

TUGAS AKHIR - EE234899

**PENJADWALAN PEMBANGKIT TENAGA LISTRIK
MENGUNAKAN *MIXED INTEGER LINEAR*
PROGRAMMING DENGAN INTEGRASI PERAMALAN
DAYA ENERGI TERBARUKAN**

NATHANAEL DYLAN WANASIDA

NRP 5022201080

Dosen Pembimbing

Dr.Eng. Rony Seto Wibowo, S.T., M.T.

NIP: 197411292000121001

Dedet Candra Riawan, S.T., M.Eng., Ph.D.

NIP: 197311192000031001

Program Studi Sarjana Teknik Elektro

Departemen Teknik Elektro

Fakultas Teknik Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026



TUGAS AKHIR - EE234899

**PENJADWALAN PEMBANGKIT TENAGA LISTRIK
MENGUNAKAN *MIXED INTEGER LINEAR*
PROGRAMMING DENGAN INTEGRASI PERAMALAN
DAYA ENERGI TERBARUKAN**

NATHANAEL DYLAN WANASIDA

NRP 5022201080

Dosen Pembimbing

Dr.Eng. Rony Seto Wibowo, S.T., M.T.

NIP: 197411292000121001

Dedet Candra Riawan, S.T., M.Eng., Ph.D.

NIP: 197311192000031001

Program Studi Sarjana Teknik Elektro

Departemen Teknik Elektro

Fakultas TEknik

Institut Teknologi Sepuluh Nopember

Surabaya

2026



FINAL PROJECT - EE234899

POWER PLANT SCHEDULING USING MIXED INTEGER LINEAR PROGRAMMING WITH INTEGRATED SOLAR POWER FORECASTING

NATHANAEL DYLAN WANASIDA

NRP 5022201080

Advisor

Dr.Eng. Rony Seto Wibowo, S.T., M.T.

NIP: 197411292000121001

Dedet Candra Riawan, S.T., M.Eng., Ph.D.

NIP: 197311192000031001

Electrical Engineering Undergraduate Study Program

Department of Electrical Engineering

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2026

LEMBAR PENGESAHAN

PENJADWALAN PEMBANGKIT TENAGA LISTRIK MENGGUNAKAN MIXED INTEGER LINEAR PROGRAMMING DENGAN INTEGRASI PERAMALAN DAYA ENERGI TERBARUKAN

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat

memperoleh gelar Sarjana Teknik pada

Program Studi S-1 Teknik Elektro

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh : **NATHANAEL DYLAN WANASIDA**

NRP. 5022201080

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr.Eng. Rony Seto Wibowo, S.T., M.T.

Pembimbing

2. Dedet Candra Riawan, S.T., M.Eng., Ph.D.

Ko-pembimbing

3. Prof. Dr. Ir. Mochamad Ashari, M.Eng.

Penguji

4. Dr. Dimas Fajar Uman Putra, S.T., M.T.

Penguji

5. Irfan Ubaidillah ST., MT.

Penguji

SURABAYA

Juli, 2026

APPROVAL SHEET

POWER PLANT SCHEDULING USING MIXED INTEGER LINEAR PROGRAMMING WITH INTEGRATED SOLAR POWER FORECASTING

FINAL PROJECT

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

By : **NATHANAEL DYLAN WANASIDA**

NRP. 5022201080

Approved by Final Project Examiner Team:

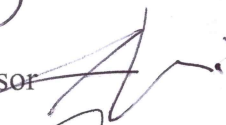
1. Dr.Eng. Rony Seto Wibowo, S.T., M.T.

Advisor



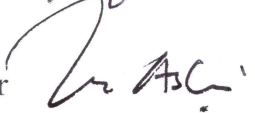
2. Dedet Candra Riawan, S.T., M.Eng., Ph.D.

Co-Advisor

 17/07/26

3. Prof. Dr. Ir. Mochamad Ashari, M.Eng.

Examiner



4. Dr. Dimas Fajar Uman Putra, S.T., M.T.

Examiner



5. Irfan Ubaidillah ST., MT.

Examiner



SURABAYA

July, 2026

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Nathanael Dylan Wanasida / 5022201080
Program studi : S-1 Teknik Elektro
Dosen Pembimbing / NIP : Dr.Eng. Rony Seto Wibowo, S.T., M.T.
/ 197411292000121001
Dedet Candra Riawan, S.T., M.Eng., Ph.D.
/ 197311192000031001

dengan ini menyatakan bahwa Tugas Akhir dengan judul “PENJADWALAN PEMBANGKIT TENAGA LISTRIK MENGGUNAKAN MIXED INTEGER LINEAR PROGRAMMING DENGAN INTEGRASI PERAMALAN DAYA ENERGI TERBARUKAN” 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, 07 Juli 2026

Mengetahui
Dosen Pembimbing


Dr.Eng. Rony Seto Wibowo,
S.T., M.T.
NIP.197411292000121001

Mahasiswa


Nathanael Dylan Wanasida
NRP.5022201080

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Nathanael Dylan Wanasida / 5022201080
Department : Electrical Engineering
Advisor / NIP : Dr.Eng. Rony Seto Wibowo, S.T., M.T.
/ 197411292000121001
Dedet Candra Riawan, S.T., M.Eng., Ph.D.
/ 197311192000031001

Hereby declare that the Final Project with the title of "POWER PLANT SCHEDULING USING MIXED INTEGER LINEAR PROGRAMMING WITH INTEGRATED SOLAR POWER FORECASTING" 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, 07 July 2026

Acknowledged
Advisor


Dr.Eng. Rony Seto Wibowo,
S.T., M.T.
NIP.197411292000121001

Student


Nathanael Dylan Wanasida
NRP.5022201080

ABSTRAK

PENJADWALAN PEMBANGKIT TENAGA LISTRIK MENGGUNAKAN MIXED INTEGER LINEAR PROGRAMMING DENGAN INTEGRASI PERAMALAN DAYA ENERGI TERBARUKAN

Nama Mahasiswa / NRP : Nathanael Dylan Wanasida / 5022201080
Departemen : Teknik ELEKTRO FTEIC - ITS
Dosen Pembimbing : Dr.Eng. Rony Seto Wibowo, S.T., M.T

Abstrak

Semakin banyak pembangkit yang menggunakan sumber energi terbarukan, sumber tersebut mempengaruhi produksi daya pembangkit tersebut. Dimana PLTS dipengaruhi oleh solar irradiasi dan PLTB dipengaruhi oleh kecepatan angin. Kedua variabel tersebut juga dipengaruhi oleh faktor luar seperti iklim dan cuaca. Dikarenakan ketidakstabilan jumlah daya yang dibangkitkan, membuat berbagai masalah seperti, di jam tertentu daya yang seharusnya dibutuhkan untuk memenuhi beban tidak dapat digunakan dikarenakan pembangkit tidak dapat mencapai potensi, atau disaat pembangkit memproduksi daya tetapi beban sudah terpenuhi dan BESS juga sudah penuh maka pembangkit tidak dapat mencapai potensi sebenarnya. Maka itu digunakan model LSTM (long-short term memory), teknik deep learning untuk memahami pola agar dapat memprediksi daya dan membuat daya yang terjadwal oleh pembangkit PLTS dan PLTB yang akhirnya dapat membuat rencana untuk hari kedepannya jadwal-jadwal pembangkit untuk memenuhi beban pada jam tertentu. untuk mencapai ini digunakan metode MILP dimana dapat menghasilkan daya continuous dan integer.

Kata kunci: *Energi Terbarukan, MILP (Mixed-Integer Linear Programming), RNN (Recurrent Neural Network), Ramalan Daya, Sistem Energi Manajemen.*

ABSTRACT

POWER PLANT SCHEDULING USING MIXED INTEGER LINEAR PROGRAMMING WITH INTEGRATED SOLAR POWER FORECASTING

Student Name / NRP : Nathanael Dylan Wanasida / 5022201080
Department : Electrical Engineering FTEIC - ITS
Advisor : Dr.Eng. Rony Seto Wibowo, S.T., M.T

Abstract

The increase of power plants using renewable energy sources is affecting power generated by the power plants itself due to the unstable nature of the energy source. Power plants using energies like, wind and solar are generating power uncontrollably. Wind energy is affected by wind speed variable and solar energy is affected by solar irradiance variable. Both variables are affected by weather, nature and climate. Due to the erratic behavior in power generated, it is causing problem such as, power generated that is supposed to fulfill a load demand cannot fill it because the power plant cannot reach its' potential due to if using solar panels, the solar panels not getting enough sun. Or another case, being the load demand is fulfilled and the BESS in the system is full the plant is forced to turn off, due to the power being generated has nowhere else to go resulted in wasted power. That is why using Deep Learning method LSTM to understand the pattern of power being generated by the renewable power plants so the model can create forecasted power. And that forecasted power can be used in dispatch scheduling for the day that comes after. To do this the model is using the MILP method to schedule the dispatching of every power plants in the power system.

Keywords: *Renewable Energy, MILP (Mixed-Integer Linear Programming), RNN (Recurrent Neural Network), Power Forecasting, Energy Management System.*

DAFTAR ISI

LEMBAR PENGESAHAN	i
APPROVAL SHEET	ii
PERNYATAAN ORISINALITAS	iii
STATEMENT OF ORIGINALITY	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
DAFTAR SIMBOL	xi
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.....	4
2.1 Hasil Penelitian Terdahulu	4
2.2 Dasar Teori.....	5
BAB 3 METODOLOGI.....	16
3.1 Formulasi Permasalahan	16
3.2 Metode yang digunakan	22
3.3 Bahan dan peralatan yang digunakan	41
3.4 Urutan pelaksanaan penelitian.....	49
BAB 4 Hasil dan Pembahasan	51
4.1 Hasil penelitian.....	51
BAB 5 Kesimpulan dan Saran	79
5.1 Kesimpulan	79
5.2 Saran	79
DAFTAR PUSTAKA	80
LAMPIRAN	81
BIODATA PENULIS	104

DAFTAR GAMBAR

Gambar 2.1 Direct, Diffuse dan Reflected Solar Irradiasi.....	6
Gambar 2.2 Rangkaian Ekvivalen Solar sel	6
Gambar 2.3 Bentuk Kurva I-V Solar Cell	8
Gambar 2.4 Contoh Heatmap Correlation Matrix.....	10
Gambar 2.5 Contoh Struktur ANN.....	11
Gambar 2.6 Struktur RNN	12
Gambar 2.7 Struktur Memory Cell LSTM	12
Gambar 2.8 Struktur Memory Cell LSTM bagian Forget Gate	13
Gambar 2.9 Struktur Memory Cell LSTM bagian Input Gate.....	13
Gambar 2.10 Struktur <i>Memory Cell</i> LSTM bagian <i>Cell State Update</i>	13
Gambar 2.11 Struktur Memory Cell LSTM bagian Output Gate	14
Gambar 3.1 Topologi Sistem Tenaga Listrik.....	16
Gambar 3.2 Overview Sistem Energi Manajemen Hybrid	16
Gambar 3.3 Sistem Peramalan Daya	17
Gambar 3.4 Potensi Daya yang Dihasilkan PLTS 1 hari Kedepan	17
Gambar 3.5 Latitude dan Longitude PLTS Lab REIDI.....	25
Gambar 3.6 Heatmap Correlation Matrix Data Cuaca	27
Gambar 3.7 Data Inverter per jam dalam rentang 1 Hari (24 Jam)	42
Gambar 3.8 Data Inverter per jam dalam rentang 7 Hari (168 Jam)	42
Gambar 3.9 Flow-Chart Programming Model LSTM Visual Studio Code.....	43
Gambar 3.10 Flow-Chart Programming Model LSTM Visual Studio Code.....	44
Gambar 4.1 Perbandingan Hasil Prediksi Solar Irradiasi dengan Nilai Nyata Solar Irradiasi tanggal 19-25 November 2025.....	52
Gambar 4.2 Perbandingan Hasil Prediksi Daya dengan Nilai Nyata Daya tanggal 19-25 November 2025	53
Gambar 4.3 Perbandingan Hasil Prediksi Daya yang diprediksi pada tanggal 13 April 2026 dengan Nilai Nyata Daya tanggal 14 April 2026	54
Gambar 4.4 Perbandingan Hasil Prediksi yang diprediksi pada tanggal 15 April 2026 dengan Nilai Nyata Solar Irradiasi tanggal 16 sampai 22 April 2026.....	55
Gambar 4.5 Hasil Power Dispatch sebelum MILP	56
Gambar 4.6 Hasil PLTS Dispatch sebelum MILP	57
Gambar 4.7 Hasil Power Dispatch setelah MILP	58
Gambar 4.8 Hasil <i>Dispatch</i> PLTS sesudah EMS	58
Gambar 4.9 Hasil nilai dispatch tiap pembangkit dalam rentang 24 jam	59
Gambar 4.10 Start Up Variabel tiap Jam tiap Pembangkit	61
Gambar 4.11 Grafik Power Balance MILP.....	62
Gambar 4.12 Grafik Energy Balance BESS (MWh).....	62
Gambar 4.13 Grafik Energy Balance SOC BESS (%)	63
Gambar 4.14 Grafik Energy Balance BESS (MWh) sebelum MILP	63
Gambar 4.15 Grafik Energy Balance SOC BESS (%) Sebelum MILP.....	64
Gambar 4.16 Grafik PLTD Selayar (DEUTZ) dengan Batasan Pmin-Pmax	64
Gambar 4.17 Grafik PLTU Selayar (CUMMINS) dengan Batasan Pmin-Pmax.....	65
Gambar 4.18 Grafik PLTU SELAYAR (DAIHATSU) dengan Batasan Pmin-Pmax	66
Gambar 4.19 Batasan Ramp Rate PLTD Selayar (DEUTZ)	67
Gambar 4.20 Batasan Ramp Rate PLTU Selayar (CUMMINS).....	67
Gambar 4.21 Batasan Ramp Rate PLTU Selayar (DAIHATSU)	68

Gambar 4.22 Jumlah Energi yang dihasilkan tiap pembangkit sebelum dan Sesudah MILP	69
Gambar 4.23 Jumlah Energi yang dihaikan tiap tipe Pembangkit sebelum dan sesudah MILP.....	69
Gambar 4.24 Perbandingan Biaya sebelum dan sesudah MILP Tiap Pembangkit dalam 1 Hari.....	70
Gambar 4.25 Perbandingan Biaya Energi sebelum dan Sesudah MILP berdasarkan tipe dalam 1 jam.....	71
Gambar 4.26 Kurva Daya nyata dan Daya Predikisi tanggal 20 April 2026 Upscaled.....	72
Gambar 4.27 Perbandingan Prediksi Daya PLTS tanggal 20 April 2026 dengan Daya Asli yang terekam pada tanggal 20 April 2026	73
Gambar 4.28 Perbandingan energi yang dihasilkan pembangkit PLTS, BESS, PLTD, dan PLTU pada tanggal 20 April 2026	73
Gambar 4.29 Perbandingan Charging/Discharging yang Diprediksi dengan yang Sebenarnya	74
Gambar 4.30 Perbandinga Total Dispatch Daya PLTU yang Diprediksi dan yang Sebenarnya	75
Gambar 4.31 Perbandingan Prediksi Energi yang dihasilakn tiap PLTU dengan yang Nyata	76
Gambar 4.32 Perbandingan Prediksi Energi yang dihasilakn tiap PLTD dengan yang Nyata_.....	77
Gambar 4.33 Perbandingan Prediksi dispacth PLTD dan yang Nyata pada tiap Jam.....	77

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	4
Tabel 2.2 Matrix korelasi antara variabel X dan Y	9
Tabel 3.1 Potensi Produksi PLTS.....	18
Tabel 3.2 Contoh Dataset Pembangkit untuk Formulasi Matrix.....	19
Tabel 3.3 Contoh Data Beban dalam 1 Hari	19
Tabel 3.4 Line Code untuk Meminta Data ke Situs	23
Tabel 3.5 Fitur dan Parameter untuk Meminta Data dari Situs.....	24
Tabel 3.6 Dataset sebelum disaring Heatmap Correlation Matrix	26
Tabel 3.7 Dataset Cuaca setelah Disaring Heatmap Correlation Matrix.....	28
Tabel 3.8 Rentang Tanggal Data Sejarah dan Data Ramalan	29
Tabel 3.9 Parameter Model LSTM dari Library Keras	30
Tabel 3.10 Spesifikasi Modul Panel PV	31
Tabel 3.11 Versi Python dan Library yang digunakan	45
Tabel 3.12 Datasheet Solar Panel digunakan di REIDI.....	45
Tabel 3.13 Datasheet Inverter digunakan di REIDI	46
Tabel 3.14 Dataset Pembangkit untuk Menguji program MILP	46
Tabel 3.15 Potensi Output PLTS untuk menguji MILP (30 Mei 2026)	47
Tabel 3.16 Data Beban Harian Pulau Selayar	48
Tabel 4.1 Tahap ke-1 Pembagian DataSejarah untuk Training dan Testing	51
Tabel 4.2 Tahap ke-2 Pembagian Data Sejarah dan Data Ramalan untuk Training dan Testing Rentang 24 Jam (1 hari)	53
Tabel 4.3 Tahap ke-2 Pembagian Data Sejarah dan Data Ramalan untuk Training dan Testing Rentang 168 Jam (7 hari)	54
Tabel 4.4 Status Tiap Pembangkit dalam Rentang 24 Jam.....	60

DAFTAR SIMBOL

I_{PV}	Arus Photovolatic (A)
I_{ph}	Arus Photocurrent (A)
I_D	Arus Diode (A)
$I_{R_{sh}}$	Arus Series (A)
$I_{R_{sh}}$	Arus Shunt (A)
V_{MPP} (V)	Maximum power voltage (V)
I_{MPP} (A)	Maximum power current (A)
V_{oc} (V)	Open-circuit voltage (A)
I_{sc} (A)	Short-circuit current (A)
T_{NOCT}	Suhu NOCT biasanya benilai 20 °C
T_{STC}	Suhu STC biasanya benilai 25 °C
$T_{ambient temp}$	Suhu Ambien (°C)
$G_{solar Irradiance}$	Solar irradiansi (W/m^2)
G_{STC}	Solar irradiansi dalam kondisi STC (W/m^2)
G_{NOCT}	Solar irradiansi dalam kondisi NOCT (W/m^2)
$NOCT$	Nominal Operating Cell Temperature (°C)
K_i	Temperature coefficeint of I_{max} (arus masksimum) (%/°C)
V_{PV}	Tegangan Photovoltaic (V)
V_{oc}	Tegangan Open Circuit (V)
q	Muatan elektron (1.602×10^{-19} Coulomb)
R_{sh}	Resistansi Shunt (ohm)
R_s	Resistansi Seri (ohm)
n	Faktor idealitas diode
k_b	Boltzman konstant (1.381×10^{-23} J/Kelvin)
E_g	Band gap energy (eV)
r_{xy}	Koefisien pearson korelasi variabel X dan Y
X_i	Variabel X ke-i
Y_i	Variabel Y ke-i
$\bar{X}$	Mean variabel X
$\bar{Y}$	Mean variabel Y
x'	Hasil normalisasi variabel x
x_i	Variabel x ke-i
x_{min}	Nilai minimum variabel x
x_{max}	Nilai maksimum variabel x
μ_i	Mean data
σ_i	Standar deviasi data
h	Hidden state
W	Weight
b	Bias
f_t	Forget gate
i_t	Input gate
σ	Sigmoid activation function
C_t	Cell state terbaru
$\hat{C}_t$	Potensi cell state
o_t	Output gate
F	Fungsi objektif
A_{eq}	Matriks A equality constrain
B_{eq}	Matriks B equality constrain
A	Matriks A inequality constrain
B	Matriks B inequality constrain
l_b	Lower bounds
u_b	Upper bounds
P_i^t	Daya dispatch pembangkit i , jam ke-t

U_i^t	Status pembangkit i , jam ke-t
SUV_i^t	Start-up variabel pembangkit i , jam ke-t
SUC_i	Start-up Cost pembangkit i
nl	Number of length (panjang data beban = 24 jam/hari)
nu	Number of units (jumlah pembangkit dalam system)
RU_i	Ramp-Up pembangkit i
RD_i	Ramp-Down pembangkit i
UT_i	Uptime pembangkit i
DT_i	Downtime pembangkit i
E_0	Energi awal BESS
E_{min}	Energi minimum Bess
E_{max}	Energi maksimum Bess
P_{BESS}^t	Daya Dispatch BESS pada jam ke-t
P_{load}^t	Daya Beban pada jam ke-t
P_{DC}	Daya DC
K_P	Temperature coefficeint of P_{max} (daya peak solar panel) (%/°C)
n_{series}	Jumlah n solar panel yang terhubung secara seri
$n_{parallel}$	Jumlah n solar panel yang terhubung paralel
$P_{DC Inverter}$	Daya DC yang masuk ke dalam inverter (Watt)
$P_{AC EMS}$	Daya aktif setelah konversi inverter (Watt)
n_{row}	Jumlah n sistem PV array (1 inverter = 1 array)
$\eta_{inverter}$	Effisiensi inverter
P_{PLTS}^t	Daya dispatch PLTS pada jam ke-t

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Sumber energi terbarukan menjadi salah satu topik yang penting beberapa tahun ini. Pembangkit tenaga dari sumber energi konvensional seperti, bahan bakar fosil dan batu bara diminta untuk diubah menggunakan sumber energi yang terbarukan (Renewable energy source). Permintaan ini berasal dari negara maju dan berkembang. Sumber energi konvensional menghasilkan berbagai jenis masalah lingkungan seperti pemanasan global, perubahan iklim, dan polusi. Dengan menggunakan sumber energi terbarukan, seperti, solar, angin, air, biomassa, geothermal, dan hydrogen untuk pembangkitan tenaga Listrik dapat mengatasi permasalahan lingkungan saat ini. Selain itu, juga dapat menghasilkan perekonomian yang lebih baik, dengan mengurangi harga dalam pembangkitan tenaga Listrik, dikarenakan menggunakan sumber yang lebih alami.(Ang et al., 2022). Salah satu energi yang sering digunakan di beberapa tahun ini adalah energi solar. Pembangkit Listrik tenaga solar sudah banyak digunakan di seluruh dunia dan akan semakin banyak di tahun kedepannya. Menurut *IRENA (International Renewable Energy Agency)*, dari tahun 2013 seluruh dunia memiliki kapasitas energi solar sebanyak 135.57 GW dan pada tahun 2022 sebanyak 1046.62 GW dengan produksi terbanyak dihasilkan oleh China sebesar 392.4 GW di tahun 2022. IRENA memberikan proyeksi pada tahun 2030 kapasitas solar PV seluruh dunia meningkat sampai 2814 GW di tahun 2030 dan 8519 GW di tahun 2050 dimana 40% adalah instalasi PV di atap-atap bangunan atau perumahan dan sisanya 60% dari pembangkit (Pourasl et al., 2023). Seperti sumber energi yang sudah ada dan digunakan, energi solar juga mempunyai tantangan yang harus dilalui untuk menghasilkan tenaga listrik. Pembangkitan energi Listrik menggunakan tenaga surya dipengaruhi oleh berbagai faktor, seperti Lokasi dan waktu penempatan panel surya. Selain itu, pembangkitan Listrik juga dipengaruhi oleh cuaca, variabel seperti suhu, kelembapan, kecepatan angin, dan awan dapat mempengaruhi performa dan efisiensi panel surya. Variabel yang disebutkan sulit diprediksi dan dikendalikan. Variabel tersebut dapat menyebabkan sistem pembangkit yang tidak seimbang, yang akhirnya mengakibatkan ketidakseimbangan antara pembangkitan tenaga dan kebutuhan beban dan menghasilkan masalah-masalah operasional.(Lim et al., 2022).

Dikarenakan banyaknya penggunaan sumber energi yang dapat diperbaharui untuk pembangkitan tenaga listrik, pemerintah mengadopsi regulasi baru untuk mengimplementasikan teknologi energi diperbaharui untuk meningkatkan efisiensi penggunaan energi dan perencanaan konservasi energi. Teknologi seperti sistem energi manajemen, yaitu sistem yang digunakan dengan tujuan mengatur sumber energi lebih dari satu, seperti berbagai macam pembangkit tenaga listrik, untuk mensuplai kebutuhan beban. Selain tujuan utama diatas, teknologi sistem energi manajemen dapat digunakan untuk memaksimalkan penggunaan tenaga yang dihasilkan pembangkit-pembangkit, proteksi komponen dari overloading, meningkatkan kestabilan dalam sistem tenaga, meminimalkan biaya produksi energi untuk pembangkit dan mengatur aliran energi untuk kebutuhan daya di waktu tertentu dan kondisi tertentu. Dikarenakan sifat-sifat sumber energi terbarukan yang

tidak stabil dan sulit diatur, semakin sulit untuk menggunakan atau membuat strategi untuk sistem energi manajemen (Olatomiwa et al., 2016).

Sumber energi terbarukan, memiliki sifat yang tidak dapat diprediksi, tapi bagaimana jika bisa, dengan kemajuan teknologi deep learning dapat digunakan untuk memprediksi daya yang dihasilkan di waktu kedepan. Inilah Tugas Akhir yang dilakukan, menggunakan peramalan daya solar menggunakan deep learning LSTM dan mengintegrasikannya dengan program sistem energi manajemen. Peramalan daya solar atau solar power forecasting yang dilakukan di tugas akhir ini adalah peramalan daya menggunakan irradiansi solar yang diprediksi melalui model *LSTM (Long Short-term Memory)* menggunakan data cuaca seperti, Suhu Ambien, kelembapan, kecepatan angin, tutupan awan, dan lainnya. Kemudian hasil prediksi irradiansi solar tersebut akan dikonversi menjadi daya prediksi yang kemudian akan dimasukkan program *energy dispatch* berbagai pembangkit, menggunakan *MILP (Mixed Integer Linear Programming)* untuk mengatur jumlah energi yang diberikan tiap pembangkit dan *BESS (Battery Energy Storage System)*. Dengan menggunakan strategi ini diharapkan dapat dilakukan perencanaan distribusi tenaga yang lebih efektif, menghindari daya yang terbuang dari baterai yang penuh dan juga menghindari pemadaman di waktu kedepan. Juga diharapkan untuk digunakan untuk sumber energi yang lain seperti energi solar.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan permasalahan yang menjadi akan diselesaikan dalam Tugas Akhir ini antara lain:

1. Bagaimana cara memprediksi daya menggunakan data ramalan cuaca dan seberapa akurat hasil ramalan tersebut ?
2. Bagaimana hasil ramalan daya dari program prediksi digunakan dalam system manajemen energi ?
3. Apakah hasil program gabungan dari prediksi daya dan system energi manajemen mencukupi fungsi objektif ?

1.3 Batasan Masalah

Batasan masalah pada Tugas Akhir ini dengan tujuan penelitian adalah:

1. Tugas Akhir ini memfokuskan akurasi peramalan daya dan seluruh strategi sistem energi manajemen
2. Data cuaca yang didapatkan berasal dari situs <https://open-meteo.com/>. Data yang digunakan dari situs tersebut didapatkan dan diproses dari ECMWF (European Centre for Medium-Range Weather Forecasts), ERA5 (ECMWF Reanalysis 5th Generation), dan CERRA (Copernicus European Regional Reanalysis)
3. Lokasi untuk konversi dan pengambilan data solar irradiansi adalah REIDI di ITS. Digunakan spesifikasi solar panel yang digunakan REIDI untuk konfigurasi. Dan juga lokasi REIDI untuk latitude dan longitude pembangkit simulasi.
4. Menggunakan 1 Pembangkit PLTS (energi terbarukan), 1 BESS, 11 Pembangkit konvensional (Energi Tidak Terbarukan) terdiri dari PLTD dan PLTU
5. Simulasi menggunakan Python di Visual Studio Code
6. Menggunakan data beban yang berbasis dengan data beban Pulau Selayar
7. Nilai batasan dataset pembangkit diasumsikan untuk bukti konsep MILP.

1.4 Tujuan

Tujuan penelitian Tugas Akhir ini adalah :

1. Mengetahui akurasi peramalan daya menggunakan daya cuaca.
2. Menggunakan hasil ramalan daya dalam program untuk economic dispatch dan Unit Commitment
3. Memastikan PLTS menggunakan semua daya yang diprediksi.

1.5 Manfaat

Manfaat dari penelitian ini adalah :

1. Dapat memprediksi daya untuk PLTS agar dapat lebih baik perencanaan dispatch daya dan menghindari pemadaman dan kehilangan daya.
2. Sebagai bentuk kontribusi dalam keseimbangan pembangkit listrik, memperbaiki ekonomi dalam sektor tenaga, dan kemajuan teknologi smart grid.
3. Sebagai referensi untuk mahasiswa lain di bidang yang mirip.

BAB 2 TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Berikut adalah hasil penelitian yang terkait yang digunakan sebagai referensi dalam penelitian yang dilakukan dalam Tugas Akhir:

Tabel 2.1 Penelitian Terkait

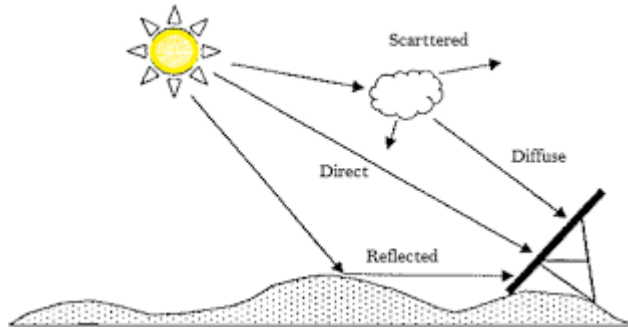
Judul	<i>Forecast-Based Energy Management for Optimal Energy Dispatch in a Microgrid</i>
Nama Peneliti	Francisco Durán, Wilson Pavón and Luis Ismael Minchala
Tahun	2024
Hasil Penelitian	Penelitian ini mengusulkan strategi sistem energi manajemen energi dispatch menggunakan peramalan beban dan daya PV dalam micro-grid. Peramalan digunakan model LSTM neural network yang dilatih menggunakan daya nyata PV dan konsumsi daya. Pengoptimalan sistem difokuskan dengan mengatur SOC dari BESS. Dilakukan analisis ekonomik dan teknis dengan komparasi strategi yang memfokuskan berdasarkan optimasi ekonomis. Pertama dilakukan peramalan daya menggunakan dua buah LSTM untuk memprediksi daya dari PV dan konsumsi daya dengan interval 1 jam. Kemudian diformulasikan masalah MILP agar mendapat hasil optimal dalam kontribusi daya. Dalam MILP terdapat objective function untuk total energy dispatch pada waktu tertentu, constrain daya aktif untuk keseimbangan daya, dan keterbatasan constrain agar BESS tidak melewati keterbatasan yang ditentukan. Hasil dari MILP ini adalah energy dispatch tiap pembangkit terhadap kebutuhan daya yang tidak melewati batasan BESS dan masih stabil keseimbangan daya.
Terkait Penelitian	Sebagai referensi peramalan daya yang digunakan dalam sistem energi manajemen.
Judul	<i>LSTM-Based Prediction of Solar Irradiance and Wind Speed for Renewable Energy Systems</i>
Nama Peneliti	AhmedA.Alguhi dan Abdullah M.Al-Shaalan
Tahun	2025
Hasil Penelitian	Penelitian ini mengusulkan menggunakan meodel LSTM untuk melakukan prediksi iradiasi solar dan kecepatan angin. Data yang digunakan dalam penelitian ini adalah data iradiasi solar dan kecepatan angin histori yang dibagi untuk pelatihan dan pengujian model, 80% untuk pelatihan dan 20% untuk pengujian. Ditentukan parameter yang digunakan dalam pembangunan model LSTM. Parameter yang ditentukan adalah <i>unit</i> atau <i>layer</i> yang digunakan untuk pembelajaran pola dan output prediksi langkah waktu berikutnya, <i>epoch</i> , untuk iterasi model agar dapat mempelajari pola yang kompleks. Juga ada parameter lain seperti <i>batch size</i> , <i>learning rate</i> , <i>gradient threshold</i> dan <i>optimizer</i> . Setelah menentukan parameter, dilakukan prediksi dan mengevaluasi model menggunakan RMSE (Root Mean Squared Error), dilakukan perbandingan prediksi dan data pengujian. Kemudian dibandingkan nilai ketepatan model LSTM dalm penelitian

	dengan model lain yang sering digunakan untuk teknik peramalan, ARIMA, SVM, GWO-LSSVM, dan Hybrid model.
Terkait Penelitian	Sebagai referensi untuk pembangunan model LSTM dan juga nilai-nilai parameter yang digunakan dalam pembangunan model.
Judul	<i>Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM</i>
Nama Peneliti	Xiangyun Qing, Yugang Niu
Tahun	2018
Hasil Penelitian	Penelitian ini adalah penelitian tentang prediksi solar iradiasi 1 hari kedepan dengan interval per 1 jam dengan menggunakan data ramalan cuaca menggunakan LSTM. Data Irradiance dikumpulkan di sebuah pembangkit listrik tenaga surya di pulau Santiago, Cape Verde. Disana terdapat 2 musim yaitu, musim berangin dari bulan Oktober-Juli dan musim hujan dari Agustus-September. Data cuaca yang dikumpulkan berasal dari Praia International Airport, terdiri dari jenis cuaca (13) (Cerah, Mendung, Hujan, Berawan dan lainnya), Suhu, Kecepatan Angin, Titik Embun, Kelembapan, dan Visibilitas. Data peramalan cuaca adalah sebagai input, yang akan mengeluarkan output iradiasi solar. Saat pelatihan model data dibagi menjadi 80% untuk pelatihan dan sisanya untuk pengujian. yang dilatih adalah nilai-nilai tiap variabel cuaca yang menghasilkan nilai iradiasi solar di waktu tertentu. Setelah pelatihan dilakukan evaluasi melalui RMSE. Hasil penelitian model LSTM dibandingkan dengan algoritma lain (Persistence, LR, dan BPNN) dari keempat algoritma tersebut, LSTM lebih akurat dari BPNN, LR, dan Persistence menurut RMSE.
Terkait Penelitian	Sebagai referensi untuk data-data cuaca yang digunakan dalam pembangunan model LSTM

2.2 Dasar Teori

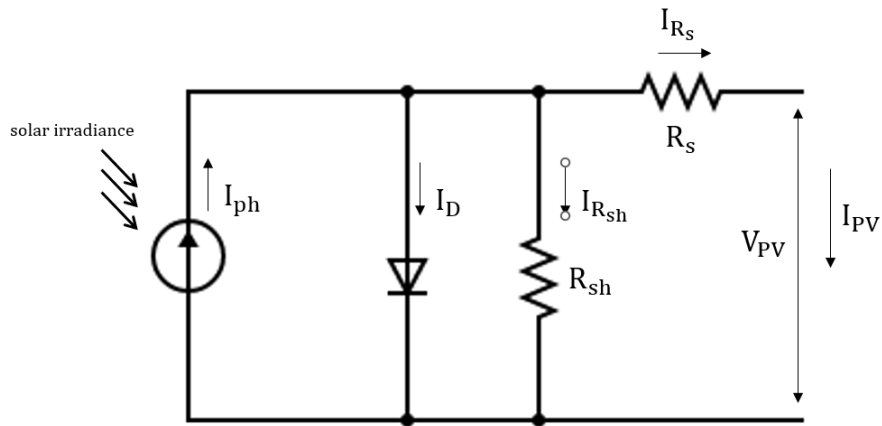
2.2.1 Energi Solar

Energi solar adalah energi yang paling bersih dan yang paling berlimpah diantara energi lain yang terbarukan. Energi solar berasal dari matahari, lebih detail nya adalah cahaya yang dihasilkan matahari. Cahaya matahari terdiri dari dua bagian, cahaya radiasi secara langsung (direct) dan secara tidak langsung (indirect). Radiasi solar secara langsung berarti secara langsung dari matahari, yang arah nya belum berubah. Sedangkan radiasi solar tidak langsung adalah cahaya matahari yang sudah terpantul (reflected) dan tersebar (diffused) . Jenis-jenis solar radiasi ini yang akan berubah menjadi tenaga listrik dengan solar panel.



