semua fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.74 Hasil T-Test Skenario 4 Temperature

Metrik	FEDFormer WEB Semua Fitur	FEDFormer WEB Selected Fitur	<i>t-stat</i>	<i>p-value</i> (one-tail)
MAE	0,15635	0,16678	-1,92989	0,04487
MSE	0,03985	0,04510	-2,01844	0,03912

Pada prediksi untuk *temperature* ini, dalam percobaan menggunakan *selected* fitur maka fitur yang digunakan hanya *dew point*, *precipitable water*, *demand* listrik (KW), GHI, *wind speed*, dan *pressure* yang memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *relative humidity* dan *cloud type* tidak dipilih karena memiliki nilai *correlation* sebesar 0,156180 dan 0,041042.

Namun demikian, pada *Tabel 4.64*, fitur *relative humidity* memiliki hubungan yang cukup kuat dengan fitur *dew point* dan *pressure* karena memiliki nilai *absolute* $\geq 0,2$ dengan nilai pearson masing-masing adalah 0,4627 dan -0,3441. Demikian pula untuk fitur *cloud type* yang memiliki hubungan cukup kuat dengan fitur *precipitable water* karena memiliki nilai *absolute* $\geq 0,2$ dengan nilai pearson *correlation*nya adalah 0,3474. Hasil temuan ini menunjukkan bahwa meski fitur *relative humidity* dan *cloud type* tidak memiliki korelasi tinggi terhadap variabel target yaitu *temperature*, akan tetapi keduanya berperan sebagai fitur pendukung pada fitur-fitur utama yang digunakan untuk prediksi target.

b. Prediksi untuk GHI

Pada *Tabel 4.44*, dalam skala asli, terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM memiliki nilai *average* MAE terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 62,28274 dan 56,12330. Dari kedua nilai tersebut, dapat terlihat bahwa model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki nilai *average* MAE terendah.

Pada *Tabel 4.75*, dilakukan uji coba *t-test* terkait model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan semua fitur dan *selected* fitur. Pada kedua metrik, *t-stat* memiliki nilai positif yaitu 9,39889 dan 8,98120 yang menandakan bahwa rata-rata model Hybrid FEDFormer WEB *with* LSTM menggunakan *selected* fitur memberikan hasil yang baik daripada model Hybrid FEDFormer WEB *with* LSTM menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,00001 pada metrik MAE dan 0,00001 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model terdapat perbedaan yang signifikan secara statistik dan model Hybrid FEDFormer WEB *with* LSTM menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.75 Hasil T-Test Skenario 4 GHI

Metrik	Hybrid FEDFormer WEB with LSTM Semua Fitur	Hybrid FEDFormer WEB with LSTM Selected Fitur	t-stat	p-value (one-tail)
MAE	62,28274	56,12330	9,39889	0,00001
MSE	11067,67200	10053,32400	8,98120	0,00001

Pada prediksi untuk GHI ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *demand listrik (KW)*, *dew point*, dan *relative humidity* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada Tabel 4.64. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand listrik*, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *demand listrik (KW)* memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand listrik (KW)*, *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Terakhir, untuk fitur *relative humidity* memiliki hubungan kuat dengan *precipitable water*, *dew point*, GHI, *wind speed*, dan *pressure*. Meskipun fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

c. Prediksi untuk *Demand listrik (KW)*

Pada Tabel 4.48, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average MAE* terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 20,42159 dan 18,29667. Dari kedua nilai tersebut, diketahui bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average MAE* terkecil.

Pada Tabel 4.76, dilakukan uji coba t-test terkait model FEDFormer WEB baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur. Pada kedua metrik, t-stat memiliki nilai positif yaitu 9,16090 dan 8,22370 yang menandakan bahwa rata-rata FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang baik daripada FEDFormer WEB menggunakan semua fitur. Selain itu, untuk p-value kedua metrik ini memiliki nilai dibawah 0,05 yaitu 0,00001 pada metrik MAE dan 0,00002 pada metrik MSE. Nilai dari p-value ini menegaskan bahwa antara kedua model memiliki perbedaan yang signifikan secara statistik dan model FEDFormer WEB menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Tabel 4.76 Hasil T-Test Skenario 4 Demand Listrik

Metrik	FEDFormer WEB Semua Fitur	FEDFormer WEB Selected Fitur	t-stat	p-value (one-tail)
MAE	20,42159	18,29667	9,16090	0,00001
MSE	724,44574	604,71874	8,22370	0,00002

Pada prediksi untuk *demand listrik (KW)* ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *dew point*, *precipitable water*, GHI, dan *pressure*

karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *cloud type*, *relative humidity*, dan *wind speed* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand listrik*, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand listrik* (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Untuk fitur *precipitable water* memiliki hubungan kuat dengan *demand listrik* (KW), *temperature*, *dew point*, *pressure*, *relative humidity*, *wind speed*, *precipitable water*, dan *cloud type*. Untuk fitur GHI memiliki hubungan kuat dengan *demand listrik*, *temperature*, *dew point*, dan *relative humidity*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand listrik*, *temperature*, *dew point*, *relative humidity*, *wind speed*, dan *precipitable water*. Meskipun fitur *cloud type*, *relative humidity*, dan *wind speed* berkorelasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

4.4.5 Pembahasan Skenario 5

Pada pembahasan ini, dilakukan analisis terkait hasil skenario 5 yang telah dilakukan pada Subbab 4.3.5. Pembahasan akan terbagi menjadi 3 target prediksi yaitu *temperature*, GHI, dan *demand listrik* (KW).

a. Prediksi untuk *Temperature*

Pada *Tabel 4.52*, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average MAE* terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 0,15635 dan 0,16678. Dari kedua nilai tersebut, didapatkan hasil bahwa model FEDFormer WEB pada percobaan menggunakan semua fitur memiliki nilai standar deviasi MAE terendah.

Analisis untuk *T-test* sama dengan subbab 4.4.4 untuk prediksi pada target *temperature*. Hal tersebut disebabkan karena membandingkan antara dua model terbaik yaitu model FEDFormer WEB pada percobaan dengan semua fitur maupun dengan *selected* fitur. Pada *Tabel 4.74*, *t-stat* memiliki nilai negatif pada kedua metrik yaitu -1,92989 dan -2,01844 yang menandakan bahwa rata-rata model FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang buruk daripada model FEDFormer WEB menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,04487 pada metrik MAE dan 0,03912 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model juga signifikan berbeda secara statistik dan model FEDFormer WEB pada semua fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Pada prediksi untuk *temperature* ini, dalam percobaan menggunakan *selected* fitur maka fitur yang digunakan hanya *dew point*, *precipitable water*, *demand listrik* (KW), GHI, *wind speed*, dan *pressure* yang memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *relative humidity* dan *cloud type* tidak dipilih karena memiliki nilai *correlation* sebesar 0,156180 dan 0,041042.

Namun demikian, pada *Tabel 4.64*, fitur *relative humidity* memiliki hubungan yang cukup kuat dengan fitur *dew point* dan *pressure* karena memiliki nilai *absolute* $\geq 0,2$ dengan nilai *pearson* masing-masing adalah 0,4627 dan -0,3441. Demikian pula untuk fitur *cloud type* yang memiliki hubungan cukup kuat dengan fitur *precipitable water* karena memiliki nilai *absolute* $\geq 0,2$ dengan nilai *pearson correlation*nya adalah 0,3474. Hasil temuan ini menunjukkan bahwa

meski fitur *relative humidity* dan *cloud type* tidak memiliki korelasi tinggi terhadap variabel target yaitu *temperature*, akan tetapi keduanya berperan sebagai fitur pendukung pada fitur-fitur utama yang digunakan untuk prediksi target.

b. Prediksi untuk GHI

Pada *Tabel 4.56*, dalam skala asli, terlihat bahwa model Hybrid FEDFormer WEB *with LSTM* memiliki nilai *average MAE* terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 62,28274 dan 56,12330. Dari kedua nilai tersebut, dapat terlihat bahwa model Hybrid FEDFormer WEB *with LSTM* pada percobaan menggunakan *selected* fitur memiliki nilai *average MAE* terendah.

Analisis untuk *T-test* sama dengan subab 4.4.4 untuk prediksi pada target GHI. Hal tersebut disebabkan karena membandingkan antara dua model terbaik yaitu model Hybrid FEDFormer WEB *with LSTM* pada percobaan dengan semua fitur maupun dengan *selected* fitur. Pada *Tabel 4.75*, *t-stat* memiliki nilai positif pada kedua matrik yaitu 9,39889 dan 8,98120 yang menandakan bahwa rata-rata model Hybrid FEDFormer WEB *with LSTM* menggunakan *selected* fitur memberikan hasil yang baik daripada model Hybrid FEDFormer WEB *with LSTM* menggunakan semua fitur. Selain itu, untuk *p-value*, kedua metrik memiliki nilai dibawah 0,05 yaitu 0,00001 pada metrik MAE dan 0,00001 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model terdapat perbedaan yang signifikan secara statistik dan model Hybrid FEDFormer WEB *with LSTM* menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Pada prediksi untuk GHI ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *demand listrik (KW)*, *dew point*, dan *relative humidity* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* tidak dipilih karena memiliki nilai *absolute* yang kecil.

Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada *Tabel 4.64*. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand listrik*, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *demand listrik (KW)* memiliki hubungan kuat dengan fitur *temperature*, *dew point*, GHI, *pressure*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand listrik (KW)*, *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Terakhir, untuk fitur *relative humidity* memiliki hubungan kuat dengan *precipitable water*, *dew point*, GHI, *wind speed*, dan *pressure*. Meskipun fitur *precipitable water*, *wind speed*, *cloud type*, dan *pressure* berkorelasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

c. Prediksi untuk *Demand listrik (KW)*

Pada *Tabel 4.60*, dalam skala asli, terlihat bahwa model FEDFormer WEB memiliki nilai *average MAE* terendah baik dari percobaan menggunakan semua fitur maupun dengan *selected* fitur yaitu sebesar 20,42159 dan 18,29667. Dari kedua nilai tersebut, diketahui bahwa model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki nilai *average MAE* terkecil.