Gambar 2.1 *Direct, Diffuse dan Reflected Solar Irradiasi*
(El Mghouchi, Youness. (2022))

Solar panel terbuat dari beberapa solar cell yang dirangkai secara seri. Solar cell inilah yang akan mengubah irradiasi solar menjadi energi listrik. Arus listrik yang dihasilkan bergantung dengan banyaknya irradiasi solar yang mengenai modul solar panel. Semakin besar irradiasi solar, semakin besar daya yang dihasilkan (Silva et al., 2024). Perubahan irradiasi solar dapat terlihat dari rangkaian ekivalen solar cell dibawah ini dan rumusnya:



Gambar 2.2 Rangkaian Ekivalen Solar sel

Dari gambar 2.2 didapatkan rumus arus PV yaitu:

$$I_{PV} = I_{R_s} \quad (2.1)$$

$$I_{PV} = I_{ph} - I_D - I_{R_{sh}} \quad (2.2)$$

Dan rumus tegangan PV yaitu:

$$V_{ph} = V_D = V_{R_{sh}} \quad (2.3)$$

$$V_{PV} = V_{R_{sh}} - V_{R_s} \quad (2.4)$$

Variabel I_{ph} dipengaruhi oleh variabel $G_{solar\ Irradiance}$:

$$I_{ph} = I_{sc} + K_i(T_{cell} - T_{STC}) \times \frac{G_{solar\ Irradiance}}{G_{STC}} \quad (2.5)$$

Variabel I_D merupakan arus diode yang dirumuskan :

$$I_D = I_0 \left[\exp \left(\frac{q(V_{PV} + I_{PV}R_s)}{nk_b T_{cell}} \right) - 1 \right] \quad (2.6)$$

Variabel I_0 adalah arus kebalikan saturasi diode yang dirumuskan:

$$I_0 = I_{RS} \left[\frac{T_{cell}}{T_{NOCT}} \right]^3 \cdot \exp \left(\frac{qE_g \left(\frac{1}{T_{NOCT}} - \frac{1}{T_{cell}} \right)}{nk_b} \right) \quad (2.7)$$

Variabel I_{RS} adalah arus kebalikan saturasi:

$$I_{RS} = \frac{I_{SC}}{\left[\exp \left(\frac{q(V_{OC})}{nk_b T_{cell}} \right) - 1 \right]} \quad (2.8)$$

Maka jika persamaan 2.6 diteruskan menggunakan persamaan 2.7 dan 2.8 maka:

$$I_D = \frac{I_{SC} \left[\exp \left(\frac{q(V_{PV} + I_{PV}R_s)}{nk_b T_{cell}} \right) - 1 \right]}{\left[\exp \left(\frac{q(V_{OC})}{nk_b T_{cell}} \right) - 1 \right]} \left[\frac{T_{cell}}{T_{NOCT}} \right]^3 \cdot \exp \left(\frac{qE_g \left(\frac{1}{T_{NOCT}} - \frac{1}{T_{cell}} \right)}{nk_b} \right) \quad (2.9)$$

Variabel $I_{R_{sh}}$ merupakan arus resistor shunt yang dirumuskan:

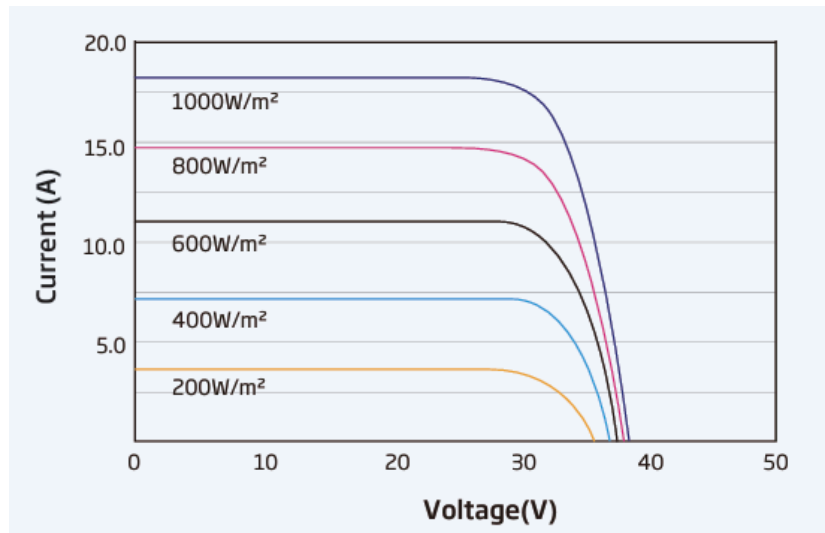
$$I_{R_{sh}} = \frac{V_{PV} + I_{PV}R_s}{R_{sh}} \quad (2.10)$$

$$V_{PV} = I_{R_{sh}} R_{sh} - I_{PV} R_s \quad (2.11)$$

Dimana T_{cell} dipengaruhi oleh variabel $T_{ambien temp}$:

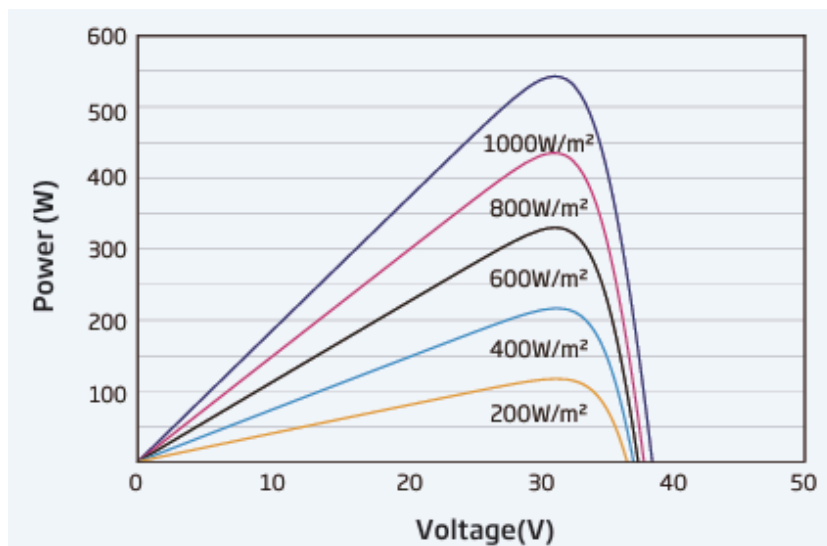
$$T_{cell} = T_{ambient temp} + (NOCT - T_{NOCT}) \frac{G_{Solar Irradiance}}{G_{NOCT}} \quad (2.12)$$

Dari persamaan 2.2, persamaan 2.4 dihasilkan kurva I-V yang dipengaruhi oleh $G_{Solar Irradiance}$, V_{PV} dan I_{PV} . Kurva suatu cell PV dapat dilihat sebagai berikut:



Gambar 2.3 Bentuk Kurva I-V Solar Cell

Dari gambar 2.3 Pada kurva I-V 1000W/m² saat $I_{PV} = I_{SC}$ berarti $V_{PV} = 0$, dan saat $V_{PV} = V_{OC}$, maka $I_{PV} = 0$, dari gambar 2.3 dan persamaan 2.9 semakin besar nilai tegangan V_{PV} mendekati nilai V_{OC} maka arus I_{PV} akan menurun sampai 0 saat tegangan PV mencapai tegangan open-circuit, ini dibuktikan dari persamaan 2.2, persamaan 2.9 dan persamaan 2.10. Secara idealnya menurut persamaan 2.11 perbandingan nilai resistansi shunt harus sangat besar agar arus yang melewatinya kecil, sedangkan resistansi seri sangat kecil agar arus dapat melewatinya. Maka dari gambar 2.3 didapatkan kurva daya maksimum:



Gambar 2.4 Kurva Daya *Maximum Power point*

Dari gambar 2.4 untuk mendapatkan nilai daya yang paling besar maka harus diketahui tegangan PV harus besar tetapi tidak mencapai nilai tegangan open-circuit. Dan juga nilai arus PV harus besar dan tidak mencapai arus short-circuit. Untuk mendapatkan tegangan V_{MPP} dan I_{MPP} dikendalikan bebannya dalam perangkat maximum power-point tracking (MPPT)

2.2.2 Battery Energy Storage System

Battery Energy Storage System (BESS) merupakan sebuah perangkat yang dapat menyerap dan mengeluarkan energi listrik. Dalam sebuah sistem energi listrik, BESS digunakan untuk menyediakan energi cadangan, dimana saat sebuah sistem tidak dapat mencapai target beban, maka energi yang tersimpan dalam BESS akan digunakan untuk memenuhi beban tersebut. Juga digunakan untuk menyerap energi saat target beban tercapai dan terdapat energi yang berlebihan. Dalam sistem energi listrik energi yang berlebihan ini berasal dari pembangkit yang menghasilkan daya yang tidak terkendali (Pembangkitan daya yang dipengaruhi oleh kondisi alam, seperti sumber angin (PLTB) dan solar (PLTS)).

2.2.3 Heatmap Correlation Matrix

Heatmap correlation matrix merupakan tabel yang memberi gambaran korelasi antara variable-variabel yang terdapat di sebuah dataset. Untuk mengisi matrix ini diperlukan perumusan yang bernama Pearson's Correlation Coefficient (r), dimana koefisien ini berada di rentang -1 dan 1.

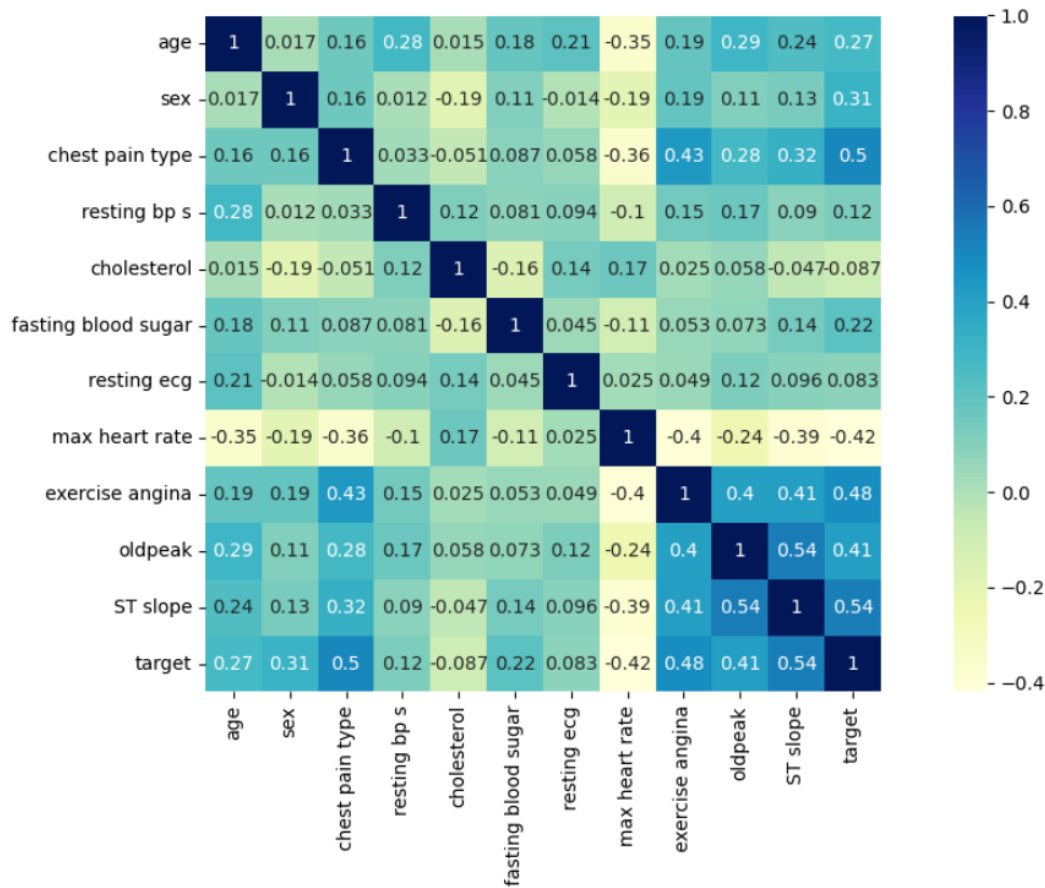
$$r_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (2.4)$$

Dibawah ini adalah contoh matrix korelasi menggunakan variabel X dan Y :

Tabel 2.2 Matrix korelasi antara variabel X dan Y

X	1	r_{XY}
Y	r_{XY}	1
Variabel	X	Y

Dibawah ini adalah contoh heatmap correlaton matrix yang menggunakan variabel dari dataset untuk mendeteksi kondisi awal serangan jantung:



Gambar 2.4 Contoh Heatmap Correlation Matrix
(Aversano et al., 2022)

2.2.4 Normalisasi Data

Normalisasi data adalah proses yang menstruktur ulang nilai-nilai data pada setiap variabel agar terlihat sama, biasanya di rentang 0 dan 1. Nilai 0 merepresentasikan data bernilai 0 pada sebuah variabel dan 1 adalah nilai maksimum pada sebuah variabel tersebut. Beberapa metode normalisasi data, diantaranya adalah metode Z-score dan min-max.

1. Metode min-max

Metode min-max mengubah dimensi fitur dalam dataset menjadi nol dan 1 atau -1 dan 1. Bisa dinyatakan dengan rumus dibawah ini:

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2.5)$$

Dimana x' adalah data yang telah dinormalisasi, x_i adalah variabel input i , x_{min} adalah nilai minimum variabel i dan x_{max} adalah nilai maksimum variabel i

2. Metode Z-score

Metode Z-score mengubah rata-rata setiap variabel dalam dataset menjadi nol dan standar deviasi menjadi 1. Bisa dinyatakan dengan rumus berikut:

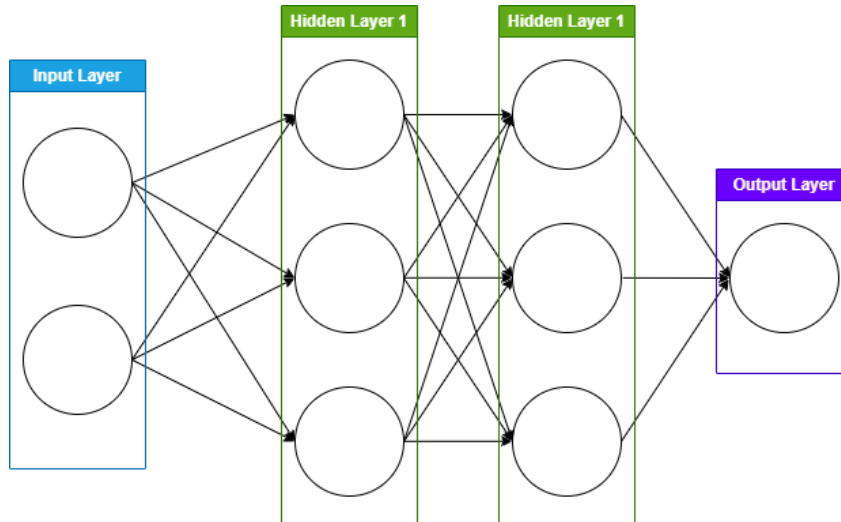
$$x' = \frac{x_i - \mu_i}{\sigma_i} \quad (2.6)$$

Dimana x' adalah data yang telah dinormalisasi, x_i adalah variabel input i , μ_i adalah rata-rata dari variabel i dan σ_i adalah standar deviasi dari variabel i

Dalam struktur RNN, normalisasi data merupakan tahap pre-processing yang dilakukan dengan tujuan untuk mengurangi bias dan mempercepat pembelajaran fitur.

2.2.5 ANN (Artificial Neural Network)

Artificial Neural Network (ANN), adalah sebuah model komputasi yang terinspirasi dari jaringan saraf pada otak. Manusia belajar, berdasarkan dari informasi yang kita punya (input) dan juga dari aksi yang kita lakukan setelah informasi-informasi tersebut diproses (output). ANN berguna sebagai alat untuk menjelaskan interaksi antara input dan output untuk melihat dan menemukan pola dalam data sekuensial yang kompleks. ANN terdiri dari 3 lapis yaitu, Input Layer, Hidden Layer, dan Output Layer(Cao, 2023).



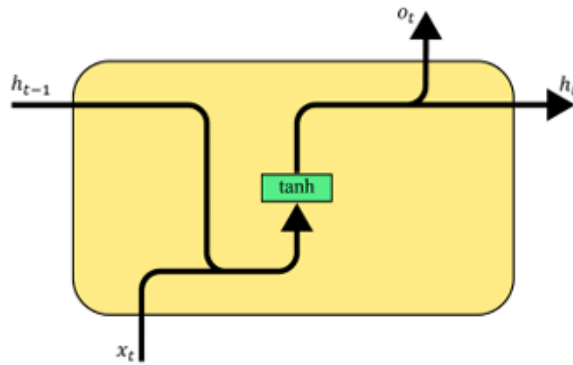
Gambar 2.5 Contoh Struktur ANN

2.2.6 RNN (Recurrent Neural Network)

Recurrent Neural Network (RNN) adalah sebuah salah satu jenis *Artificial Neural Network (ANN)* yang struktur dasarnya unit. Unit yang sama digunakan pada setiap tahap waktu dan input terdiri dari data pada waktu saat ini (x_t). *Hidden Layer* pada tahap waktu t (h_t) dapat dirumuskan sebagai berikut :

$$h_t = \tan(W_h h_{t-1} + W_x x_t + b) \quad (2.7)$$

W_h adalah matriks bobot untuk *Hidden Layer* pada waktu t , h_{t-1} adalah *Hidden layer* pada waktu sebelumnya, W_x adalah bobot untuk input dalam *Hidden Layer* pada waktu t , b adalah vektor bias. Dapat dilihat dari rumus bahwa h_t menggunakan fungsi aktivasi tanh. Rumus h_t tersebut bisa dilihat dari gambar cell hidden layer dibawah ini:

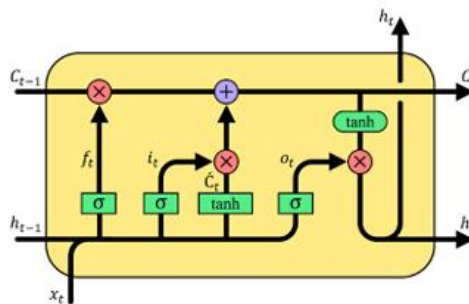


Gambar 2.6 Struktur RNN
(Ghimire et al., 2022)

RNN masih mempunyai kekurangan yaitu, permasalahan Vanishing gradient sehingga sulit untuk mempelajari ketergantungan jangka panjang. Untuk mengatasi masalah ini RNN menggunakan variasi yang dikembangkan, yaitu LSTM (Long-Short Term Memory) dan GRU (Gate Recurrent Unit). LSTM mengenalkan *memory cells* dan *gates* (input, forget, dan output) untuk memilih informasi yang diingat atau dilupakan.

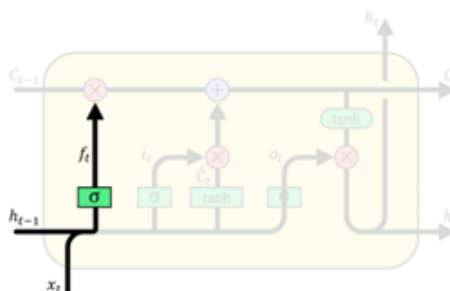
2.2.7 LSTM (Long Short Term Memory)

LSTM merupakan variasi dari RNN dengan menggantikan hidden layer dengan memory cell khusus. Sama dengan ANN dan RNN, LSTM memiliki 3 lapisan utama, *Input layer*, *Hidden Layer*, dan *Output Layer*. Penggantian Hidden layer tradisional RNN, diubah menjadi memory cell khusus untuk LSTM.



Gambar 2.7 Struktur *Memory Cell* LSTM
(Ghimire et al., 2022)

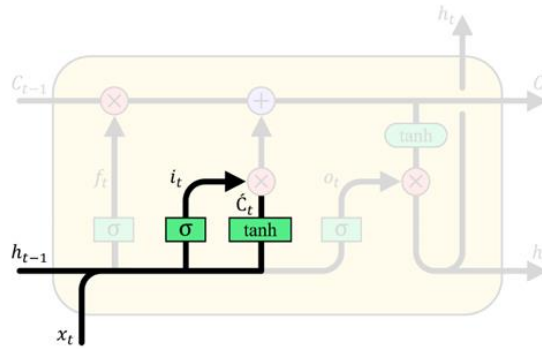
Berdasarkan Gambar 2.8 Struktur *Memory cell* LSTM terdiri dari 3 buah gate, *input gate*, *forget gate*, dan *output gate*. Dimulai dari forget gate (f_t) :



Gambar 2.8 Struktur *Memory Cell* LSTM bagian *Forget Gate*
(Ghimire et al., 2022)

$$f_t = \sigma(W_f [h_{t-1}; x_t] + b_f) \quad (2.8)$$

Persamaan tersebut adalah persamaan forget gate. Pada forget gate digunakan fungsi aktivasi sigmoid agar f_t berada dalam rentang 0 sampai 1, dimana ditentukan seberapa banyak informasi yang masuk dari Memory cell sebelumnya (C_{t-1}) dilupakan. Selanjutnya akan berlangsung ke Input gate, dimana gate ini berfungsi untuk menentukan informasi baru apa saja yang perlu ditambahkan ke sel memory dengan menggunakan fungsi sigmoid, Persamaan input gate (i_t) adalah sebagai berikut:

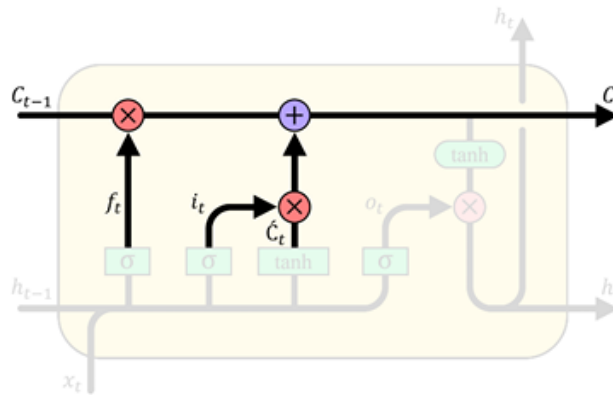


Gambar 2.9 Struktur *Memory Cell* LSTM bagian *Input Gate*
(Ghimire et al., 2022)

$$i_t = \sigma(W_i [h_{t-1}; x_t] + b_i) \quad (2.9)$$

$$\hat{C}_t = \tanh(W_c [h_{t-1}; x_t] + b_c) \quad (2.10)$$

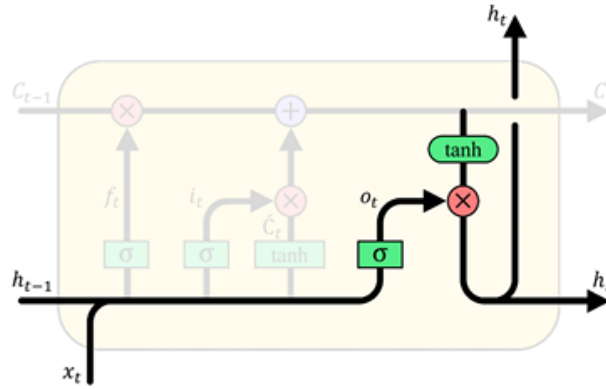
$\hat{C}_t$ adalah nilai baru yang akan ditambahkan ke status sel. Sebelum menuju output gate dilakukan pembaharuan cell (C_t). Dapat dilihat pada gambar berikut:



Gambar 2.10 Struktur *Memory Cell* LSTM bagian *Cell State Update*
(Ghimire et al., 2022)

$$C_t = (f_t \times C_{t-1}) + (i_t \times \hat{C}_t) \quad (2.11)$$

$(f_t \times C_{t-1})$ adalah informasi yang masih dipertahankan dari Memory cell sebelumnya, dan $(i_t \times \hat{C}_t)$ adalah informasi baru yang ditambahkan ke status cell. Dimana kedua ini akan dijumlahkan menghasilkan status cell baru C_t . Yang terakhir adalah output Gate.



Gambar 2.11 Struktur Memory Cell LSTM bagian Output Gate (Ghimire et al., 2022)

$$o_t = \sigma(W_o [h_{t-1}; x_t] + b_o) \quad (2.12)$$

$$h_t = h_t \times \tanh(C_t) \quad (2.13)$$

Pada Output Gate, fungsi aktivasi sigmoid digunakan untuk mengatur output memory cell untuk menghasilkan hidden state. Output gate berfungsi untuk mengeluarkan hidden state baru dan juga hidden state baru yang akan digunakan sebagai h_{t-1} untuk memory cell berikutnya (Qing & Niu, 2018).

2.2.8 MAE (Mean Absolute Error)

Mean Absolute Error (MAE) adalah salah satu metrik yang digunakan untuk mengukur akurasi dari model prediksi. MAE menghitung rata-rata besar error antaran nilai prediksi dan nilai asli. Persamaan MAE dapat dilihat sebagai berikut

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (2.14)$$

Dimana n adalah jumlah data, y_i adalah nilai asli, $\hat{y}_i$ adalah nilai prediksi. $|y_i - \hat{y}_i|$ adalah nilai mutlak error, sehingga menghindari nilai error positif dan negatif. MAE mengukur kesalahan prediksi dalam satuan yang sama dengan data asli (Dhaked et al., 2023). Misalkan variabel Suhu prediksi dan asli memiliki MAE = 2.5 berarti tiap kali prediksi suhu memiliki besar error rata-rata sekitar 2.5°C antara nilai asli dan prediksi.

2.2.9 MILP (Mixed-Integer Linear Programming)

Mixed-Integer Linear Programming adalah model optimasi matematika yang digunakan untuk mencari solusi pada masalah kompleks. Solusi tersebut dapat berupa nilai bersifat integer

atau continuous(Durán et al., 2024). Masalah kompleks berupa fungsi objektif dan memiliki constrain. Berikut adalah persamaan fungsi objektif.

$$F(x) = \min_x f^T x \quad (2.15)$$

Fungsi objektif mempunyai persamaan linear constrain:

$$A_{eq}x = B_{eq} \quad (2.16)$$

Dan juga mempunyai pertidaksamaan linear constarin:

$$Ax \leq B \quad (2.17)$$

Sebagai batasan solusi variabel diantara lower bounds (l_b) dan upper bounds (u_b):

$$l_b \leq x \leq u_b \quad (2.18)$$

Solusi-solusi x dari metode MILP dapat bersifat integer dan continuous jika solusi variable ingin bersifat integer maka diperlukan persamaan intcon.

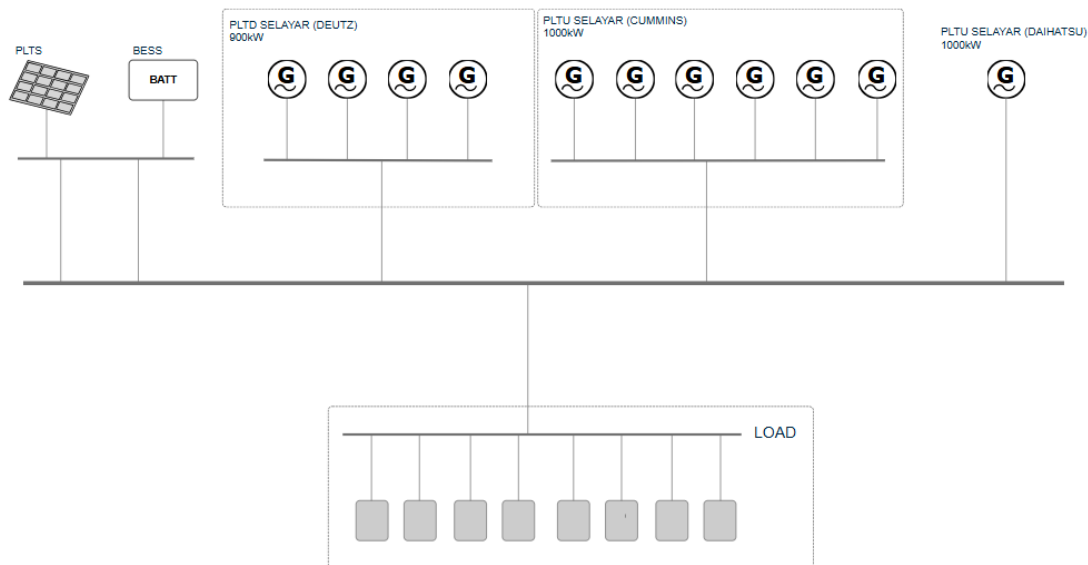
$$x \text{ (intcon)} \in \{\dots, -1, 0, 1, \dots\} \quad (2.19)$$

Dari persamaan 2.9 solusi x bersifat binary yang berarti nilai variable x hanya bisa bernilai 0 atau 1.

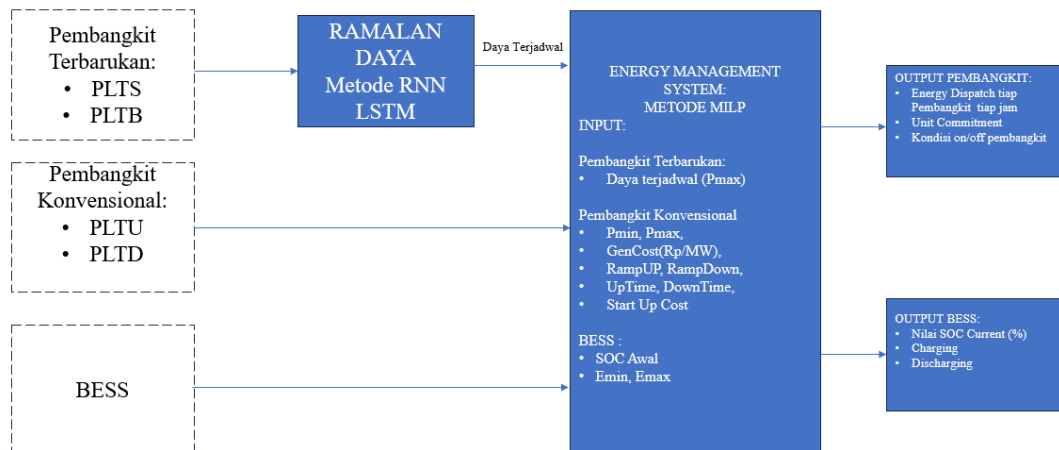
BAB 3 METODOLOGI

3.1 Formulasi Permasalahan

Dalam penelitian ini terdapat dua permasalahan yaitu, memprediksi daya dan menggunakan hasil prediksi tersebut untuk Unit Commitment. Dibawah ini adalah bentuk sistem yang direncanakan untuk melakukan penelitian ini:



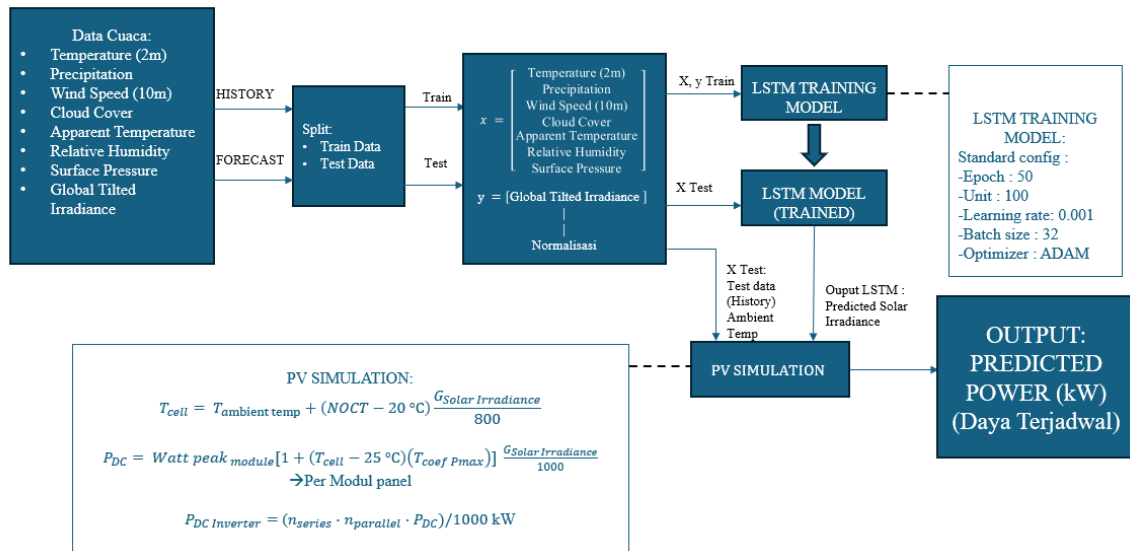
Gambar 3.1 Topologi Sistem Tenaga Listrik



Gambar 3.2 Overview Sistem Energi Manajemen Hybrid

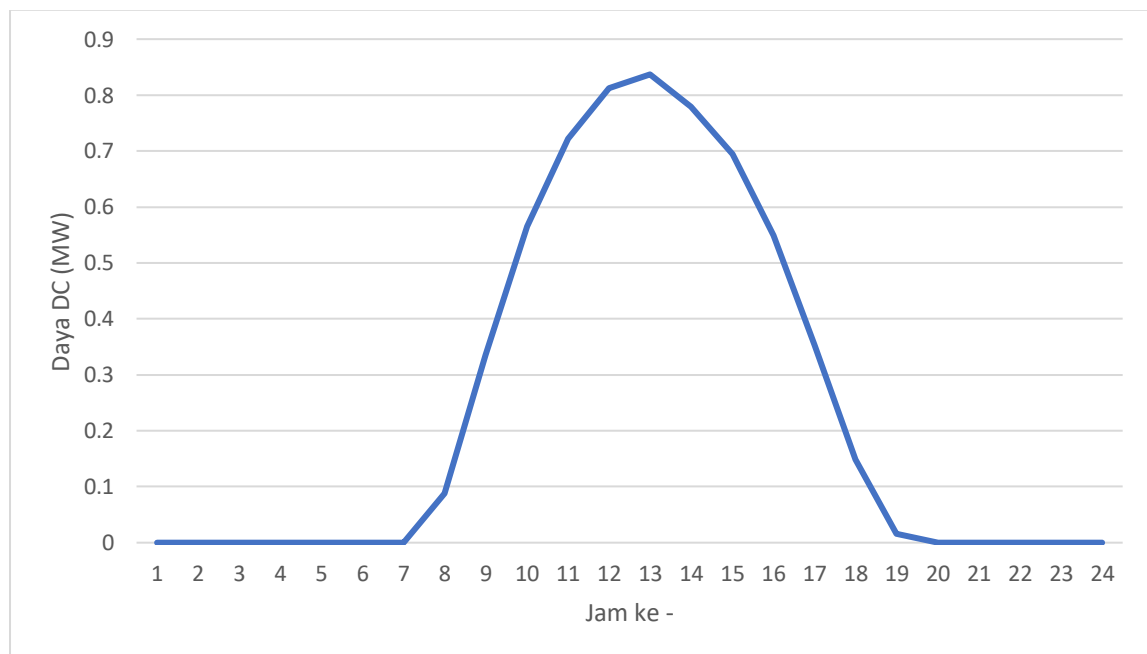
3.1.1 Tahap Ramalan Daya

Pada penelitian dari gambar 3.2 diperlukan potensi daya (ramalan daya untuk hari kedepan) sebelum masuk ke tahap Sistem Manajemen Energi. Untuk itu, diperlukan model peramalan daya untuk menghasilkan hasil tersebut:



Gambar 3.3 Sistem Peramalan Daya

Dari Gambar 3.3 bisa dilihat model sistem peramalan daya. Pada model ini digunakan model LSTM sebagai alat deep learning untuk mempelajari pola-pola solar irradiance berdasarkan input data cuaca, setelah melakukan prediksi solar irradiasi, informasi tersebut digunakan sebagi input simulasi model PV array untuk menghasilkan daya DC dari ramalan solar irradiance dan ramalan suhu hari kedepan. Hasil dari sistem ini adalah potensi daya yang dihasilkan hari-hari kedepan dengan interval tiap jam.



Gambar 3.4 Potensi Daya yang Dihasilkan PLTS 1 hari Kedepan

Gambar 3.4 adalah potensi daya yang dapat diproduksi oleh PLTS untuk 1 hari, tentu saja grafik pada gambar 3.4 belum tentu benar tetapi sudah ada gambaran untuk potensi penghasilannya untuk lebih siap menghadapi hari kedepan.

Tabel 3.1 Potensi Produksi PLTS

Jam	Produksi PLTS (MW)
0 - 1	0.0000
1 - 2	0.0000
2 - 3	0.0000
3 - 4	0.0000
4 - 5	0.0000
5 - 6	0.0000
6 - 7	0.0000
7 - 8	0.0877
8 - 9	0.3349
9 - 10	0.5652
10 - 11	0.7215
11 - 12	0.8126
12 - 13	0.8370
13 - 14	0.7786
14 - 15	0.6948
15 - 16	0.5496
16 - 17	0.3536
17 - 18	0.1481
18 - 19	0.0162
19 - 20	0.0000
20 - 21	0.0000
21 - 22	0.0000
22 - 23	0.0000
23 - 24	0.0000

Tabel 3.1 adalah nilai-nilai potensi Daya kurva yang dihasilkan oleh PLTS tiap jam dalam rentang 1 hari dari gambar 3.4.

3.1.2 Tahap MILP

Dari tahap ramalan daya hasil potensi daya pembangkit PLTS digunakan untuk formulasi fungsi objektif dan persamaan dan pertidaksamaan MILP. Untuk membuat fungsi objektif dan persamaan dan pertidaksamaan batasan diperlukan data-data tiap pembangkit yang ada pada sistem tenaga listrik dibawah ini adalah contoh data pembangkit yang digunakan untuk mencapai tujuan ini:

Tabel 3.2 Contoh Dataset Pembangkit untuk Formulasi Matrix

PEMBANGKIT	Pmin	Pmax	Gen Cost (Rp/Mw)	RU	RD	UT	DT	STC
PLTS	0.0	P_{PLTS}^t	Rp -	50000.0	50000	4.0	2.0	Rp -
PLTD	0.5	5.0	Rp 3,801,435.9	4.0	4.0	4.0	2.0	Rp 2,121,120.0
PLTU	0.5	2.0	Rp 3,700,000.0	3.0	3.0	2.0	1.0	Rp 3,181,680.0
BESS	-2.0	2.0	Rp -	4.0	4.0	1.0	1.0	Rp -
Total Pmax		9.0						
BESS	Emin	Emax						
	0.0	5.0						

Tabel 3.3 Contoh Data Beban dalam 1 Hari

Jam	Load Data (Mw)	Jam
1	4.0	0 - 1
2	4.0	1 - 2
3	5.0	2 - 3
4	4.0	3 - 4
5	4.5	4 - 5
6	4.5	5 - 6
7	4.3	6 - 7
8	4.5	7 - 8
9	5.0	8 - 9
10	5.5	9 - 10
11	5.0	10 - 11
12	5.0	11 - 12
13	5.0	12 - 13
14	5.5	13 - 14
15	5.5	14 - 15
16	5.0	15 - 16
17	5.0	16 - 17
18	7.0	17 - 18
19	7.0	18 - 19
20	6.5	19 - 20
21	6.0	20 - 21
22	6.5	21 - 22
23	6.0	22 - 23
24	5.5	23 - 24

Sebelum memulai pemodelan harus ditetapkan dari data pembangkit pada tabel 3.2 dan tabel 3.3 sebagai berikut bahwa:

1. $nu = 4$ (*jumlah pembangkit dalam sistem*)
2. $i = [1,2,3,4] \rightarrow PLTS, PLTD, PLTU, BESS$
3. $nl = 24$ (*jam dalam 1 hari*) $\rightarrow$ *dari data beban*
4. $t = [1,2,3, \dots, 24] \rightarrow$ *jam ke 1, jam ke 2, \dots, jam ke 24*

Berdasarkan informasi-informasi diatas maka dapat memformulasikan:

1. Fungsi Objektif

Unit commitment yang bertujuan untuk menghasilkan solusi biaya operasi sistem paling rendah dengan variabel daya dispatch, status pembangkit tiap jam, dan biaya tiap kali menyalakan pembangkit. Fungsi objektif dapat dirumuskan:

$$F(P_i^t, U_i^t, SUC_i^t) = \text{Min} \sum_{t=1}^{nl=24} \sum_{i=1}^{nu=4} [F_{Pi}(P_i^t) + F_{Ui}(U_i^t) + F_{SUCi}(SUC_i^t)] \quad (3.1)$$

2. Pertidaksamaan intcon

Pertidaksamaan agar variable yang bernilai integer U_i^t Dimana hanya bernilai 0 atau 1 tidak melebihi nilai 1:

$$U_i^t \leq 1 \quad (3.2)$$

3. Batasan daya yang dihasilkan

Dari gambar 3.5 tiap pembangkit memiliki daya minimum dan daya maksimum. Ini adalah variable untuk merepresentasikan daya minimal yang harus dibangkitkan oleh pembangkit saat kondisi nyala dan daya yang tidak boleh dibangkitkan pembangkit melewati batas maksimum. Dapat dirumuskan dalam pertidaksamaan berikut:

$$U_i^t P_{i,min} \leq P_i^t \leq U_i^t P_{i,max} \quad (3.3)$$

4. Batasan Ramp rate

Ramp rate terdiri dari ramp up dan ramp down yaitu Batasan perbedaan daya yang dibangkitkan tiap jam oleh pembangkit. Dapat dirumuskan dalam pertidaksamaan berikut:

Jika perbedaan daya negative (daya turun):

$$P_i^{(t-1)} - P_i^t \leq RD_i \quad (3.4)$$

Jika perbedaan daya positif (daya naik):

$$-P_i^{(t-1)} + P_i^t \leq RU_i \quad (3.5)$$

5. Batasan Uptime

Uptime berarti waktu minimal pembangkit tiap kali menyala. Sebagai contoh jika uptime sebuah pembangkit 4 jam, maka setiap kali kondisi pembangkit dari OFF ke ON, pembangkit harus tetap ON minimal selama 4 jam.

$$\sum_{j=t}^{t+UT_i-1} u(t) \geq UT_i(u(t) - u(t-1)) \quad (3.6)$$

6. Batasan Downtime

Kebalikannya dari batasan uptime, contohnya jika Downtime bernilai 2 jam perbedaannya adalah tiap kali kondisi pembangkit dari ON ke OFF maka pembangkit harus dalam kondisi OFF selama 2 jam

$$\sum_{j=t}^{t+DT_i-1} u(t) \leq DT_i[1 + (u(t) - u(t-1))] \quad (3.7)$$

7. Pertidaksamaan Startup Cost

Setiap kali pembangkit mengalami kondisi dari OFF ke ON maka pembangkit terkena biaya Startup, biasanya terdapat dalam pembangkit bertenaga uap untuk *warm-up* pembangkit.

Agar variabel tidak bernilai negatif diberi pertidaksamaan:

$$SUV_i^t \geq 0 \quad (3.8)$$

Agar pertidaksamaan hanya aktif saat kondisi pembangkit berubah dari OFF ke ON:

$$SUV_i^t \geq SUC_i[(-u(t-1) + u(t))] \quad (3.9)$$

8. Batasan kondisi SOC BESS

Agar nilai SOC berada diantara SOC minimum yaitu 0 dan SOC maksimum 100% diberi persamaan:

$$E_{min} \leq E_0 - \sum_{t=1}^j P_{BESS}^t \times (t - (t-1)) \leq E_{max} \quad (3.10)$$

9. Persamaan Power Balance

Persamaan daya yang dibangkitkan tiap jam tiap pembangkit yang ada dalam sistem harus jumlahnya memenuhi data beban pada jam tersebut. Maka diberikan pertidaksamaan sebagai berikut:

$$P_{load}^t = \sum_{i=1}^{nu=4} P_i^t \quad (3.11)$$

10. Persamaan BESS siap untuk hari berikut

Agar BESS siap digunakan untuk keesokan harinya, dimana SOC BESS tidak digunakan tiap harinya maka diberi persamaan agar SOC pada jam pertama di hari sama dengan jam terakhir di hari atau jam ke-24:

$$\sum_{t=1}^{t=24} P_{BESS}^t \times (t - (t - 1)) = 0 \quad (3.12)$$

Persamaan dan pertidaksamaan diatas digunakan untuk pemodelan matrix yang akan dipakai pada MILP, F adalah matrix fungsi objektif. Matrix A, B Aeq, Beq yaitu matrix-matrix inequality dan equality constrain.

3.2 Metode yang digunakan

Dalam penelitian ini digunakan berbagai metode untuk menyaring data-data dan juga menyesuaikan data untuk siap dimasukkan ke dalam program prediksi dan program sistem manajemen energi. Metode-metode yang digunakan termasuk, API Request, untuk meminta data yang dibutuhkan untuk prediksi solar irradiance dari situs <https://open-meteo.com/>. Data yang diambil berupa data cuaca sejarah dan data cuaca ramalan. Heatmapping Correlation, untuk menyaring variabel data cuaca yang relevan untuk memprediksi solar irradiance, MinMax Scaling untuk menormalisikan data cuaca agar siap menjadi input LSTM. Metode MILP (Mixed Integer Linear Programming) untuk menjadi basis program sistem manajemen energi.

3.2.1 API REQUEST

Application Programming Interface (API) Request adalah sebuah pesan atau arahan digital yang dikirim oleh sebuah program yang biasanya digunakan untuk meminta data. Sebagai contoh adalah program seperti YouTube, tiap kali kita mengklik suatu video, dikirim pesan untuk memutar video yang kita klik. Dalam penelitian ini API Call digunakan untuk meminta data cuaca sejarah dan ramalan. Bisa dilihat kode yang digunakan di gambar dibawah ini:

Tabel 3.4 Line Code untuk Meminta Data ke Situs

<pre>import datetime yesterday = datetime.date.today() - datetime.timedelta(days=1) print(yesterday) import openmeteo_requests import pandas as pd import requests_cache from retry_requests import retry # Setup the Open-Meteo API client with cache and retry on error cache_session = requests_cache.CachedSession('.cache', expire_after = -1) retry_session = retry(cache_session, retries = 5, backoff_factor = 0.2) openmeteo = openmeteo_requests.Client(session = retry_session)</pre>	API Request Command Line
<pre>url = "https://archive-api.open-meteo.com/v1/archive" params = { "latitude": -7.2850234, "longitude": 112.7961397, "start_date": "2024-01-01", "end_date": f"{yesterday}", "hourly": ["rain", "cloud_cover", "temperature_2m", "relative_humidity_2m", "wind_speed_10m", "surface_pressure", "precipitation", "apparent_temperature", "global_tilted_irradiance"], "timezone": "Asia/Bangkok", "wind_speed_unit": "ms", "timeformat": "unixtime", "tilt": 12.251, } responses = openmeteo.weather_api(url, params=params)</pre>	OpenMeteo Parameter
<pre>response = responses[0] print(f"Coordinates: {response.Latitude()}°N {response.Longitude()}°E") print(f"Elevation: {response.Elevation()} m asl") print(f"Timezone: {response.Timezone()} {response.TimezoneAbbreviation()}") print(f"Timezone difference to GMT+0: {response.UtcOffsetSeconds()}s") # Process hourly data. The order of variables needs to be the same as requested. hourly = response.Hourly() hourly_rain = hourly.Variables(0).ValuesAsNumpy() hourly_cloud_cover = hourly.Variables(1).ValuesAsNumpy() hourly_temperature_2m = hourly.Variables(2).ValuesAsNumpy() hourly_relative_humidity_2m = hourly.Variables(3).ValuesAsNumpy() hourly_wind_speed_10m = hourly.Variables(4).ValuesAsNumpy() hourly_surface_pressure = hourly.Variables(5).ValuesAsNumpy() hourly_precipitation = hourly.Variables(6).ValuesAsNumpy() hourly_apparent_temperature = hourly.Variables(7).ValuesAsNumpy() hourly_global_tilted_irradiance = hourly.Variables(8).ValuesAsNumpy() hourly_data = {"date": pd.date_range(start = pd.to_datetime(hourly.Time(), unit = "s", utc = True), end = pd.to_datetime(hourly.TimeEnd(), unit = "s", utc = True), freq = pd.Timedelta(seconds = hourly.Interval()), inclusive = "left")} hourly_data["temperature_2m"] = hourly_temperature_2m hourly_data["relative_humidity_2m"] = hourly_relative_humidity_2m hourly_data["apparent_temperature"] = hourly_apparent_temperature hourly_data["precipitation"] = hourly_precipitation hourly_data["rain"] = hourly_rain hourly_data["surface_pressure"] = hourly_surface_pressure hourly_data["cloud_cover"] = hourly_cloud_cover hourly_data["wind_speed_10m"] = hourly_wind_speed_10m hourly_data["global_tilted_irradiance"] = hourly_global_tilted_irradiance hourly_dataframe = pd.DataFrame(data = hourly_data) print("\nHourly data\n", hourly_dataframe)</pre>	Pembuatan Dataframe

Dari Tabel 3.4 adalah garis perintah untuk meminta data fitur yang diinginkan dari situs berdasarkan parameter yang ditentukan. Dibawah ini adalah parameter dan fitur yang digunakan pada penelitian ini.

Tabel 3.5 Fitur dan Parameter untuk Meminta Data dari Situs

Fitur	Parameter
Datetime hourly	Latitude
Temperature_2m	Longitude
Apparent Temperature	Tilt angle
Precipitation	Timezone
Rain	Unit wind speed
Surface Pressure	Start date
Cloud cover	End date
Wind speed _10m	
Global tilted irradiance	

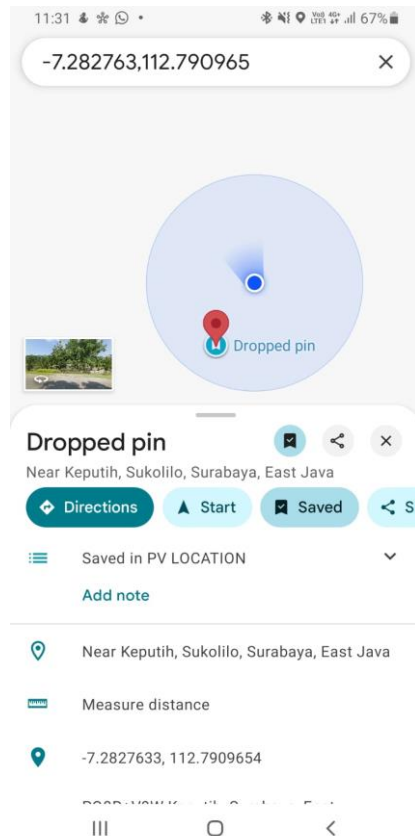
Fitur yang diinginkan dalam penelitian ini adalah sebagai berikut:

1. Datetime hourly
Tanggal dan waktu di lokasi lab REIDI untuk menentukan nilai fitur-fitur lain di tanggal dan jam tertentu
2. Temperature_2m
Suhu ambien yang berketinggian 2 meter dari permukaan,
3. Apparent Temperature
Suhu yang dirasakan orang-orang, dimisalkan suhu ambien adalah 25°C tetapi yang dirasakan orang-orang adalah 30°C dikarenakan faktor lain seperti humiditas/kelembapan, intensitas sinar UV, kualitas udara, kecepatan angin, dan lainnya
4. Precipitation
Fitur yang menandakan sebuah cairan (hujan) ataupun padat (salju) yang turun dari langit yang menutupi permukaan yang diukur dengan unit mm, berarti jika nilai precipitation hujan 1 mm, berarti hujan akan menutupi permukaan setinggi 1mm. Biasanya nilai precipitation yang tinggi adalah salju yang bisa mencapai 100mm yaitu 10cm, yang berarti salju dapat menutupi permukaan setinggi 10cm.
5. Rain
Fitur rain sama dengan precipitation tetapi hanya khusus untuk hujan, sedangkan fitur precipitation adalah hujan dan salju.
6. Surface Pressure
Surface Pressure adalah tekanan atmosferik yang berada pada lokasi tertentu di bumi, dalam penelitian ini lokasi tersebut adalah lab REIDI.
7. Cloud Cover yaitu persentase penutupan awan. Wind speed_10m adalah kecepatan angin yang berketinggian 10 meter dari permukaan bumi.
8. Global Tilted Irradiance adalah Solar irradiasi berdasarkan sudut kemiringan panel surya untuk mendapat nilai sudut kemiringan ini diselesaikan dengan sudut kemiringan pemasangan solar panel di REIDI.

Dibawah ini adalah parameter-parameter yang ditentukan dalam pengambilan fitur-fitur yang ada diatas:

1. Latitude dan Longitude

Latitude adalah garis horizontal yang digunakan untuk mengukur jarak utara dan selatan dari khatulistiwa, Longitude adalah garis vertikal untuk mengukur jarak timur dan barat dari meridian utama. Dalam penelitian ini digunakan untuk memetakan lokasi REIDI agar dapat diberikan data cuaca di lokasi REIDI. Latitude dan Longitude lab REIDI berturut-turut adalah -7.2827633 dan 112.7909654.



Gambar 3.5 Latitude dan Longitude PLTS Lab REIDI

2. Tilt Angle

Tilt angle adalah parameter untuk global tilted irradiance khusus untuk panel surya. Sudut kemiringan panel surya berpengaruh dalam penerimaan solar irradiance dari matahari. Dalam penelitian ini Tilt angle solar panel di lab REIDI adalah 12.251°.

3. Timezone

Zona waktu digunakan untuk fitur datetime, dikarenakan lab REIDI terletak di Surabaya, Indonesia maka zona waktu yang digunakan adalah Asia/Bangkok, yaitu GMT+7

4. Unit Wind Speed

Unit Wind speed adalah parameter yang digunakan untuk merubah satuan kecepatan angin, pada penelitian ini, digunakan satuan meter/s tetapi bisa diubah menjadi km/h dan juga mph (mile per hour)

5. Start date dan end date

Start date dan end date adalah parameter yang digunakan untuk meminta data-data dalam rentang start date sampai end date. Bergantung kegunaannya bisa digunakan untuk mengambil data yang sudah terekam atau data ramalan cuaca.

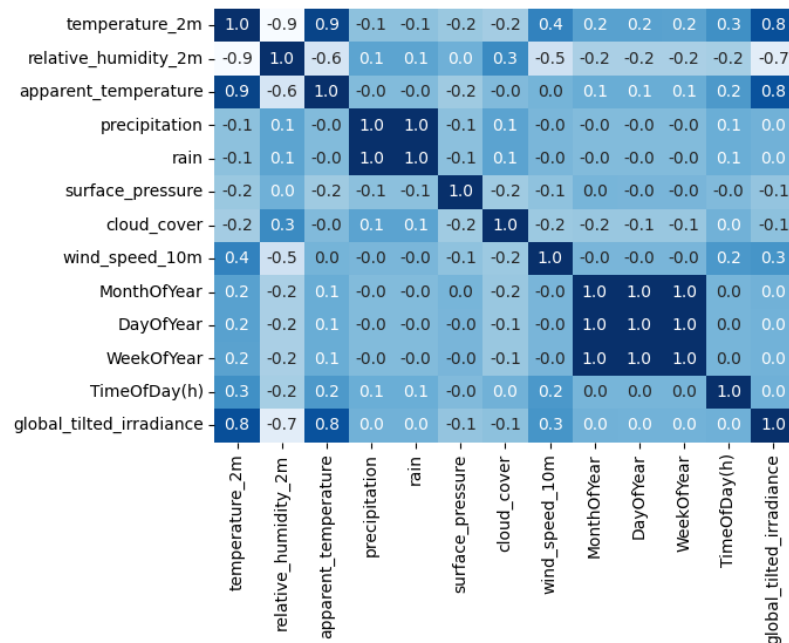
Tabel 3.6 Dataset sebelum disaring Heatmap Correlation Matrix

Date _Time	temp er_2 m	relative_hu midity_2m	apparent_t emperature	precip itation	r ain	surface_ pressure	cloud_ cover	wind_sp eed_10m	global_tilted _irradiance
1/2/2 024 0:00	25.3	95.3	30.1	1.1	1. 1	1008.7	100.0	2.9	0.0
1/2/2 024 1:00	24.9	96.5	30.8	0.9	0. 9	1008.6	100.0	0.7	0.0
1/2/2 024 2:00	25.0	95.3	30.7	0.2	0. 2	1008.2	100.0	1.0	0.0
1/2/2 024 3:00	24.9	96.2	30.3	0.0	0. 0	1008.0	100.0	1.7	0.0
1/2/2 024 4:00	25.0	94.8	30.0	0.0	0. 0	1008.2	100.0	2.1	0.0
1/2/2 024 5:00	25.1	93.6	30.1	0.0	0. 0	1008.7	100.0	1.9	0.0
1/2/2 024 6:00	25.5	92.0	30.5	0.1	0. 1	1008.9	79.0	2.0	14.4
1/2/2 024 7:00	26.6	88.0	32.1	0.0	0. 0	1009.5	100.0	1.6	189.9
1/2/2 024 8:00	29.1	78.6	35.5	0.0	0. 0	1009.0	43.0	0.6	438.2
1/2/2 024 9:00	30.5	74.5	37.4	0.1	0. 1	1009.5	100.0	0.8	644.7
1/2/2 024 10:00	31.4	71.6	39.2	0.0	0. 0	1009.0	100.0	1.1	770.5
1/2/2 024 11:00	32.0	67.6	39.6	0.0	0. 0	1008.2	99.0	2.6	925.2
1/2/2 024 12:00	32.4	64.5	39.1	0.0	0. 0	1007.5	100.0	2.6	843.6
1/2/2 024 13:00	30.5	71.0	36.8	1.0	1. 0	1007.3	99.0	3.6	906.0
1/2/2 024 14:00	30.4	69.7	35.6	0.0	0. 0	1006.4	69.0	4.0	779.1
1/2/2 024 15:00	29.8	75.7	35.2	0.1	0. 1	1006.2	80.0	2.9	608.4
1/2/2 024 16:00	29.9	74.6	35.7	0.1	0. 1	1006.1	84.0	1.6	337.4
1/2/2 024 17:00	29.5	75.0	34.8	0.2	0. 2	1006.8	96.0	2.2	226.1

Tabel 3.6 adalah data-data sample yang diterima dari situs.

3.2.2 Heatmap Correlation

Correlation Heatmap adalah sebuah alat visual yang dapat menampilkan relasi antar variabel-variabel yang ada di dalam dataset. Dalam penelitian ini metode correlation heatmap digunakan untuk mengetahui variabel apa saja yang berelasi dengan solar irradiance. Variabel yang tidak memiliki relasi tidak akan digunakan dalam input proses prediksi. Dibawah ini adalah contoh Heatmap Correlation yang digunakan dalam dataset yang diambil dari situs <https://open-meteo.com/>.



Gambar 3.6 Heatmap Correlation Matrix Data Cuaca

Dapat dilihat dari gambar, di kolom dan baris 'global_tilted_irradiance' nilai korelasi tiap-tiap variabel. Dalam penelitian ini nilai korelasi yang bernilai 0 di baris dan kolom 'global_tilted_irradiance' akan dihapus dan tidak akan dimasukkan dalam input model prediksi. Menurut gambar diatas, variabel 'precipitation', 'rain', 'MonthOfYear', 'DayOfYear', 'WeekOfYear', dan 'TimeOfDay(h)' akan dihapus dari dataset. Maka fitur-fitur yang digunakan dalam dataset adalah 'temperature_2m', 'relative humidity', 'apparent temperature', 'surface pressure', 'cloud cover', 'wind speed', dan 'global tilted irradiance'. Dibawah ini adalah perbedaan dataset yang sudah dan belum disaring:

Tabel 3.7 Dataset Cuaca setelah Disaring Heatmap Correlation Matrix

temper_2m	relative_humidity_2m	apparent_temperature	surface_pressure	cloud_cover	wind_speed_10m	global_tilted_irradiance
25.3	95.3	30.1	1008.7	100.0	2.9	0.0
24.9	96.5	30.8	1008.6	100.0	0.7	0.0
25.0	95.3	30.7	1008.2	100.0	1.0	0.0
24.9	96.2	30.3	1008.0	100.0	1.7	0.0
25.0	94.8	30.0	1008.2	100.0	2.1	0.0
25.1	93.6	30.1	1008.7	100.0	1.9	0.0
25.5	92.0	30.5	1008.9	79.0	2.0	14.4
26.6	88.0	32.1	1009.5	100.0	1.6	189.9
29.1	78.6	35.5	1009.0	43.0	0.6	438.2
30.5	74.5	37.4	1009.5	100.0	0.8	644.7
31.4	71.6	39.2	1009.0	100.0	1.1	770.5
32.0	67.6	39.6	1008.2	99.0	2.6	925.2
32.4	64.5	39.1	1007.5	100.0	2.6	843.6
30.5	71.0	36.8	1007.3	99.0	3.6	906.0
30.4	69.7	35.6	1006.4	69.0	4.0	779.1
29.8	75.7	35.2	1006.2	80.0	2.9	608.4
29.9	74.6	35.7	1006.1	84.0	1.6	337.4
29.5	75.0	34.8	1006.8	96.0	2.2	226.1

Dari Tabel 3.6 dataset inilah yang akan menjadi input untuk model LSTM. Bisa dilihat perbedaan variabel-variabel yang tidak digunakan dari gambar 3.8 dan gambar 3.10

3.2.3 Data Preparation

Data yang kita gunakan dibagi menjadi dua yaitu dataset cuaca sejarah (data real yang sudah terekam) dan data forecast (data ramalan cuaca). Terdapat 2 tahap dalam pembuatan model LSTM, yang pertama adalah tahap konfigurasi, yaitu melatih model LSTM menggunakan konfigurasi standard. Cara melatihnya adalah menggunakan data sejarah saja (data yang terekam) yang dimulai dari tanggal 01 Januari 2024 sampai tanggal terakhir data terekam. Dalam tahap pertama data sejarah dibagi menjadi data 48 jam terakhir dari data sejarah sebagai testing dan sisanya digunakan sebagai training hasilnya adalah solar irradiance dan dikonversi ke daya yang masuk ke inverter. Setelah itu akan dievaluasi dengan data solar irradiance dan daya yang sudah terekam. Dalam Tahap kedua digunakan semua data sejarah yang terekam untuk training dan semua data forecast 2 hari kedepan dari tanggal data terakhir terekam untuk testing. Data-data ini dalam tahap kedua, digunakan sebagai input model LSTM dengan konfigurasi model tahap pertama. Hasil dari Tahap kedua adalah daya dan solar irradiance yang belum terekam. Dikarenakan belum terekam maka harus ditunggu sampai data real terekam. Berikut adalah rentang data yang digunakan untuk tahap pertama dan kedua:

Tabel 3.8 Rentang Tanggal Data Sejarah dan Data Ramalan

Current example: 25 December 2025	Date	Training	Testing
Tahap ke-1		DataSejarah [01-01-2024 sampai 22-12-2025]	DataSejarah [23-12-2025 sampai 24-12-2025]
Tahap ke-2		DataSejarah [01-01-2024 sampai 24-12-2025]	DataForecast [25-12-2025 sampai 26-12-2025]

Pembagian data input variabel LSTM dapat dipresentasikan sebagai berikut:

$$x = \begin{bmatrix} \text{temperature_2m} \\ \text{relative humidity} \\ \text{apparent temperature'} \\ \text{surface pressure} \\ \text{cloud cover} \\ \text{wind speed} \end{bmatrix}, y = [\text{global tilted irradiance}] \quad (3.13)$$

Dimana nilai x adalah jumlah fitur yang ada dalam input model LSTM dan y adalah jenis output dalam model LSTM.

3.2.4 Normalisasi Data

Normalisasi data adalah proses yang menstandarkan ukuran data yang bertujuan untuk mengurangi bias pada RNN LSTM. Metode yang digunakan pada penelitian ini adalah metode min-max. Metode min-max mengubah dimensi fitur yang ada pada dataset menjadi rentang 0 sampai 1 (Alguhi & Al-Shaalan, 2025). Dilakukan normalisasi pada seuruh dataset yang sudah disaring dengan Heatmap Correlation Matrix. Metode normalisasi data yang digunakna adalah metode min-max scaling:

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (3.14)$$

3.2.5 Long-Short Term Memory

Long-Short Term Memory (LSTM) dalam penelitian ini digunakan untuk meramalkan solar irradiance. Sebelum dataset digunakan dalam model LSTM, data perlu dibagi dan juga dinormalisasikan. Dalam pemrograman Python model LSTM digunakan dari library keras. Dari library keras dimiliki modul bawaan LSTM yang dapat dikonfigurasi untuk model training. Dibawah ini adalah parameter-parameter digunakan untuk konfigurasi model LSTM keras.

Tabel 3.9 Parameter Model LSTM dari *Library Keras*

Model	Sequential
Activation Function	ReLu (Rectified Linear unit)
Look back	24
features	13
optimizer	Adam (Adaptive Moment Estimation)
Loss type	Mean squared error
Learning rate	0.001
Batch size	32

Nilai-nilai parameter diatas adalah nilai standar yang digunakan dalam pengkonfigurasian model LSTM(Alguhi & Al-Shaalan, 2025).

Setelah mendapatkan hasil prediksi solar irradiance, nilai-nilai variabel tersebut dimasukkan sebagai input model PV sekaligus dengan ramalan suhu dari data ramalan cuaca. Rumus model matematika solar panel per modul adalah sebagai berikut:

$$P_{DC} = Watt\ peak_{module} [1 + (T_{cell} - T_{STC})(K_P)] \frac{G_{Solar\ Irradiance}}{1000} \quad (3.15)$$

Dimana P_{DC} adalah nilai daya DC per modul solar panel sesudah MPPT. $T_{coef\ Pmax}$ adalah suhu koefisien dari daya maksimum modul panel PV yang memiliki nilai $-0.34\%/^{\circ}C$ dari rumus diatas memiliki arti setiap derajat celsius daya maksimum berkurang sebesar 0.34% . maka semakin besar suhu operasi solar cell (T_{cell}) semakin besar pengurangan daya yang dihasilkan oleh modul. Untuk menghitung suhu operasi solar cell dirumuskan berikut:

$$T_{cell} = T_{ambient\ temp} + (NOCT - T_{NOCT}) \frac{G_{Solar\ Irradiance}}{800} \quad (3.16)$$

Jadi untuk input model PV diperlukan input $T_{ambient\ temp}$ dari data ramalan cuaca dan $G_{Solar\ Irradiance}$ dari hasil prediksi LSTM. Variabel seperti NOCT (Nominal Operating Cell Temperature), $T_{NOCT} = 20^{\circ}C$, $Watt\ peak_{module}$ dan $T_{coef\ Pmax}$ dan $T_{STC} = 25^{\circ}C$ dapat dilihat di spesifikasi modul solar panel pada Tabel 3.9 dan datasheet di Lampiran.

Tabel 3.10 Spesifikasi Modul Panel PV

ELECTRICAL DATA	STC	NOCT
PEAK POWER (WP)	550 Wp	417 Wp
MAXIMUM POWER VOLTAGE V_{MPP} (V)	31.60 V	29.30 V
MAXIMUM POWER CURRENT I_{MPP} (A)	17.40 A	14.19 A
OPEN-CIRCUIT VOLTAGE V_{oc} (V)	37.90 V	35.70 V
SHORT-CIRCUIT CURRENT I_{sc} (A)	18.52 A	14.92 A
TEMPERATURE RATINGS		
NOCT (NORMAL OPERATING CELL TEMPERATURE)	43 °C(±2 °C)	
TEMPERATURE COEFFICEINT OF P _{MAX}	-0.34 %/°C	
TEMPERATURE COEFFICEINT OF V_{oc}	-0.25 %/°C	
TEMPERATURE COEFFICEINT OF I_{sc}	-0.04 %/°C	
MAXIMUM RATINGS		
OPERATIONAL TEMPERATURE	43 °C(±2 °C)	
MAXIMUM SYSTEM VOLTAGE	1500 V DC	
MAXIMUM SERIES FUSE RATING	30 A	

Setelah konversi nilai input $T_{ambient temp}$ dan $G_{Solar Irradiance}$ menjadi Daya DC (Watt). Maka data daya DC di sizing sesuai konfigurasi pemasangan panel PV di lab REIDI. Pemasangan solar panel di PV 26 seri dan 2 paralel, 26 seri menghadap utara dan 26 seri menghadap selatan. Konfigurasi 26 seri dan 2 paralel digunakan 1 inverter untuk konversi tenag listrik DC ke AC. Dikarenakan 26 solar panel menghadap ke utara dan selatan maka diasumsikan Panel-panel dibagian utara dan selatan menghasilkan daya yang sama. Maka sizing untuk PLTS REIDI di 1 inverter dirumuskan:

$$P_{DC Inverter} = (n_{series} \cdot n_{parallel} \cdot P_{DC})/1000 \text{ kW} \quad (3.17)$$

$P_{DC Inverter}$ adalah daya total yang dihasilkan oleh PV array yang masuk ke dalam 1 buah inverter dengan $n_{series} = 26$ dan $n_{parallel} = 2$ variabel memiliki nilai tersebut karena akan digunakan untuk mensimulasikan 1 buah baris dalam satu kluster Agri di laboratorium PLTS REIDI. Hasil daya dari simulasi digunakan untuk evaluasi hasil daya prediksi dengan daya asli yang belum dan sudah terekam.

$$P_{AC EMS} = n_{row} P_{DC Inverter} \cdot \eta_{inverter} \text{ (kW)} \quad (3.18)$$

$P_{AC EMS}$ adalah daya prediksi sizing yang digunakan untuk perencanaan MILP. Pada peneleitian ini $n_{row} = 50$. Untuk evaluasi akurasi model digunakan PLTS REIDI sebelum di scale-up (dikali 50) yang memiliki kapasitas 28.6 kWp dan untuk percobaan program MILP digunakan PLTS yang sudah di scale-up, memiliki kapastias 1.43MWp

3.2.6 Mixed Integer Linear Programming

Masalah MILP pada visual studio code diselesaikan menggunakan modul bawaan library scipy. Berikut adalah input dari MILP di VSCode:

Ini adalah baris kode Python yang digunakan untuk fungsi MILP:

```
res = milp(c=c, bounds=bounds, integrality=integrality, constraints=constraints)
```

1. c adalah fungsi objektif yang digunakan untuk memformulasikan unit commitment
2. $bounds$ adalah Batasan solusi x yaitu $l_b \leq x \leq u_b$ Dimana $u_b = inf$ dan $l_b = -inf$
3. Integrality adalah sebuah matriks berbaris satu dengan panjang sebesar matriks fungsi objektif, dimana variable-variabel dalam matriks integrality bernilai 0 atau 1. Dimana 0 merepresentasikan continuous dan 1 merepresentasikan intcon.
4. Constrain berisi matrix A, B, A_{eq}, b_{eq}
5. A adalah matrix yang berisi koefisien linear dari Batasan inequality constrain
6. B adalah matrix yang memiliki satu kolom berisi konstanta dari Batasan inequality constrain
7. A_{eq} adalah matrix yang berisi koefisien linear dari persamaan equality constrain
8. b_{eq} adalah matrix yang memiliki satu kolom berisi konstanta dari persamaan equality constrain

Untuk mendapatkan hasil proses MILP digunakan perintah kode dibawah ini:

1. Res.x adalah perintah untuk memulai proses MILP dan mendapatkan hasil variable-variabel yang meminimalkan fungsi objektif.
2. Res.fun adalah perintah untuk hasil optimal dari fungsi objektif yaitu hasil minimum atau maksimum fungsi objektif setelah fungsi objektif diisi variable-variabel solusi
3. Res.status adalah perintah untuk memberikan nilai indikasi status keberhasilan atau kegagalan pencarian solusi optimal. Dalam VSCode kode status 0 berarti solusi ditemukan. Kode status 1 berarti optimasi terlalu lama dan mencapai batasannya sebelum mencapai solusi. Kode status 2 berarti masalah fungsi objektif tidak ada solusi. Kode status 3 berarti variable fungsi objektif bernilai infinity.

3.2.7 Pemodelan Matrix Mixed Integer Linear Programming

Dalam penelitian ini diberikan sebuah dataset pemabngkit-pembangkit yang digunkan. Dan pembangkit-pembangkit tersebut memiliki kondisi-kondisi saat beroperasi. Tujuan utama dari pemodelan ini adalah fungsi objektif dengan memenuhi kondisi-kondisi yang ada pada dataset pembangkit-pembangkit pada tabel 3.1, tabel 3.2 dan tabel 3.3.

3.2.7.1 Model Fungsi Objektif

Fungsi objektif adala unit commmitment yaitu, fungsi biaya yang disusun ke dalam sebuah matriks (f). f berisi koefisien linear dari biaya pembangkit per MW dan biaya start up cost tiap kali pembangkit menyala. Fungsi objektif matrix dapat dirumuskan:

$$F(P_i^t, U_i^t, SUV_i^t) = \text{Min} \sum_{t=1}^{nl=24} \sum_{i=1}^{nu=4} [F_{Pi}(P_i^t) + F_{Ui}(U_i^t) + F_{SUCi}(SUV_i^t)] \quad (3.19)$$

Dikarenakan U_i^t merepresentasikan status on/off sebuah pembangkit maka $F_{Ui} = 0$. Jadi matrix fungsi objektif untuk unit commitment adalah:

$$F \text{ Matrix} = [[F_{P1}(P_1^1) \quad \dots \quad F_{P4}(P_4^{24})] \quad \dots \quad [F_{SUC1}(SUV_1^1) \quad \dots \quad (F_{SUC4} SUV_4^{24})]]$$

3.2.7.2 Pemodelan Equality Constrain

Equality constrain dalam penelitian ini digunakan untuk memformulasikan kondisi Dimana semua demand beban terpenuhi dan kondisi BESS harus siap untuk keesokan harinya. Untuk memudahkan dan agar memuat matrix maka hanya digunakan variable i (pembangkit) dan bentuk matrix dibentuk menurut persamaan 2.16:

$$i = [1,2,3,4] \rightarrow PLTS, PLTD, PLTU, BESS$$

1. Power Balance

Keseimbangan daya adalah salah satu persamaan yang digunakan agar daya yang dibangkitkan tiap pembangkit pada tiap jam-nya memenuhi kebutuhan daya tiap jam-nya maka persamaannya dalah sebagai berikut sebagai berikut:

$$P_{load}^t = \sum_{i=1}^{nu=4} P_i^t \quad (3.20)$$

$$P_1^1 + P_2^1 + P_3^1 + P_4^1 = P_{load}^1 \quad \text{JAM KE} - 1 \quad (3.21)$$

$$P_1^2 + P_2^2 + P_3^2 + P_4^2 = P_{load}^2 \quad \text{JAM KE} - 2 \quad (3.22)$$

$$P_1^3 + P_2^3 + P_3^3 + P_4^3 = P_{load}^3 \quad \text{JAM KE} - 3 \quad (3.23)$$

Dari persamaan (3.20), (3.21), (3.22), (3.23) membuat pola matrix sebagai berikut:

$$A_{eq} = \begin{bmatrix} P_1^1 & P_2^1 & P_3^1 & P_4^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_1^2 & P_2^2 & P_3^2 & P_4^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_1^3 & P_2^3 & P_3^3 & P_4^3 \end{bmatrix}$$

$$B_{eq} = \begin{bmatrix} P_{load}^1 \\ P_{load}^2 \\ P_{load}^3 \end{bmatrix}$$

2. BESS Ready for next day

BESS harus siap untuk tiap harinya maka pada saat hari berakhir BESS harus terisi sesuai sama dengan SOC baterai jam pertama dimulai hari. Maka persamaan total dispatch BESS tiap jam nya adalah sebagai berikut:

$$\sum_{t=1}^{t=24} P_{BESS}^t = 0 \quad (3.24)$$

Dari persamaan (3.24) pemodelan matrixnya hanya menggunakan $i = 4$ yaitu BESS dan dalam matrix tersebut adalah P_{BESS}^t yang berarti dispatch BESS yang juga berarti

perbedaan SOC. Yang diinginkan adalah SOC saat jam pertama $t=1$ sama dengan jam terakhir $t=24$ maka matrix akan berbentuk:

$$\mathbf{A}_{eq} = [0 \ 0 \ 0 \ P_{BESS}^1 \ 0 \ 0 \ 0 \ P_{BESS}^2 \ \dots \ 0 \ 0 \ 0 \ P_{BESS}^{24}]$$

$$\mathbf{B}_{eq} = [0]$$

3.2.7.3 Pertidaksamaan Inequality Constrain

Inequality constrain merepresentasikan Batasan-batasan yang harus dipenuhi saat pengoperasian pembangkit-pembangkit yang ada dalam sistem. Inequality Constrain terdiri dari, Daya minimal dan daya maksimum pembangkit, Ramp up dan Ramp Down Pembangkit, Up dan Down Time Pembangkit, Kondisi status on/off pembangkit, SOC Baterai dan Start-up Cost pembangkit. Bentuk pertidaksamaan inequality constrain dibuat menurut persamaan (2.17).

1. Batasan daya minimum dan maksimum

Tiap pembangkit memiliki batasan saat kondisi ON, Dimana pembangkit harus membangkitkan minimal nilai daya and juga tidak boleh melewati suatu nilai daya maka itu diberikan persamaan sebagai berikut:

$$U_i^t P_{i,min} \leq P_i^t \leq U_i^t P_{i,max} \quad (3.25)$$

Dari pertidaksamaan (3.25) dibentuk menjadi 2 buah persamaan agar terbentuk seperti persamaan (2.17):

$$P_i^t - U_i^t P_{i,max} \leq 0 \quad (3.26)$$

$$-P_i^t + U_i^t P_{i,min} \leq 0 \quad (3.27)$$

Dikarenakan P_1 adalah daya PLTS maka $P_{1,max}^1$ adalah daya maksimum PLTS pada jam ke-1, pada jam ke-2 dan setiap jam berikutnya daya maksimum PLTS berubah-ubah menurut potensi produksi PLTS di gambar 3.5. Dimana pembangkit lain tiap jam nya memiliki daya maksimum yang sama tiap jam nya. Jadi khusus untuk pembangkit PLTS memiliki persamaan sebagai berikut:

$$P_{PLTS=1}^t - U_{PLTS=1}^t P_{PLTS=1,max}^t \leq 0 \quad (3.28)$$

$$-P_{PLTS=1}^t + U_{PLTS=1}^t P_{PLTS=1,min}^t \leq 0 \quad (3.29)$$

Dari persamaan (3.26), (3.27), (3.28), dan (3.29) akan membentuk pola matrix sebagai berikut:

$$A = \begin{bmatrix} P_1^1 & 0 & 0 & -U_1^1 P_{1,max}^1 & 0 & 0 \\ -P_1^1 & 0 & 0 & -U_1^1 P_{1,max}^1 & 0 & 0 \\ 0 & P_2^1 & 0 & 0 & -U_2^1 P_{2,max} & 0 \\ 0 & -P_2^1 & 0 & 0 & U_2^1 P_{2,min} & 0 \\ 0 & 0 & P_3^1 & 0 & 0 & -U_3^1 P_{3,max} \\ 0 & 0 & -P_3^1 & 0 & 0 & U_3^1 P_{3,min} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

2. Batasan integer

Agar nilai status on/off pembangkit hanya berada di nilai nol atau satu maka dibuat persamaan sebagai berikut:

$$U_i^t \leq 1 \quad (3.30)$$

Dari persamaan (3.30) maka dibentuk pola matrix:

$$A = \begin{bmatrix} U_1^1 & 0 & 0 & 0 & 0 \\ 0 & U_2^1 & 0 & 0 & 0 \\ 0 & 0 & U_3^1 & 0 & 0 \\ 0 & 0 & 0 & U_4^1 & 0 \\ 0 & 0 & 0 & 0 & U_1^2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

3. Batasan ramp up dan ramp down

Batasan ramp up dan ramp down memberikan batasan untuk tiap daya yang diproduksi pembangkit tiap jamnya tidak memiliki perbedaan tertentu. Dimulai dari pembangkit jam ke-2 dan jam ke-1. Maka nilai waktu $t = [2, 3, \dots, 24]$. Dan memiliki persamaan untuk batasan ramp up:

$$P_i^{(t-1)} - P_i^t \leq RD_i \quad (3.31)$$

Dan persamaan ramp down adalah:

$$-P_i^{(t-1)} + P_i^t \leq RU_i \quad (3.32)$$

Dari persamaan (3.31) dibentuk pola matrix Batasan ramp down sebagai berikut:

$$A = \begin{bmatrix} P_1^1 & 0 & 0 & 0 & -P_1^2 & 0 & 0 & 0 & 0 \\ 0 & P_2^1 & 0 & 0 & 0 & -P_2^2 & 0 & 0 & 0 \\ 0 & 0 & P_3^1 & 0 & 0 & 0 & -P_3^2 & 0 & 0 \\ 0 & 0 & 0 & P_4^1 & 0 & 0 & 0 & -P_4^2 & 0 \\ 0 & 0 & 0 & 0 & P_1^2 & 0 & 0 & 0 & -P_1^3 \end{bmatrix},$$

$$B = \begin{bmatrix} RD_1 \\ RD_2 \\ RD_3 \\ RD_4 \\ RD_1 \end{bmatrix}$$

dan dari persamaan 3.32 dibentuk pola matrix Batasan ramp up sebagai berikut:

$$A = \begin{bmatrix} -P_1^1 & 0 & 0 & 0 & P_1^2 & 0 & 0 & 0 & 0 \\ 0 & -P_2^1 & 0 & 0 & 0 & P_2^2 & 0 & 0 & 0 \\ 0 & 0 & -P_3^1 & 0 & 0 & 0 & P_3^2 & 0 & 0 \\ 0 & 0 & 0 & -P_4^1 & 0 & 0 & 0 & P_4^2 & 0 \\ 0 & 0 & 0 & 0 & -P_1^2 & 0 & 0 & 0 & P_1^3 \end{bmatrix},$$

$$B = \begin{bmatrix} RU_1 \\ RU_2 \\ RU_3 \\ RU_4 \\ RU_1 \end{bmatrix}$$

Persamaan 3.31 berlaku sebagai Batasan untuk daya yang dibangkitkan pada jam sebelumnya t-1 lebih kecil dari daya yang dibangkitkan pada jam t. Persamaan 3.32 berlaku sebagai Batasan untuk daya yang dibangkitkan pada jam t-1 lebih besar dari daya yang dibangkitkan pada jam t.