Analisis untuk *T-test* sama dengan subab 4.4.4 untuk prediksi pada target GHI. Hal tersebut disebabkan karena membandingkan antara dua model terbaik yaitu model FEDFormer WEB

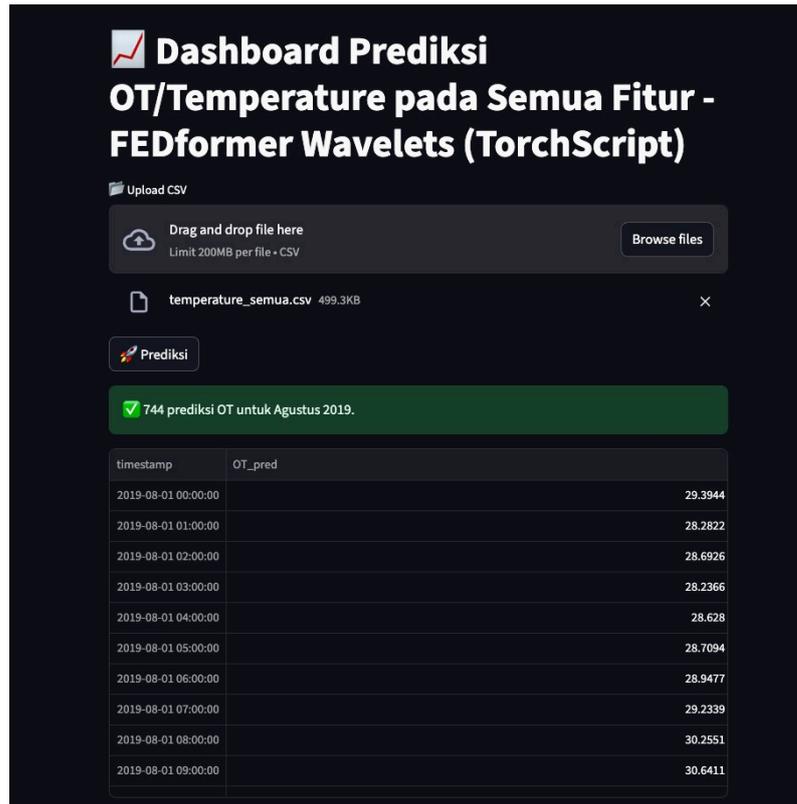
pada percobaan dengan semua fitur maupun dengan *selected* fitur. Pada Tabel 4.76, *t-stat* memiliki nilai positif pada kedua metrik yaitu 9,16090 dan 8,22370 yang menandakan bahwa rata-rata FEDFormer WEB menggunakan *selected* fitur memberikan hasil yang baik daripada FEDFormer WEB menggunakan semua fitur. Selain itu, untuk *p-value* kedua metrik ini memiliki nilai dibawah 0,05 yaitu 0,00001 pada metrik MAE dan 0,00002 pada metrik MSE. Nilai dari *p-value* ini menegaskan bahwa antara kedua model memiliki perbedaan yang signifikan secara statistik dan model FEDFormer WEB menggunakan *selected* fitur memberikan kesalahan yang lebih kecil bagi kedua metrik.

Pada prediksi untuk *demand* listrik (KW) ini, dalam percobaan menggunakan *selected* fitur, fitur yang digunakan adalah *temperature*, *dew point*, *precipitable water*, GHI, dan *pressure* karena memiliki nilai *absolute* $\geq 0,2$ dan telah dijelaskan pada Subbab 3.2. Fitur *cloud type*, *relative humidity*, dan *wind speed* tidak dipilih karena memiliki nilai *absolute* yang kecil.

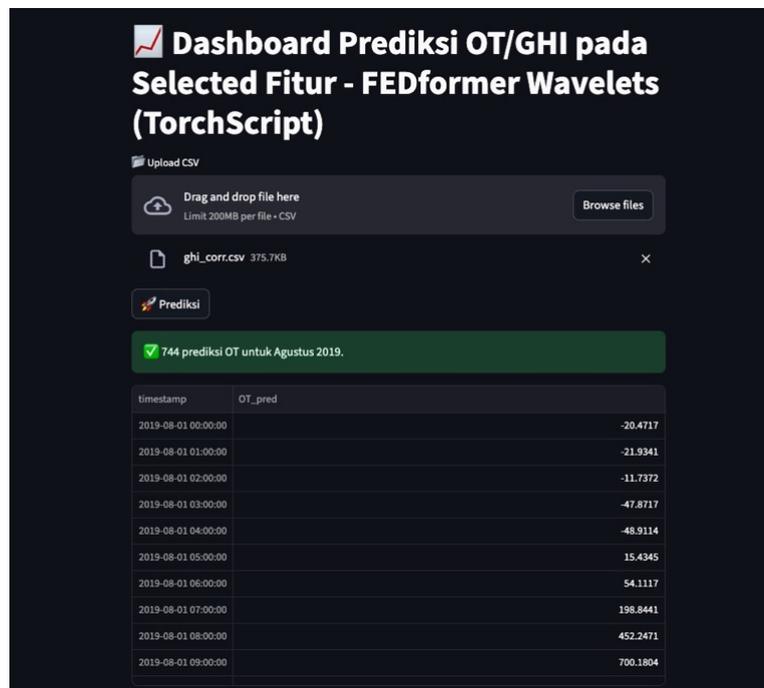
Dilakukan analisis terkait fitur-fitur yang dipilih dari *correlation matrix* untuk mencari hubungan kuat (nilai *absolute* $\geq 0,2$) antara fitur yang dipilih dengan fitur lainnya menggunakan *pearson correlation* yang terdapat pada Tabel 4.64. Untuk fitur *temperature* memiliki hubungan kuat dengan fitur *demand* listrik, *dew point*, GHI, *pressure*, *wind speed*, dan *precipitable water*. Untuk fitur *dew point* memiliki hubungan kuat dengan *temperature*, *demand* listrik (KW), *pressure*, *wind speed*, *relative humidity*, GHI, dan *precipitable water*. Untuk fitur *precipitable water* memiliki hubungan kuat dengan *demand* listrik (KW), *temperature*, *dew point*, *pressure*, *relative humidity*, *wind speed*, *precipitable water*, dan *cloud type*. Untuk fitur GHI memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, dan *relative humidity*. Terakhir, untuk fitur *pressure* memiliki hubungan kuat dengan *demand* listrik, *temperature*, *dew point*, *relative humidity*, *wind speed*, dan *precipitable water*. Meskipun fitur *cloud type*, *relative humidity*, dan *wind speed* berkolerasi tinggi terhadap fitur lainnya, hasil eksperimen menunjukkan bahwa tidak semua interaksi antar fitur berkontribusi untuk memberikan nilai yang terbaik pada performa model, sehingga proses seleksi fitur dengan *correlation matrix* tetap efektif.

4.4.6 Pembahasan Hasil *Marine Predator Algorithm*

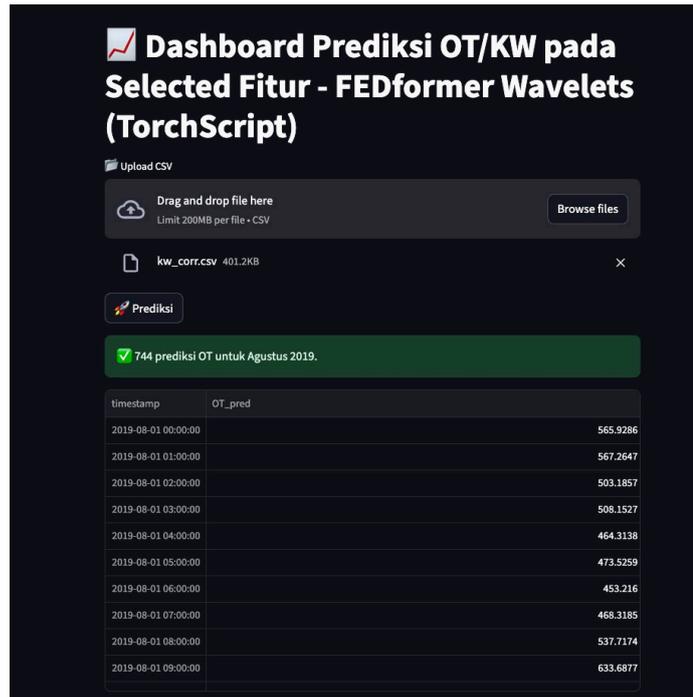
Dari hasil skenario 5, yang telah dijelaskan pada Subbab 4.4.5, diambil model terbaik untuk masing-masing target prediksi. Pada target *temperature*, diambil hasil prediksi dari model FEDFormer WEB pada percobaan menggunakan semua fitur pada sample ke-3 berdasarkan Tabel 4.52 karena merupakan nilai terendah dibanding keempat sample lainnya. Untuk pengambilan target GHI, diambil hasil prediksi dari model Hybrid FEDFormer WEB with LSTM pada percobaan menggunakan *selected* fitur pada sample ke-5 berdasarkan Tabel 4.56 karena merupakan nilai terendah dibanding keempat sample lainnya. Untuk pengambilan target *demand* listrik (KW), diambil hasil prediksi dari model FEDFormer WEB pada percobaan menggunakan *selected* fitur pada sample ke-2 berdasarkan Tabel 4.60 karena merupakan nilai terendah dibanding keempat *sample* lainnya. Hasil dari prediksi untuk target *temperature* terlihat pada Gambar 4.96 yang berupa *datetime* mulai dari tanggal 1 Agustus 2019 pukul 00:00 hingga 31 Agustus 2019 pukul 23:00. Hasil dari prediksi untuk target GHI terlihat pada Gambar 4.97 yang juga berupa *datetime* mulai dari tanggal 1 Agustus 2019 pukul 00:00 hingga 31 Agustus 2019 pukul 23:00. Terakhir, untuk hasil dari prediksi target *demand* listrik (KW) terlihat pada Gambar 4.98 yang juga berupa *datetime* mulai dari tanggal 1 Agustus 2019 pukul 00:00 hingga 31 Agustus 2019 pukul 23:00.