4. Batasan *downtime*

Batasan *downtime* adalah batasan minimal waktu OFF pembangkit ketika berubah dari status ON berubah menjadi status OFF. Jika pembangkit pada jam t-1 ber kondisi ON dan pada jam t ber kondisi OFF, maka pembangkit tersebut harus dalam kondisi OFF minimal selama t+DT-1. Dibawah ini adalah parameter i dan t yang digunakan agar mudah untuk formulasi pola matrix:

$$i = [PLTS, PLTD, PLTU, BESS]$$

$$i = [1, 2, 3, 4]$$

$$t = [2, 3, \dots, (nl - 2)]$$

Dibawah ini adalah contoh ketika ($i = 1, t = 2, DT_1 = 4$):

$$\sum_{j=t}^{t+DT_i-1} u(t) \leq DT_i[1 + (u(t) - u(t-1))] \quad (3.33)$$

$$u(2) + u(3) + u(4) + u(5) \leq (1 + [u(2) - u(1)])DT_1 \quad (3.34)$$

Dari persamaan 3.34 diteruskan untuk membentuk persamaan 2.17:

$$U_1^2 + U_1^3 + U_1^4 + U_1^5 \leq (1 + [U_1^2 - U_1^1])4 \quad (3.35)$$

Dari persamaan 3.35 diteruskan lagi menjadi:

$$4U_1^1 - 3U_1^2 + U_1^3 + U_1^4 + U_1^5 \leq 4 \quad (3.36)$$

Dari persamaan 3.36 ketika ($i = 1, t = 2, DT_1 = 4$) membentuk pola matrix sebagai berikut:

$$A = \begin{bmatrix} 4U_1^1 & -3U_1^2 & U_1^3 & U_1^4 & U_1^5 & 0 & 0 & 0 & 0 \\ 0 & 4U_1^2 & -3U_1^3 & U_1^4 & U_1^5 & U_1^6 & 0 & 0 & 0 \\ 0 & 0 & 4U_1^3 & -3U_1^4 & U_1^5 & U_1^6 & U_1^7 & 0 & 0 \\ 0 & 0 & 0 & 4U_1^4 & -3U_1^5 & U_1^6 & U_1^7 & U_1^8 & 0 \\ 0 & 0 & 0 & 0 & 4U_1^5 & -3U_1^6 & U_1^7 & U_1^8 & U_1^9 \end{bmatrix},$$

$$B = \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}$$

Menurut variabel $t = [2, 3, \dots, (nl-2)]$, waktu hanya sampai $nl-2$ untuk mengatasi sliding window berhenti di akhir kolom matrix. Dalam formulasi sebenarnya yang digunakan dalam penelitian ini kolom matriks sampai jam ke-24.

5. Batasan uptime

Batasan uptime adalah batasan minimal waktu ON pembangkit ketika berubah dari status OFF berubah menjadi status ON. Jika pembangkit pada jam $t-1$ berkondisi OFF dan pada jam t berkondisi ON maka pembangkit tersebut harus tetap dalam kondisi ON minimal selama $t+UT-1$. Dibawah ini adalah parameter i dan t yang digunakan agar mudah untuk formulasi pola matrix:

$$i = [PLTS, PLTD, PLTU, BESS]$$

$$i = [1, 2, 3, 4]$$

$$t = [2, 3, \dots, (nl-2)]$$

Dibawah ini adalah contoh ketika ($i = 1, t = 2, UT_1 = 4$):

$$\sum_{j=t}^{t+UT_i-1} u(t) \geq UT_i(u(t) - u(t-1)) \quad (3.37)$$

$$u(2) + u(3) + u(4) + u(5) \geq (u(2) - u(1))UT_1 \quad (3.38)$$

Dari persamaan 3.38 diteruskan untuk membentuk persamaan 2.17:

$$U_1^2 + U_1^3 + U_1^4 + U_1^5 \geq (U_1^2 - U_1^1)4 \quad (3.39)$$

Dari persamaan 3.39 diteruskan lagi menjadi:

$$-4U_1^1 + 3U_1^2 - U_1^3 - U_1^4 - U_1^5 \leq 0 \quad (3.40)$$

Dari persamaan 3.40 ketika ($i = 1, t = 2, DT_1 = 4$) membentuk pola matrix sebagai berikut:

$$\mathbf{A} = \begin{bmatrix} -4U_1^1 & 3U_1^2 & -U_1^3 & -U_1^4 & -U_1^5 & 0 & 0 & 0 & 0 \\ 0 & -4U_1^2 & 3U_1^3 & -U_1^4 & -U_1^5 & -U_1^6 & 0 & 0 & 0 \\ 0 & 0 & -4U_1^3 & 3U_1^4 & -U_1^5 & -U_1^6 & -U_1^7 & 0 & 0 \\ 0 & 0 & 0 & -4U_1^4 & 3U_1^5 & -U_1^6 & -U_1^7 & -U_1^8 & 0 \\ 0 & 0 & 0 & 0 & -4U_1^5 & 3U_1^6 & -U_1^7 & -U_1^8 & -U_1^9 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

6. Batasan variable startup

Variable startup harus lebih besar dari nol, dikarenakan nilai tidak boleh bernilai negative. Variable start-up berupa nilai biaya ketika kondisi pada jam t-1 OFF dan pada jam t berkondisi ON. Ini merepresentasikan batas nilai variable aktif hanya ketika berubah dari kondisi OFF ke ON. Dibawah ini digunakan 2 buah pertidaksamaan untuk memenuhi batasan pembangkit startup variable lebih dari nol dan hanya berlaku ketika melakukan kondisi OFF ke ON. Dibawah ini adalah persamaan berikut:

$$SUV_i^t \geq 0 \quad (3.41)$$

Dari persamaan 3.41 dibentuk menurut persamaan 2.17:

$$-SUV_i^t \leq 0 \quad (3.42)$$

Dari persamaan 3.42 digunakan untuk membuat pola matrix sebagai berikut:

$$A = \begin{bmatrix} -SUV_1^1 & 0 & 0 & 0 & 0 \\ 0 & -SUV_2^1 & 0 & 0 & 0 \\ 0 & 0 & -SUV_3^1 & 0 & 0 \\ 0 & 0 & 0 & -SUV_4^1 & 0 \\ 0 & 0 & 0 & 0 & -SUV_1^2 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Untuk membuat pertidaksamaan batasan untuk berlaku hanya ketika pembangkit melakukan kondisi OFF di jam $t-1$ dan kondisi ON di jam t maka digunakan persamaan sebagai berikut:

Agar mudah memformulasi matrix maka digunakan parameter i dan t dibawah ini:

$$i = [PLTS, PLTD, PLTU, BESS]$$

$$i = [1, 2, 3, 4]$$

$$t = [2, 3, \dots, (nl - 2)]$$

Persamaan untuk batasan start up bergantung dengan kondisi adalah sebagai berikut :

$$SUV_i^t \geq SUC_i[(-u(t-1) + u(t))] \quad (3.43)$$

Dari persamaan 3.43 digunakan untuk membentuk pola matrix ketika ($i = 1, t = 2$):

$$SUV_i^t \geq SUC_i[(-u(1) + u(2))] \quad (3.44)$$

$$SUV_1^2 \geq SUC_1[-U_1^1 + U_1^2] \quad (3.45)$$

Dari persamaan 3.45 diteruskan lagi untuk membentuk persamaan 2.17:

$$-SUC_1U_1^1 + SUC_1U_1^2 - SUV_1^2 \leq 0 \quad (3.46)$$

Dari persamaan 3.46 dibentuk pola matrix nya:

$$A = \begin{bmatrix} -SUC_1U_1^1 & 0 & 0 & 0 & SUC_1U_1^2 & 0 & 0 & 0 & 0 \\ 0 & -SUC_1U_2^1 & 0 & 0 & 0 & SUC_1U_2^2 & 0 & 0 & 0 \\ 0 & 0 & -SUC_1U_3^1 & 0 & 0 & 0 & SUC_1U_3^2 & 0 & 0 \\ 0 & 0 & 0 & -SUC_1U_4^1 & 0 & 0 & 0 & SUC_1U_4^2 & 0 \\ 0 & 0 & 0 & 0 & -SUC_1U_1^2 & 0 & 0 & 0 & SUC_1U_1^3 \end{bmatrix}$$

$$\begin{bmatrix} -SUV_1^2 & 0 & 0 & 0 & 0 \\ 0 & -SUV_2^2 & 0 & 0 & 0 \\ \dots & 0 & 0 & -SUV_3^2 & 0 \\ 0 & 0 & 0 & -SUV_4^2 & 0 \\ 0 & 0 & 0 & 0 & -SUV_1^3 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

7. Batasan SOC BESS

Selain batasan daya yang masuk dan keluar dari BESS juga ada batasan agar energi dalam BESS berada di rentang SOC minimum dan maksimum untuk mensimulasikan baterai tidak melebihi kondisi SOC BESS 100% dan tidak mengurangi kondisi SOC 0%. Dibawah ini adalah persamaan yang digunakan untuk mensimulasikan kondisi tersebut:

$$E_{min} \leq E_0 - \sum_{t=1}^{j=24} P_{BESS}^t \times (t - (t - 1)) \leq E_{max} \quad (3.47)$$

Agar membentuk persamaan 2.17 dibagi menjadi dua untuk batasan BESS agar tidak dibawah minimum dan diatas maksimum saat melakukan charging dan discharging di tiap waktu. Persamaan dan matriks agar tidak dibawah SOC minimum BESS adalah sebagai berikut:

$$\sum_{t=1}^{j=24} P_{BESS}^t \times \Delta t \leq -E_{min} + E_0 \quad (3.48)$$

Dari persamaan 3.48 digunakan untuk membentuk pola matrix:

$$A = \begin{bmatrix} P_{BESS}^1 & 0 & 0 & 0 \\ P_{BESS}^1 & P_{BESS}^2 & 0 & 0 \\ P_{BESS}^1 & P_{BESS}^2 & P_{BESS}^3 & 0 \\ P_{BESS}^1 & P_{BESS}^2 & P_{BESS}^3 & P_{BESS}^4 \end{bmatrix}$$

$$B = \begin{bmatrix} -E_{min} + E_0 \\ -E_{min} + E_0 \\ -E_{min} + E_0 \\ -E_{min} + E_0 \end{bmatrix}$$

Agar kondisi SOC tidak melebihi batas maksimum SOC 100% maka dbentuk persamaan dari persamaan 3.49. dibawah ini adalah persamaan dan matrix agar tidak melebihi SOC maximum BESS:

$$-\sum_{t=1}^j P_{BESS}^t \times \Delta t \leq E_{max} - E_0 \quad (3.49)$$

Dari persamaan 3.49 digunakan untuk membentuk pola matrix:

$$A = - \begin{bmatrix} P_{BESS}^1 & 0 & 0 & 0 \\ P_{BESS}^1 & P_{BESS}^2 & 0 & 0 \\ P_{BESS}^1 & P_{BESS}^2 & P_{BESS}^3 & 0 \\ P_{BESS}^1 & P_{BESS}^2 & P_{BESS}^3 & P_{BESS}^4 \end{bmatrix}$$

$$B = \begin{bmatrix} E_{max} - E_0 \\ E_{max} - E_0 \\ E_{max} - E_0 \\ E_{max} - E_0 \end{bmatrix}$$

3.3 Bahan dan peralatan yang digunakan

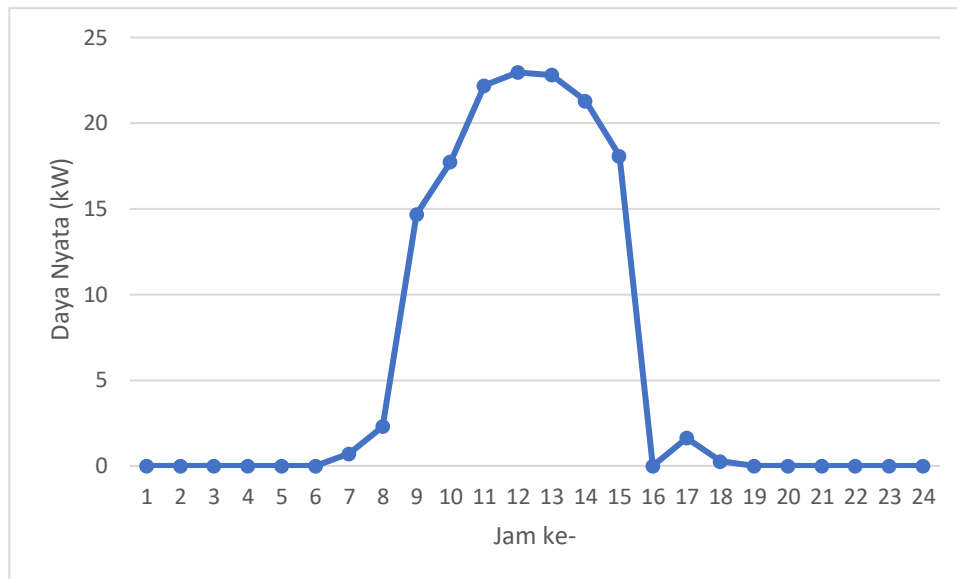
Dibawah ini adalah berbagai peralatan yang digunakan untuk mengumpulkan, menganalisis, dan mengevaluasi data. Berikut adalah penjelasan detail mengenai bahan dan peralatan yang digunakan selama penelitian.

3.3.1 Dataset cuaca sejarah dan ramalan daya

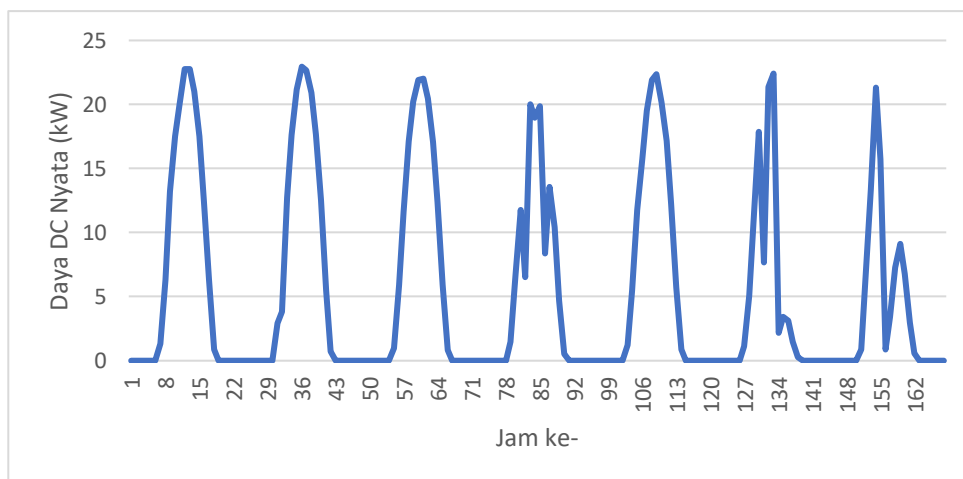
Dataset cuaca yang didapatkan dari situs <https://open-meteo.com/>. Dengan menentukan parameter-parameter seperti latitude dan longitude untuk mendapatkan dataset cuaca yang terkeam (sejarah) dan yang belum terkeam (ramalan) di lokasi PLTS yaitu lab REIDI ITS. Untuk mendapatkan parameter tersebut dapat dilihat dari google maps dari Gambar 3.7. Gambar tersebut menunjukkan lokasi dan koordinat dari google map yang menunjukan lab REIDI. Dengan menggunakan API Request didapatkan data cuaca sejarah dan ramalan, yang kemudian disaring menggunakan metode Heatmap Correlation matrix menghasilkan dataset dengan fitur-fitur yang berkorelasi dengan solar irradiasi. Dapat dilihat hasil data yang digunakan setelah penyaringan ada pada tabel 3.7

3.3.2 Dataset Inverter REIDI

Dataset inverter REIDI digunakan untuk pemodelan simulasi PV di REIDI agar mendapat hasil yang dapat mengevaluasi data daya asli produksi PV di REIDI dengan hasil prediksi daya dari model prediksi daya di penelitian ini. Dalam penelitian ini digunakan 1 Inverter REIDI yang terhubung dengan panel terhubung dengan konfigurasi 26 seri dan 2 paralel. Dibawah ini adalah dataset yang diambil dari inverter Huwaei SUN2000-25KTL-M5 yang digunakan di PLTS lab REIDI. Dari dataset Inverter REIDI yang digunakan adalah kolom Date_time, dan P_dc,. Fitur-fitur ini yang akan digunakan untuk evaluasi hasil prediksi daya pada tahap pertama dan tahap kedua pembuatan model LSTM untuk prediksi daya. Dibawah ini adalah dataset yang digunakan untuk evaluasi hasil peramalan daya pada penelitian ini:



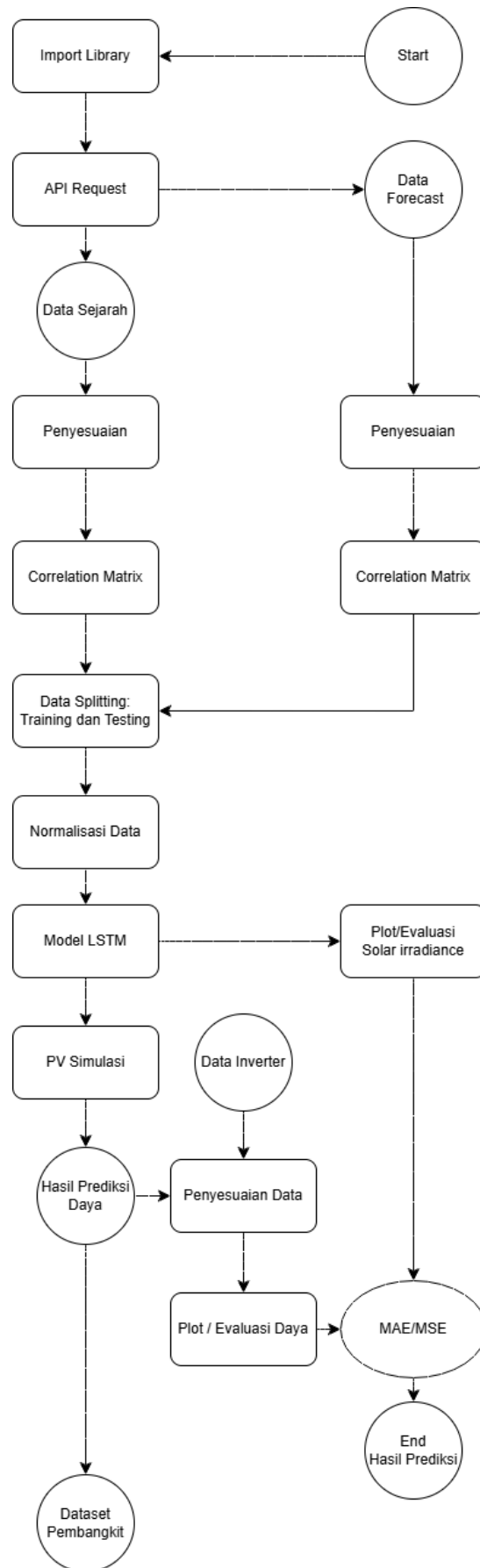
Gambar 3.7 Data Inverter per jam dalam rentang 1 Hari (24 Jam)



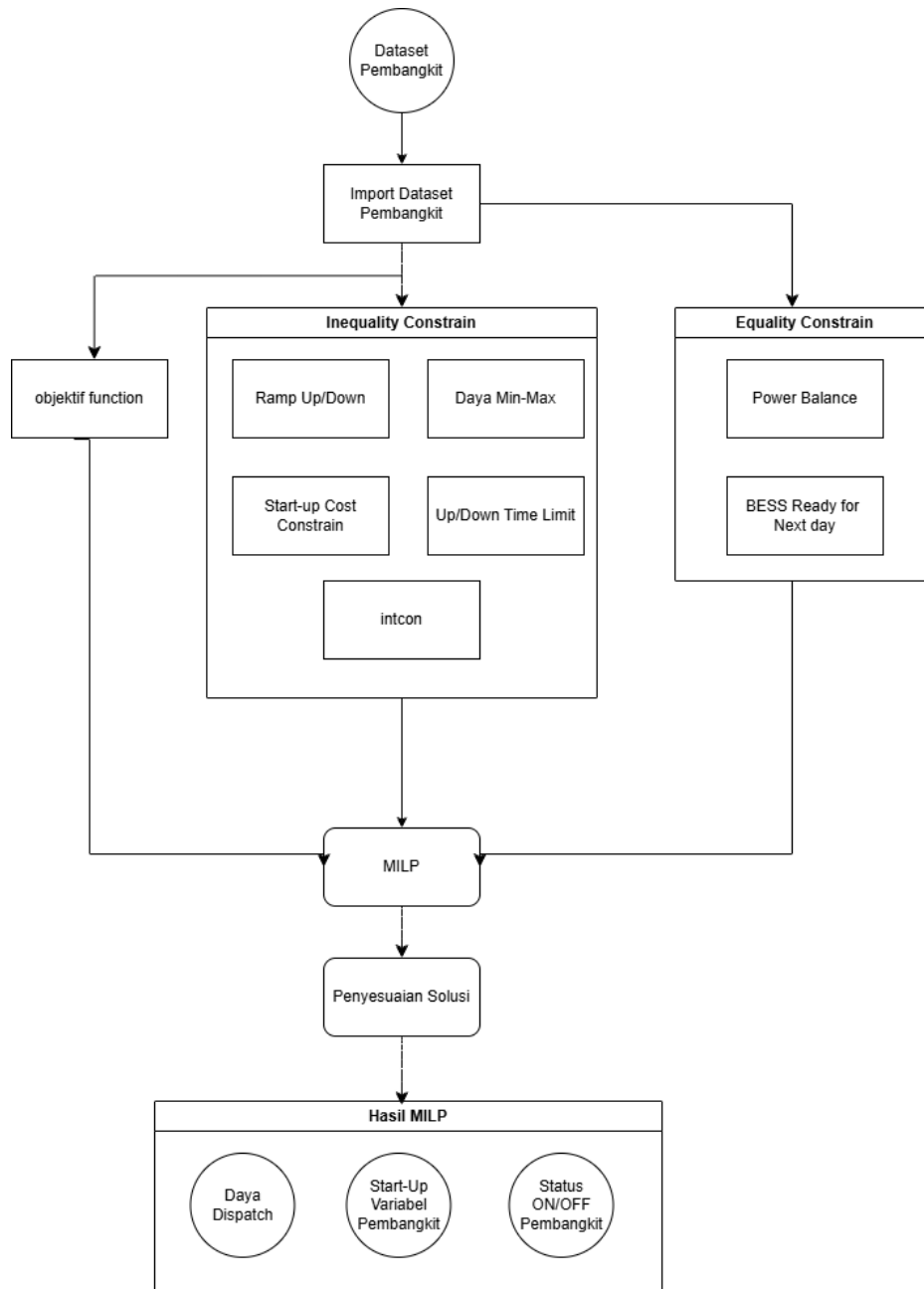
Gambar 3.8 Data Inverter per jam dalam rentang 7 Hari (168 Jam)

3.3.3 Visual Studio Code

Visual Studio Code adalah sebuah program gratis yang dikembangkan oleh Microsoft dengan kemampuan untuk menulis kode dengan bahasa pemrograman seperti C, C++, Python, Java, HTML dan lainnya. Visual Studio Code dapat digunakan untuk berbagai tujuan, seperti analisa data, pengembangan website, pengembangan AI dan lainnya. Visual Studio Code pada penelitian ini digunakan untuk menyaring data mentah yang didapatkan dari REIDI, menyesuaikan dataset, membuat model prediksi daya, dan membuat program MILP. Dibawah ini adalah flow-chart penggunaan program Visual Studio Code:



Gambar 3.9 Flow-Chart Programming Model LSTM Visual Studio Code



Gambar 3.10 Flow-Chart Programming Model LSTM Visual Studio Code

Dari gambar 3.9 dan gambar 3.10 dibuat kode dalam penelitian ini. Kode dapat dilihat dalam bab Lampiran

3.3.4 Bahasa Pemrograman Python

Python adalah sebuah bahasa pemrograman yang digunakan dalam penelitian ini. Python adalah salah satu bahasa pemrograman yang paling populer maka terdapat banyak masukan dari pengembang-pengembang lain yang berkontribusi menggunakan bahasa ini. Bahasa pemrograman Python juga memiliki berbagai library yang digunakan untuk penelitian ini:

Tabel 3.11 Versi *Python* dan *Library* yang digunakan

Python 3.13.9	Jupyter_core 5.9.1
Pandas 2.3.3	Openmeteo_requests 1.7.4
Numpy 2.3.4	Openmeteo_sdk 1.23.0
Keras 3.12.0	Scipy 1.16.3
Matplotlib 3.10.7	Scikit-learn 1.7.2
Seaborn 0.13.2	

3.3.5 Datasheet Solar Panel dan Inverter yang Digunakan REIDI

Dalam pembuatan model simulasi PV diperlukan data-data di panel surya yang digunakan REIDI dan juga data inverter untuk mengetahui efisiensi untuk emndapatkan daya AC. Datasheet PV digunakan untuk validasi data-data yang ada di inverter dan juga digunakan sebagai parameter pembuatan simulasi PV. Datasheet inverter yang digunakan adalah datasheet inverter yang terhubung dengan solar panel yang di ada di datsheet solar PV.

Tabel 3.12 Datasheet Solar Panel digunakan di REIDI

MODEL NAME	VERTEX BACKSHEET MONOCRYSTALLINE MODULE	
PRODUCT NO.	TSM-DE19-555	
ELECTRICAL DATA	STC	NOCT
PEAK POWER (WP)	550 Wp	417 Wp
MAXIMUM POWER VOLTAGE V_{MPP} (V)	31.60 V	29.30 V
MAXIMUM POWER CURRENT I_{MPP} (A)	17.40 A	14.19 A
OPEN-CIRCUIT VOLTAGE V_{oc} (V)	37.90 V	35.70 V
SHORT-CIRCUIT CURRENT I_{sc} (A)	18.52 A	14.92 A
TEMPERATURE RATINGS		
NOCT (NORMAL OPERATING CELL TEMPERATURE)	43 °C(±2 °C)	
TEMPERATURE COEFFICEINT OF P _{MAX}	-0.34 %/°C	
TEMPERATURE COEFFICEINT OF V_{oc}	-0.25 %/°C	
TEMPERATURE COEFFICEINT OF I_{sc}	-0.04 %/°C	
MAXIMUM RATINGS		
OPERATIONAL TEMPERATURE	43 °C(±2 °C)	
MAXIMUM SYSTEM VOLTAGE	1500 V DC	
MAXIMUM SERIES FUSE RATING	30 A	
MECHANICAL DATA		
SOLAR CELLS	Mono Crystalline	
NO. OF CELLS	110 cells	
WEIGHT	28.6kg	
ENCAPSULANT MATERIAL	EVA/POE	
GLASS	3.2 mm High Transmission, AR Coated Heat Strengthened Glass	
BACKSHEET	white	
FRAME	35mm Anodized Aluminium Alloy	
CABLES	photovoltaic Technology Cable 4.0 mm ² portrait: 280/280 mm Landscape: 1400/1400 mm	
CONNECTORS	MC4 EVO2 / TS4*	

Tabel 3.13 Datasheet Inverter digunakan di REIDI

MODEL NAME	HUAWEI SUN2000-25-KTL-M5
ELECTRICAL DATA	
MAXIMUM INPUT VOLTAGE	1100 VDC
MAXIMUM INPUT CURRENT	30A
MPPT RANGE	200-1000VDC
OUTPUT NOMINAL VOLTAGE	220/380 VAC
OUTPUT CURRENT	42.0A/380Vac 39.9A/400Vac 38.5A/415Vac
NOMINAL OPERATING FREQUENCY	50/60Hz
OUTPUT RATED POWER	25kW
MAX OUTPUT RATED POWER	27.5 kVA
OPERATION TEMPERATURE RANGE	-25 ~ + 60 °C

3.3.6 Dataset pembangkit, beban dan potensi PLTS untuk menguji program MILP

Dataset pembangkit, beban, dan potensi PLTS yang digunakan untuk pemodelan matrix MILP. Pada penelitian ini digunakan dataset berikut:

Tabel 3.14 Dataset Pembangkit untuk Menguji program MILP

PEMBANGKIT	Pmin (MW)	Pmax (MW)	Gen Cost (Rp/MWh)	RU	RD	UT	DT	STC
PLTS	0	P_{PLTS}^t	0	500	500	1	1	Rp 0
PLTD SELAYAR #01 (DEUTZ)	0.5	0.9	Rp 3,801,435.9	0.35	0.35	4	2	Rp 0
PLTD SELAYAR #02 (DEUTZ)	0.5	0.9	Rp 3,801,435.9	0.35	0.35	4	2	Rp 0
PLTD SELAYAR #03 (DEUTZ)	0.5	0.9	Rp 3,801,435.9	0.35	0.35	4	2	Rp 0
PLTD SELAYAR #04 (DEUTZ)	0.5	0.9	Rp 3,801,435.9	0.35	0.35	4	2	Rp 0
PLTU SELAYAR #01 (DAIHATSU)	0.1	0.22	Rp 4,086,470.5	0.1	0.1	4	2	Rp 3,535,200.0
PLTU SELAYAR #01 (CUMMINS)	0.5	1	Rp 3,615,142.5	0.4	0.3	2	1	Rp 1,944,360.0
PLTU SELAYAR #02 (CUMMINS)	0.5	1	Rp 3,615,142.5	0.4	0.3	2	1	Rp 1,944,360.0
PLTU SELAYAR #03 (CUMMINS)	0.5	1	Rp 3,615,142.5	0.4	0.3	2	1	Rp 1,944,360.0
PLTU SELAYAR #04 (CUMMINS)	0.5	1	Rp 3,615,142.5	0.4	0.3	2	1	Rp 1,944,360.0
PLTU SELAYAR #05 (CUMMINS)	0.5	1	Rp 3,615,142.5	0.4	0.3	2	1	Rp 1,944,360.0
PLTU SELAYAR #06 (CUMMINS)	0.5	1	Rp 3,700,000.0	0.4	0.3	2	1	Rp 3,181,680.0
BESS	-0.2	0.2	0	500	500	1	1	Rp 0

Untuk nilai-nilai batasan yang ada pada pembangkit di tabel 3.14 tidak realistis dan hanya digunakan untuk bukti konsep pemrograman MILP. Secara realita kapasitas PLTU lebih besar dari 1 MW, tidak ada PLTU yang memiliki kapasitas yang sangat kecil. PLTS dan BESS juga tidak memiliki batasan Ramp Up dan Ramp Down dan juga tidak memiliki batasan Uptime dan Downtime. PLTD secara realitanya juga tidak memiliki batasan Ramp Up dan Ramp Down yang sangat kecil.

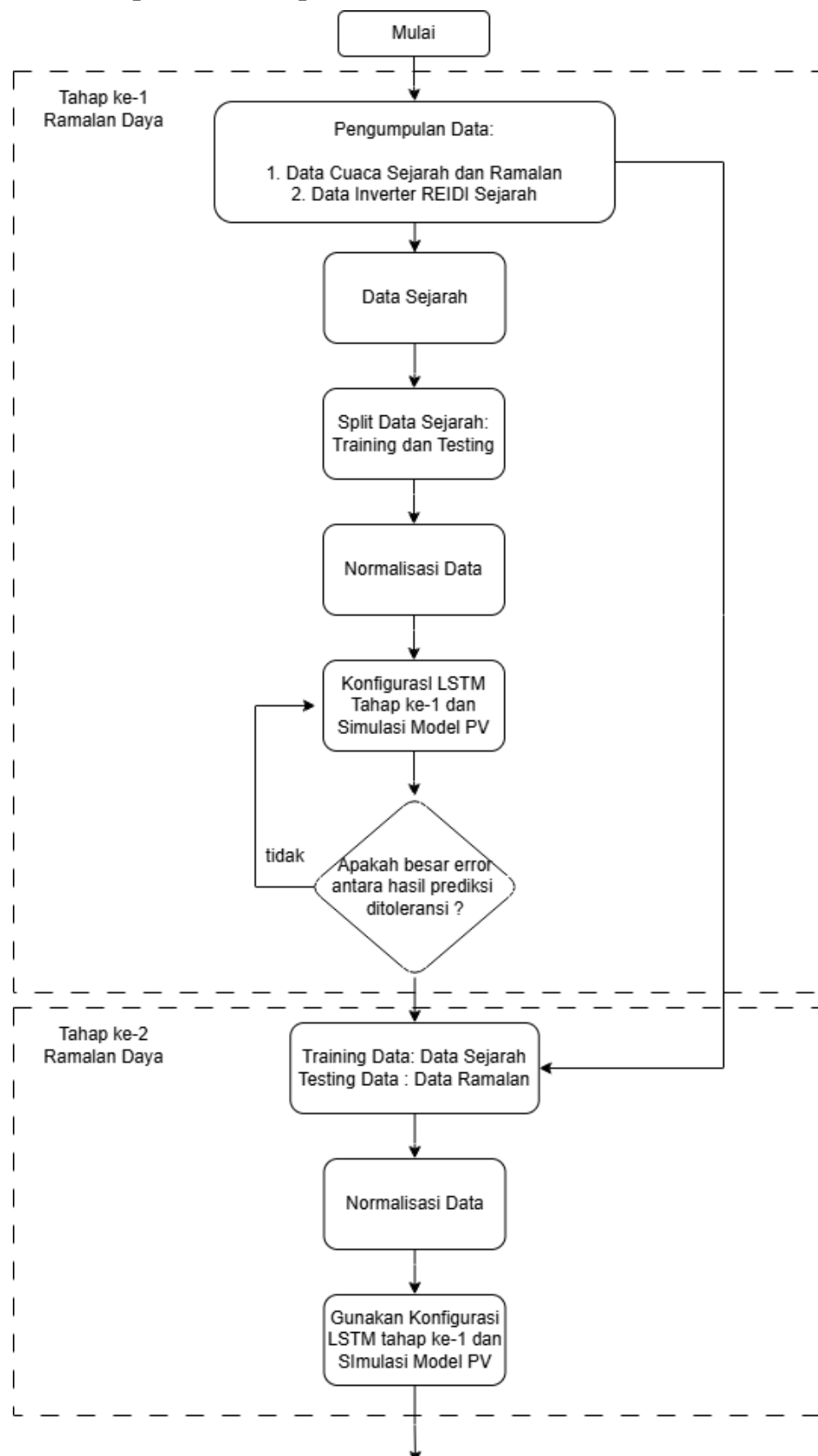
Tabel 3.15 Potensi Output PLTS untuk menguji MILP (30 Mei 2026)

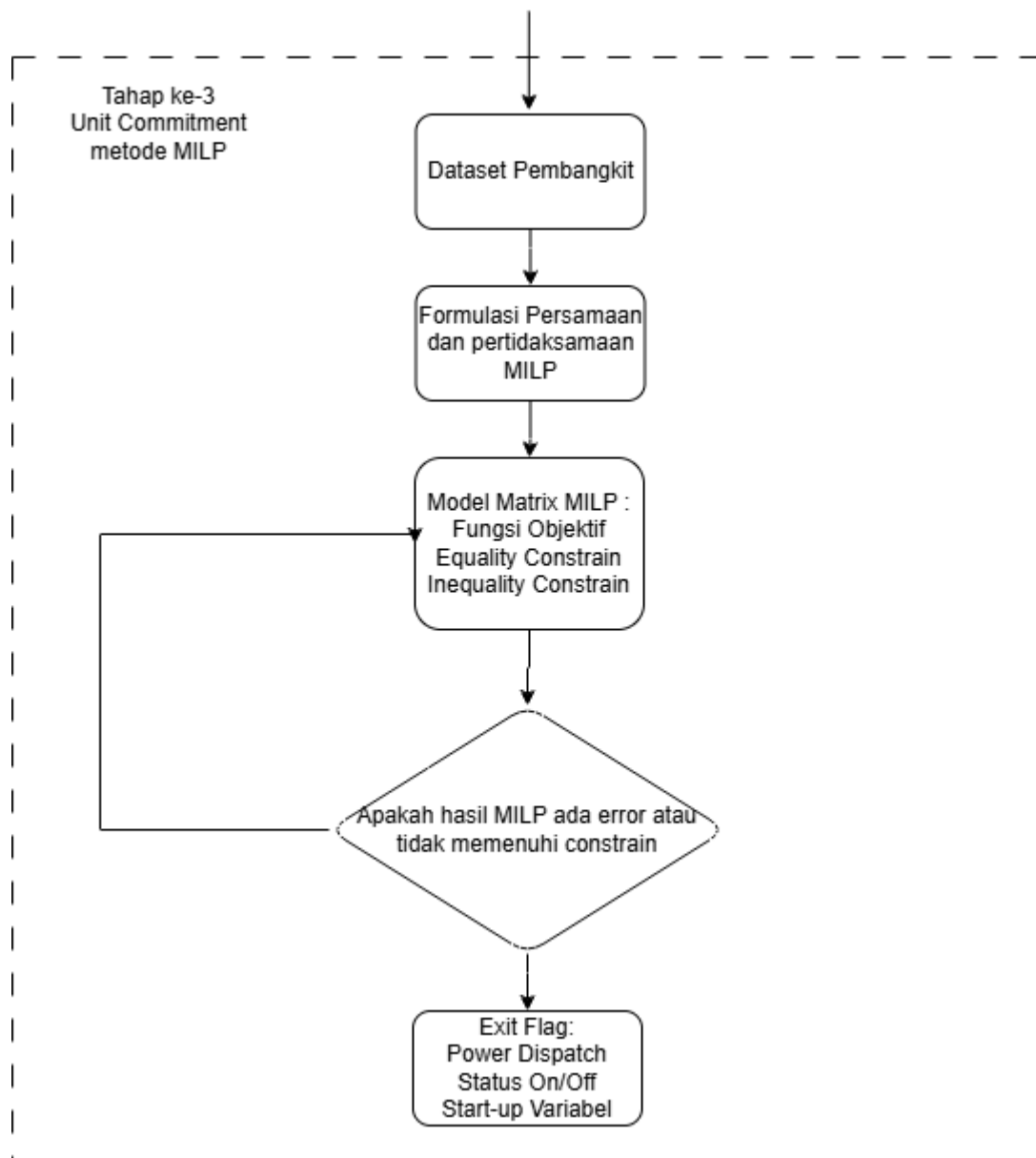
Jam	Output Daya Prediksi PLTS (MW) Kapasitas (1.43 MWp)
0 - 1	0.000
1 - 2	0.000
2 - 3	0.000
3 - 4	0.000
4 - 5	0.000
5 - 6	0.000
6 - 7	0.017
7 - 8	0.202
8 - 9	0.422
9 - 10	0.612
10 - 11	0.787
11 - 12	0.947
12 - 13	1.023
13 - 14	0.822
14 - 15	0.512
15 - 16	0.426
16 - 17	0.256
17 - 18	0.065
18 - 19	0.006
19 - 20	0.000
20 - 21	0.000
21 - 22	0.000
22 - 23	0.000
23 - 24	0.000

Tabel 3.16 Data Beban Harian Pulau Selayar

Jam	Load Data (MW)	Jam
1	4.917	0 - 1
2	4.697	1 - 2
3	4.6145	2 - 3
4	4.466	3 - 4
5	4.5815	4 - 5
6	4.477	5 - 6
7	4.323	6 - 7
8	4.51	7 - 8
9	4.774	8 - 9
10	4.8015	9 - 10
11	4.917	10 - 11
12	5.1205	11 - 12
13	5.1205	12 - 13
14	5.2305	13 - 14
15	5.236	14 - 15
16	5.1975	15 - 16
17	5.192	16 - 17
18	6.6275	17 - 18
19	7.095	18 - 19
20	6.7815	19 - 20
21	6.149	20 - 21
22	6.3745	21 - 22
23	5.94	22 - 23
24	5.577	23 - 24

3.4 Urutan pelaksanaan penelitian





BAB 4 Hasil dan Pembahasan

4.1 Hasil penelitian

Pada BAB 4 adalah hasil-hasil penelitian yang ada. Hasil-hasil tersebut didapatkan pada tahap-tahap yang dituliskan di metodologi. Di bawah ini adalah hasil-hasil yang didapatkan dari penelitian:

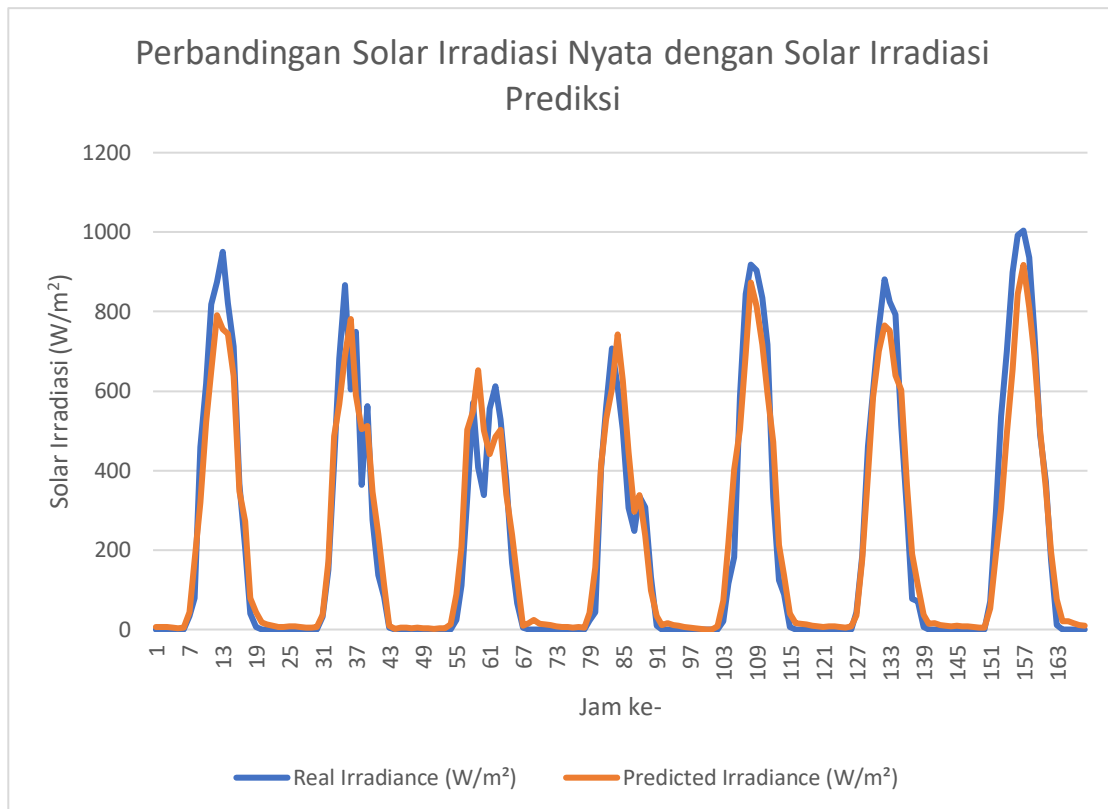
1. Tahap 1 ramalan daya, menggunakan data sejarah cuaca untuk training dan testing model LSTM
2. Tahap 2 ramalan daya, menggunakan data sejarah cuaca untuk training dan data ramalan cuaca untuk testing model LSTM
3. Power dispatch tiap pembangkit dari tiap jam dari MILP
4. Start-up variabel tiap pembangkit pada tiap jam dari MILP
5. Status on/off tiap pembangkit pada tiap jam
6. Evaluasi batasan daya minimum dan maksimum MILP tiap pembangkit pada tiap jam
7. Evaluasi batasan ramp up dan ramp down MILP tiap pembangkit pada tiap jam
8. Power balance Sistem
9. Energy Balance
10. Perbandingan Dispatch menggunakan daya yang diprediksi dan daya nyata dari PLTS.

4.1.1 Hasil Model Ramalan Daya Tahap 1

Hasil dari tahap 1 pembuatan model ramalan solar iradiasi sebelum dijadikan input model simulasi PV Array yang berkapasitas 28,6 kWp dimana berarti energi yang dihasilkan tiap hari, berdasarkan peak sun hours di Surabaya 4 - 5 jam, adalah 114.4 – 143 kWh per hari. Pada tahap ini digunakan data sejarah untuk training dan testing model LSTM. Di bawah ini adalah perbandingan kurva hasil prediksi dan kurva nilai nyata solar iradiasi dan daya pada tanggal 19-25 November 2025.

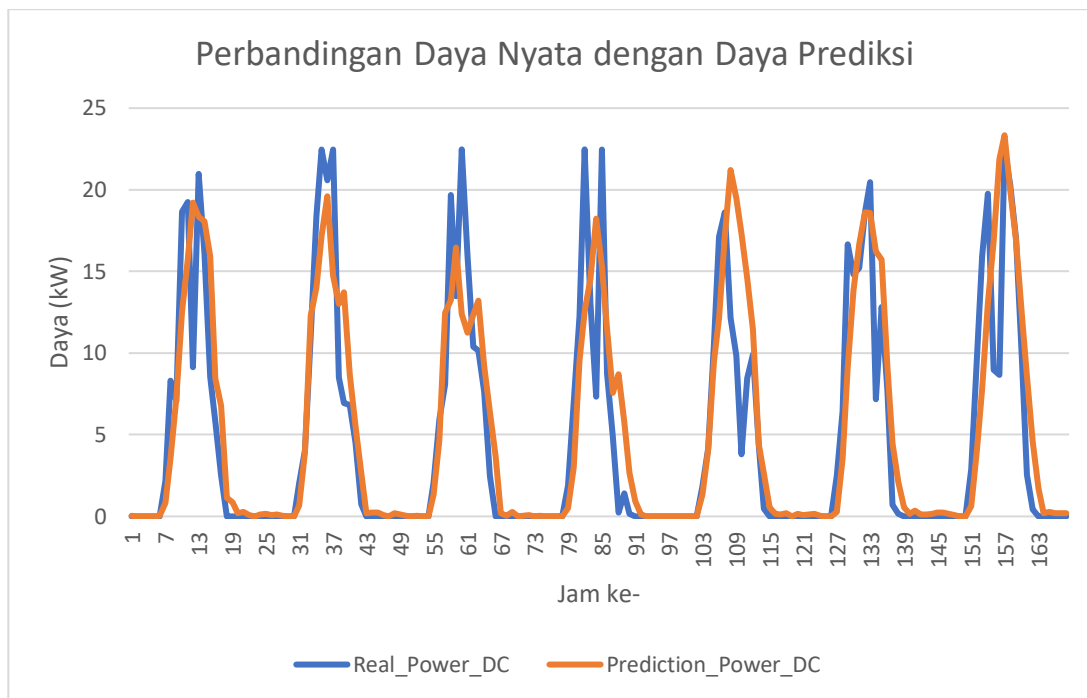
Tabel 4.1 Tahap ke-1 Pembagian Data Sejarah untuk *Training* dan *Testing*

	Training	Testing
Tahap ke-1	Data Sejarah [01-01-2024 sampai 17-11-2025]	Data Sejarah [18-11-2025 sampai 25-11-2025]



Gambar 4.1 Perbandingan Hasil Prediksi Solar Irradiasi dengan Nilai Nyata Solar Irradiasi tanggal 19-25 November 2025

Hasil dari gambar 4.1 garis berwarna biru merepresentasikan hasil prediksi solar irradiance dan garis oranye merepresentasikan nilai solar irradiasi nyata yang sudah terekam. Dikarenakan di Lab REIDI terdapat beberapa data yang tidak terekam atau kosong maka data solar irradiasi asli diambil dari situs <https://open-meteo.com/>. Perbedaan hasil solar irradiasi dari sensor tidak berbeda jauh dari data dari situs <https://open-meteo.com/>. Maka data dari situs dapat digunakan sebagai data asli. Berdasarkan gambar 4.1 dilakukan prediksi selama 7 hari, 168 jam, dimana dapat terlihat solar irradiasi mulai terlihat dimulai pada jam ke-7 pagi dan berakhir di jam ke-19. Ini dapat terlihat pada juga di hari berikutnya pada jam ke-31(Hari ke -2 jam ke-7) dan jam ke-43 (Hari ke-2, jam ke-24). Selama 7 hari diprediksikan solar irradiasi memuncak di jam ke -13 tiap hari nya selama 7 hari. Juga dapat terlihat model LSTM mencoba untuk menyesuaikan kurva-nya di hari ke-4. Jumlah solar irradiance yang diprediksikan diterima di lokasi REIDI untuk 7 hari tersebut adalah $37,614.6 \text{ Wh/m}^2$ dengan rata-rata 5373.515 Wh/m^2 per hari. Kenyataannya pada 7 hari tersebut jumlah energi solar irradiasi yang diterima di lokasi adalah $37,454.36 \text{ Wh/m}^2$ dengan rata-rata per hari 5350.623 Wh/m^2 . Rata-rata per hari yang sebenarnya dan juga yang diprediksi sesuai dengan solar irradiasi per hari di indonesia dimana peak sun hours berada pada jangka 4-6 jam per hari. Dengan menggunakan rumus Pearson Correlation coefficient untuk mengevaluasi hasil prediksi. Untuk hasil prediksi solar irradiasi dinyatakan bahwa hasil prediksi memiliki akurasi rata-rata 97.26% dari hasil asli-nya setiap 24 Jam.



Gambar 4.2 Perbandingan Hasil Prediksi Daya dengan Nilai Nyata Daya tanggal 19-25 November 2025

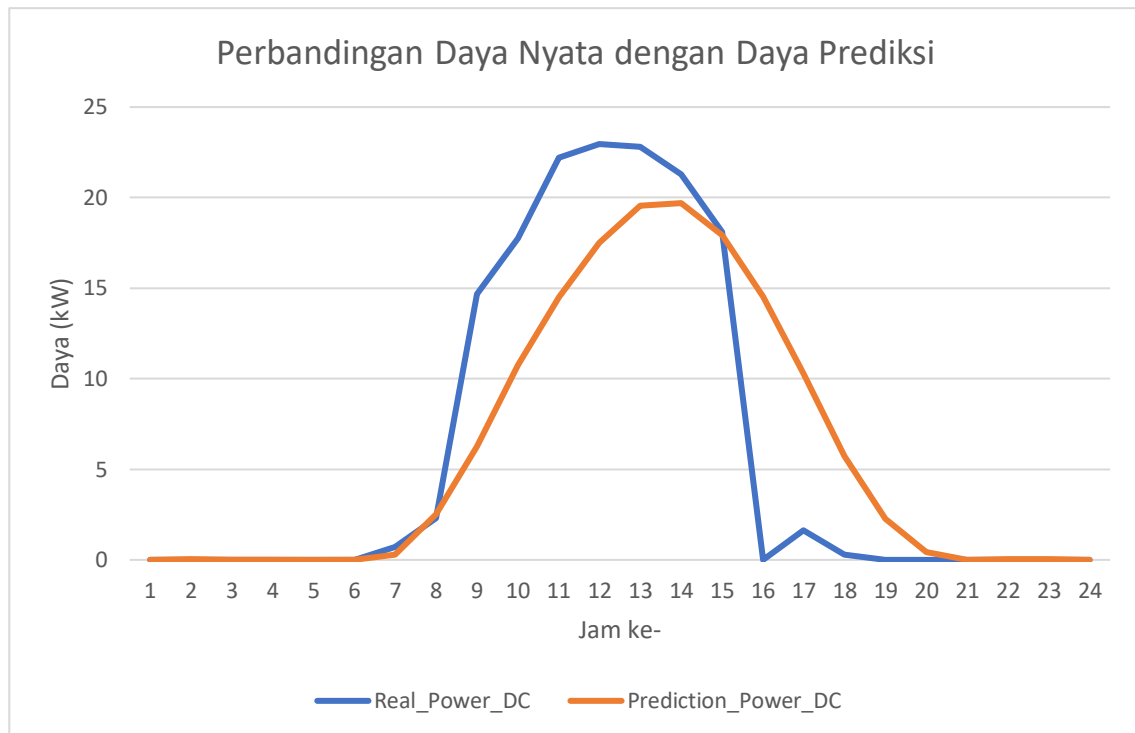
Dari gambar 4.2 adalah hasil prediksi daya dibandingkan dengan daya asli. Hasil prediksi daya didapatkan melalui hasil prediksi solar irradiasi dan ramalan suhu dari data ramalan cuaca yang digunakan sebagai input di PV simulasi REIDI. Menurut kurva nya hasil dari simulasi PV masih sulit dalam menanganinya Daya pada saat jam tertentu dimana daya tiba-tiba drop. Jumlah energi yang diprediksikan di lokasi REIDI untuk 7 hari tersebut adalah 905.71 kWh dengan rata-rata 129.39 kWh per hari. Kenyataannya pada 7 hari tersebut Jumlah energi yang dibangkitkan PLTS yang diterima di lokasi adalah 830.51 kWh dengan rata-rata per hari 118.64 kWh. Dengan menggunakan rumus Pearson Correlation coefficient untuk mengevaluasi hasil prediksi. Untuk hasil prediksi Daya dinyatakan bahwa hasil prediksi memiliki akurasi rata-rata 87.47% dari hasil asli-nya setiap 24 Jam.

4.1.2 Hasil Model Ramalan Daya tahap 2

Pada tahap 2 dalam pembuatan model prediksi daya digunakan seluruh data sejarah sebagai training dan data ramalan untuk testing. Berbeda dengan tahap pertama, daya prediksi pada tahap kedua daya asli harus ditunggu karena belum terekam atau belum dihasilkan sampai tanggal ditesting selesai.

Tabel 4.2 Tahap ke-2 Pembagian Data Sejarah dan Data Ramalan untuk *Training* dan *Testing* Rentang 24 Jam (1 hari)

	Training	Testing
Tahap ke-2	DataSejarah [01-01-2024 sampai 12-04-2026]	DataRamalan [13-04-2026 sampai 14-04-2026]

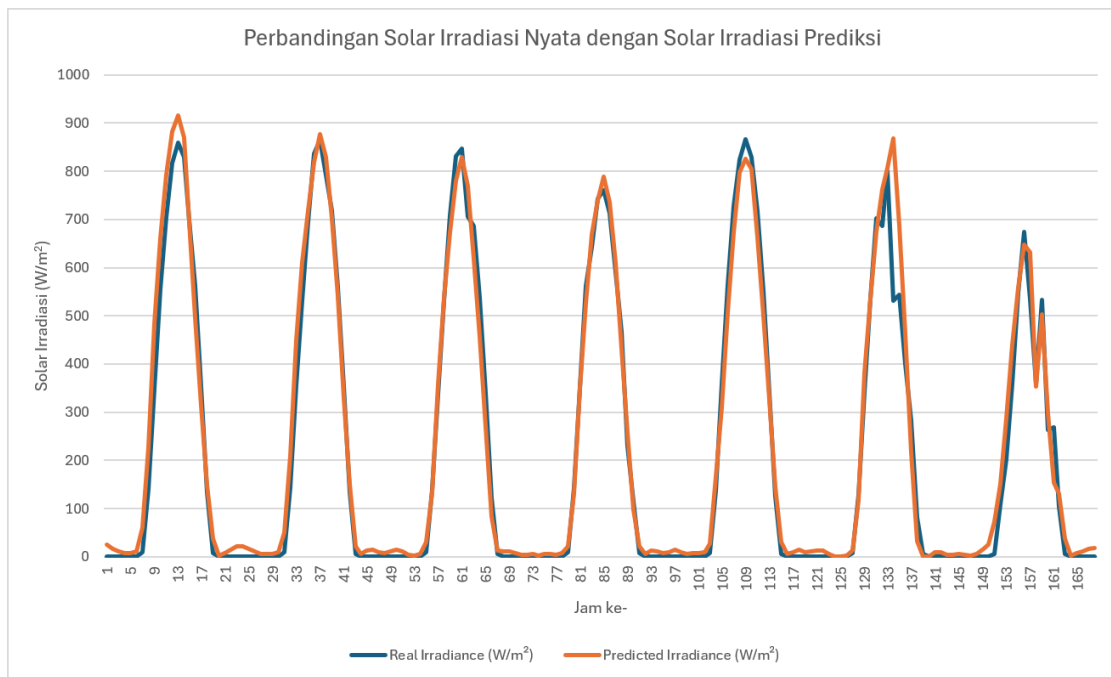


Gambar 4.3 Perbandingan Hasil Prediksi Daya yang diprediksi pada tanggal 13 April 2026 dengan Nilai Nyata Daya tanggal 14 April 2026

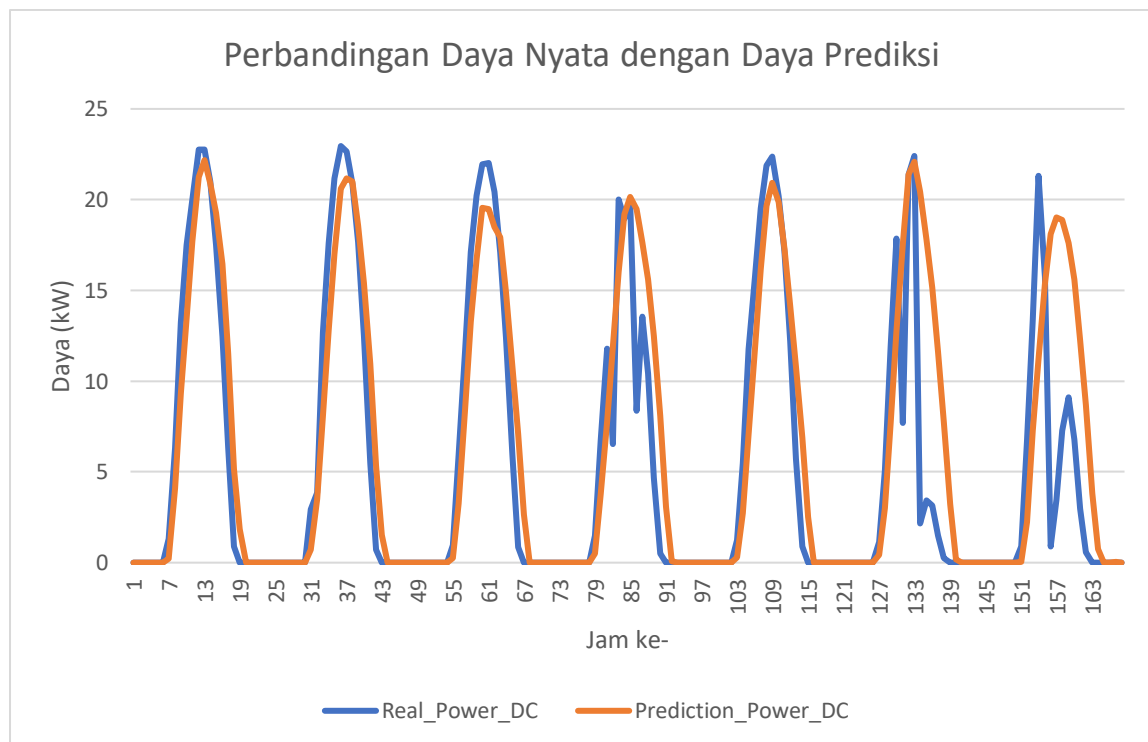
Dari gambar 4.3 Dapat terlihat bahwa pada kurva data real menghasilkan daya yang lebih banyak dari yang diprediksi dari jam ke-7 sampai jam ke-15. Pada saat jam ke-16 diprediksikan daya akan menurun secara bertahap tetapi secara nyata-nya pada saat jam ke-16 daya drop ke 0 dan dibawah kurva prediksi sampai ke jam ke-20. Jumlah energi yang diprediksikan di lokasi REIDI untuk hari tersebut adalah $142.41 kWh$. Kenyataannya pada hari tersebut Jumlah energi yang dibangkitkan PLTS yang di lokasi adalah $144.71 kWh$. Dengan menggunakan rumus Pearson Correlation coefficient untuk mengevaluasi hasil prediksi, dinyatakan bahwa hasil prediksi daya memiliki akurasi 85.67% per hari dari hasil asli-nya.

Tabel 4.3 Tahap ke-2 Pembagian Data Sejarah dan Data Ramalan untuk *Training* dan *Testing* Rentang 168 Jam (7 hari)

	Training	Testing
Tahap ke-2	DataSejarah [01-01-2024 sampai 14-04-2026]	DataRamalan [15-04-2026 sampai 22-04-2026]



Gambar 4.4 Perbandingan Hasil Prediksi yang diprediksi pada tanggal 15 April 2026 dengan Nilai Nyata Solar Irradiasi tanggal 16 sampai 22 April 2026



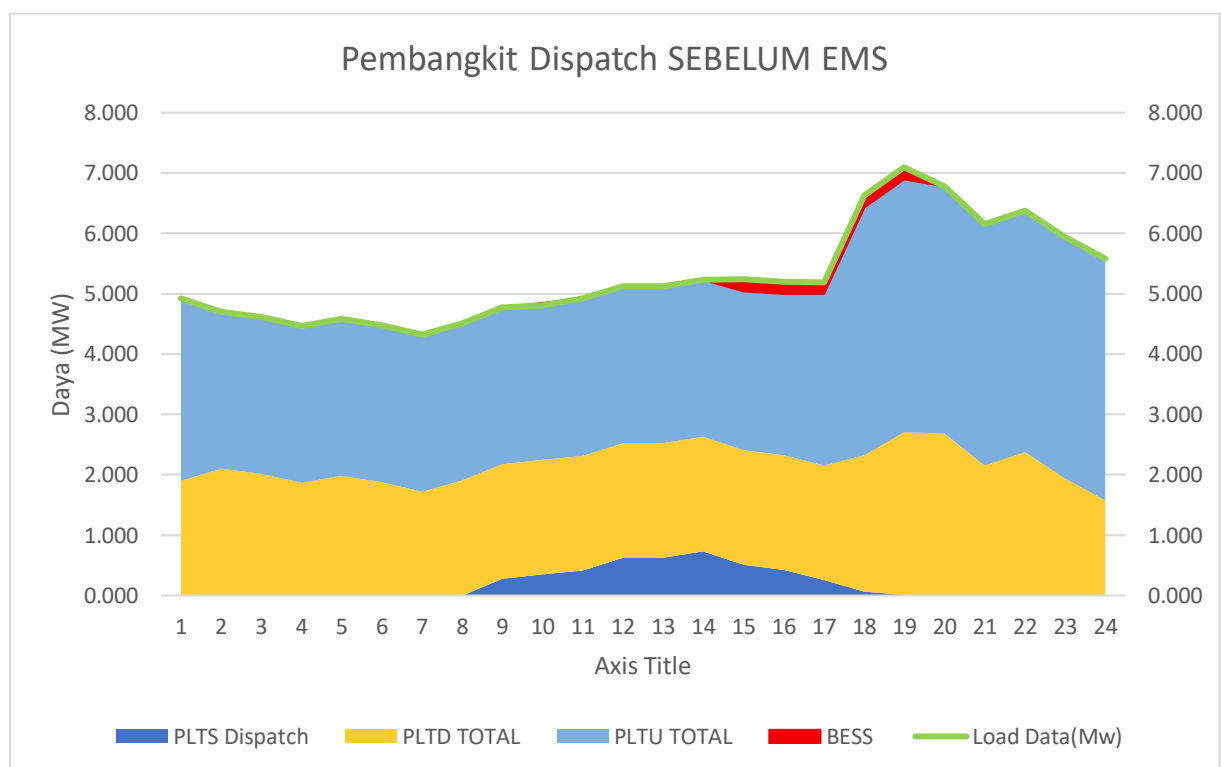
Gambar 4.5 Perbandingan Hasil Prediksi Daya yang diprediksi pada tanggal 15 April 2026 dengan Nilai Nyata Daya tanggal 16 sampai 22 April 2026

Berdasarkan gambar 4.4 dapat dilihat model memiliki prediksi dimana memiliki nilai solar irradiasi yang lebih dari nol tetapi seharusnya nilai nyata-nya adalah nol dikarenakan jam-nya. Tetapi walaupun seperti itu sebagai input simulasi PV hasil prediksi PV tidak mengalami perubahan dimana masih menghasilkan daya nol. Pada hari ke-7 grafik gambar 4.4 dapat dilihat model mencoba untuk menyesuaikan hasil-nya dengan nilai nyata. Diamana dapat terlihat di

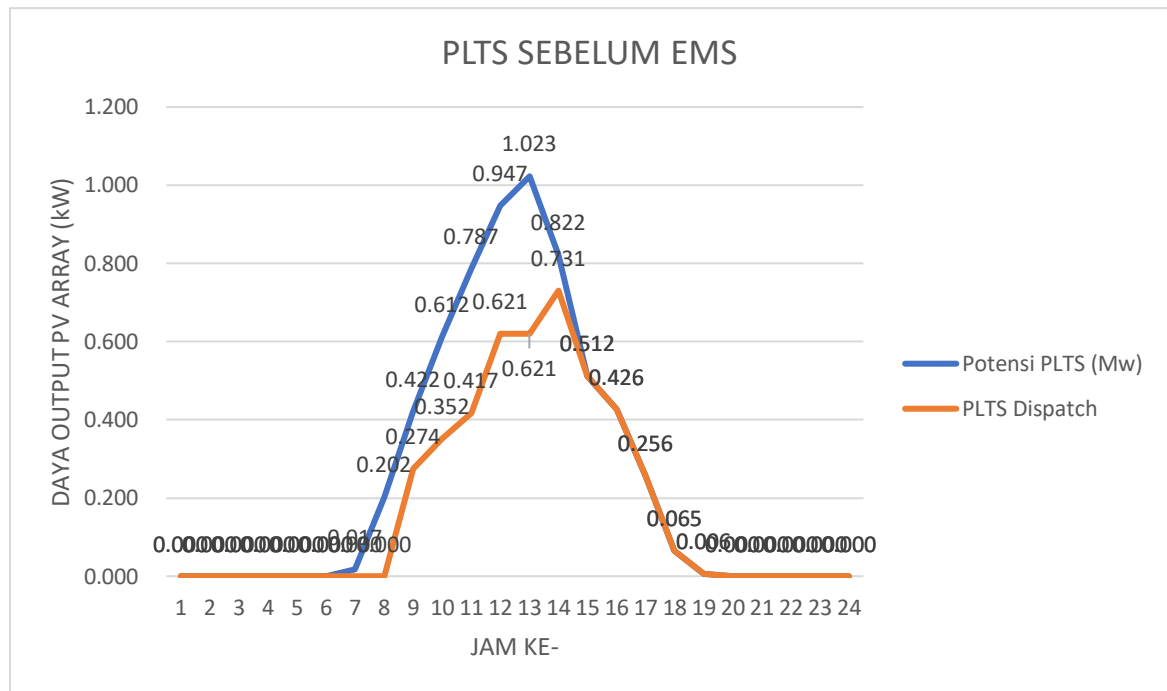
hari ke-7 grafik gambar 4.5 Hasil prediksi daya tidak mencoba untuk menyesuaikan seperti hasil prediksi pada gambar 4.4. ini dikarenakan saat memasukkan input ke dalam model LSTM yang sudah terlatih digunakan data ramalan suhu ambien sebagai input simulasi PV berbeda dengan tahap ke-1 dimana suhu ambien pada hari-hari si tahap ke-1 sudah terekam. Jumlah energi yang diprediksikan di lokasi REIDI untuk 7 hari tersebut adalah 1090.70 kWh dengan rata-rata 155.82 kWh per hari. Kenyataannya pada 7 hari tersebut Jumlah energi yang dibangkitkan PLTS yang diterima di lokasi adalah 943.79 kWh dengan rata-rata per hari 134.83 kWh. Dengan menggunakan rumus Pearson Correlation coefficient untuk mengevaluasi hasil prediksi. Untuk hasil prediksi Solar Irradiasi dinyatakan bahwa hasil prediksi memiliki akurasi rata-rata 99.07% setiap 24 jam (1 hari) dari hasil asli-nya. Untuk hasil prediksi Daya dinyatakan bahwa hasil prediksi memiliki akurasi rata-rata 85.41% setiap 24 jam (1 hari) dari hasil asli-nya.

4.1.3 Hasil Simulasi Unit Commitment

Hasil prediksi daya digunakan untuk membuat model MILP dengan untuk menggunakan semua potensi daya yang dibangkitkan economic dispatch dan unit commitment. Dibawah ini adalah hasil potensi penggunaan daya yang dihasilkan sebelum dan sesudah melewati model MILP. Gambar dibawah ini adalah kurva sebelum EMS

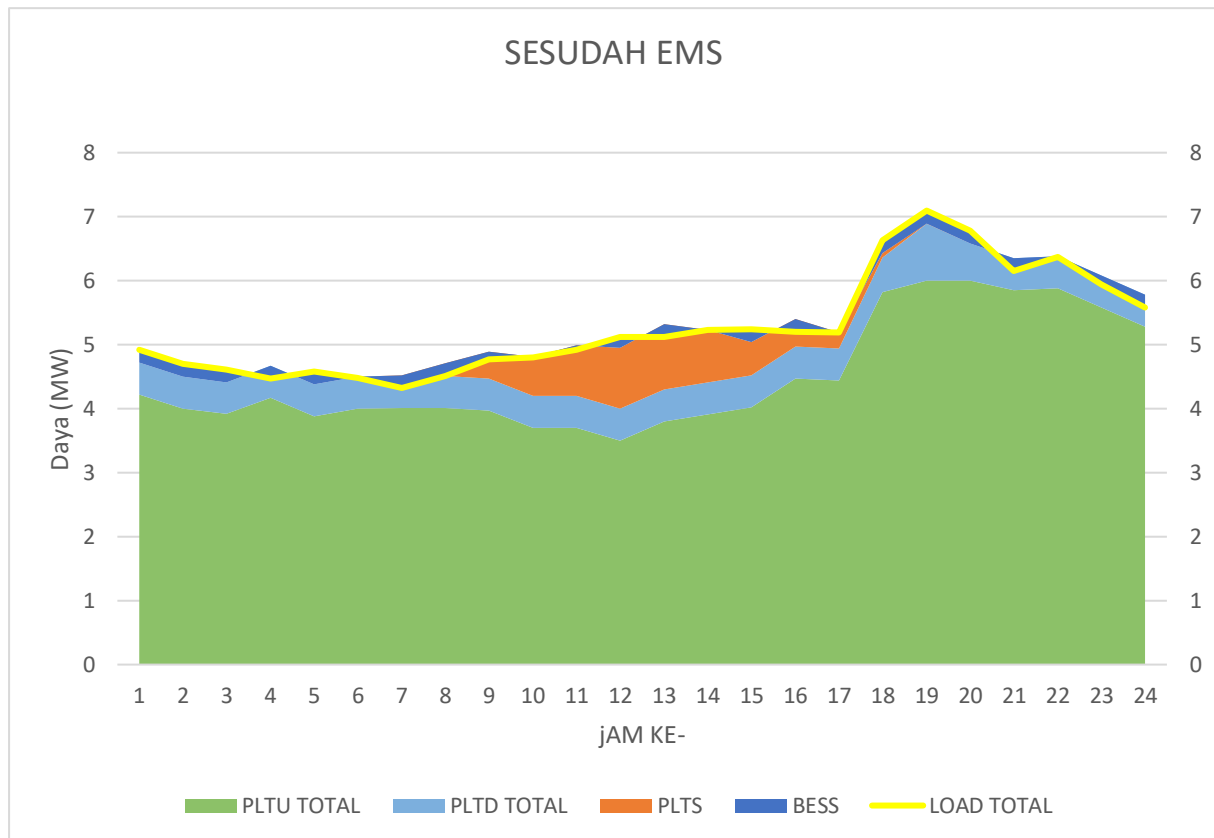


Gambar 4.5 Hasil *Power Dispatch* sebelum MILP

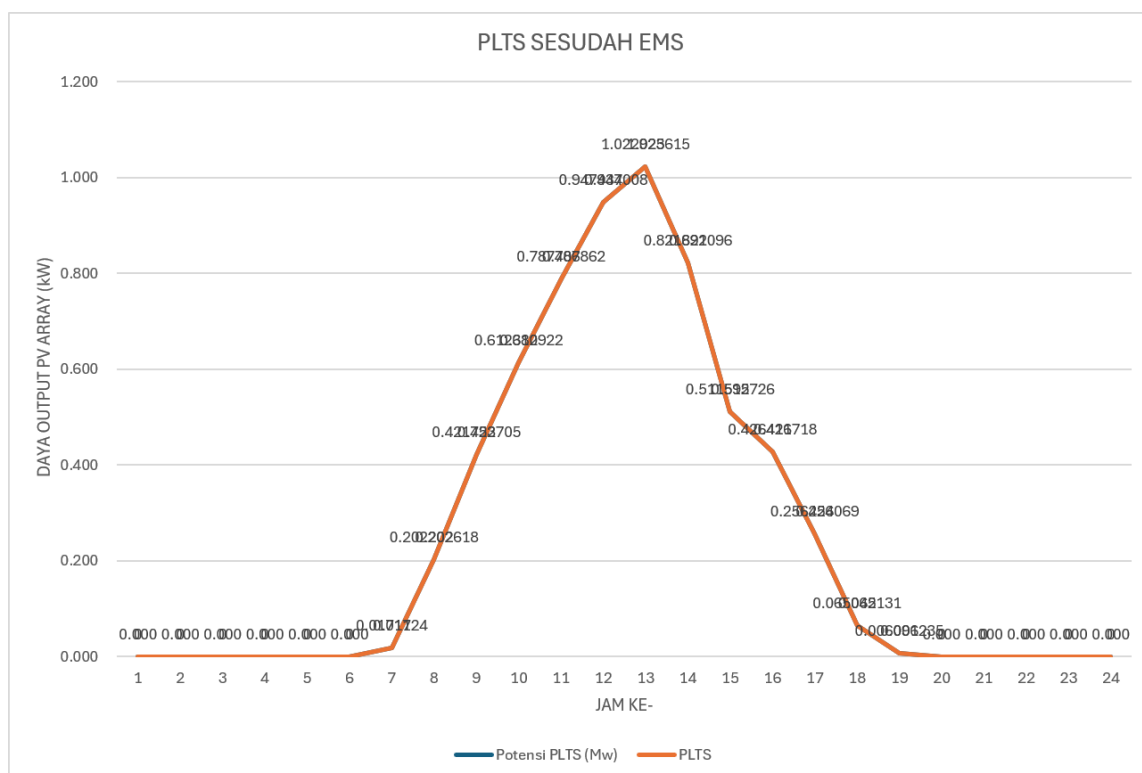


Gambar 4.6 Hasil PLTS *Dispatch* sebelum MILP

Dari gambar 4.6 dan gambar 4.7 PLTS tidak dapat mengembangkan daya sebanyak potensinya. Hasil dispatch PLTS pada gambar 4.6 dapat dilihat pada jam ke-7 sampai ke-14 tidak mencapai potensi-nya, sebaga contoh adalh potensi PLTS pada jam ke-13 dimana solar irradiasi yang paling tinggi PLTS dapat mengeluarkan output sebesar 1.023 MW tetapi dikarenakan BESS SOC sudah penuh dan karena beban sudah hampir terpenuhi oleh pembangkit lain maka PLTS hanya menghasilkan energi untuk daya yang kurang untuk memenuhi beban pada jam tersebut yaitu, 0.621 MW . Dengan program MILP dispatch PLTS akan diprioritaskan agar PLTS dapat mendispatch daya sesuai dengan potensinya pada jam tersebut.



Gambar 4.7 Hasil *Power Dispatch* setelah MILP



Gambar 4.8 Hasil *Dispatch* PLTS sesudah EMS

Dari gambar 4.7 dan 4.8 semua potensi yang dibangkitkan PLTS digunakan semua menyesuaikan dengan load data. Dari kedua gambar tersebut juga menunjukkan energi yang

dibangkitkan PLTU lebih tinggi setelah MILP dan penurunan pembangkitan daya PLTD. Semua hasil dispatch pembangkit juga harus memenuhi tiap constarin yang ditentukan saat memformulasikan model MILP dibawah ini adalah hasil-hasil yang memenuhi tiap batasan-batasan terseut.

4.1.3.1 Hasil Power Dispatch

Dibawah ini adalah hasil power dispatch yang dibangkitkan tiap pembangkit pada tiap jam. Pembangkit yang dapat dalam sistem terdiri dari PLTS, 4 PLTD (DEUTZ), 1 PLTU (DAIHATSU), 6 PLTU (CUMMINS) dan 1 BESS.