Gambar 4.96 Hasil Prediksi Antarmuka Temperature



Gambar 4.97 Hasil Prediksi Antarmuka GHI



Gambar 4.98 Hasil Prediksi Antarmuka Demand Listrik

Selanjutnya, dilakukan perhitungan untuk mendapatkan *solar hour generation* (SG) sesuai pada Persamaan 2.7 berdasarkan nilai *temperature* dan GHI yang didapatkan. Perhitungan dilakukan menggunakan *excel* yang terlihat pada Gambar 4.99. Untuk fitur GHI, awalnya pada hasil prediksi memiliki satuan $\frac{w^2}{m^2}$ lalu diubah agar menjadi $\frac{kw^2}{m^2}$. Setelah menyesuaikan satuan, maka dilakukan perhitungan untuk SG. Hasil dari prediksi *demand* listrik (KW) dan perhitungan SG akan menjadi data untuk masuk dilakukan optimasi dengan MPA.

datetime	temperature	ghi(w2/m2)	ghi(kw2/m2)	sg	kw
2019-08-01 00:00:00	28,8567579	0	0	0	599,244081
2019-08-01 01:00:00	28,6845746	0	0	0	566,368694
2019-08-01 02:00:00	28,639407	0	0	0	523,959686
2019-08-01 03:00:00	28,4874741	0	0	0	488,038025
2019-08-01 04:00:00	28,4851089	0	0	0	469,411086
2019-08-01 05:00:00	28,7636243	0	0	0	442,661512
2019-08-01 06:00:00	29,2452295	86,4398457	0,08643985	0,07276035	425,802236
2019-08-01 07:00:00	29,9221295	219,893922	0,21989392	0,18445468	438,643438
2019-08-01 08:00:00	30,5163008	454,28341	0,45428341	0,37990809	482,028345
2019-08-01 09:00:00	31,3398943	703,046382	0,70304638	0,58545376	585,78692
2019-08-01 10:00:00	31,4682992	787,772401	0,7877724	0,65557341	701,796775
2019-08-01 11:00:00	31,2635847	841,901325	0,84190132	0,70135986	747,15818
2019-08-01 12:00:00	31,1674101	859,422121	0,85942212	0,71631127	754,537924
2019-08-01 13:00:00	31,0405572	799,454239	0,79945424	0,6667653	735,643607
2019-08-01 14:00:00	30,9610546	692,412903	0,6924129	0,5777268	731,018651
2019-08-01 15:00:00	30,8685393	572,705425	0,57270542	0,4780746	734,437651
2019-08-01 16:00:00	30,5158519	421,876581	0,42187658	0,35280772	739,251626
2019-08-01 17:00:00	30,2104255	209,663305	0,20966331	0,17561297	702,161416
2019-08-01 18:00:00	29,6941528	69,3136839	0,06931368	0,05821068	634,88356
2019-08-01 19:00:00	29,4343353	0	0	0	594,281208
2019-08-01 20:00:00	29,3872337	0	0	0	573,302136
2019-08-01 21:00:00	29,2657147	0	0	0	589,491156
2019-08-01 22:00:00	29,2576452	0	0	0	592,336046

Gambar 4.99 Excel Perhitungan SG

Pada umumnya, MPA ini mencari nilai *fitness* terbesar. Namun pada penelitian ini, dibutuhkan nilai *fitness* terkecil karena nilai *fitness* merupakan *total* biaya yang dikeluarkan sesuai pada Persamaan 2.8 dan ingin didapatkan biaya paling minimum. Pada pengolahan untuk mendapatkan populasi, tidak bisa secara langsung menetapkan bahwa suatu populasi layak untuk di-*training*. Perlu dilakukan pengecekan apakah di dalam suatu populasi terdapat setidaknya dua *chromosome* yang tidak memiliki nilai negatif dalam komponen *surge soc list* karena apabila negatif maka baterai tersebut tidak layak untuk *supply* energi di jam tersebut yang telah dijelaskan pada Subbab 2.2.5. Harus dipastikan bahwa kapasitas baterai dan kapasitas solar PV layak untuk *supply* energi selama satu bulan.

Selanjutnya, hasil dari *fitness function* ini akan mengalami tiga kali fase yaitu fase eksplorasi, fase transisi, dan fase eksploitasi yang terdapat pada *Gambar 4.85*. Pada fase eksplorasi, nilai *mean* berubah-ubah. Untuk nilai *best fitness* dan *top fitness* sedang melakukan eksplorasi untuk mendapatkan nilai *fitness* terendah. Kemudian, dilakukan masa transisi dimana disini nilai *mean* mulai terlihat stabil dan nilai *top fitness* serta *best fitness* juga stabil. Terakhir, pada fase eksploitasi, mulai fokus untuk memperhalus solusi yang sudah diperoleh. Hal ini ditandai dengan penurunan drastis pada nilai *mean fitness* hingga mendekati nilai *top fitness* dan *best fitness*. Disini, telah mengalami proses konvergensi pada nilai *fitness*. Pada fase ini, variabilitas populasi menurun dan eksplorasi terhadap ruang solusi baru menjadi terbatas, sehingga MPA lebih banyak melakukan eksplorasi terhadap solusi terbaik yang telah ditemukan sebelumnya.

Dari keseluruhan *best chromosome* yang telah didapatkan yang terdapat pada *Tabel 4.63*, terlihat bahwa kapasitas PV 7303 dan kapasitas BESS 10689 memiliki *total cost* terkecil yaitu sebesar \$14.459.293,1528 NTD. Dari kapasitas PV dan BESS tersebut, dilakukan prediksi *charging* dan *discharging* pada Bulan Agustus 2019 yang terdapat pada *Gambar 4.86*. Hasilnya adalah pada siang hari, energi listrik di-*supply* oleh PV dan apabila terdapat sisa energi listrik maka baterai akan melakukan *charging*. Pada saat malam hari, baterai melakukan *discharging* dengan batasan dari *range* 20% hingga 80% untuk menjaga umur baterai. Pada saat baterai melakukan *idle*, maka baterai tidak akan melakukan *discharging* karena baterai tepat berada di daya 20%. Pada saat baterai full, maka baterai tidak bisa melakukan *charging* karena baterai tepat berada pada daya 80%. Kondisi kapan saat baterai melakukan *idle* terdapat pada *Tabel 4.77*.

Tabel 4.77 Hasil Waktu Kapan Baterai Idle

Tanggal	Waktu	Energi baterai (kW)	Energi PV (kW)	Kebutuhan energi listrik yang tidak ter-supply (kW)
20 Agustus 2019	04:00:00	2137,8	0	466,15393
20 Agustus 2019	05:00:00	2137,8	0	450,33802
20 Agustus 2019	06:00:00	2137,8	378,02645	437,77408
30 Agustus 2019	06:00:00	2137,8	428,84190	430,66247

Pada *Tabel 4.77*, meski energi baterai memenuhi kebutuhan energi listrik yang dibutuhkan, BESS tidak bisa melakukan *discharging* karena BESS harus memiliki energi dari *range* 20% hingga 80%. Batas energi minimum yang didapatkan untuk BESS adalah 2137,8. Selain itu, sebagai contoh pada data tanggal 20 Agustus dan 30 Agustus di jam 6, terdapat energi dari PV namun energinya tidak memenuhi kebutuhan listrik yang dibutuhkan di jam tersebut. Oleh karena itu, terjadi *blackout* atau mati listrik dalam gedung saat tidak ada energi listrik yang memenuhi kebutuhan.

Untuk baterai dalam kondisi *full* dapat terlihat pada *Gambar 4.87*. Diambil contoh untuk Tanggal 1 Agustus 2019 pukul 10:00. Baterai tidak bisa melakukan pengisian daya lagi dari

solar PV karena berada di daya maksimum yang dapat dimiliki oleh baterai yaitu di batas 80% dengan nilai 8551,2.

4.5 Diskusi

Penelitian yang dilakukan dibatasi pada musim panas tahun 2019 mulai dari Bulan April 2019 hingga Bulan Agustus 2019. Untuk prediksi dari *timeseries* dilakukan selama Bulan Agustus 2019 berdasarkan penelitian dari (Hernando C. S, 2023). Untuk hasil dari MPA, difokuskan pada optimal sizing selama Bulan Agustus 2019.

Pada skenario pertama dimana membandingkan antara model FEDFormer FEB dan FEDFormer WEB baik pada percobaan menggunakan semua fitur maupun *selected* fitur. Hasilnya adalah untuk target *temperature*, maka model FEDFormer WEB pada percobaan dengan semua fitur memiliki hasil yang baik. Untuk target GHI, model FEDFormer WEB dengan percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik. Terakhir, untuk target *demand* listrik, model FEDFormer WEB dengan percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik.

Perbedaan ini dapat dijelaskan dari sisi pemilihan fitur dan kemampuan model dalam mengekstraksi informasi dari fitur-fitur tersebut. Pada prediksi *temperature*, penggunaan seluruh fitur memberi keunggulan karena fitur-fitur seperti *relative humidity* dan *cloud type* yang tidak masuk dalam *selected* fitur ternyata memiliki hubungan yang cukup kuat terhadap fitur-fitur utama seperti *dew point*, *pressure*, dan *precipitable water*. Hal ini terlihat dari nilai Pearson *Correlation* yang signifikan, yaitu antara *relative humidity* dengan *dew point* dengan nilai 0,4627 dan *pressure* dengan nilai -0,3441. Selain itu, target *cloud type* juga memiliki hubungan yang cukup kuat dengan *precipitable water* dengan nilai 0,3474.