Jam ke-	PLTS	PLTD SELAY	PLTD SELA	PLTD SELA	PLTD SELAYAR	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	BESS
1	0	0	0.5	0	0	0	1	0.5	0.717	1	0.5	0.5	0.2
2	0	0	0.5	0	0	0	1	0.5	0.697	0.8	0.5	0.5	0.2
3	0	0	0.5	0	0	0	0.9145	0.5	1	0.5	0.5	0.5	0.2
4	0	0	0.5	0	0	0	1	0.5	1	0.5	0.666	0.5	-0.2
5	0	0	0.5	0	0	0	1	0.5	0.7	0.5	0.6815	0.5	0.2
6	0	0	0.5	0	0	0	0.7	0.8	0.5	0.5	1	0.5	-0.023
7	0.017112	0	0.5	0	0	0	1	0.5	0.805888	0.5	0.7	0.5	-0.2
8	0.202203	0	0.5	0	0	0	1	0.5	0.507797	0.5	1	0.5	-0.2
9	0.421754	0	0.5	0	0	0	0.7	0.9	0.571025	0.5	0.8	0.5	-0.118778208
10	0.612381	0	0.5	0	0	0	0.8	0.6	0.5	0.8	0.5	0.5	-0.010880922
11	0.787407	0	0.5	0	0	0	0.5	0.5	0.9	0.5	0.8	0.5	-0.070406862
12	0.947434	0	0.5	0	0	0	0.5	0.5	1	0.5	0.5	0.5	0.173065992
13	1.022926	0	0.5	0	0	0	0.5	0.5	0.897574	0.9	0.5	0.5	-0.2
14	0.821691	0	0.5	0	0	0	0.8	0.5	0.597574	0.657919	0.853316	0.5	0
15	0.511596	0	0.5	0	0	0	0.5	0.5	0.971088	1	0.553316	0.5	0.2
16	0.426412	0	0.5	0	0	0	0.9	0.9	0.671088	0.9	0.6	0.5	-0.2
17	0.256424	0	0.5	0	0	0	1	0.6	0.735576	0.6	1	0.5	0
18	0.065042	0	0.538909	0	0	0	1	1	1	1	1	0.823549	0.2
19	0.006091	0	0.888909	0	0	0	1	1	1	1	1	1	0.2
20	0	0	0.5815	0	0	0	1	1	1	1	1	1	0.2
21	0	0	0.5	0	0	0	1	1	1	1	1	0.849	-0.2
22	0	0	0.5	0	0	0	1	1	1	1	1	0.88225	-0.00775
23	0	0	0.5	0	0	0	1	1	1	1	1	0.58225	-0.14225
24	0	0	0.5	0	0	0	1	1	1	0.777	1	0.5	-0.2

Gambar 4.9 Hasil nilai dispatch tiap pembangkit dalam rentang 24 jam

Dapat terlihat dari gambar 4.9 dari 4 pembangkit PLTD (DEUTZ) hanya ada 1 saja yang mendispatch daya, dan mendispatch nya dari jam ke-1 samapai jam ke-24. Pembangkit PLTU (DAIHASTU) tidak mengeluarkan daya sama sekali. Pembangkit PLTU (CUMMINS) mendispatch daya setiap jam-nya. PLTS mendispatch daya nya yang hanya bisa dibangkitkan pada waktu tersebut.

4.1.3.2 Uptime dan Downtime

Dari hasil MILP, mendapatkan hasil status on/off tiap pemabnagkit di tiap jam yang ada di dalam sistem tenaga listrik:

Tabel 4.4 Status Tiap Pembangkit dalam Rentang 24 Jam

J A M k e -	P L T S	PLTD SELA YAR #01 (DEU TZ)	PLTD SELA YAR #02 (DEU TZ)	PLTD SELA YAR #03 (DEU TZ)	PLTD SELA YAR #04 (DEU TZ)	PLTU SELA YAR #01 (DAIH ATSU)	PLTU SELA YAR #01 (CUM MINS)	PLTU SELA YAR #02 (CUM MINS)	PLTU SELA YAR #03 (CUM MINS)	PLTU SELA YAR #04 (CUM MINS)	PLTU SELA YAR #05 (CUM MINS)	PLTU SELA YAR #06 (CUM MINS)	BES S (SO C AW AL 99%)
1	1	0	1	0	0	0	1	1	1	1	1	1	1
2	1	0	1	0	0	0	1	1	1	1	1	1	1
3	1	0	1	0	0	0	1	1	1	1	1	1	1
4	1	0	1	0	0	0	1	1	1	1	1	1	1
5	1	0	1	0	0	0	1	1	1	1	1	1	1
6	1	0	1	0	0	0	1	1	1	1	1	1	1
7	1	0	1	0	0	0	1	1	1	1	1	1	1
8	1	0	1	0	0	0	1	1	1	1	1	1	1
9	1	0	1	0	0	0	1	1	1	1	1	1	1
10	1	0	1	0	0	0	1	1	1	1	1	1	1
11	1	0	1	0	0	0	1	1	1	1	1	1	1
12	1	0	1	0	0	0	1	1	1	1	1	1	1
13	1	0	1	0	0	0	1	1	1	1	1	1	1
14	1	0	1	0	0	0	1	1	1	1	1	1	1
15	1	0	1	0	0	0	1	1	1	1	1	1	1
16	1	0	1	0	0	0	1	1	1	1	1	1	1
17	1	0	1	0	0	0	1	1	1	1	1	1	1
18	1	0	1	0	0	0	1	1	1	1	1	1	1
19	1	0	1	0	0	0	1	1	1	1	1	1	1
20	2	0	1	0	0	0	1	1	1	1	1	1	1
21	2	0	1	0	0	0	1	1	1	1	1	1	1
22	2	0	1	0	0	0	1	1	1	1	1	1	1
23	2	0	1	0	0	0	1	1	1	1	1	1	1
24	2	0	1	0	0	0	1	1	1	1	1	1	1

Dapat dilihat dari Tabel 4.4 tiap pembangkit memenuhi batasan uptime dan downtime. Dimana Uptime PLTD Selayar (DEUTZ) adalah 4 jam dan downtimw nya adalah 2 jam. PLTU Selayar

(CUMMINS) memiliki uptime 2 jam dan downtime 1 jam. Dan PLTU Selayar (DAIHATSU) memiliki batasan uptime 4 jam dan downtime 2 jam. Semua pembangkit PLTU akan selalu menyala tiap jam nya sementara dari PLTD hanya ada 1 Pembangkit yang hanya menyala terus. Jika dilihat dari power balance dan BESS, PLTS selalu menyala tetapi dikarenakan waktu terbit dan tenggelam matahari, daya yang dibangkitkan sama dengan nol. Untuk BESS juga sama seperti PLTS dimana status selalu menyala dan dapat dilihat dari power dispatch dari jam ke-1 dan ke-24 selalu mengeluarkan atau memasukkan daya.

4.1.3.3 Hasil Start-up Cost

Dikarenakan hasil Start-up cost berhubungan dengan status on/off tiap pembangkit, dan karena tidak ada pembangkit yang mengalami kondisi dari off ke on atau dari 0 ke 1 maka mendapatkan hasil seperti dibawah ini:

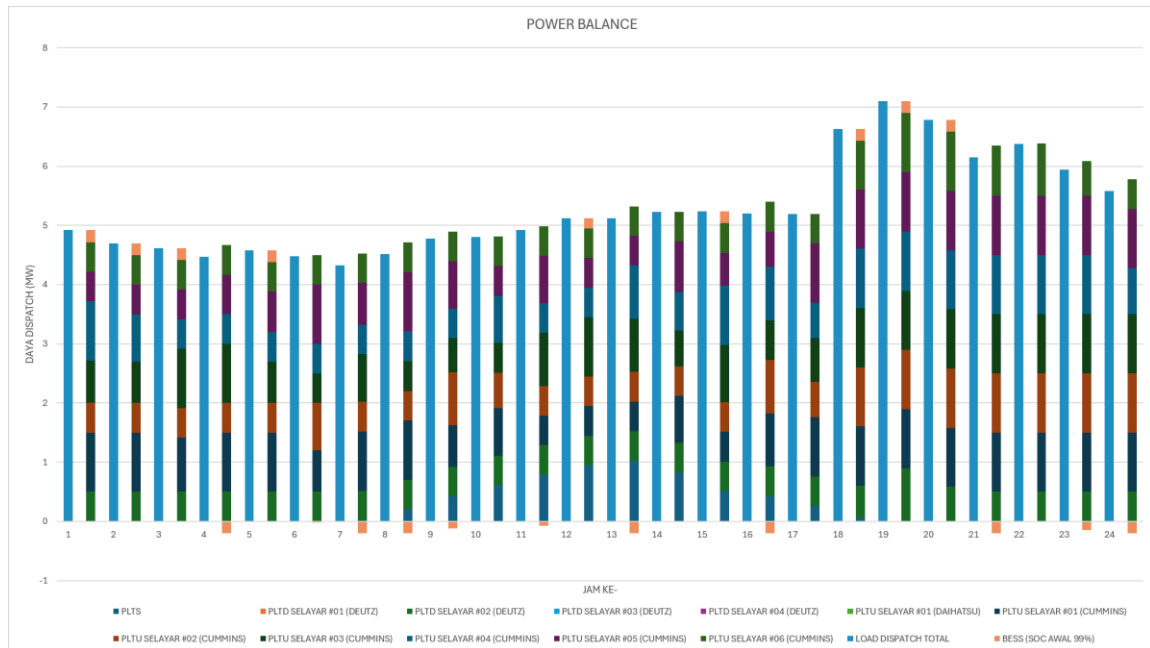
Jam ke	PLTS	PLTD SELA	PLTD SELA	PLTD SELA	PLTD SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	PLTU SELA	BESS (SOC AWAL 99%)
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 4.10 Start Up Variabel tiap Jam tiap Pembangkit

Dari gambar 4.10 hasil untuk Start-up Variabel semua sama dengan nol. Walaupun nol masih memenuhi syarat dimana variabel start-up sama dengan atau lebih dari nol. Tetapi jika ada yang mengalami dari status OFF ke ON start-up variabel yang memiliki nilai hanya PLTU karena yang hanya ada konstanta-nya hanya pada pembangkit PLTU.

4.1.3.4 Power Balance

Dari hasil solusi unit commitment dengan menggunakan metode MILP tertera pada gambar 4.9. dari gambar tersebut terdapat 2 bar dimana bar pertama adalah besar load data dan bar kedua yang memiliki berbagai warna adalah jumlah daya yang didispatch dan direpresentasikan dengan pembangkit-pembangkit yang ada dalam sistem.

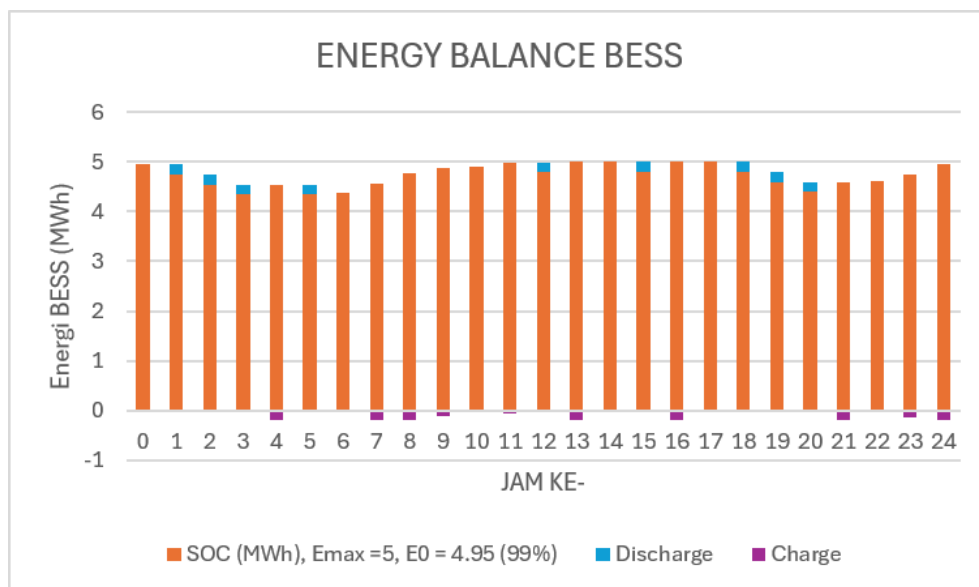


Gambar 4.11 Grafik *Power Balance MILP*

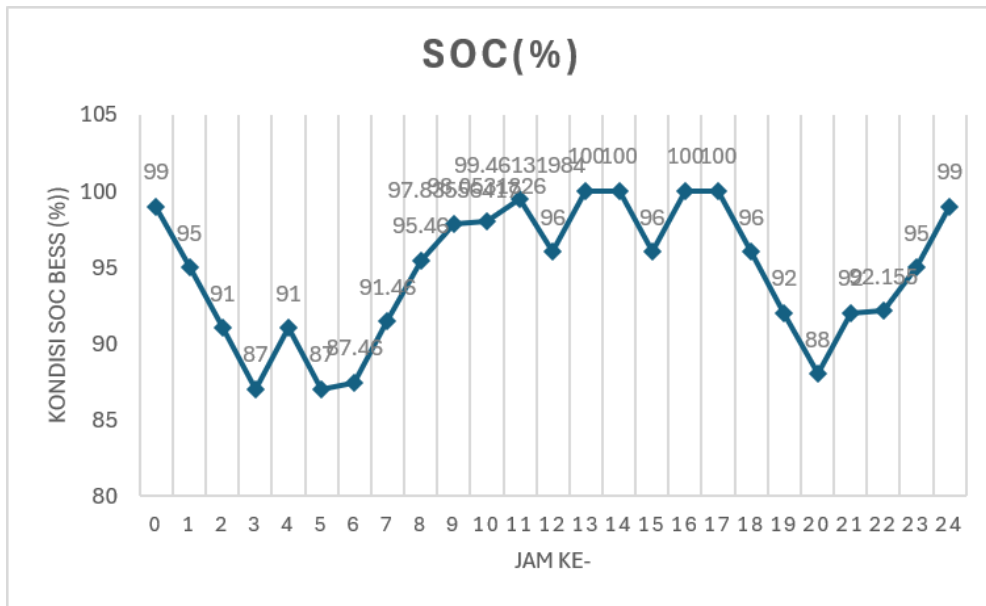
Dapat dilihat dari grafik dispatch (gambar 4.11) pembangkit tiap jam memenuhi data beban tiap jam nya. Dapat disimpulkan bahwa desain MILP memenuhi kondisi batasan Power Balance dimana tiap daya yang dibangkitkan semua pembangkit yang ada dalam sistem terpenuhi. Dari grafik ini juga memiliki bar yang dibawah nol, itu daalah bar yang merepresentasikan dispatch BESS, dalam kasus ini nilai dibawah nol berarti BESS dalam kondisi charging. Dari gambar 4.11 dapat terlihat BESS dalam kondisi charging pada jam ke-3, jam ke-4, jam ke-8, jam ke-9, jam ke-10, jam ke-11, jam ke14, jam ke-15, jam ke-17, jam ke-21, jam ke-23 dan jam ke-24.

4.1.3.5 Energy Balance

Energy Balance pada sistem adalah kondisi energy dalam sebuah baterai dalam rentang 1 hari, yang berarti adalah SOC baterai pada BESS. Dapat dilihat pada gambar dibawah ini kondisi energi BESS dan kondisi SOC BESS.

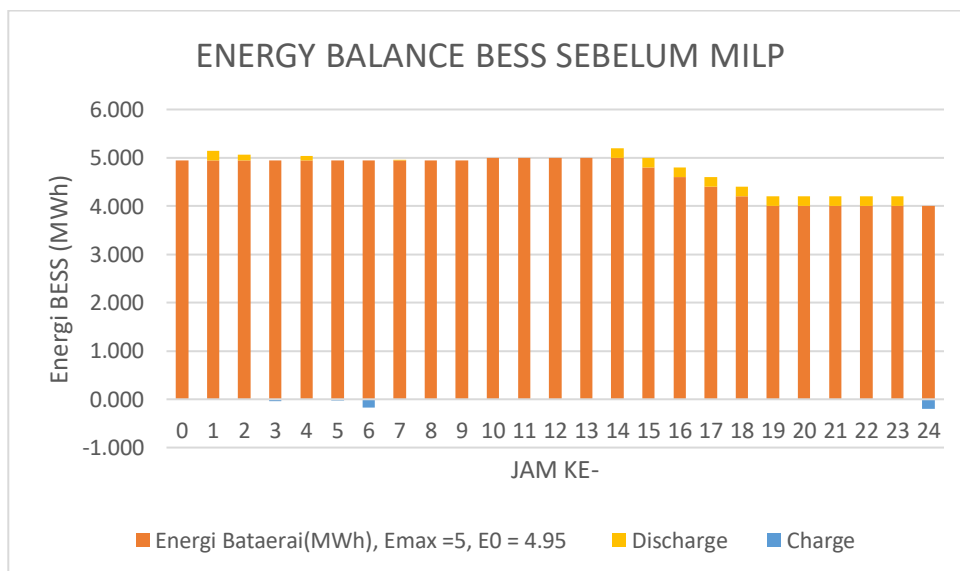


Gambar 4.12 Grafik *Energy Balance BESS (MWh)*

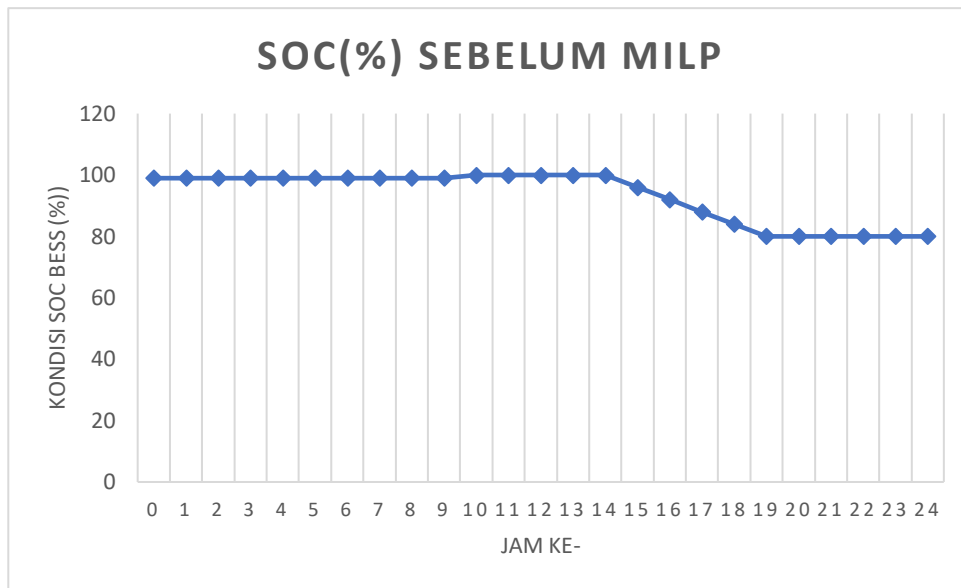


Gambar 4.13 Grafik *Energy Balance SOC BESS (%)*

Dari gambar 4.12 dan gambar 4.13 diperlihatkan hasil BESS setelah diproses oleh MILP. Dapat dilihat bahwa SOC pada jam ke-0 sama dengan di jam ke-24, yang berarti memenuhi persamaan BESS agar BESS siap untuk digunakan dihari kedepannya. Daya yang disimpan pada jam pertama hari dimulai harus sama dengan daya yang disimpan pada jam di akhir hari. Dari gambar 4.10 juga dapat dilihat tiap kali discharge dan charge BESS mengikuti batasan daya minimum dan maksimum BESS yaitu 0.2 untuk discharge dan -0.2 untuk charge. Perbedaan Energy balance sebelum dilakukan MILP akan terlihat seperti gambar dibawah ini



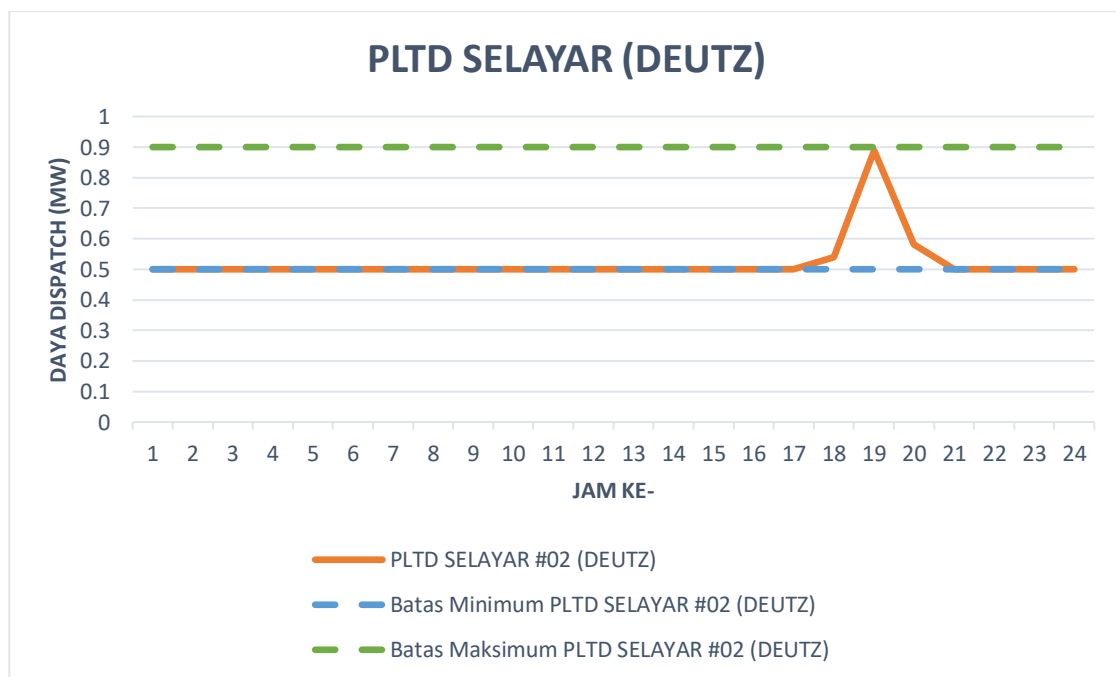
Gambar 4.14 Grafik *Energy Balance BESS (MWh) sebelum MILP*



Gambar 4.15 Grafik Energy Balance SOC BESS (%) Sebelum MILP

4.1.3.6 Batasan Dispatch minimum dan maksimum tiap pembangkit

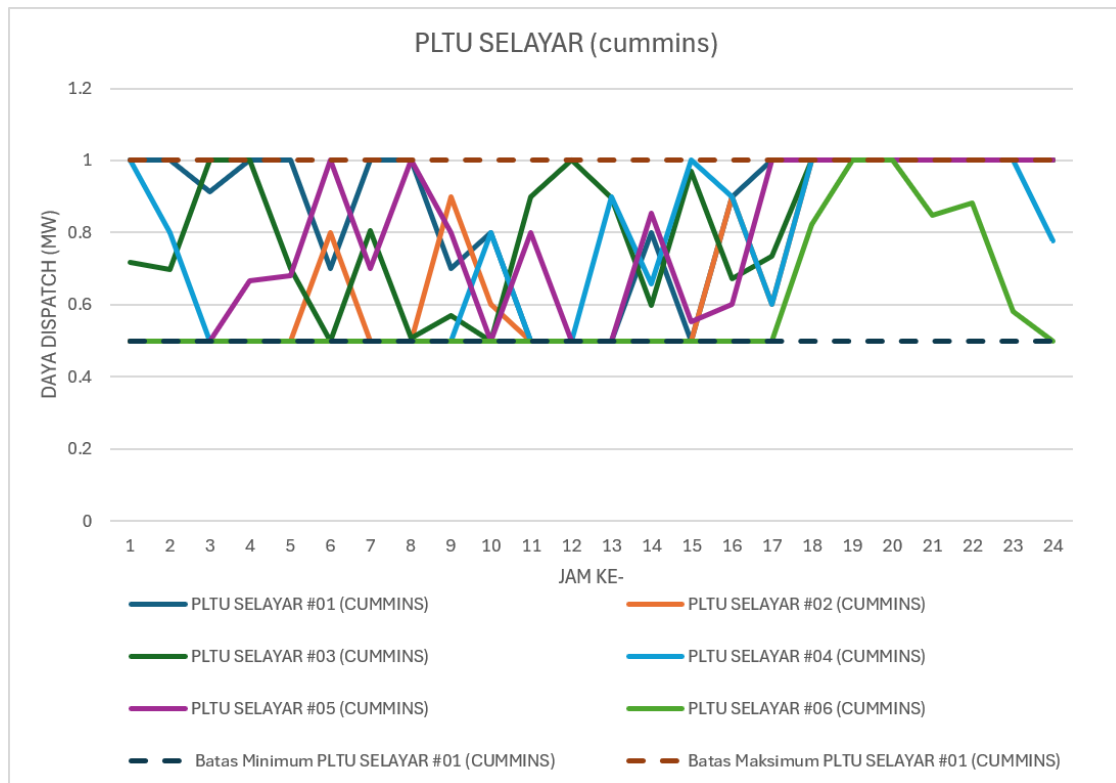
Hasil power dispatch tiap pembangkit masih ahrus mengikut rentang daya minimum dan daya maksimum jika mengalami kondisi ON pada waktunya. dibawah ini adalah grafik yang merepresentasikan power dispatch nya dan apakah mereka melewati batasan minimum dan maksimumnya.



Gambar 4.16 Grafik PLTD Selayar (DEUTZ) dengan Batasan Pmin-Pmax

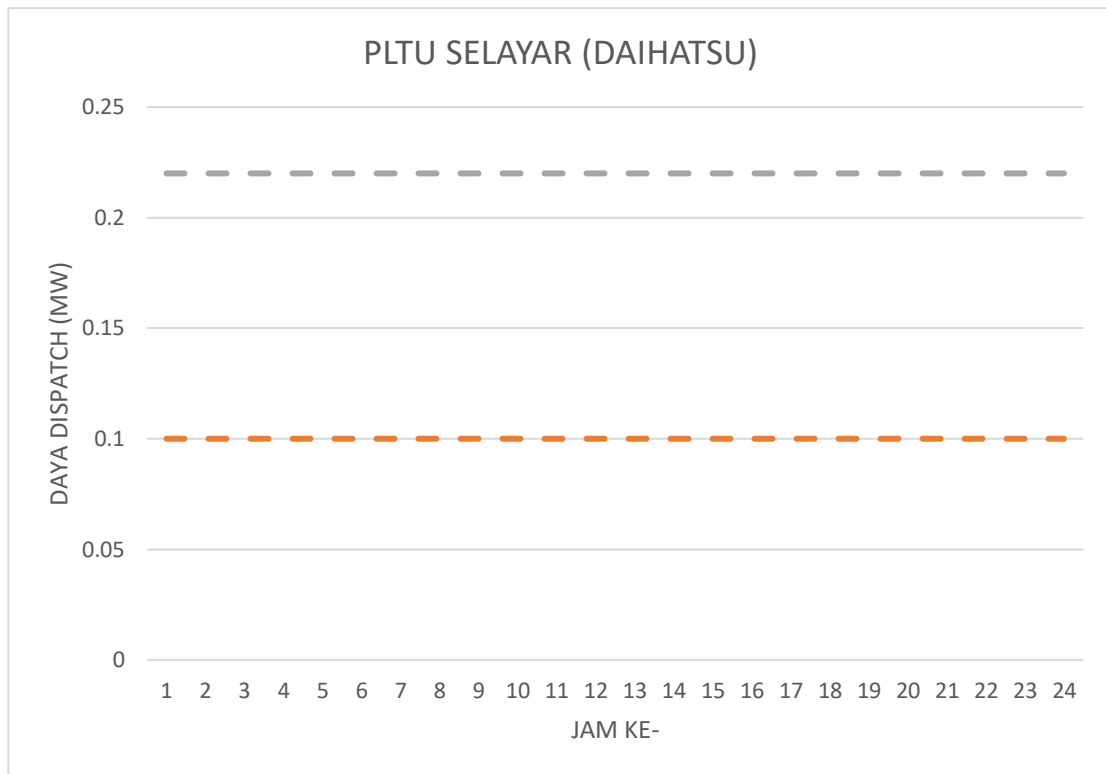
Pada Gambar 4.16 merupakan garfik PLTD Selayar yang memiliki batasan daya minimum 0.5 dan daya maksimum 0.9. Terlihat dari grafik hanya PLTD SELAYAR #02(DEUTZ) yang hnaya melakukan power dispatch dan sisanya nol atau dalam kondisi OFF. Menurut gambar 4.10. kurva PLTD SELAYAR #02 (DEUTZ) berada di rentang 0.5 dan 0.9 maka dinyatakan pembangkit PLTD memenuhi constrain daya minimum dan daya maksimum ketika kondisi

ON. Tentu saja ada pembangkit lain yang memiliki rentang daya minimum dan maksimum yang berbeda seperti PLTU (CUMMINS) dimana rentang daya minimum dan maksimumnya adalah 0.5 dan 1.



Gambar 4.17 Grafik PLTU Selayar (CUMMINS) dengan Batasan Pmin-Pmax

Dari Tabel 4.4 dan gambar 4.18 dinyatakan bahwa 6 PLTU selayar (CUMMINS) menyala tiap jam-nya maka perubahan-dispatch tiap jam-nya berada di dalam rentang 0.5 dan 1.0. Dapat dilihat dari gambar 4.17 PLTU SELAYAR #01-06 (CUMMINS) semua berada dalam garis konstan yang bernilai 0.5 dan 1.0 yang berarti hasil power dispatch PLTU SELAYAR (CUMMINS) berhasil. Yang terakhir adalah pembangkit PLTU SELAYAR#01 (DAIHATSU) yang memiliki rentang daya minimum dan maksimum 0.1 dan 0.22

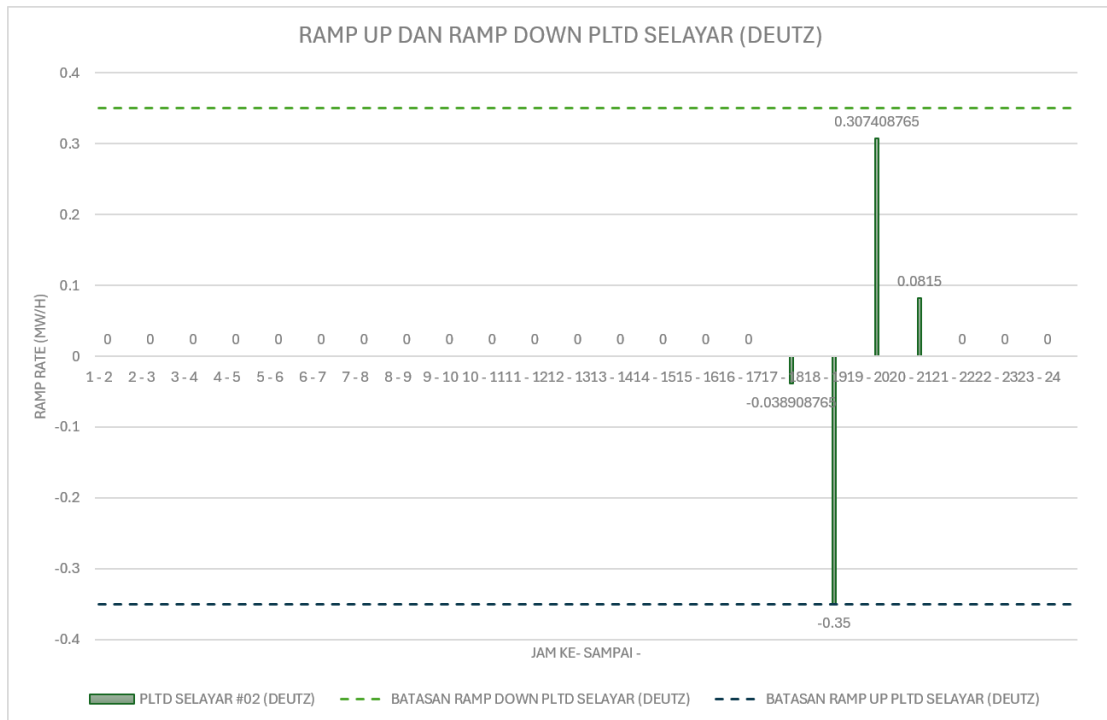


Gambar 4.18 Grafik PLTU SELAYAR (DAIHATSU) dengan Batasan Pmin-Pmax

Dari gambar 4.18 dan Tabel 4.4. PLTU SELAYAR (DAIHATSU) bernilai nol dan tidak aktif tiap jam-nya. Solusi MILP menyatakan bahwa PLTU SELAYAR (DAIHATSU) tidak digunakan untuk mencapai kebutuhan beban. Dari gambar 4.18 secara teknis benar dikarenakan PLTU SELAYAR(DAIHATSU) tidak dalam kondisi ON dari jam ke-1 sampai ke-24 maka tidak ada apa-pun dalam rentang 0.1 dan 0.22 dan hanya memiliki nilai nol karena kondisinya selalu OFF dalam penyelesaian solusi MILP.

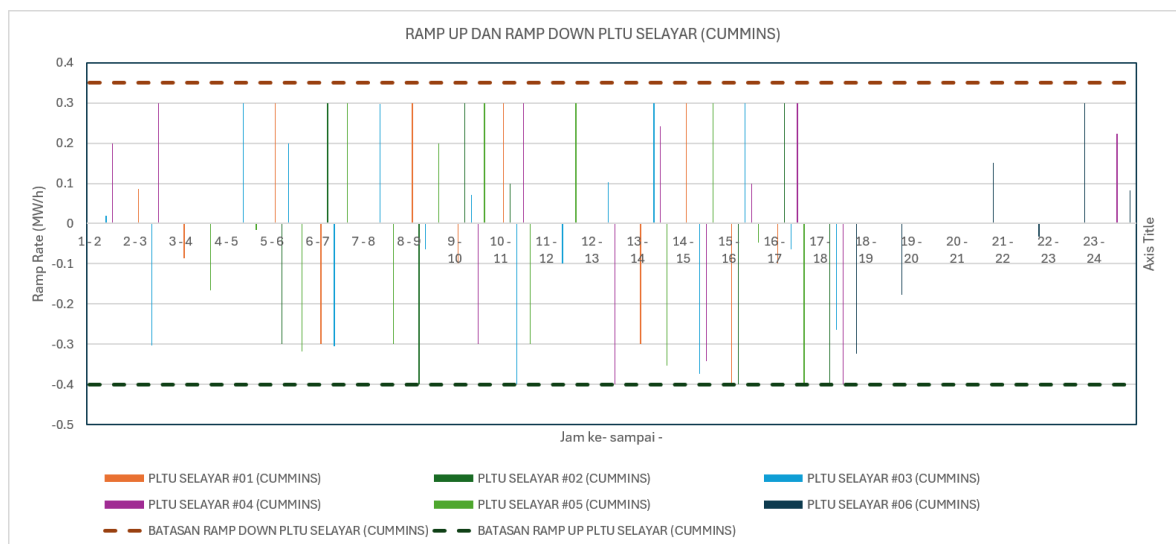
4.1.3.7 Ramp Rate

Dari hasil power dispatch program MILP memenuhi batasan ramp up dan ramp down dapat dilihat pada grafik dibawah ini untuk pembangkit PLTD dan PLTU. Untuk pembangkit PLTS dan BESS tidak memiliki batasan ramp-rate.



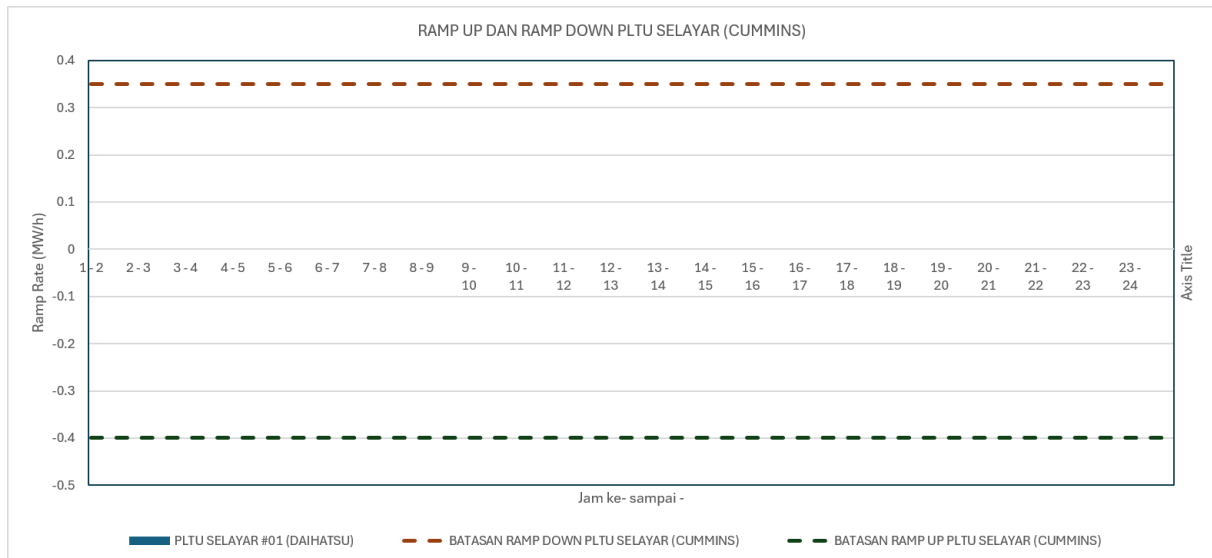
Gambar 4.19 Batasan *Ramp Rate* PLTD Selayar (DEUTZ)

Dari gambar 4.19 PLTD Selayar (DEUTZ) memenuhi batasan ramp rate nya dimana perbedaan daya dispatch pada tiap pembangkit tiap jamnya memenuhi nilai ramp up atau ramp down nilai batasan ramp rate yaitu berada di rentang -0.35 dan 0.35 MW/h.