Di sisi lain, untuk prediksi GHI dan *demand* listrik, *selected* fitur dengan *correlation matrix* dapat mengurangi noise dan redundansi informasi yang tidak relevan sehingga meningkatkan performa model. Hal ini menunjukkan bahwa efektivitas *selected* fitur bergantung pada target prediksi serta interaksi antar fitur. Selain itu, pemrosesan menggunakan *wavelet transform* memberi keuntungan tambahan. Metode *Wavelet* yang digunakan memiliki kemampuan yang lebih baik dalam menangkap karakteristik lokal dari data *timeseries*. *Wavelet decomposition* memungkinkan terbentuknya pemodelan informasi dalam domain waktu dan frekuensi secara simultan dan membuat lebih adaptif dalam menyesuaikan terhadap pola seasonal atau anomali jangka pendek.

Pada skenario kedua, terkait perbandingan antara model Hybrid FEDFormer FEB *with* LSTM dan Hybrid FEDFormer WEB *with* LSTM baik pada percobaan menggunakan semua fitur maupun *selected* fitur, hasilnya adalah untuk target *temperature* memiliki model terbaik pada Hybrid FEDFormer WEB *with* LSTM dengan percobaan menggunakan *selected* fitur. Selain itu, untuk target GHI, model Hybrid FEDFormer WEB *with* LSTM pada *selected* fitur juga memiliki nilai terbaik dan untuk target *demand* listrik berlaku hal yang sama. Hal ini menunjukkan bahwa dengan adanya *selected* fitur menggunakan *correlation matrix*, maka memberikan hasil yang bagus dalam model tanpa mengurangi kualitas informasi penting terutama ketika dikombinasikan dengan arsitektur LSTM. Selain itu, dibanding dengan *fourier*, *wavelet* masih memiliki keunggulan dalam menangkap dinamika *local* dari data *timeseries* yang kompleks. Berbeda dengan *Fourier transform* yang cenderung fokus pada analisis *global* di domain frekuensi, *wavelet* mampu menganalisis sinyal dalam domain waktu dan frekuensi secara bersamaan.

Pada skenario ketiga, terkait perbandingan antara FEDFormer FEB dan Hybrid FEDFormer FEB *with* LSTM baik pada percobaan menggunakan semua fitur maupun dengan *selected* fitur, untuk target *temperature* terdapat model FEDFormer FEB dengan percobaan menggunakan *selected* fitur yang memiliki nilai yang lebih baik. Untuk target GHI, model Hybrid FEDFormer FEB *with* LSTM dengan percobaan menggunakan *selected* fitur juga memiliki nilai yang lebih baik. Untuk target *demand* listrik, model FEDFormer FEB dengan percobaan menggunakan *selected* fitur memiliki hasil yang baik. Dari ketiga target tersebut, keseluruhan model yang memiliki nilai yang baik berada pada percobaan menggunakan *selected* fitur yang menandakan bahwa struktur model mampu memprediksi dengan baik pada fitur-fitur yang memiliki hubungan kuat dengan target.

Pada skenario keempat, terdapat percobaan menggunakan model FEDFormer WEB dan model Hybrid FEDFormer WEB *with* LSTM pada masing-masing percobaan dengan semua fitur dan *selected* fitur. Hasilnya adalah untuk target *temperature*, model FEDFormer WEB pada semua fitur memiliki hasil yang lebih baik. Hal ini kemungkinan disebabkan oleh model yang lebih ringan yaitu tanpa LSTM mampu memaksimalkan kemampuan wavelet untuk mengekstraksi karakteristik *local* dan periodik yang ada dalam dataset.

Untuk target GHI, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik. Hal ini menunjukkan bahwa penambahan LSTM pada arsitektur *wavelet decoder* dapat membantu menangkap dinamika jangka panjang yang lebih kompleks dari dataset dan hal ini memperkuat dugaan bahwa penggunaan LSTM cocok untuk data dengan pola sekuensial dan temporal yang tidak sepenuhnya ditangkap oleh *wavelet* aja.

Untuk target *demand* listrik, model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik. Hal ini menunjukkan bahwa untuk prediksi beban listrik, kompleksitas tambahan dari LSTM tidak memberikan keunggulan signifikan dibanding model yang lebih sederhana. Secara keseluruhan, hasil pada skenario ini menunjukkan bahwa kombinasi pemilihan fitur dan pemilihan arsitektur model perlu disesuaikan dengan karakteristik target prediksi. Model tanpa LSTM cenderung unggul dalam kasus di mana pola data dapat ditangkap melalui transformasi *wavelet* saja, sedangkan penambahan LSTM menjadi bermanfaat ketika prediksi membutuhkan pemahaman terhadap ketergantungan temporal yang lebih panjang seperti pada prediksi GHI.

Pada skenario kelima, dilakukan perbandingan antara keseluruhan model yang dilakukan dalam penelitian ini yaitu model LSTM, hybrid LSTM *with* Transformer, FEDFormer FEB, Hybrid FEDFormer FEB *with* LSTM, FEDFormer WEB, dan Hybrid FEDFormer WEB *with* LSTM. Hasilnya sama seperti dengan skenario keempat dimana untuk target *temperature*, model FEDFormer WEB pada semua fitur memiliki hasil yang lebih baik. Untuk target GHI, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik. Terakhir, untuk target *demand* listrik, model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik.

Pada implementasi *Marine Predator Algorithm*, didapatkan hasil bahwa kapasitas PV 7303 dan kapasitas BESS 10689 memiliki *total cost* terkecil yaitu sebesar \$14.459.293,1528 NTD. Nilai ini merupakan hasil dari proses eksplorasi dan eksploitasi yang dilakukan oleh MPA terhadap ruang solusi, dengan mempertimbangkan *fitness function* yang telah ditentukan, yaitu meminimalkan *total* biaya sistem secara keseluruhan. Pemilihan kapasitas PV sebesar 7303 kW memungkinkan sistem untuk menghasilkan energi yang cukup selama periode produksi maksimum, sementara kapasitas BESS sebesar 10689kW memberikan ruang penyimpanan

yang memadai untuk menjaga kontinuitas suplai listrik, terutama pada malam hari atau saat kondisi cuaca buruk. Hasil dari penelitian ini menunjukkan bahwa metode gabungan antara prediksi *timeseries* dan optimasi metaheuristik (MPA) mampu memberikan solusi terintegrasi dalam merancang sistem energi terbarukan yang efisien, baik dari segi operasional BESS maupun dari sisi biaya investasi.

[Halaman Ini Sengaja Dikosongkan]

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian ini, terdapat beberapa jawaban dari tujuan penelitian ini. Kesimpulan yang didapat adalah sebagai berikut:

1. Metode Prediksi waktu untuk *charging* dan *discharging* BESS didesain dengan mengembangkan berbagai model yaitu LSTM, hybrid LSTM *with* Transformer, FEDFormer FEB, Hybrid FEDFormer FEB *with* LSTM, FEDFormer WEB, dan Hybrid FEDFormer WEB *with* LSTM. Dataset yang digunakan berasal dari NSRDB dan NTUST berupa fitur *weather*, radiasi, dan *demand* listrik. Hasil prediksi dari pelatihan model ini berupa fitur target *temperature*, GHI, dan *demand* listrik difokuskan selama Bulan Agustus 2019.
2. Kinerja metode prediksi waktu dari keenam model (LSTM, hybrid LSTM *with* Transformer, FEDFormer FEB, Hybrid FEDFormer FEB *with* LSTM, FEDFormer WEB, dan Hybrid FEDFormer WEB *with* LSTM) dievaluasi menggunakan dataset dengan keseluruhan fitur berupa domain *weather*, radiasi, dan *demand* listrik ataupun sebagian fitur yang diseleksi dari *correlation matrix*. Pemilihan model terbaik ini dilakukan berdasarkan metrik evaluasi MAE, MSE, dan uji *t-test*. Hasil eksperimen menunjukkan bahwa untuk target *temperature*, model FEDFormer WEB pada semua fitur memiliki hasil yang lebih baik dengan nilai *average* MAE adalah 0,15635. Untuk target GHI, model Hybrid FEDFormer WEB *with* LSTM pada percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik dengan nilai *average* MAE adalah 56,12330. Terakhir, untuk target *demand* listrik, model FEDFormer WEB pada percobaan menggunakan *selected* fitur memiliki hasil yang lebih baik dengan nilai *average* MAE adalah 18,29667.
3. Optimasi kapasitas PV dan BESS dilakukan menggunakan *Marine Predator Algorithm* (MPA) dengan *input* berupa data hasil prediksi untuk *temperature*, GHI, dan *demand* listrik. Dari proses optimasi ini, diperoleh kapasitas PV sebesar 7303 kW dan kapasitas BESS sebesar 10689 kW yang mampu menghasilkan *total* biaya minimum sebesar \$14.459.293,15 NTD. Konfigurasi ini mampu memenuhi kebutuhan listrik masyarakat dengan efisiensi biaya dan tetap menjaga batas SOC BESS pada rentang 20% hingga 80%.