Gambar 4.20 Batasan *Ramp Rate* PLTU Selayar (CUMMINS)

Hasil dispatch tiap jam pembangkit memenuhi batasan ramp up dan ramp down selama kurva berada di dalam batasan garis yang konstan maka pembangkit tersebut memenuhi batasan ramp rate nya yaitu 0.35 dan -0.4 MW/h

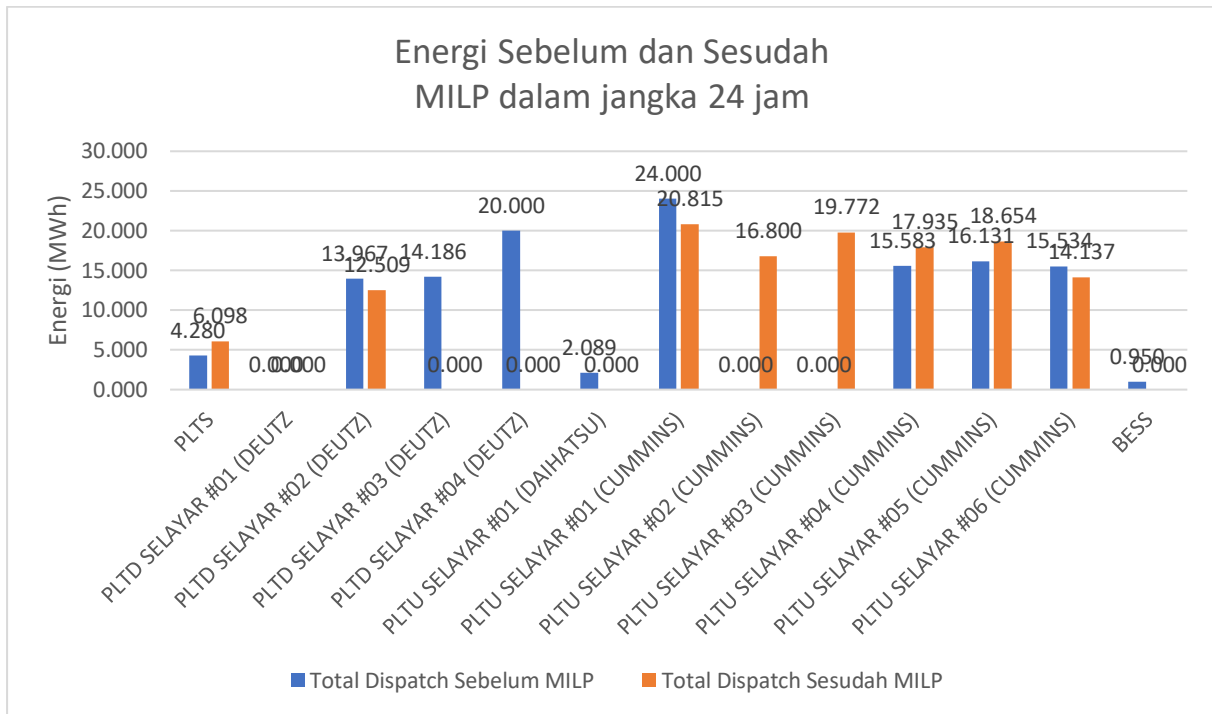


Gambar 4.21 Batasan *Ramp Rate* PLTU Selayar (DAIHATSU)

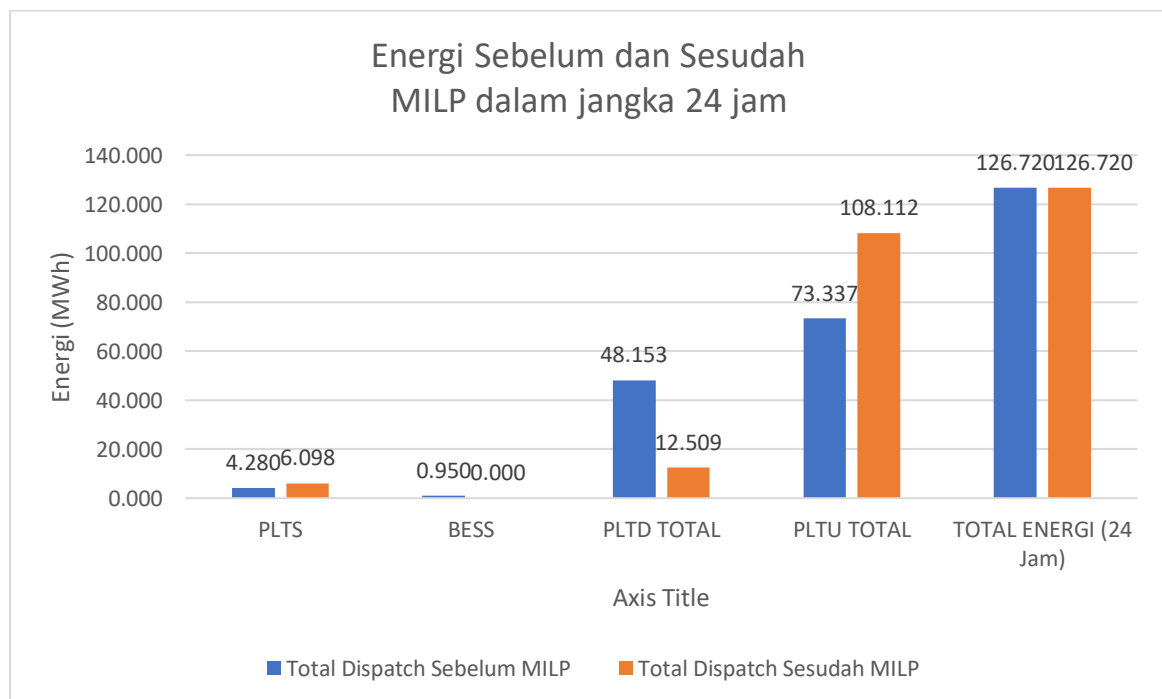
Hasil dispatch tiap jam dalam pembangkit PLTU Selayar (DAIHATSU) memenuhi batasannya yaitu 0.1 dan -0.1. PLTU Selayar (DAIHATSU) tidak memiliki perbedaan dispatch tiap jam-nya karena kondisi PLTU Selayar (DAIHATSU) selama tiap jam-nya tidak menyala atau OFF.

4.1.3.8 Energi Pembangkit dalam 1 hari

Dengan bantuan MILP dapat juga dilihat perbandingan energi yang dihasilkan tiap pembangkit dalam 1 hari sebelum dan sesudah MILP:

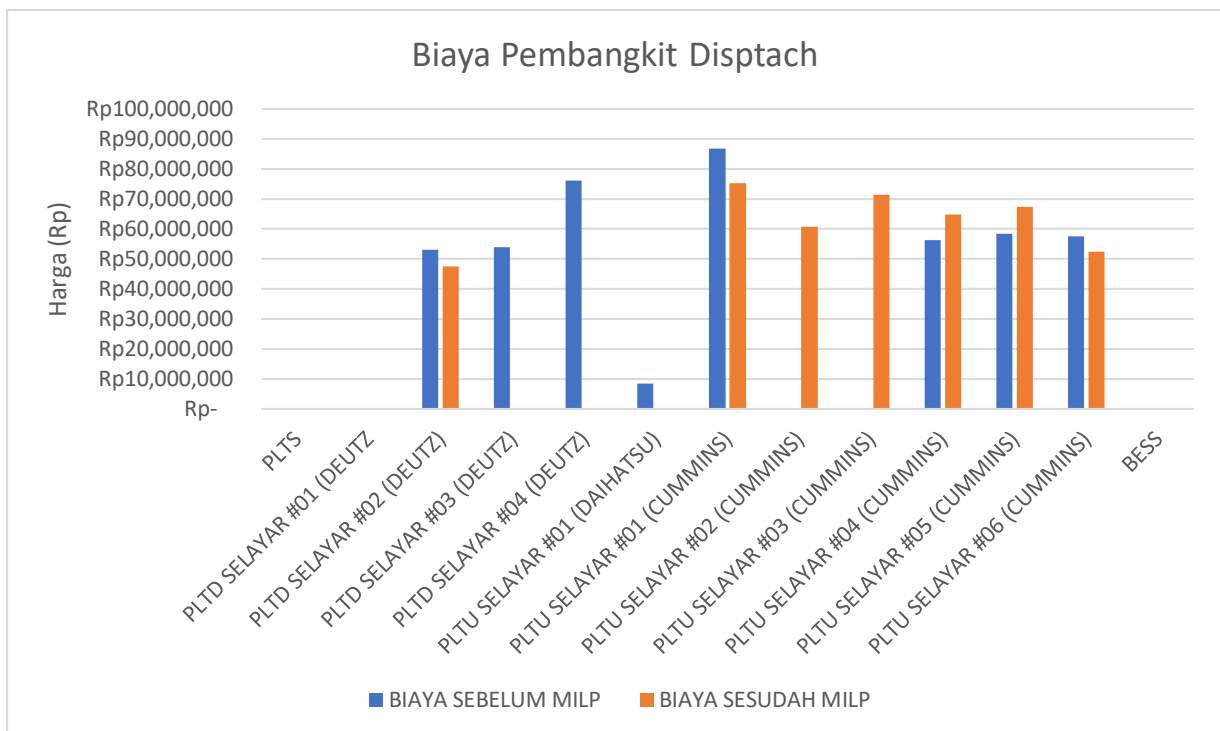


Gambar 4.22 Jumlah Energi yang dihasilkan tiap pembangkit sebelum dan Sesudah MILP

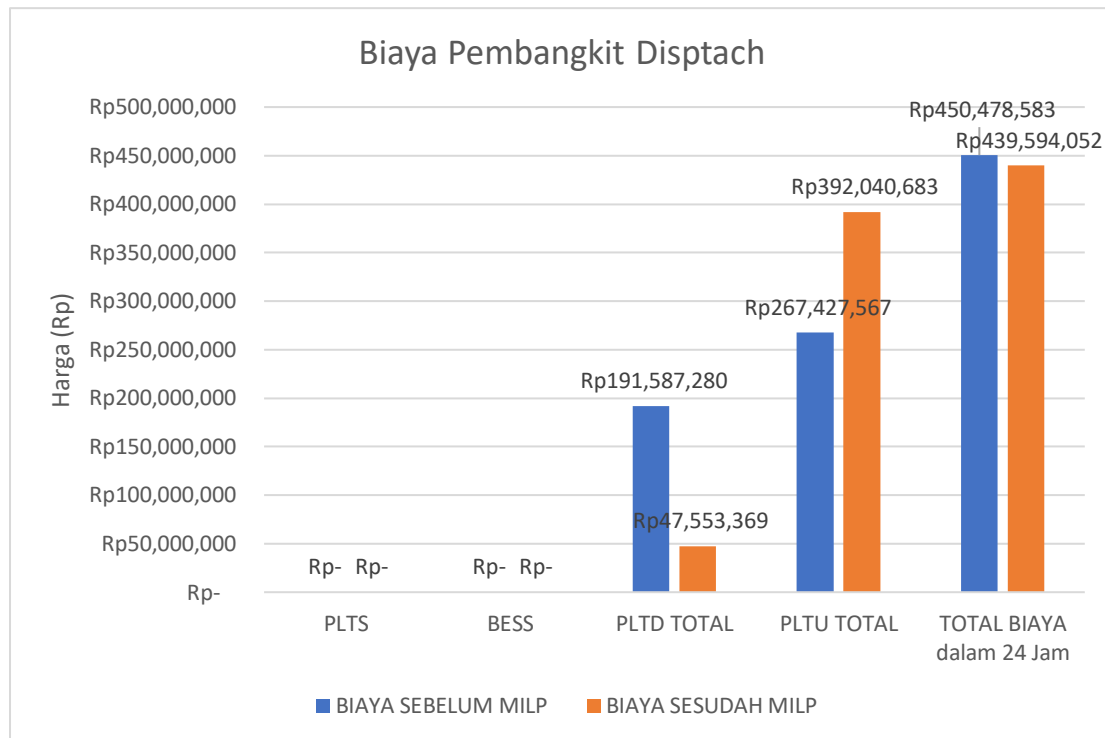


Gambar 4.23 Jumlah Energi yang dihaikan tiap tipe Pembangkit sebelum dan sesudah MILP

Dari gambar 4.23 dan gambar 4.24 dapat dilihat sebelum MILP penggunaan pembangkit PLTS lebih kecil daripada setelah MILP. Dalam kolom BESS perubahan energi BESS sebelum MILP mengalami perubahan sebesar 0.95 MWh dalam 1 hari, sementara setelah MILP kondisi Energi BESS tidak Mengalami perubahan. Untuk pembangkit konvensional PLTU dan PLTD, sebelum MILP pembangkit PLTD yang paling banyak menghasilkan daya, sedangkan setelah MILP PLTU yang paling banyak menghasilkan daya. Ini dikarenakan di fungsi objektif yang diprioritaskan adalah pembangkit yang memiliki biaya terendah yaitu PLTU. Maka itu penggunaan PLTD menurun setelah digunakan MILP untuk energi dispatch, dari pemakaian 48.153 MWh menjadi 12.509 MWh. Walaupun pembangkit sebelum MILP tetap memenuhi daya beban tetapi masih tidak efisien yang dapat menyebabkan penambahan biaya pembangkit, pembangkit PLTS yang tidak dapat mencapai potensinya dan kelalaian penggunaan BESS yang dapat menyebabkan tidak dapat digunakan saat diperlukan. Dari hasil energi yang dibangkitkan dapat ditemukan juga perbedaan harga sebelum dan sesudah MILP.



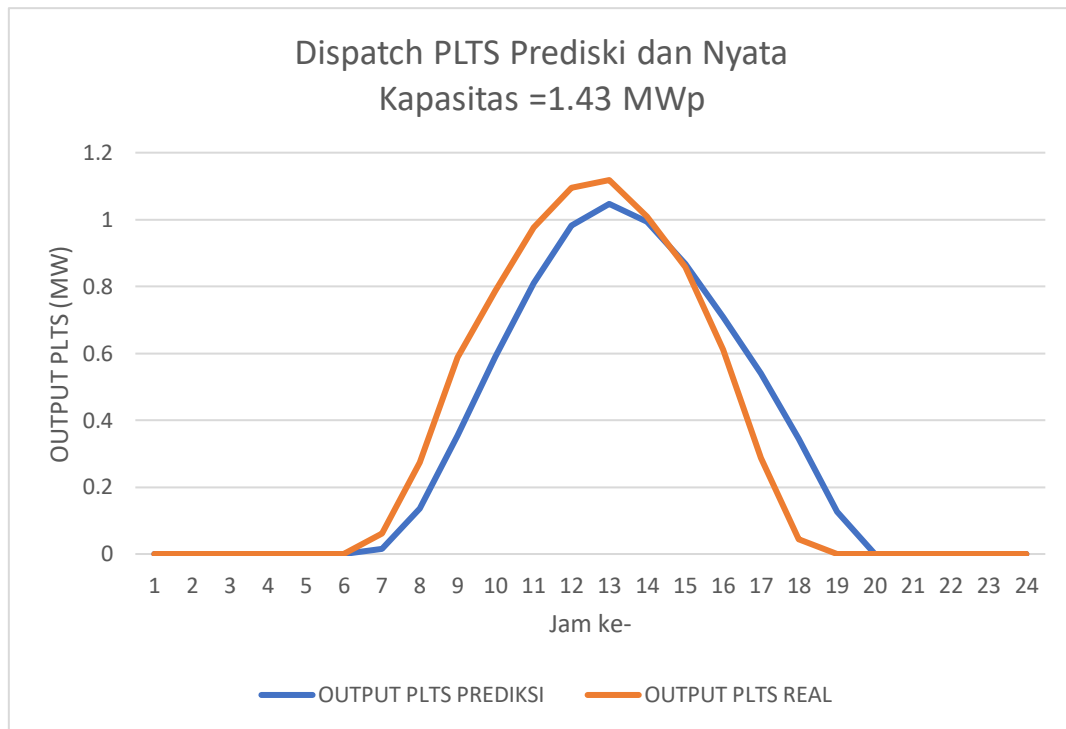
Gambar 4.24 Perbandingan Biaya sebelum dan sesudah MILP Tiap Pembangkit dalam 1 Hari



Gambar 4.25 Perbandingan Biaya Energi sebelum dan Sesudah MILP berdasarkan tipe dalam 1 jam

Dari gambar 4.24 dapat terlihat biaya-biaya yang dihasilkan tiap pembangkit dalam 1 hari. PLTS bernilai nol karena sumber-nya adalah energi solar, BESS tidak ada dikarenakan digunakan sebagai alat menyimpan dan mengeluarkan daya bukan menghasilkan daya. Dimana PLTU dan PLTD memiliki biaya pembangkitan dikarenakan bahan bakar pembangkit-nya. PLTU berbahan bakar batu bara dan PLTD berbahan bakar Diesel. Hasil biaya diatas tidak dengan start-up Cost dikarenakan hasil dari kedua dispatch di PLTU sebelum dan sesudah MILP menyala dari awal sampai akhir dan tidak ada yang berhenti dan memulai di saat jam tertentu. Dapat dilihat pada gambar 4.25 bahwa dengan MILP total Biaya lebih murah walaupun PLTU setelah MILP biaya-nya semakin mahal tetapi ditanggulangi dengan biaya PLTD yang lebih sedikit dari Sebelum MILP. Sebelum MILP pembangkit-pembangkit menghasilkan biaya sebesar Rp450,478,583 dalam 24 jam (1 hari) dan setelah MILP menghasilkan biaya sebesar Rp439,594,052 dalam 24 jam (1 hari)

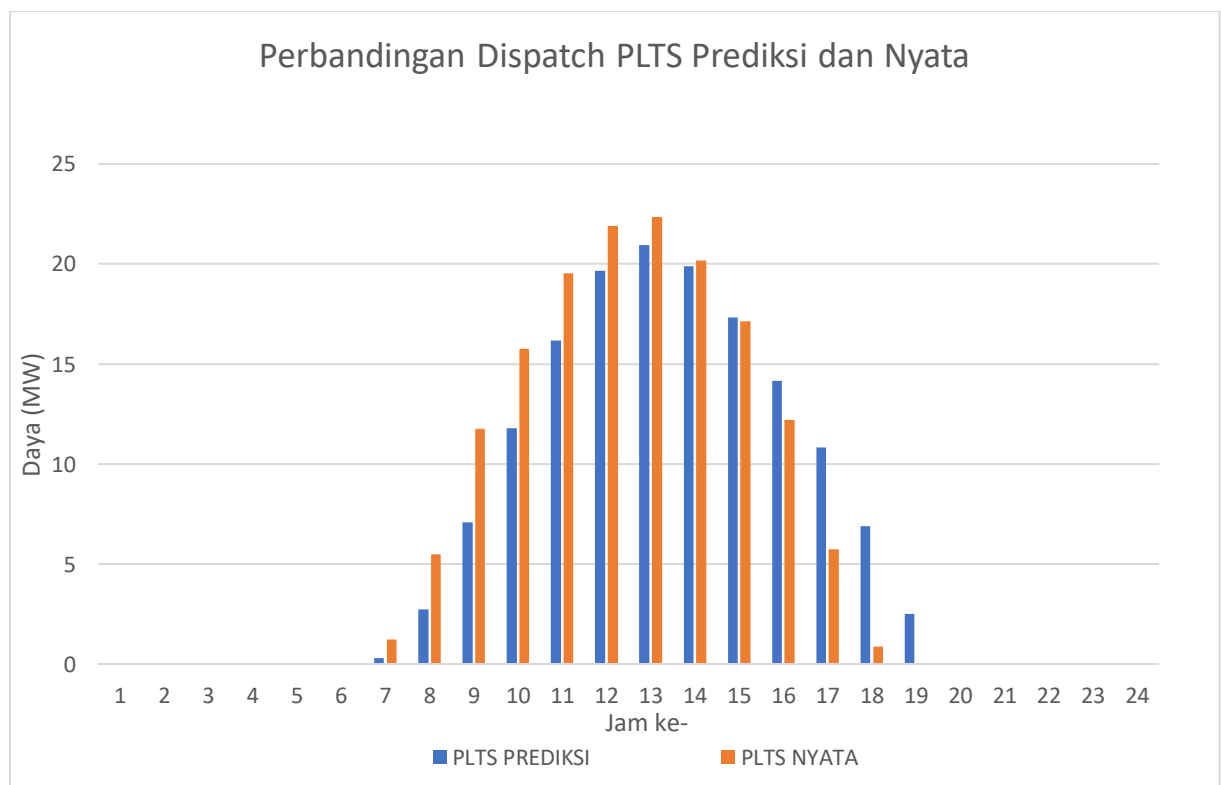
4.1.4 Perbandingan Hasil Unit Commitment menggunakan Daya Nyata dan Daya Prediksi
 Dari sub-bab 4.1.3. dinyatakan bahwa hasil dispatch, status on/off dan Start-up Variabel memenuhi semua batasan/ constrain yang diterapkan dalam dataset pembangkit. Untuk selanjutnya adalah membandingkan hasil dispatch pembangkit lain jika diberi input Daya Nyata dan daya yang diprediksi pada algoritma MILP. Dibawah ini adalah daya nyata yang dihasilkan dan yang diprediksi pada tanggal 20 April 2026 di percobaan tahap ke-2 Ramalan daya dan yang sudah di scale up untuk menyesuaikan dengan dataset pembangkit tabel 3.14 yang digunakan dalam simulasi di sub-bab 4.1.3



Gambar 4.26 Kurva Daya nyata dan Daya Prediksi tanggal 20 April 2026 Upscaled

4.1.4.1 Perbandingan Dispatch PLTS

Dikarenakan keakuratan hasil model prediksi tidak 100% maka hasil dari model tidak sama dengan hasil nyata-nya maka itu akan berdampak pada hasil hasil dispatch oleh program MILP. Dibawah ini adalah gambar grafik perbandingan jumlah daya yang seharusnya di dispatch tiap jam dan yang dipredikisikan di dispatch tiap jam .

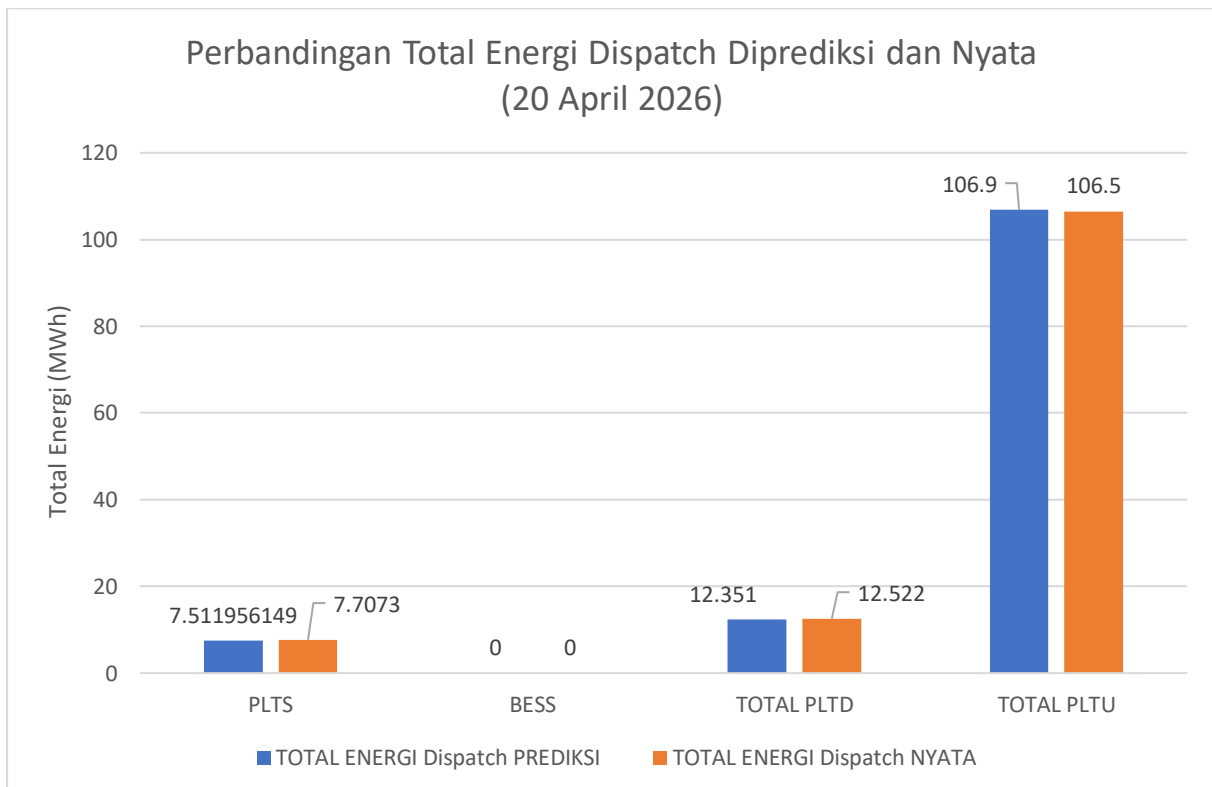


Gambar 4.27 Perbandingan Prediksi Daya PLTS tanggal 20 April 2026 dengan Daya Asli yang terekam pada tanggal 20 April 2026

Dari gambar 4.26 dapat dilihat juga dari gambar 4.25, program MILP tetap menggunakan semua daya yang diprediksi dan juga daya yang asli. Dari gambar 4.26 dapat terlihat ada daya yang diprediksi dibawah daya nyata seperti pada jam ke-7 sampai jam ke-14. Dan juga ada prediksi dimana hasilnya diatas daya yang nyata-nya seperti, pada jam ke-15 sampai ke-19. Dari nilai-nilai yang ditunjukkan pada gambar 4.26 juga berdampak pada efek penjadwalan yang diprediksi yang akhir-nya berbeda dengan penjadwalan yang seharusnya. PLTS diprediksi menghasilkan energi sebanyak 7.512 MWh kenyataannya menghasilkan dalam sehari 7.707 MWh. Jadi besar error prediksi jumlah energi adalah 0.195 MWh yang berarti 2.53% berbeda dari asli-nya.

4.1.4.2 Jumlah Energi perediksi dan Energi Nyata yang dibangkitkan dalam 1 hari

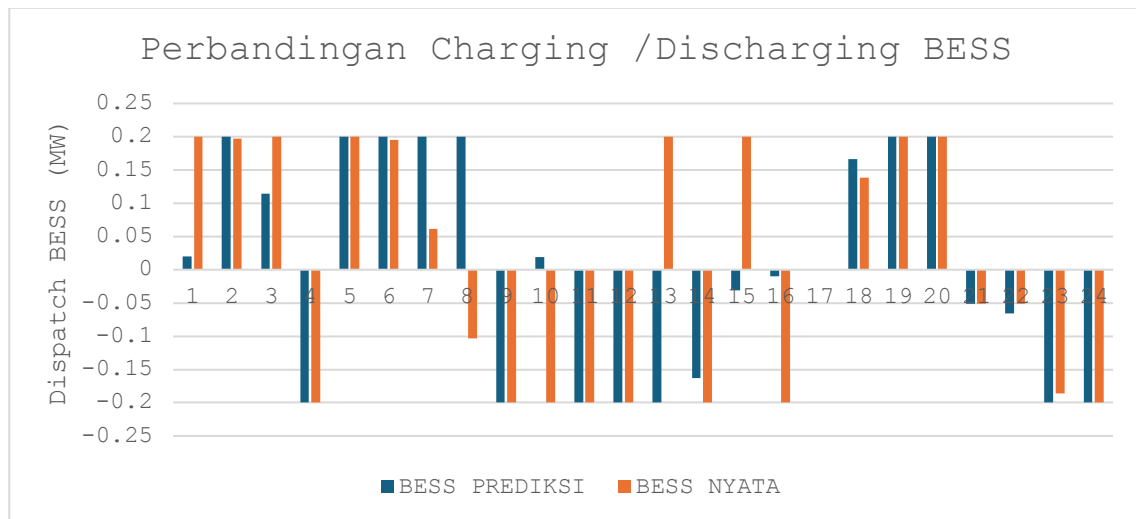
Dalam gambar 4.26 didapatkan jumlah energi yang seharusnya dibangkitkan oleh PLTS dan juga jumlah energi yang diprediksi dibangkitkan oleh PLTS dalam 1 hari. Jumlah energi tersebut juga memiliki dampak pada jumlah energi yang dibangkitkan oleh pembangkit PLTD dan PLTU pada sistem. Dapat dilihat pada gambar berikut:



Gambar 4.28 Perbandingan energi yang dihasilkan pembangkit PLTS, BESS, PLTD, dan PLTU pada tanggal 20 April 2026

Dapat dilihat dari gambar 4.27 Jumlah energi yang diprediksikan oleh PLTS adalah 7.512 MWh tetapi sebenarnya PLTS menghasilkan energi sebesar 7.707 MWh pada tanggal tersebut. Karena itu, diprediksikan PLTD menghasilkan energi sebesar 12.351 MWh pada hari itu, tetapi sebenarnya PLTD perlu menghasilkan 12.522 MWh. Sedangkan PLTU diprediksikan menghasilkan 106.9 MWh tetapi sebenarnya PLTU menghasilkan 106.5 kWh pada hari tersebut. BESS terlihat tidak menghasilkan energi apapun pada gambar 4.27, ini dikarenakan persamaan

3.24 dimana dalam 1 hari tersebut jumlah energi yang di discharge dan yang di charge sama sehingga BESS siap untuk hari berikutnya. Walaupun terlihat seperti itu, terdapat perbedaan Charging dan Discharging yang diprediksi dan yang sebenarnya pada hari itu.

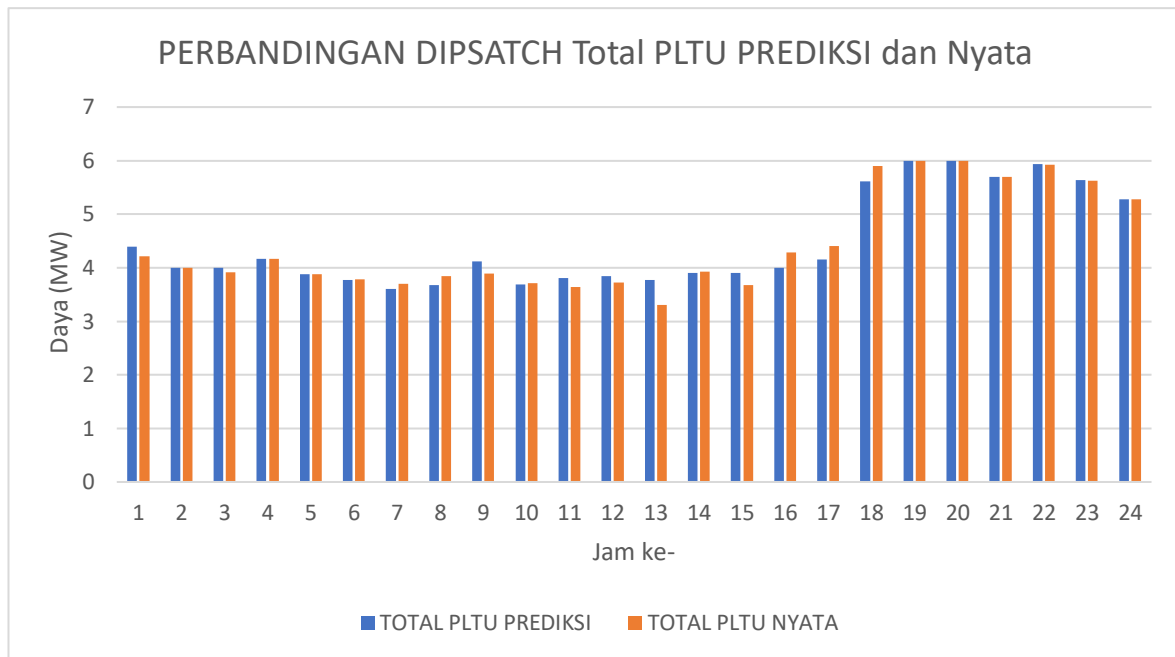


Gambar 4.29 Perbandingan Charging/Discharging yang Diprediksi dengan yang Sebenarnya

Dilihat pada gambar 4.28 BESS diprediksikan untuk melakukan charging saat jam ke-8 tetapi asli-nya pada jam ke-8 BESS ditentukan untuk melakukan discharging. Ini juga terjadi pada jam ke-10, ke-13, dan jam ke-15. Selain perbedaan jadwal charging dan discharging, ada juga perbedaan daya yang masuk dan keluar saat melakukan charging atau discharging. Pada jam ke-1 diprediksikan BESS melakukan discharging sebesar 0.02MW tetapi asli-nya melakukan charging sebesar 0.2MW. Ini juga terjadi pada jam ke-2, ke-3, ke-7, ke-14, ke-18, ke-22, dan jam ke-23. Dikarenakan pada fungsi objektif program MILP adalah biaya termurah, maka dispatch yang diprioritaskan adalah PLTS dan BESS, dikarenakan tidak memiliki biaya pembangkitan (Rp/MW), setelah PLTS dan BESS yang diprioritaskan adalah PLTU dan yang terakhir adalah PLTD, dikarenakan biaya pembangkitan PLTU lebih murah daripada PLTD.

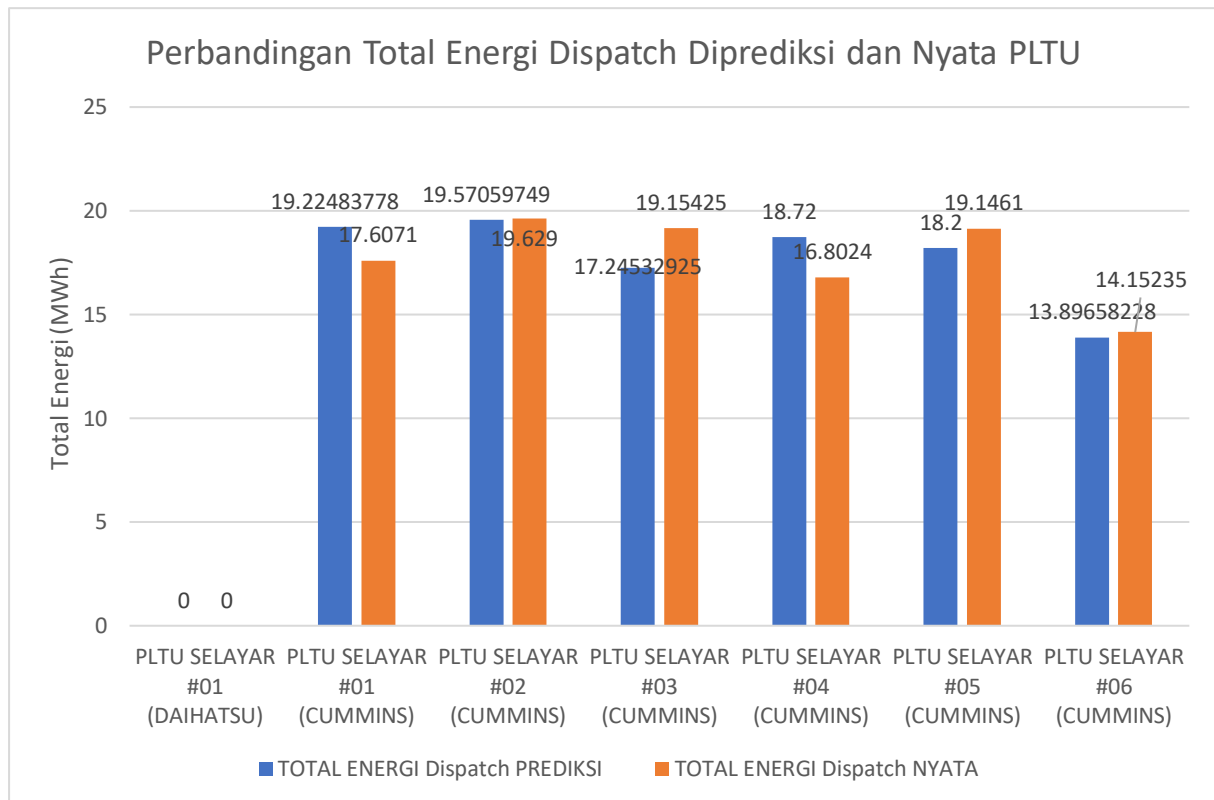
4.1.4.3 Perbandingan dispatch PLTU yang diprediksi dan yang Nyata

Pada sub-bab sebelumnya dinyatakan setelah PLTS dan BESS selanjutnya yang diprioritaskan adalah PLTU. Dalam sistem ini PLTU memiliki 7 pembangkit. Dimana 6 memiliki rating dan biaya pembangkitan yang sama dan 1 yang memiliki rating paling rendah dan biaya pembangkitan yang paling tinggi dari pembangkit lain, maka PLTU tersebut dinyatakan tidak dinyalakan untuk hari itu. Dari gambar 4.27 dinyatakan perbedaan prediksi Jumlah energi yang dihasilkan dan Jumlah energi yang asli nya dihasilkan pada hari itu adalah 0.4 MWh. Secara keseluruhan energi yang diprediksi dihasilkan dan yang sebenarnya dihasilkan tidak beda jauh. Tetapi yang terpenting adalah daya yang digunakan tiap jam-nya.



Gambar 4.30 Perbandinga Total Dispatch Daya PLTU yang Diprediksi dan yang Sebenarnya

Dari gambar 4.30 terdapat pola seperti charging dan discharging BESS, jika error dispatch BESS semakin besar di jam tertentu maka error jumlah dispatch untuk PLTU juga semakin besar. Dari gambar 4.29 dan gamabr 4.30 pada jam ke-2, jam ke-4 sampai jam ke-6, jam ke-14 dan jam ke-19 sampai ke-24 PLTU dan BESS tidak memiliki perbedaan dispatch pada jam-jam tersebut. Kecuali pada jam ke-2 dan ke-22 dimana error BESS sangat kecil maka tidak berpengaruh pada dispatch PLTS di jam ke-2 dan jam ke-22. Walaupun total disaptch PLTU memiliki pola yang berhubungan dengan BESS, pembangkit PLTU tersendiri menghasilkan jumlah energi yang berbeda pada hari itu. Dibawah ini adalah grafik energi yang dihasilkan tiap pembangkit PLTU pada hari itu

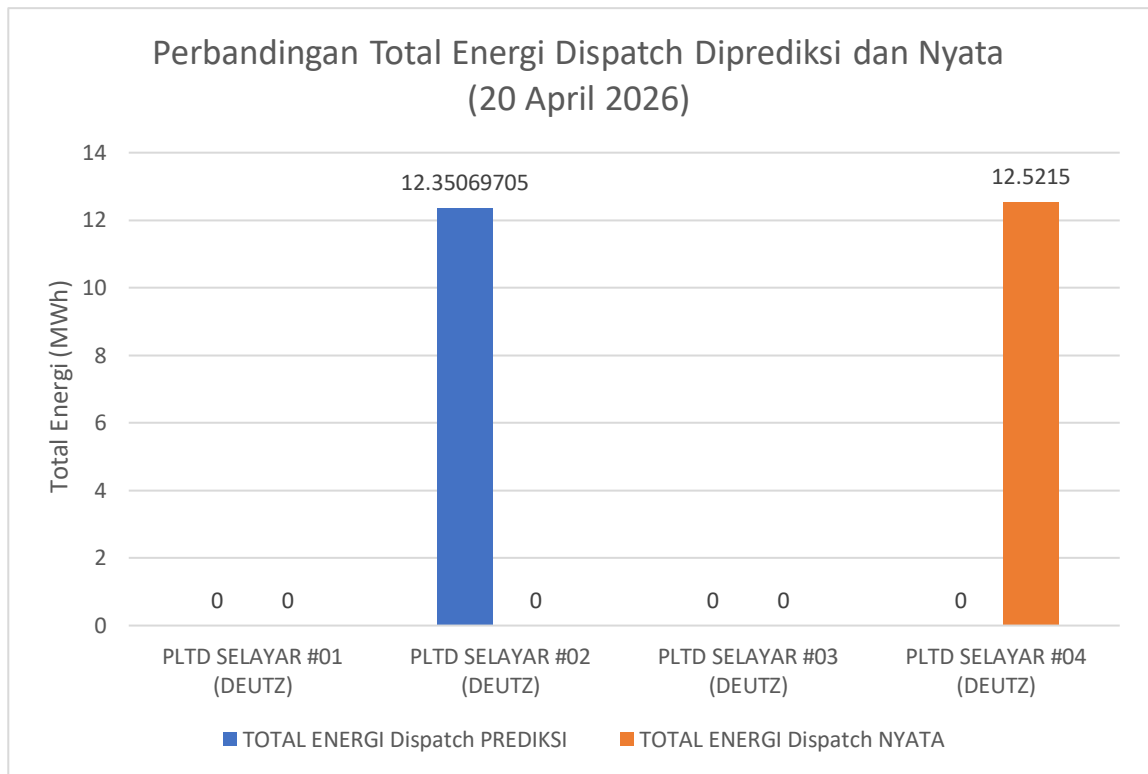


Gambar 4.31 Perbandingan Prediksi Energi yang dihasilkan tiap PLTU dengan yang Nyata

Dari gambar 4.30 dapat terlihat tiap pembangkit PLTU memiliki error yang berbeda dengan error yang paling kecil adalah PLTU SELAYAR #02 (Cummins) dan PLTU yang tidak memiliki error adalah PLTU SELAYAR #01(DAIHATSU), dikarenakan prioritas pengaktifan pembangkit tersebut adalah yang paling rendah diantara pembangkit-pembangkit lain. Untuk melihat dispatch tiap pembangkit PLTU yang diprediksi dan yang Nyata dapat dilihat di bab LAMPIRAN.