5.2 Saran

Adapun saran yang dilakukan setelah melakukan penelitian ini adalah:

1. Target prediksi tidak hanya pada Bulan Agustus 2019 saja, melainkan bisa diterapkan juga selama satu tahun.
2. Perlu dilakukan *hyperparameter tuning* agar hanya terdapat satu model yang unggul dari ketiga target prediksi yang diinginkan yaitu *temperature*, GHI, dan *demand* listrik.
3. Perlu dilakukan pencegahan agar tidak terjadi baterai *idle* saat dibutuhkan energi listrik dengan kondisi dimana kapasitas PV tidak mampu memenuhi kebutuhan energi listrik dan baterai tidak bisa melakukan *discharging* karena berada di minimum *capacity* dari range 20% hingga 80%.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Ali, P. O., & Kim, K. N. (2024). Analysis of Indonesia's priority selection: Energy transition, energy-related measures, mining governance, and resource transition using the analytic hierarchy process (AHP). *Energy for Sustainable Development*, 83, 101559. <https://doi.org/10.1016/j.esd.2024.101559>
- Alorf, A. (2023). A survey of recently developed metaheuristics and their comparative analysis. *Engineering Applications of Artificial Intelligence*, 117, 105622. <https://doi.org/10.1016/j.engappai.2022.105622>
- Amini, M., Khorsandi, A., Vahidi, B., Hosseinian, S. H., & Malakmahmoudi, A. (2021). Optimal sizing of battery energy storage in a microgrid considering capacity degradation and replacement year. *Electric Power Systems Research*, 195, 107170. <https://doi.org/10.1016/j.epsr.2021.107170>
- Andayani, F., Theng, L. B., Tsun, M. T., & Chua, C. (2022). Hybrid LSTM-Transformer Model for Emotion Recognition From Speech Audio Files. *IEEE Access*, 10, 36018–36027. <https://doi.org/10.1109/ACCESS.2022.3163856>
- Barrow, D. F. (2018). Off Grid: Meaning, Technologies, and Applications. *Journal of Architectural/Planning Research and Studies (JARS)*, 5(1), 155–170. <https://doi.org/10.56261/jars.v5i1.169241>
- Bovera, F., Spiller, M., Zatti, M., Rancilio, G., & Merlo, M. (2023). Development, validation, and testing of advanced mathematical models for the optimization of BESS operation. *Sustainable Energy, Grids and Networks*, 36, 101152. <https://doi.org/10.1016/j.segan.2023.101152>
- Bowen, T., Chernyakhovskiy, I., & Denholm, P. (2019). *Grid-Scale Battery Storage*.
- Chollet, F. (2021). *DEEP LEARNING with Python* (J. Stout & F. Buontempo, Eds.; Second Edition).
- Erdiwansyah, Gani, A., Mamat, R., Bahagia, Nizar, M., Yana, S., Mat Yasin, M. H., Muhibbuddin, & Rosdi, S. M. (2024). Prospects for renewable energy sources from biomass waste in Indonesia. *Case Studies in Chemical and Environmental Engineering*, 10, 100880. <https://doi.org/10.1016/j.cscee.2024.100880>
- Gambella, C., Ghaddar, B., & Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3), 807–828. <https://doi.org/10.1016/j.ejor.2020.08.045>
- Hernando C. S, P. (2023). *Optimal Sizing and Scheduling Battery Storage System and Solar Photovoltaic Grid Connection based on Levelized Cost of Electricity using a Mathematical Model and Recurrent Neural Network* [Master's thesis]. National Taiwan University of Science and Technology.

- Kamoon, A., Song, H., Keshavarzian, K., Levy, K., Jalili, M., Wilkinson, R., Yu, X., McGrath, B., & Meegahapola, L. (2023). Machine learning based energy demand prediction. *Energy Reports*, 9, 171–176. <https://doi.org/10.1016/j.egy.2023.09.151>
- Khalis Syahril Suryana, Syahla Anisah, & Aceng Komarudin Mutaqin. (2024). Uji Dua Rata-Rata Waktu Belajar Mandiri Antara Mahasiswa Laki-Laki dan Perempuan. *Jurnal Riset Statistika*, 103–110. <https://doi.org/10.29313/jrs.v4i2.5002>
- Mugemanyi, S., Qu, Z., Rugema, F. X., Dong, Y., Wang, L., Bananeza, C., Nshimiyimana, A., & Mutabazi, E. (2023). Marine predators algorithm: A comprehensive review. *Machine Learning with Applications*, 12, 100471. <https://doi.org/10.1016/j.mlwa.2023.100471>
- Nair, U. R., Sandelic, M., Sangwongwanich, A., Dragicevic, T., Costa-Castello, R., & Blaabjerg, F. (2021). An Analysis of Multi Objective Energy Scheduling in PV-BESS System Under Prediction Uncertainty. *IEEE Transactions on Energy Conversion*, 36(3), 2276–2286. <https://doi.org/10.1109/TEC.2021.3055453>
- Paradongan, H. T., Hakam, D. F., Wiryono, S. K., Prahastono, I., Aditya, I. A., Banjarnahor, K. M., Sinisuka, N. I., & Asekomeh, A. (2024). Techno-economic feasibility study of solar photovoltaic power plant using RETScreen to achieve Indonesia energy transition. *Heliyon*, 10(7), e27680. <https://doi.org/10.1016/j.heliyon.2024.e27680>
- Rainio, O., Teuhon, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1), 6086. <https://doi.org/10.1038/s41598-024-56706-x>
- Rana, M. M., Uddin, M., Sarkar, M. R., Shafiullah, G. M., Mo, H., & Atef, M. (2022). A review on hybrid photovoltaic – Battery energy storage system: Current status, challenges, and future directions. *Journal of Energy Storage*, 51, 104597. <https://doi.org/10.1016/j.est.2022.104597>
- Roy, D. K., Hossain, M. A., Haque, M. P., Alataway, A., Dewidar, A. Z., & Mattar, M. A. (2024). Automated Model Selection Using Bayesian Optimization and the Asynchronous Successive Halving Algorithm for Predicting Daily Minimum and Maximum Temperatures. *Agriculture*, 14(2), 278. <https://doi.org/10.3390/agriculture14020278>
- Sankaran, G., & Venkatesan, S. (2022). An overview of Lithium-Ion batteries for electric mobility and energy storage applications. *IOP Conference Series: Earth and Environmental Science*, 1042(1), 012012. <https://doi.org/10.1088/1755-1315/1042/1/012012>
- Suradhaniwar, S., Kar, S., Durbha, S. S., & Jagarlapudi, A. (2021). Time Series Forecasting of Univariate Agrometeorological Data: A Comparative Performance Evaluation via One-Step and Multi-Step Ahead Forecasting Strategies. *Sensors*, 21(7), 2430. <https://doi.org/10.3390/s21072430>
- Susilowati, Y., Hardiyasanti, D. C., Widianingrum, S., Endrasari, F., Djamari, D. W., Bahar, A. H., Wahono, J., & Veza, I. (2023). Carbon credit and economic feasibility analysis of biomass-solar PV-battery power plant for application in Indonesia remote area. *Renewable Energy*, 219, 119383. <https://doi.org/10.1016/j.renene.2023.119383>

- Symeonidou, M. M., Zioga, C., & Papadopoulos, A. M. (2021). Life cycle cost optimization analysis of battery storage system for residential photovoltaic panels. *Journal of Cleaner Production*, 309, 127234. <https://doi.org/10.1016/j.jclepro.2021.127234>
- Tovar Rosas, M. A., Pérez, M. R., & Martínez Pérez, E. R. (2022). Itineraries for charging and discharging a BESS using energy predictions based on a CNN-LSTM neural network model in BCS, Mexico. *Renewable Energy*, 188, 1141–1165. <https://doi.org/10.1016/j.renene.2022.02.047>
- Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8), 5929–5955. <https://doi.org/10.1007/s10462-020-09838-1>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*.
- Zaki Abdullah, M., Kalid Jassim, A., Noori Hummadi, F., & Majid M. Al Khalidy, M. (2024). NEW STRATEGIES FOR IMPROVING NETWORK SECURITY AGAINST CYBER ATTACK BASED ON INTELLIGENT ALGORITHMS. *Journal of Engineering and Sustainable Development*, 28(3), 342–354. <https://doi.org/10.31272/jeasd.28.3.4>
- Zhang, R., Lee, M., Huang, L., & Ni, M. (2024). Optimization of battery energy storage system (BESS) sizing in different electricity market types considering BESS utilization mechanisms and ownerships. *Journal of Cleaner Production*, 470, 143317. <https://doi.org/10.1016/j.jclepro.2024.143317>
- Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). *FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting*.
- Zolfaghari, M., Ghaffarzadeh, N., & Ardakani, A. J. (2019). Optimal sizing of battery energy storage systems in off-grid micro grids using convex optimization. *Journal of Energy Storage*, 23, 44–56. <https://doi.org/10.1016/j.est.2019.02.027>

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Kode semu 5.1 Algoritma Series Decomposition

```
Algoritma 5: Series Decomposition
1:  DEFINE CLASS series_decomp:
2:    INHERIT FROM nn.Module
3:    INITIALIZE(kernel_size):
4:      INITIALIZE moving_avg = moving_avg(kernel_size, stride = 1) #CALL Algorithm 6
5:    FUNCTION forward(x):
6:      #Input x shape: [Batch, Seq_len, Features]
7:      #Step 1: Compute trend component
8:      COMPUTE moving_mean = moving_avg(x) #CALL Algorithm 6
9:      #shape of moving_mean same as X
10:     #Step 2: Compute seasonal component
11:     COMPUTE seasonal = x - moving_mean
12:     RETURN:
13:       seasonal AND moving_mean
14:
15:  DEFINE CLASS series_decomp_multi:
16:    INHERIT FROM nn.Module
17:    INITIALIZE(kernel_size_list):
18:      INITIALIZE moving_mean_list = []
19:      FOR EACH kernel_size IN kernel_size_list:
20:        APPEND moving_avg(kernel_size, stride = 1) TO moving_avg_list #CALL Algorithm 6
21:      END FOR
22:      INITIALIZE linear_layer = Linear(in_features=1, out_features=length(kernel_size_list))
23:    FUNCTION forward(x):
24:      INITIALIZE moving_mean_list = []
25:      FOR each func IN moving_avg_list:
26:        COMPUTE kernel_mean = func(x) #apply moving average
27:        APPEND kernel_mean.unsqueeze(-1) TO moving_mean_list
28:      END FOR
29:      CONCATENATE moving_mean_list ALONG last dimension → moving_mean_cat
30:      COMPUTE attention_weights = Softmax(linear_layer(x.unsqueeze(-1)), dim = -1)
31:      COMPUTE weighted_mean = SUM over kernels:
32:        Moving_mean_cat * attention_weights
33:      REDUCE last dimension to get final moving_mean
34:      Shape: [Batch, Seq_len, Features]
35:      COMPUTE seasonal = x - weighted_mean
36:      RETURN seasonal, weighted_mean
```

Kode semu 5.2 Algoritma Moving Average

```
Algoritma 6: Moving Average
1:  DEFINE CLASS moving_avg:
2:    INHERIT FROM nn.Module
3:    INITIALIZE(kernel_size, stride):
4:      STORE kernel_size = kernel_size
5:      INITIALIZE avg = nn.AvgPool1d(kernel_size, stride, padding)
6:
7:    FUNCTION forward(x):
8:      #Input x shape: [Batch, Seq_len, Features]
9:      #Step 1: Padding
10:     COMPUTE left_pad = kernel_size - 1 - floor((kernel_size - 1) / 2)
11:     COMPUTE right_pad = floor((kernel_size - 1) / 2)
12:     REPLICATE first timestep x[:, 0:1, :] → left_pad times → front padding
13:     REPLICATE last timestep x[:, -1:, :] → right_pad times → end padding
14:     CONCATENATE front, x, and end along time dimension → padded_x
15:     #Step 2: Apply Average Pooling
16:     PERMUTE padded_x to shape [Batch, Features, Seq_len]
```

```

17: APPLY 1D AveragePooling → output shape [Batch, Features, Seq_len]
18: #Step 3: Revert back to original shape
19: PERMUTE back to [Batch, Seq_Len, Features]
20: RETURN smoothed sequence (highlighting trend)

```

Kode semu 5.3 Algoritma DataEmbedding *without* Positional Encoding

Algoritma 7: DataEmbedding *without* Positional Encoding

```

1: DEFINE CLASS DataEmbedding_wo_pos:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(c_in, d_model, embed_type, freq, dropout):
4:     INITIALIZE value_embedding = TokenEmbedding(c_in, d_model) #CALL Algoritma 8
5:     INITIALIZE position_embedding = PositionalEmbedding(d_model) #CALL Algoritma 9
6:     IF embed_type IS NOT 'timeF':
7:       INITIALIZE temporal_embedding = TemporalEmbedding(d_model, embed_type, freq) #CALL
           Algoritma 10
8:     ELSE:
9:       INITIALIZE temporal_embedding = TimeFeatureEmbedding(d_model, embed_type, freq)
           #CALL Algoritma 11
10:    INITIALIZE dropout = nn.Dropout(dropout)
11:
12:    FUNCTION forward(x, x_mark):
13:      #value + temporal embedding (no positional encoding)
14:      COMPUTE embedded = value_embedding(x) + temporal_embedding(x_mark)
15:      APPLY dropout to embedded
16:      RETURN embedded

```

Kode semu 5.4 Algoritma TokenEmbedding

Algoritma 8: TokenEmbedding

```

1: DEFINE CLASS TokenEmbedding:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(c_in, d_model):
4:     SET padding based ON PyTorch version
5:     INITIALIZE tokenConv = nn.Conv1d(
6:       in_channels=c_in, out_channels=d_model, kernel_size=3, padding=circular)
7:     APPLY nn.init.kaiming_normal_ to Conv1d weights
8:
9:     FUNCTION forward(x):
10:      PERMUTE x to shape [Batch, Features, Seq_Len]
11:      APPLY Conv1d
12:      TRANSPOSE output back to [Batch, Seq_Len, d_model]
13:      RETURN embedded tensor

```

Kode semu 5.5 Algoritma PositionalEmbedding

Algoritma 9: PositionalEmbedding

```

1: DEFINE CLASS PositionalEmbedding:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(d_model, max_len=5000):
4:     CREATE pe = zero tensor of shape [max_len, d_model]
5:     CREATE position = tensor of shape [0,1, ..., max_len-1]
6:     COMPUTE div_term using exponential decay on even indices
7:     COMPUTE:
8:       pe[:, 0::2] = sin(position * div_term)
9:       pe[:, 1::2] = cos(position * div_term)
10:    ADD batch dimension to pe
11:    STORE buffer = pe
12:
13:    FUNCTION forward(x):
14:      RETURN pe[:, :seq_len] (shape matched to input)

```

Kode semu 5.6 Algoritma TemporalEmbedding

Algoritma 10: TemporalEmbedding

```
1: DEFINE CLASS TemporalEmbedding:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(d_model, embed_type= 'fixed', freq = 'h'):
4:     DEFINE:
5:       minute_size = 4
6:       hour_size = 24
7:       weekday_size = 7
8:       day_size = 32
9:       month_size = 13
10:    SELECT embedding layer class:
11:      IF embed_type == 'fixed' :
12:        FixedEmbedding()
13:      ELSE:
14:        nn.Embedding
15:    INITIALIZE embeddings:
16:      IF freq == 't':
17:        minute_embed = Embed(minute_size, d_model)
18:        hour_embed = Embed(hour_size, d_model)
19:        weekday_embed = Embed(weekday_size, d_model)
20:        day_embed = Embed(day_size, d_model)
21:        month_embed = Embed(month_size, d_model)
22:
23:    FUNCTION forward(x):
24:      CONVERT x to long integer type
25:      COMPUTE minute_x = minute_embed(x[:, :, 4]) IF hasattr('minute_embed') ELSE 0
26:      COMPUTE hour_x = hour_embed(x[:, :, 3])
27:      COMPUTE weekday_x = weekday_embed(x[:, :, 2])
28:      COMPUTE day_x = day_embed(x[:, :, 1])
29:      COMPUTE month_x = month_embed(x[:, :, 0])
30:      SUM res = hour_x + weekday_x + day_x + month_x + minute_x
31:      RETURN res
```

Kode semu 5.7 Algoritma TimeFeatureEmbedding

Algoritma 11: TimeFeatureEmbedding

```
1: DEFINE CLASS TimeFeatureEmbedding:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(d_model, embed_type= 'timeF', freq = 'h'):
4:     MAP freq to input dimension (d_inp) using freq_map
5:     INITIALIZE embed = nn.Linear(input=d_inp, output=d_model)
6:
7:     FUNCTION forward(x):
8:       APPLY linear projection to x
9:       RETURN embedded features
```

Kode semu 5.8 Algoritma FourierBlock

Algoritma 12: FourierBlock

```
1: DEFINE CLASS FourierBlock:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(in_channels, out_channels, seq_len, modes, mode_select method):
4:     GET index = get_frequency_modes(seq_len, modes, mode_select_method)
5:     COMPUTE scale = (1 / (in_channels * out_channels))
6:     INITIALIZE weights:
7:       shape = [8, in_channels/8, out_channels/8, number of modes]
8:       data type = complex
9:
10:    FUNCTION compl_mul1d(self, input, weights):
11:      #Perform complex matrix multiplication using Einstein summation
12:      RETURN einsum("bhi,hio->bho", input, weights)
```

```

13:
14:   FUNCTION forward(q, k, v, mask):
15:     #Input 1: tensor shape [Batch, Length, Heads, Embedding]
16:     PERMUTE q to shape [Batch, Heads, Embedding, Length] → x
17:     APPLY 1D real FFT to x along time dimension → x_ft
18:     INITIALIZE zero tensor for transformed Fourier domain output (out_ft):
19:     shape = [Batch, Heads, Embedding, FrequencyLength]
20:     FOR each selected frequency index i:
21:       out_ft[..., i] = complex multiplication of x_ft[..., i] and weights[..., i]
22:     APPLY inverse FFT (irfft) to out_ft along time dimension → x (return to time domain)
23:     RETURN x and None

```

Kode semu 5.9 Algoritma GetFrequencyModes

Algoritma 13: GetFrequencyModes

```

1:   DEFINE FUNCTION get_frequency_modes(seq_len, modes=64, mode_select_method='random'):
2:     SET modes = min(modes, seq_len//2)
3:     IF mode_select_method == 'random':
4:       CREATE index = list(range(0, seq_len//2))
5:       SHUFFLE index list randomly
6:       SELECT first 'modes' elements from index list
7:     ELSE:
8:       CREATE index list = [0, 1, 2, ..., modes - 1]
9:       SORT index list in ascending order
10:      RETURN index list

```

Kode semu 5.10 Algoritma FourierCrossAttention

Algoritma 14: FourierCrossAttention

```

1:   DEFINE CLASS FourierCrossAttention:
2:     INHERIT FROM nn.Module
3:     INITIALIZE(in_channels, out_channels, seq_len_q, modes, mode_select_method, activation,
4:               policy=0):
5:       SET activation function
6:       STORE in_channels = in_channels
7:       STORE out_channels = out_channels
8:       SELECT frequency modes for query and key/value sequences:
9:       index_q = get_frequency_modes(seq_len_q, modes, mode_select_method)
10:      index_kv = get_frequency_modes(seq_len_kv, modes, mode_select_method)
11:      DISPLAY selected query and key/value modes and indices
12:      COMPUTE scale factor = 1 / (in_channels * out_channels)
13:      INITIALIZE learnable complex weights:
14:      shape = [8, in_channels/8, out_channels/8, number_of_query_modes]
15:
16:      FUNCTION compl_mulld(input, weights):
17:        PERFORM complex matrix multiplication with einsum: "bhi,hio->bho"
18:        RETURN result
19:
20:      FUNCTION forward(q, k, v, mask):
21:        # q, k, v shape: [Batch, Length, Heads, Embedding]
22:        PERMUTE q, k, v to shape: [Batch, Heads, Embedding, Length]
23:        APPLY 1D FFT on q and k:
24:        xq_ft = rFFT of q
25:        xk_ft = rFFT of k
26:        EXTRACT selected frequency modes from q and k:
27:        xq_ft_ = gather frequencies using index_q
28:        xk_ft_ = gather frequencies using index_kv
29:        COMPUTE attention weights in frequency domain:
30:        xqk_ft = einsum over xq_ft_ and xk_ft_
31:        APPLY activation function:
32:        IF 'tanh': elementwise tanh
33:        ELSEIF 'softmax': magnitude softmax + convert to complex
34:        ELSE: raise error

```

```

34: COMPUTE value-weighted sum in frequency domain:
35:     xqkv_ft = einsum(xqk_ft, xk_ft_)
36:     xqkvw = einsum(xqkv_ft, weights1)
37: INITIALIZE zero tensor for final frequency-domain output → out_ft
38: SCATTER weighted outputs to corresponding frequency indices in out_ft
39: APPLY inverse FFT (irfft) to return to time domain:
40:     out = irfft(out_ft) with scaling
41: RETURN output and None

```

Kode semu 5.11 Algoritma MultiWaveletTransform

```

Algoritma 15: MultiWaveletTransform
1: DEFINE CLASS MultiWaveletTransform:
2:     INHERIT FROM nn.Module
3:     INITIALIZE(ich, k, alpha, c, nCZ, L, base, attention_dropout):
4:     STORE parameters: k (wavelet size), c (channels), L (levels), nCZ (number of cascade layers)
5:     INITIALIZE Linear projection:
6:         - Lk0: projects input from ich → c × k
7:         - Lk1: projects back from c × k → ich
8:     INITIALIZE multiwavelet cascade:
9:         MWT_CZ = list of nCZ modules, each a MWT_CZ1d instance with given parameters
10:    STORE ich as instance variable
11:
12:    FUNCTION forward(queries, keys, values, attn_mask):
13:    EXTRACT shapes:
14:        B = batch, L = query length, H = heads, E = embedding
15:        S = key/value length, D = dimension
16:    IF query length L > key/value length S:
17:        PAD keys and values with zeros to match L
18:    ELSE:
19:        TRUNCATE keys and values to match L
20:    FLATTEN values to shape [B, L, H × D] → values
21:    APPLY linear projection Lk0 to values → [B, L, c, k]
22:    FOR i in range(nCZ):
23:        PASS through MWT_CZ[i] → V
24:        IF not last layer:
25:            APPLY ReLU activation
26:        FLATTEN V back and apply Lk1 → [B, L, ich]
27:    RESHAPE V to [B, L, H, D]
28:    RETURN (V, None)

```

Kode semu 5.12 Algoritma MultiWaveletCross

```

Algoritma 16: MultiWaveletCross
1: DEFINE CLASS MultiWaveletCross:
2:     INHERIT FROM nn.Module
3:     INITIALIZE(in_channels, out_channels, seq_len_q, seq_len_kv, modes, c, k, ich, L, base,
4:     mode_select_method, initializer, activation):
5:     STORE internal parameters: c, k, L
6:     # Generate wavelet filter banks based on selected base (e.g., Legendre)
7:     H0, H1, G0, G1, PHI0, PHI1 = get_filter(base, k)
8:     COMPUTE reconstructed filters:
9:     H0r = H0 × PHI0
10:    G0r = G0 × PHI0
11:    H1r = H1 × PHI1
12:    G1r = G1 × PHI1
13:    ZERO OUT near-zero values for stability
14:    SET max_item = 3
15:    # Initialize 4 Fourier-based cross attention layers
16:    attn1, attn2, attn3, attn4 = instances of FourierCrossAttentionW with same parameters
17:    # Feedforward projection layers
18:    INITIALIZE:
19:    T0: Linear layer on wavelet output

```

```

19:     ec_s, ec_d: wavelet transform filters (even/odd split)
20:     rc_e, rc_o: wavelet reconstruction filters
21:     INITIALIZE:
22:         Lk, Lq, Lv: project input to wavelet channels
23:         out: final output projection layer
24:     STORE modes
25:
26:     FUNCTION forward(q, k, v, mask):
27:         EXTRACT shape: B=batch, N=query_len, H=heads, E=embed_dim
28:         FLATTEN q, k, v along heads and embed: shape → [B, N, H×E]
29:         PROJECT q, k, v → shape [B, N, c, k] via Lq, Lk, Lv
30:         ALIGN length between q and k/v via zero-padding or truncation
31:         EXTEND q, k, v to nearest power-of-two length by repeating front section
32:         INITIALIZE lists:
33:             Ud_q, Ud_k, Ud_v: detail and coarse decomposition tuples
34:             Us_q, Us_k, Us_v: only coarse part
35:             Ud, Us: final outputs from cross attention layers
36:         # Wavelet decomposition
37:         FOR each level in ns - L:
38:             APPLY wavelet_transform on q, k, v
39:             STORE detail and coarse parts
40:         # Cross attention in frequency domain
41:         FOR each level:
42:             APPLY attn1 and attn2 on detail q/k/v → sum to Ud
43:             APPLY attn3 on coarse q/k/v → Us
44:         # Final attention at base level
45:         v = attn4(q, k, v)
46:         # Reconstruction from wavelet domain
47:         FOR each level from last to first:
48:             v += Us[i]
49:             CONCATENATE v and Ud[i] along feature dim
50:             APPLY inverse wavelet via evenOdd()
51:         PROJECT v to original dimension → shape [B, N, H×E]
52:         RETURN v and None
53:
54:     FUNCTION wavelet_transform(x):
55:         SPLIT x into even and odd indexed parts
56:         CONCATENATE along feature dim
57:         APPLY wavelet filters (ec_d, ec_s) → get detail and smooth
58:         RETURN detail, smooth
59:
60:     FUNCTION evenOdd(x):
61:         ASSERT x has 2×k features
62:         APPLY reconstruction filters (rc_e and rc_o)
63:         COMBINE even and odd back into original sequence order
64:         RETURN reconstructed x

```

Kode semu 5.13 Algoritma *AutoCorrelationLayer*

```

Algoritma 17: AutoCorrelationLayer
1:  DEFINE CLASS AutoCorrelationLayer:
2:  INHERIT FROM nn.Module
3:  INITIALIZE(correlation, d_model, n_heads, d_keys=None, d_values=None):
4:  SET d_keys and d_values to d_model // n_heads IF not provided
5:  STORE given correlation function/module (e.g., FourierBlock or MultiWaveletCross)
6:  INITIALIZE:
7:  query_projection = nn.Linear(d_model, d_keys * n_heads)
8:  key_projection = nn.Linear(d_model, d_keys * n_heads)
9:  value_projection = nn.Linear(d_model, d_values * n_heads)
10: out_projection = nn.Linear(d_values * n_heads, d_model)
11: STORE n_heads = n_heads
12:
13: FUNCTION forward(queries, key, values, attn_mask):
14: EXTRACT shapes: B = batch, L = query_len, S = key_len, H = number of heads

```

```

15: # Step 1: Linear projections and reshape
16: PROJECT query_projection WITH queries → [B, L, H, d_keys]
17: PROJECT key_projection WITH keys → [B, S, H, d_keys]
18: PROJECT value_projection WITH values → [B, S, H, d_values]
19: # Step 2: Apply the internal correlation mechanism
20: APPLY inner_correlation(queries, keys, values, attn_mask)
21: → RETURNS: output per head [B, L, H, d_values], attention scores
22: # Step 3: Merge heads and project to final output
23: RESHAPE output → [B, L, H * d_values]
24: PROJECT final output → [B, L, d_model]
25: RETURN projected output, attention scores

```

Kode semu 5.14 Algoritma EncoderLayer

Algoritma 18: EncoderLayer

```

1: DEFINE CLASS EncoderLayer:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(attention, d_model, d_ff, moving_avg, dropout, activation):
4:     SET d_ff = 4 × d_model IF not specified
5:     STORE provided attention module (e.g., AutoCorrelationLayer)
6:     INITIALIZE:
7:       conv1: 1D convolution from d_model → d_ff (feedforward part)
8:       conv2: 1D convolution from d_ff → d_model (projection back)
9:     IF moving_avg is a list:
10:      INITIALIZE decomp1 and decomp2 using series_decomp_multi()
11:     ELSE:
12:      INITIALIZE decomp1 and decomp2 using series_decomp()
13:     SET dropout layer with given rate
14:     SET activation function:
15:       IF activation == 'relu': using relu
16:       ELSE: gelu
17:
18:   FUNCTION forward(x, attn_mask):
19:     # Step 1: Self-attention
20:     APPLY attention block on x (as query, key, value) → new_x, attn
21:     ADD residual connection with dropout:
22:       x = x + dropout(new_x)
23:     # Step 2: First decomposition
24:     DECOMPOSE x using decomp1 → keep seasonal component (x)
25:     # Step 3: Position-wise Feedforward
26:     TRANSPOSE x to match Conv1d input: [B, D, L]
27:     APPLY conv1 → activation → dropout
28:     APPLY conv2 → dropout
29:     TRANSPOSE back to [B, L, D]
30:     ADD x and y
31:     # Step 4: Second decomposition
32:     DECOMPOSE (x + y) using decomp2 → final output
33:     RETURN final representation (res), and attention scores (attn)

```

Kode semu 5.15 Algoritma Encoder

Algoritma 19: Encoder

```

1: DEFINE CLASS Encoder:
2:   INHERIT FROM nn.Module
3:   INITIALIZE(attn_layers, conv_layers, norm_layer):
4:     STORE attn_layers as list of EncoderLayer modules
5:     IF conv_layers is provided:
6:       STORE conv_layers as list

```

```

7:     ELSE:
8:         SET conv_layers = None
9:         STORE optional normalization layer (norm_layer)
10:
11:     FUNCTION forward(x, attn_mask):
12:         INITIALIZE empty list attns to store attention scores
13:         IF conv_layers exists:
14:             FOR each pair of (attn_layer, conv_layer) in attn_layers and conv_layers:
15:                 APPLY attn_layer on input x → output x and attention attn
16:                 APPLY conv_layer to x (e.g., smoothing or projection)
17:                 STORE attn in attns list
18:             APPLY final attn_layer (last one) without conv_layer → x, attn
19:             STORE attn in attns list
20:         ELSE:
21:             FOR each attn_layer in attn_layers:
22:                 APPLY attn_layer to x → x, attn
23:                 STORE attn in attns list
24:         IF norm_layer is defined:
25:             APPLY norm_layer to x
26:         RETURN final output x and list of all attention maps attns

```

Kode semu 5.16 Algoritma DecoderLayer

```

Algoritma 20: DecoderLayer
1:  DEFINE CLASS DecoderLayer:
2:  INHERIT FROM nn.Module
3:  INITIALIZE(self_attention, cross_attention, d_model, c_out, d_ff, moving_avg, dropout, activation):
4:  SET d_ff = 4 × d_model IF not specified
5:  STORE self_attention module (e.g. AutoCorrelationLayer)
6:  STORE cross_attention module
7:  INITIALIZE:
8:  conv1: 1D convolution from d_model → d_ff (position-wise FFN)
9:  conv2: 1D convolution from d_ff → d_model
10: INITIALIZE decomposition modules:
11: decomp1: after self-attention
12: decomp2: after cross-attention
13: decomp3: after feedforward network
14: (Use series_decomp or series_decomp_multi based on config)
15: INITIALIZE:
16: dropout layer
17: projection: final 1D convolution to project trend → c_out
18: (uses kernel size 3 and circular padding)
19: activation function: ReLU or GELU
20:
21: FUNCTION forward(x, cross, x_mask, cross_mask):
22: # Step 1: Self-Attention + Residual + Dropout
23: APPLY self_attention on x → self_attended
24: x = x + dropout(self_attended)
25: # Step 2: Decomposition 1 (after self-attention)
26: x, trend1 = decomp1(x)
27: # Step 3: Cross-Attention + Residual + Dropout
28: APPLY cross_attention on x and cross → cross_attended
29: x = x + dropout(cross_attended)
30: # Step 4: Decomposition 2 (after cross-attention)
31: x, trend2 = decomp2(x)
32: # Step 5: Feedforward network
33: y = TRANSPOSE x for Conv1d → [B, D, L]
34: y = conv1(y) → activation → dropout
35: y = conv2(y) → dropout
36: TRANSPOSE back y to [B, L, D]
37: x, trend3 = decomp3(x + y)

```

```

38:     # Step 6: Combine all trend components
39:     residual_trend = trend1 + trend2 + trend3
40:     # Step 7: Project residual_trend → c_out
41:     PERMUTE residual_trend to [B, D, L]
42:     APPLY projection convolution (kernel=3, padding='circular')
43:     TRANSPOSE back to [B, L, c_out]
44:     RETURN:
45:         x: the decomposed seasonal output
46:         residual_trend: projected trend component

```

Kode semu 5.17 Algoritma Decoder

Algoritma 21: Decoder

```

1:     DEFINE CLASS Decoder:
2:     INHERIT FROM nn.Module
3:     INITIALIZE(layers, d_model, norm_layer, projection, lstm_hidden, lstm_layers, lstm_dropout):
4:     STORE layers as a list of DecoderLayer modules
5:     STORE optional pre-normalization layer (pre_norm)
6:     STORE optional output projection layer (projection)
7:     SET hidden_size = lstm_hidden IF provided ELSE use d_model
8:     INITIALIZE LSTM layer:
9:         input_size = d_model
10:        hidden_size = hidden_size
11:        num_layers = lstm_layers
12:        batch_first = True
13:        dropout = lstm_dropout (only if lstm_layers > 1)
14:        IF hidden_size != d_model:
15:            INITIALIZE lstm_proj: Linear layer from hidden_size → d_model
16:        ELSE:
17:            SET lstm_proj = Identity() (no change)
18:        INITIALIZE post_norm = LayerNorm(d_model)
19:
20:    FUNCTION forward(x, cross, x_mask=None, cross_mask=None, trend=None):
21:        FOR each DecoderLayer in self.layers:
22:            APPLY DecoderLayer on x with cross input and masks
23:            ADD returned residual_trend to trend
24:        IF pre_norm is defined:
25:            APPLY pre_norm to x
26:        # LSTM Processing
27:        PASS x through LSTM → x_lstm
28:        APPLY lstm_proj to x_lstm to ensure dimension = d_model
29:        APPLY post_norm to x_lstm
30:        IF projection layer is defined:
31:            APPLY projection to x
32:        RETURN:
33:            x: final decoder output
34:            trend: accumulated trend component

```

Kode semu 5.18 Fitness Function

Algoritma 24: Fitness Function

```

1:     DEFINE CONSTANTS:
2:     SYSTEM_LOSS = 0.14
3:     BETA_0 = 3832, BETA_1 = 0.68, BETA_2 = 1.64
4:     BESS_CH_EFF = 0.95, BESS_DCH_EFF = 0.9
5:     COST_CAP_BESS = 50000, ALPHA_I = 0.03, EFF = 0.95
6:     POWER_RATING = 320, ENERGY_RATING = 360
7:     BESS_INSTALLATION_COST = 15
8:     BESS_OPERATION_MAINTENANCE = 5
9:     CCAP = 50000

```

```

10: PV_INSTALLATION_COST = 1200
11: PV_OPERATION_MAINTENANCE = 0.04
12:
13: DEFINE FUNCTION fitness(population):
14:   READ dataset from './dataset/sg_load_wavelet_ms.csv'
15:   PARSE values as (load, sg)
16:   INITIALIZE population_iterated = []
17:   FOR EACH chromosome IN population:
18:     cappv ← chromosome['cappv_desimal']
19:     ebess ← chromosome['ebess_desimal']
20:     sg_total ← SUM of all sg in dataset
21:     INITIALIZE pbess_list, ppv_out_list
22:     FOR EACH (load, sg) IN dataset:
23:       ppv_out = sg × cappv
24:       pbess = ppv_out - load
25:       APPEND ppv_out TO ppv_out_list
26:       APPEND pbess TO pbess_list
27:     INITIALIZE pdischarge_list, pcharge_list
28:     FOR EACH pbess IN pbess_list:
29:       IF pbess < 0:
30:         pdischarge = -pbess / BESS_DCH_EFF
31:       ELSE:
32:         pdischarge = 0
33:       IF pbess > 0:
34:         pcharge = pbess × BESS_CH_EFF
35:       ELSE:
36:         pcharge = 0
37:       APPEND pdischarge TO pdischarge_list
38:       APPEND pcharge TO pcharge_list
39:     INITIALIZE:
40:       state_of_charge_list, depth_of_discharge_list, ncycle_list
41:       ce_soc_list, cbt_list, excess_soc_list, surge_soc_list
42:     FOR i FROM 0 TO LENGTH(pdischarge_list) - 1:
43:       SET initial state_of_charge = 0.8 IF i == 0 ELSE previous value
44:       COMPUTE result_soc = SoC transition equation
45:       CLAMP result_soc within [0, 1]
46:       IF result_soc < 0.2:
47:         ADD result_soc TO surge_soc_list
48:       ELSE:
49:         APPEND 0 TO surge_soc_list
50:       IF result_soc > 0.8:
51:         ADD result_soc TO excess_soc_list
52:       ELSE:
53:         APPEND 0 TO excess_soc_list
54:       APPEND result_soc TO state_of_charge_list
55:       IF discharge occurred:
56:         COMPUTE depth_of_discharge = 1 - result_soc
57:         COMPUTE ncycle using battery degradation formula
58:         COMPUTE ce_soc = CCAP / ncycle
59:         COMPUTE cbt = ce_soc - previous_ce_soc
60:       ELSE:
61:         SET values = 0
62:       APPEND computed values TO respective lists
63:     max_pcharge = MAX(pcharge_list)
64:     max_pdischarge = MAX(pdischarge_list)
65:     pbess_max = MAX(max_pcharge, max_pdischarge)
66:     total_pbess = SUM(pbess_list)
67:     # Cost & Penalty Calculation
68:     cost_pv = (sg_total × cappv × PV_OPERATION_MAINTENANCE) +
69:       (PV_INSTALLATION_COST × cappv)
70:     installation_cost = (pbess_max × (POWER_RATING +
71:       BESS_OPERATION_MAINTENANCE)) +
72:       (ebess × (ENERGY_RATING + BESS_INSTALLATION_COST))
73:     degradation_cost = SUM(cbt_list)
74:     cost_bess = installation_cost + degradation_cost
75:     total_cost = cost_pv + cost_bess

```

```

75:      # Determine if invalid (surge detected)
76:      erase_flag = True IF any value in surge_soc_list < 0 ELSE False
77:      # Append result to population_iterated
78:      APPEND {
79:          "fitness": total_cost,
80:          "cappv_desimal": cappv,
81:          "ebess_desimal": ebess,
82:          "erase": erase_flag,
83:          "cpv": cost_pv,
84:          "ic": installation_cost,
85:          "dc": degradation_cost,
86:          "cbess": cost_bess,
87:          "TC": total_cost
88:      } TO population_iterated
89:      RETURN population_iterated

```

Tabel 5.1 Pengakuan Penggunaan Alat Berbasis AI

No.	Nama Alat AI	Platform	Fungsi yang Digunakan	Bagian Skripsi yang Terpengaruh
1	ChatGPT	OpenAI	Membantu dalam menjelaskan alur kode	Bab 3 (Pengembangan Model)

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis dilahirkan di Surabaya, 03 Desember 2002, merupakan anak kedua dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu di SD Islam Sabilillah Sidoarjo, SMP Al-Falah Deltasari dan SMAN 15 Surabaya. Setelah lulus dari SMAN tahun 2021, Penulis mengikuti SBMPTN dan diterima di Departemen Teknik Informatika FTEIC - ITS pada tahun 2021 dan terdaftar dengan NRP 5025211147.

Di Departemen Teknik Informatika Penulis sempat aktif di beberapa kegiatan seperti menjadi Staff Ahli pada Schematics NLC dan menjadi admin lab NETICS