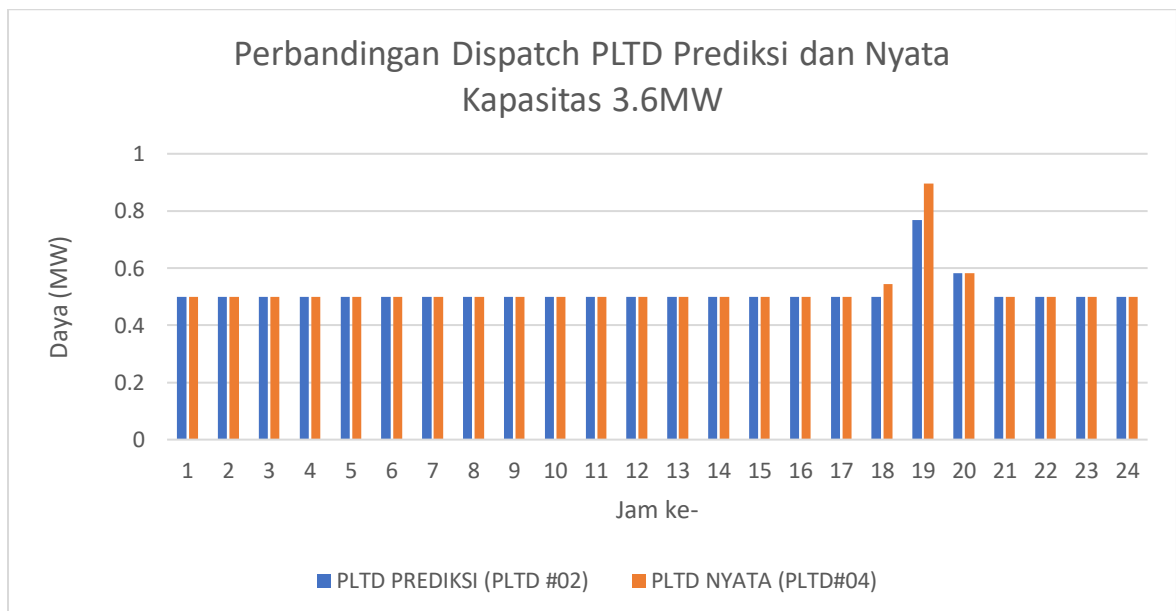
4.1.4.4 Perbandingan dispatch PLTD yang diprediksi dan yang Nyata

Setelah PLTU yang diprioritaskan adalah PLTD dimana dalam penelitian ini PLTD yang digunakan saat prediksi berbeda dengan yang asli. Tetapi Jumlah energi yang dihasilkan dalam 1 hari hampir sama dan error dispatch daya tiap jam-nya memiliki error paling kecil. Dapat dilihat pada grafik berikut



Gambar 4.32 Perbandingan Prediksi Energi yang dihasilkan tiap PLTD dengan yang Nyata

Dari gambar 4.31 program MILP menentukan saat prediksi menggunakan PLTD SELAYAR #02 dan yang nyata-nya ditentukan PLTD SELAYAR #04. Sebenarnya hasil ini berada di dalam rentang ekspektasi dikarenakan rating dan biaya pembangkitan tiap pembangkit PLTD sama. Untuk perbandingan tiap jam-nya dapat dilihat pada gambar berikut



Gambar 4.33 Perbandingan Prediksi dispatch PLTD dan yang Nyata pada tiap Jam

Dari gambar 4.32 dapat terlihat error terdapat pada jam ke-18 dan jam ke-19 saja ini dikarenakan pada jam-jam lain PLTD hanya mendispatch daya minimum-nya dikarenakan daya

beban pada jam-jam tersebut sudah terpenuhi oleh pembangkit yang memiliki prioritas yang lebih tinggi. Ini dapat menyatakan pengaruh error PLTS semakin mengecil semakin kecil prioritas suatu pembangkit, tetapi juga ada pengaruhnya terhadap beban. Jika beban besarnya menyebabkan tiap pembangkit pada tiap jam-nya aktif apakah pengaruh error PLTS pada pembangkit-pembangkit lain.

BAB 5 Kesimpulan dan Saran

5.1 Kesimpulan

5.1.1 Hasil Ramalan Daya

Dari Penelitian ini dapat disimpulkan, untuk hasil evaluasi prediksi solar irradiance memiliki akurasi sekitar 97% dengan nilai solar irradiasi nyata. Sedangkan hasil prediksi daya memiliki akurasi rata-rata 85% dengan nilai daya nyata. Disimpulkan untuk hasil peramalan daya masih dibawah nilai minimum yaitu 95%. Walaupun tidak mencapai akurasi minimum, hasil prediksi daya masih cukup bagus masih bisa digunakan sebagai input output daya PLTS tiap jam dalam pembuatan algoritma MILP.

5.1.2 Hasil MILP

1. Dari penelitian ini model MILP bekerja dengan baik, semua nilai dispatch pembangkit memenuhi semua Batasan yang ditentukan dari dataset pembangkit, termasuk daya minimum dan maksimum, Ramp up dan ramp down, uptime dan downtime, Star-up cost dan juga BESS. Juga memenuhi power balance dan energy balance.
2. Efek dari error ramalan daya mempengaruhi hasil MILP yang diprediksi dan yang sebenarnya. Pembangkit BESS yang paling terpengaruh dalam penelitian ini, dapat dilihat dari penjadwalan charging/discharging nya.

5.2 Saran

Dibawah ini adalah saran-saran yang dapat dilakukan untuk digunakan dalam penelitian selanjutnya:

1. Data solar irradiance yang nyata, yang seharusnya diambil dari lab REIDI, tidak dapat digunakan dikarenakan banyak data yang tidak terekam dikarenakan masalah teknis di pembangkit. Disarankan untuk mendapatkan data solar irradiasi pada pembangkit yang berfungsi lebih baik, dikarenakan lab pembangkit REIDI masih baru.
2. Sensitivitas hasil MILP terhadap akurasi prediksi.
3. Dibuat weather station dan menambahkan variable seperti kelembapan, dan lainnya untuk merekam data secara real dimana terekam tiap detiknya untuk dapat memprediksi daya bukan hanya tiap jam, tetapi juga dalam interval 30 menit, atau 5 menit atau bahkan 1 menit.
4. Untuk pengujian MILP jika pembangkit memiliki biaya pembangkitan yang sama dapat diubah dengan perbedaan sekecil mungkin agar dapat diprioritaskan untuk jenis pembangkit yang sama dan rating yang sama.
5. Pengujian MILP perlu diberikan batasan berapa lama pembangkit bisa dalam kondisi ON agar tidak selalu dalam kondisi ON.
6. Dalam mempelajari efek program ramalan daya terhadap program MILP diperlukan variasi dimana PLTS menangani sebagian atau lebih beban dari pada PLTU dan PLTD.

DAFTAR PUSTAKA

- Alguhi, A. A., & Al-Shaalan, A. M. (2025). LSTM-Based Prediction of Solar Irradiance and Wind Speed for Renewable Energy Systems. *Energies*, 18(17). <https://doi.org/10.3390/en18174594>
- Ang, T. Z., Salem, M., Kamarol, M., Das, H. S., Nazari, M. A., & Prabakaran, N. (2022). A comprehensive study of renewable energy sources: Classifications, challenges and suggestions. In *Energy Strategy Reviews* (Vol. 43). Elsevier Ltd. <https://doi.org/10.1016/j.esr.2022.100939>
- Cao, J. (2023). Artificial Neural Network Models for Image Recognition. In *Highlights in Science, Engineering and Technology MAME* (Vol. 2023).
- Dhaked, D. K., Dadhich, S., & Birla, D. (2023). Power output forecasting of solar photovoltaic plant using LSTM. *Green Energy and Intelligent Transportation*, 2(5). <https://doi.org/10.1016/j.geits.2023.100113>
- Durán, F., Pavón, W., & Minchala, L. I. (2024). Forecast-Based Energy Management for Optimal Energy Dispatch in a Microgrid. *Energies*, 17(2). <https://doi.org/10.3390/en17020486>
- Ghimire, S., Deo, R. C., Wang, H., Al-Musaylh, M. S., Casillas-Pérez, D., & Salcedo-Sanz, S. (2022). Stacked LSTM Sequence-to-Sequence Autoencoder with Feature Selection for Daily Solar Radiation Prediction: A Review and New Modeling Results. *Energies*, 15(3). <https://doi.org/10.3390/en15031061>
- Lim, S. C., Huh, J. H., Hong, S. H., Park, C. Y., & Kim, J. C. (2022). Solar Power Forecasting Using CNN-LSTM Hybrid Model. *Energies*, 15(21). <https://doi.org/10.3390/en15218233>
- Olatomiwa, L., Mekhilef, S., Ismail, M. S., & Moghavvemi, M. (2016). Energy management strategies in hybrid renewable energy systems: A review. In *Renewable and Sustainable Energy Reviews* (Vol. 62, pp. 821–835). Elsevier Ltd. <https://doi.org/10.1016/j.rser.2016.05.040>
- Pourasl, H. H., Barenji, R. V., & Khojastehnezhad, V. M. (2023). Solar energy status in the world: A comprehensive review. In *Energy Reports* (Vol. 10, pp. 3474–3493). Elsevier Ltd. <https://doi.org/10.1016/j.egy.2023.10.022>
- Qing, X., & Niu, Y. (2018). Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy*, 148, 461–468. <https://doi.org/10.1016/j.energy.2018.01.177>
- Silva, O. N., Kumaragamage, C., Perera, M., & SPR Arachchige, U. (2024). Solar Energy Technologies: A Complete review of the Solar system technologies. In *J. Res. Technol. Eng* (Vol. 5, Number 1).

LAMPIRAN

Kode peramalan solar irradiasi untuk simulasi PV array

<pre> import os import datetime import IPython import IPython.display import matplotlib as mpl import matplotlib.pyplot as plt import numpy as np import pandas as pd import seaborn as sns import tensorflow as tf from datetime import datetime from pytz import timezone import pytz from sklearn.metrics import mean_absolute_error from sklearn.preprocessing import MinMaxScaler mpl.rcParams['figure.figsize'] = (8, 6) mpl.rcParams['axes.grid'] = False # to display all columns pd.options.display.max_columns = None pd.options.display.max_rows = None </pre>	Import Library
<pre> import datetime yesterday = datetime.date.today() - datetime.timedelta(days=1) print(yesterday) import openmeteo_requests import pandas as pd import requests_cache from retry_requests import retry # Setup the Open-Meteo API client with cache and retry on error cache_session = requests_cache.CachedSession('.cache', expire_after = - 1) retry_session = retry(cache_session, retries = 5, backoff_factor = 0.2) openmeteo = openmeteo_requests.Client(session = retry_session) # Make sure all required weather variables are listed here # The order of variables in hourly or daily is important to assign them correctly below url = "https://archive-api.open-meteo.com/v1/archive" params = { </pre>	API Request History

```

"latitude": -7.2850234,
"longitude": 112.7961397,
"start_date": "2024-01-01",
"end_date": f"{yesterday}",
"hourly": ["rain", "cloud_cover", "temperature_2m",
"relative_humidity_2m", "wind_speed_10m", "surface_pressure",
"precipitation", "apparent_temperature", "global_tilted_irradiance"],
"timezone": "Asia/Bangkok",
"wind_speed_unit": "ms",
"timeformat": "unixtime",
"tilt": 12.251,
}
responses = openmeteo.weather_api(url, params=params)

# Process first location. Add a for-loop for multiple locations or weather
models
response = responses[0]
print(f'Coordinates: {response.Latitude()}°N {response.Longitude()}°E')
print(f'Elevation: {response.Elevation()} m asl")
print(f'Timezone:
{response.Timezone()} {response.TimezoneAbbreviation()}")
print(f'Timezone difference to GMT+0:
{response.UtcOffsetSeconds()}s")

# Process hourly data. The order of variables needs to be the same as
requested.
hourly = response.Hourly()
hourly_rain = hourly.Variables(0).ValuesAsNumpy()
hourly_cloud_cover = hourly.Variables(1).ValuesAsNumpy()
hourly_temperature_2m = hourly.Variables(2).ValuesAsNumpy()
hourly_relative_humidity_2m = hourly.Variables(3).ValuesAsNumpy()
hourly_wind_speed_10m = hourly.Variables(4).ValuesAsNumpy()
hourly_surface_pressure = hourly.Variables(5).ValuesAsNumpy()
hourly_precipitation = hourly.Variables(6).ValuesAsNumpy()
hourly_apparent_temperature = hourly.Variables(7).ValuesAsNumpy()
hourly_global_tilted_irradiance = hourly.Variables(8).ValuesAsNumpy()

hourly_data = {"date": pd.date_range(
    start = pd.to_datetime(hourly.Time(), unit = "s", utc = True),
    end = pd.to_datetime(hourly.TimeEnd(), unit = "s", utc = True),
    freq = pd.Timedelta(seconds = hourly.Interval()),
    inclusive = "left"
)}

hourly_data["temperature_2m"] = hourly_temperature_2m
hourly_data["relative_humidity_2m"] = hourly_relative_humidity_2m
hourly_data["apparent_temperature"] = hourly_apparent_temperature
hourly_data["precipitation"] = hourly_precipitation
hourly_data["rain"] = hourly_rain
hourly_data["surface_pressure"] = hourly_surface_pressure

```

<pre> hourly_data["cloud_cover"] = hourly_cloud_cover hourly_data["wind_speed_10m"] = hourly_wind_speed_10m hourly_data["global_tilted_irradiance"] = hourly_global_tilted_irradiance hourly_dataframe = pd.DataFrame(data = hourly_data) print("\nHourly data\n", hourly_dataframe) </pre>	
<pre> from datetime import datetime, timezone, timedelta ## History data_history = pd.DataFrame(hourly_dataframe) data_history['Date_Time'] = pd.to_datetime(hourly_dataframe['date'], unit='s').dt.tz_localize(None) col_to_move = data_history.pop('Date_Time') data_history = data_history.drop(columns=['date']) data_history.insert(0, 'Date_Time', col_to_move) Original_data_history = data_history.copy() data_history['Date_Time'] = data_history['Date_Time'].dt.tz_localize('UTC') data_history['Date_Time'] = data_history['Date_Time'].dt.tz_convert('Asia/Jakarta') data_history['Date_Time'] = data_history['Date_Time'].dt.tz_localize(None) ##data_history = data_history.iloc[12395:].reset_index(drop=True) Datetime = data_history['Date_Time'] ## start date forecast final_datetime = Datetime.iloc[-1] start_forecast_date = final_datetime.date() forecast_time = final_datetime.time() print(start_forecast_date, forecast_time) </pre>	Penyesuaian Data sejarah
<pre> import datetime n_day_ahead = 1 tomorrows = start_forecast_date + datetime.timedelta(days= 1 + n_day_ahead) import openmeteo_requests import pandas as pd import requests_cache from retry_requests import retry # Setup the Open-Meteo API client with cache and retry on error cache_session = requests_cache.CachedSession('.cache', expire_after = 3600) retry_session = retry(cache_session, retries = 5, backoff_factor = 0.2) openmeteo = openmeteo_requests.Client(session = retry_session) # Make sure all required weather variables are listed here </pre>	API Request Forecast

```

# The order of variables in hourly or daily is important to assign them
correctly below
url = "https://api.open-meteo.com/v1/forecast"
params = {
    "latitude": -7.2850234,
    "longitude": 112.7961397,
    "hourly": ["temperature_2m", "relative_humidity_2m",
"apparent_temperature", "precipitation", "rain", "surface_pressure",
"cloud_cover", "wind_speed_10m", "global_tilted_irradiance"],
    "timezone": "Asia/Bangkok",
    "tilt": 12.251,
    "wind_speed_unit": "ms",
    "start_date": f"{start_forecast_date}",
    "end_date": f"{tomorrows}",
}
responses = openmeteo.weather_api(url, params=params)

# Process first location. Add a for-loop for multiple locations or weather
models
response = responses[0]
print(f"Coordinates: {response.Latitude()}°N {response.Longitude()}°E")
print(f"Elevation: {response.Elevation()} m asl")
print(f"Timezone:
{response.Timezone()} {response.TimezoneAbbreviation()}")
print(f"Timezone difference to GMT+0:
{response.UtcOffsetSeconds()}s")

# Process hourly data. The order of variables needs to be the same as
requested.
hourly = response.Hourly()
hourly_temperature_2m = hourly.Variables(0).ValuesAsNumpy()
hourly_relative_humidity_2m = hourly.Variables(1).ValuesAsNumpy()
hourly_apparent_temperature = hourly.Variables(2).ValuesAsNumpy()
hourly_precipitation = hourly.Variables(3).ValuesAsNumpy()
hourly_rain = hourly.Variables(4).ValuesAsNumpy()
hourly_surface_pressure = hourly.Variables(5).ValuesAsNumpy()
hourly_cloud_cover = hourly.Variables(6).ValuesAsNumpy()
hourly_wind_speed_10m = hourly.Variables(7).ValuesAsNumpy()
hourly_global_tilted_irradiance = hourly.Variables(8).ValuesAsNumpy()

hourly_data = {"date": pd.date_range(
    start = pd.to_datetime(hourly.Time(), unit = "s", utc = True),
    end = pd.to_datetime(hourly.TimeEnd(), unit = "s", utc = True),
    freq = pd.Timedelta(seconds = hourly.Interval()),
    inclusive = "left"
)}

hourly_data["temperature_2m"] = hourly_temperature_2m
hourly_data["relative_humidity_2m"] = hourly_relative_humidity_2m
hourly_data["apparent_temperature"] = hourly_apparent_temperature

```

<pre> hourly_data["precipitation"] = hourly_precipitation hourly_data["rain"] = hourly_rain hourly_data["surface_pressure"] = hourly_surface_pressure hourly_data["cloud_cover"] = hourly_cloud_cover hourly_data["wind_speed_10m"] = hourly_wind_speed_10m hourly_data["global_tilted_irradiance"] = hourly_global_tilted_irradiance hourly_dataframe = pd.DataFrame(data = hourly_data) print("\nHourly data\n", hourly_dataframe) </pre>	
<pre> ## Forecast data_forecast = pd.DataFrame(hourly_dataframe) data_forecast['Date_Time'] = pd.to_datetime(hourly_dataframe['date'], unit='s').dt.tz_localize(None) col_to_move = data_forecast.pop('Date_Time') data_forecast = data_forecast.drop(columns=['date']) data_forecast.insert(0, 'Date_Time', col_to_move) data_forecast['Date_Time'] = data_forecast['Date_Time'].dt.tz_localize('UTC') data_forecast['Date_Time'] = data_forecast['Date_Time'].dt.tz_convert('Asia/Jakarta') data_forecast['Date_Time'] = data_forecast['Date_Time'].dt.tz_localize(None) ## filter mask = (data_forecast['Date_Time'] == final_datetime) matching_indices = data_forecast[mask].index.item() data_forecast = data_forecast.iloc[(matching_indices+1):].reset_index(drop = True) ##dikurang 23 untuk menghasilkan len(data_forecast = 24) Original_data_forecast = data_forecast.copy() datetime_forecast = data_forecast['Date_Time'] </pre>	Penyesuaian Data Forecast
<pre> # Check null values data_history.isnull().sum() data_forecast.isnull().sum() </pre>	Check for null
<pre> data_history['MonthOfYear'] = data_history['Date_Time'].dt.strftime('%m').astype(int) data_history['DayOfYear'] = data_history['Date_Time'].dt.strftime('%j').astype(int) data_history['WeekOfYear'] = data_history['Date_Time'].dt.strftime('%U').astype(int) data_history['TimeOfDay(h)'] = data_history['Date_Time'].dt.hour data_history = data_history.drop(['Date_Time'], axis=1) data_history['global_tilted_irradiance'] = data_history.pop('global_tilted_irradiance') data_forecast['MonthOfYear'] = data_forecast['Date_Time'].dt.strftime('%m').astype(int) data_forecast['DayOfYear'] = data_forecast['Date_Time'].dt.strftime('%j').astype(int) </pre>	Penyesuaian kolom Datetime

<pre> data_forecast['WeekOfYear'] = data_forecast['Date_Time'].dt.strftime('%U').astype(int) data_forecast['TimeOfDay(h)'] = data_forecast['Date_Time'].dt.hour data_forecast = data_forecast.drop(['Date_Time'], axis=1) data_forecast['global_tilted_irradiance'] = data_forecast.pop('global_tilted_irradiance') </pre>	
<pre> sns.heatmap(data_history.corr(),annot=True, cbar=False, cmap='Blues', fmt='.1f') </pre>	Heatmap correlation Matrix
<pre> ##Dropping Unnecessary Features data_history = data_history.drop(['rain', 'precipitation','MonthOfYear','DayOfYear', 'WeekOfYear', 'TimeOfDay(h)'], axis=1) data_forecast = data_forecast.drop(['rain', 'precipitation','MonthOfYear','DayOfYear', 'WeekOfYear', 'TimeOfDay(h)'], axis=1) </pre>	Drop fitur yang tidak diperlukan
<pre> days_reduced = 0 day_start = (24*days_reduced) l = len(data_history) m = len(data_forecast) train_data_forecast = data_history[day_start:] test_data_forecast = data_forecast num_features_forecast = data_forecast.shape[1] </pre>	Split data
<pre> def split_series_X(series, n_past, n_future): X = list() for window_start in range(len(series)): past_end = window_start + n_past future_end = past_end + n_future if future_end > len(series): break past = series[window_start:past_end, :] X.append(past) return np.array(X) def split_series_Y(series, n_past, n_future): y = list() for window_start in range(len(series)): past_end = window_start + n_past future_end = past_end + n_future if future_end > len(series): break future = series[past_end:future_end] y.append(future) return np.array(y) ##Scaling (MinMAx Scaler) train_forecast = train_data_forecast.copy() scalers={} for i in train_forecast.columns: scaler = MinMaxScaler(feature_range=(0,1)) </pre>	Min-Max (Normalisasi)

<pre> s_s = scaler.fit_transform(train_forecast[i].values.reshape(-1,1)) s_s=np.reshape(s_s,len(s_s)) scalers['scaler_'+ i] = scaler train_forecast[i]=s_s test_forecast = test_data_forecast.copy() ##test = test.drop(columns=['time']) for i in train_forecast.columns: scaler = scalers['scaler_'+i] s_s = scaler.transform(test_forecast[i].values.reshape(-1,1)) s_s=np.reshape(s_s,len(s_s)) scalers['scaler_'+i] = scaler test_forecast[i]=s_s train_forecast.info() test_forecast.info() X_train_forecast_window = train_forecast.drop('global_tilted_irradiance', axis=1) Y_train_forecast_window = train_forecast['global_tilted_irradiance'].to_numpy().reshape(-1, 1) X_test_forecast_window = test_forecast.drop('global_tilted_irradiance', axis=1) n_past = 24 n_future = 1 n_features_X = len(X_train_forecast_window.columns) n_features_Y = 1 X_train_forecast = split_series_X(X_train_forecast_window.values,n_past, n_future) y_train_forecast = split_series_Y(Y_train_forecast_window,n_past, n_future) X_test_forecast = split_series_X(X_test_forecast_window.values,n_past, n_future) X_train_forecast_r = X_train_forecast.reshape((X_train_forecast.shape[0], X_train_forecast.shape[1],n_features_X)) y_train_forecast_r = y_train_forecast.reshape((y_train_forecast.shape[0], y_train_forecast.shape[1], n_features_Y)) X_test_forecast_r = X_test_forecast.reshape((X_test_forecast.shape[0], X_test_forecast.shape[1],n_features_X)) </pre>	
<pre> from matplotlib.dates import MonthLocator, DateFormatter, YearLocator from keras.models import Sequential from keras.losses import MeanSquaredError from keras.metrics import RootMeanSquaredError from keras.layers import Dense, LSTM from keras.callbacks import ModelCheckpoint from keras.optimizers import Adam </pre>	LSTM Konfigurasi Model dan Training

<pre> from keras.models import load_model ##LSTM Config epochs = 50 units = 100 learning_rate = 0.001 batch_size = 32 def train_neural_network(X, y): model = Sequential() model.add(LSTM(units=units, activation='relu', input_shape=(n_past, n_features_X))) model.add(Dense(units=1)) model.compile(optimizer='adam', loss='mse') model.fit(X, y, epochs=epochs, batch_size=batch_size, validation_split=0.2) model.compile(loss=MeanSquaredError(), optimizer=Adam(learning_rate=learning_rate), metrics=[RootMeanSquaredError()]) return model model_forecast = train_neural_network(X_train_forecast_r, y_train_forecast_r) model_forecast.summary() </pre>	
<pre> predictions_forecast = model_forecast.predict(X_test_forecast_r) prediction_forecast_data = pd.Series(scaler.inverse_transform(predictions_forecast).squeeze()) prediction_forecast_data = prediction_forecast_data.clip(lower=0) </pre>	Testing trained model
<pre> ## Result Dataframe Making step_forecast = len(test_data_forecast)-len(prediction_forecast_data) Date_forecast = datetime_forecast ##finishing dataframe df_forecast = pd.DataFrame({'Date': Date_forecast}) df_forecast = df_forecast.iloc[step_forecast:] df_forecast = df_forecast.reset_index(drop = True) df_forecast['Predicted Irradiance (W/m²)'] = prediction_forecast_data df_forecast['Predicted Irradiance (W/m²)'] = df_forecast['Predicted Irradiance (W/m²)'].clip(lower=0) </pre>	Penyesuaian hasil testing
<pre> ## Plot function def plot_forecast(pred, label, period, y_label): # Convert period to datetime if it's not already period = pd.to_datetime(period) # Plotting plt.figure(figsize=(20, 10)) plt.plot(period, pred, label='Predictions', marker='o', color='blue') # Adding labels and title plt.xlabel('Date') </pre>	Plot Hasil solar irradiasi

<pre> plt.ylabel(y_label) plt.title(label) # Adding legend plt.legend() if label == "Train Data": # Format X-axis to show only years plt.gca().xaxis.set_major_locator(YearLocator()) plt.gca().xaxis.set_major_formatter(DateFormatter('%Y')) elif label == "Test Data": # Format X-axis to show only months plt.gca().xaxis.set_major_locator(MonthLocator()) plt.gca().xaxis.set_major_formatter(DateFormatter('%b %Y')) # Rotate x-axis labels for better visibility plt.gcf().autofmt_xdate() # Show the plot plt.show() plot_forecast(df_forecast['Predicted Irradiance (W/m²)'], 'Test Data', df_forecast['Date'], 'Solar Irradiance') </pre>	
<pre> ## Input data_temp = data_forecast.iloc[24:].reset_index(drop=True) Ambient_Temp = np.array(data_temp['temperature_2m']) G_Solar_Irradiance = np.array(df_forecast['Predicted Irradiance (W/m²)']) Tc_1 = ((43-20)/800) Tc_2 = Tc_1*G_Solar_Irradiance Tc_Cell_temp = Ambient_Temp + Tc_2 Vmp = 31.6 Voc = 37.9 Imp = 17.40 Isc = 18.52 Module_Efficiency = 0.21 #21% WattPeakPV = 550 ## di datasheet 550 digunakan Vmp*Imp untuk konfiguarsi koneksi seri dan paralel TempCoeff_Pmax = -0.0034 TempCoeff_Voc = -0.0025 TempCoeff_Isc = 0.0004 Delta_Temp = Tc_Cell_temp - 25 P_Dc1 = Delta_Temp * TempCoeff_Pmax P_Dc2 = 1 + P_Dc1 PowerDC_module = WattPeakPV * P_Dc2 * (G_Solar_Irradiance/1000) #Daya DC / Panel PV Modul #Sizing </pre>	Simulasi PV Array

<pre> PV_Series = 26 PV_Paralel = 2 Inverter_eff = 0.98 # 98% PV_rows = 50 Pdc = (PowerDC_module * PV_Series * PV_Paralel)/1000000 Pdc = Pdc * PV_rows df_final_Ppv = df_forecast.copy() df_final_Ppv['Pred_P_DC'] = Pdc df_final_Ppv['Pred_P_AC'] = Pdc * Inverter_eff df_final_Ppv['Pred_Solar Irradiance'] = df_forecast['Predicted Irradiance (W/m²)'] plot_forecast(df_final_Ppv['Pred_P_AC'], 'Test Data', df_final_Ppv['Date'], 'Power DC(MW)') </pre>	
<pre> df_final_Ppv.to_excel('Prediksi_Daya .xlsx', index = False, sheet_name='Prediksi_Daya') df_forecast.to_excel('Prediksi_Solar Irradinace .xlsx', index = False, sheet_name='Solar Irradiance') </pre>	Export excel
<pre> ##Data Prep import re import numpy as np import pandas as pd from scipy.linalg import block_diag from scipy.optimize import milp, Bounds, LinearConstraint data = pd.read_excel('C:/DATA TA/EMS code/DATA GEN.xlsx') data.head() ## Gen Data GenData = pd.DataFrame() GenData['Pmin'] = data['Pmin'].values GenData['Pmax'] = data['Pmax'].values GenData['Gen Cost (Rp/Mw)'] = data['Gen Cost (Rp/Mw)'].values GenData['Ramp Up'] = data['RU'].values GenData['Ramp Down'] = data['RD'].values GenData['Up Time'] = data['UpT'].values GenData['Down Time'] = data['DownT'].values GenData['STC'] = data['STC'].values GenData.drop(labels=[12,13], axis=0, inplace=True) TotaPmax = GenData['Pmax'].sum() GenData_array = np.array(GenData) GenData_array = np.vstack((GenData, np.array([-0.2, 0.2, 0, 50000, 50000, 1, 1, 0]))) GenData = pd.DataFrame(GenData_array, columns = ['Pmin', 'Pmax', 'Gen Cost (Rp/Mw)', 'Ramp Up', 'Ramp Down', 'Up Time', 'Down Time', 'STC']) ## Load Data </pre>	Import library dan dataset pembangkit

<pre> LoadData = pd.read_excel('C:/DATA TA/EMS code/Hourly LoadData.xlsx') ## PLTS Data PLTSData = pd.read_excel('C:/DATA TA/EMS code/PLTSDataT.xlsx') #PLTSData['Produksi (Mw)'] = df_final_Ppv['Pred_P_AC'] ## BESS Data # BESSData = np.array([-0.2, 0.2, 0, 5]) # Pmin, Pmax (MW), Emin, Emax nu = len(GenData_array) nl = len(LoadData) </pre>	
<pre>intcon = np.arange(((nu*nl)), (3*nu*nl), 1).tolist()</pre>	Intcon
<pre> ## Objective function HH = list() II = pd.DataFrame() for i in range(nu): HH.append(GenData.iloc[i,2]) HH = pd.DataFrame(HH) II = pd.concat([HH] * nl, ignore_index=True) ## Startup Cost JJ = list() KK = pd.DataFrame() for i in range(nu): JJ.append(GenData.iloc[i,7]) JJ = pd.DataFrame(JJ) KK = pd.concat([JJ] * nl, ignore_index=True) df_zeros = pd.DataFrame(np.zeros((nu*nl,1))) fungsi = pd.concat([II, df_zeros, KK, df_zeros], ignore_index=True) ##Turn data frames to arrays HH = np.array(HH).flatten() II = np.array(II).flatten() JJ = np.array(JJ).flatten() KK = np.array(KK) fungsi = np.array(fungsi).flatten() </pre>	Fungsi objektif
<pre> ## Max and min generation A = pd.DataFrame() B = np.empty((0,0)) UC1 = pd.DataFrame({-1, 1}) for i in range(nu): UC2 = np.vstack((-1 * GenData.iloc[i,1], GenData.iloc[i,0])) A = block_diag(A, UC1) B = block_diag(B, UC2) AA = pd.DataFrame() </pre>	Batasan output pembangkit

<pre> BB = np.empty((0,0)) BBB = B.copy() for i in range(nl): B[0,0] = -PLTSDData.iloc[i,1] B[1,0] = 0 AA = block_diag(AA, A) BB = block_diag(BB, B) B = BBB.copy() </pre>	
<pre> ## integer variables should be less than or equal to 1 CC_one_array = np.ones((nu*nl,1), dtype=int) CC = np.diag(CC_one_array.flatten()) DD = np.array(np.zeros((nu*nl, nu*nl))) ## Ramp up Limit RU = np.array(np.zeros((nu*(nl-1), nu*nl))) KRU = [] GenData_RU = GenData['Ramp Up'].values for j in range(nl-1): for i in range(nu): RU[(j)*nu+i, (j)*nu+i] = 1 RU[(j)*nu+i, (j+1)*nu+i] = -1 KRU = np.concatenate((KRU, GenData_RU), axis=0) KRU = KRU.reshape(-1,1) ## Ramp down Limit RD = np.array(np.zeros((nu*(nl-1), nu*nl))) KRD = [] GenData_RD = GenData['Ramp Down'].values for j in range(nl-1): for i in range(nu): RD[(j)*nu+i, (j)*nu+i] = -1 RD[(j)*nu+i, (j+1)*nu+i] = 1 KRD = np.concatenate((KRD, GenData_RD), axis=0) KRD = KRD.reshape(-1,1) UP = np.vstack((RU, RD)) KUP = np.vstack((KRU, KRD)) KUP_len = len(KUP) JJ = np.zeros((len(KUP), nu*nl)) </pre>	Ramp UP, Ramp Down
<pre> UT_4arr = np.array([]) UT4 = np.empty((0, nu*nl)) KUT4 = np.array([]) for i in range(nu): if GenData.iloc[i,5] != 1: UTTT4 = np.empty((0, nu*nl)) for j in range(nl-2): KUT4 = np.concatenate((KUT4, [0]), axis=0) </pre>	Uptime

<pre> for k in range(nl): UTT_4 = np.zeros(nu) if k == j: if k+1 < (nl- GenData.iloc[i,5]+1): UTT_4[i] = -GenData.iloc[i,5] else: UTT_4[i] = (-GenData.iloc[i,5]) + ((k+1) - nl + GenData.iloc[i,5]) elif k == j + 1: if k+1 < (nl - GenData.iloc[i,5]+1+1): UTT_4[i] = GenData.iloc[i,5]-1 else: UTT_4[i] = GenData.iloc[i,5] - ((k+1) - nl + GenData.iloc[i,5]) elif (k > j + 1) and (k <= GenData.iloc[i,5] + j): UTT_4[i] = -1 UT_4arr = np.hstack((UT_4arr, UTT_4)) UTTT4 = np.vstack((UTTT4, UT_4arr)) UT_4arr = np.array([]) UT4 = np.vstack((UT4, UTTT4)) KUT4 = KUT4.reshape(-1,1) AAUT4 = np.zeros((len(KUT4), nu*nl)) </pre>	
<pre> UT_5arr = np.array([]) UT5 = np.empty((0, nu*nl)) KUT5 = np.array([]) for i in range(nu): if GenData.iloc[i,6] != 1: UTTT5 = np.empty((0, nu*nl)) KT = 1 for j in range(nl-2): if j < (nl - GenData.iloc[i,6]+1): KUT5 = np.hstack((KUT5, GenData.iloc[i,6])) else: KUT5 = np.hstack((KUT5, GenData.iloc[i,6]-KT)) KT = KT + 1 for k in range(nl): UTT = np.zeros(nu) if k == j: if k < (nl- GenData.iloc[i,6]+1): UTT[i] = -GenData.iloc[i,6] </pre>	Downtime

<pre> else: UTT[i] = -GenData.iloc[i,6] + (k - (nl + GenData.iloc[i,6])) elif k == j + 1: if k < (nl - GenData.iloc[i,6]+1+1): UTT[i] = GenData.iloc[i,6]-1 else: UTT[i] = GenData.iloc[i,6]-1 - (k - (nl + GenData.iloc[i,6])) elif (k > j + 1) and (k <= GenData.iloc[i,6] + j): UTT[i] = -1 else : UTT[i] = 0 UT_5arr = np.hstack((UT_5arr, UTT)) UTTT5 = np.vstack((UTTT5, UT_5arr)) UT_5arr = np.array([]) UT5 = np.vstack((UT5, UTTT5)) UT5 = -UT5 AAUT5 = np.zeros((len(KUT5), nu*nl)) </pre>	
<pre> SUCVar_array = np.ones((1,nu*nl), dtype=int) SUCVar = -np.diag(SUCVar_array.flatten()) KonsSUCVar = np.zeros((nu*nl, nu*nl), dtype=int) UT = np.array([[]]) UT1 = np.array([[]]) UT6 = np.empty((0, nu*nl)) UT7 = np.empty((0, nu*nl)) KUT6 = np.array([]) KUT7 = np.array([]) for i in range(nu): UTTTT = np.empty((0,nu*nl)) UTT3 = np.empty((0, nu*nl)) for j in range(nl-1): KUT6 = np.concatenate((KUT6, [0]), axis=0) KUT7 = np.concatenate((KUT7, [0]), axis=0) # SA = 0 # for k in range(nl): # UTT = np.zeros((1, nu)) # if k == GenData.iloc[i,6] + j: # UTT[0,i] = GenData.iloc[i,7] # elif k >= j and k < GenData.iloc[i,6] + j: </pre>	Start Up Cost

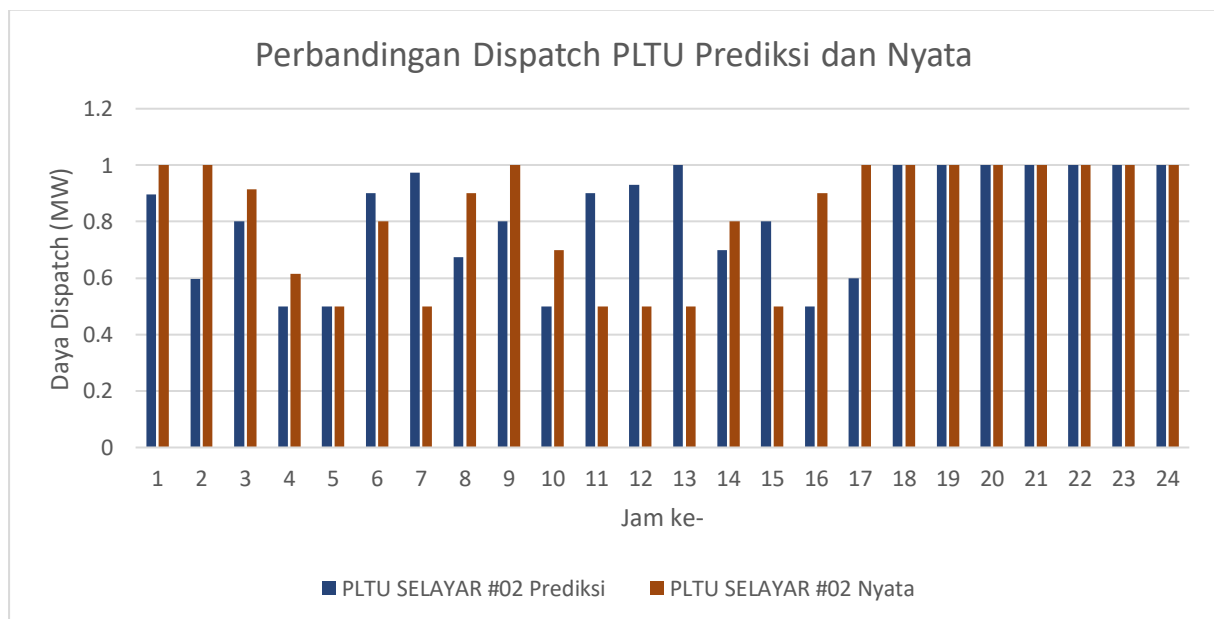
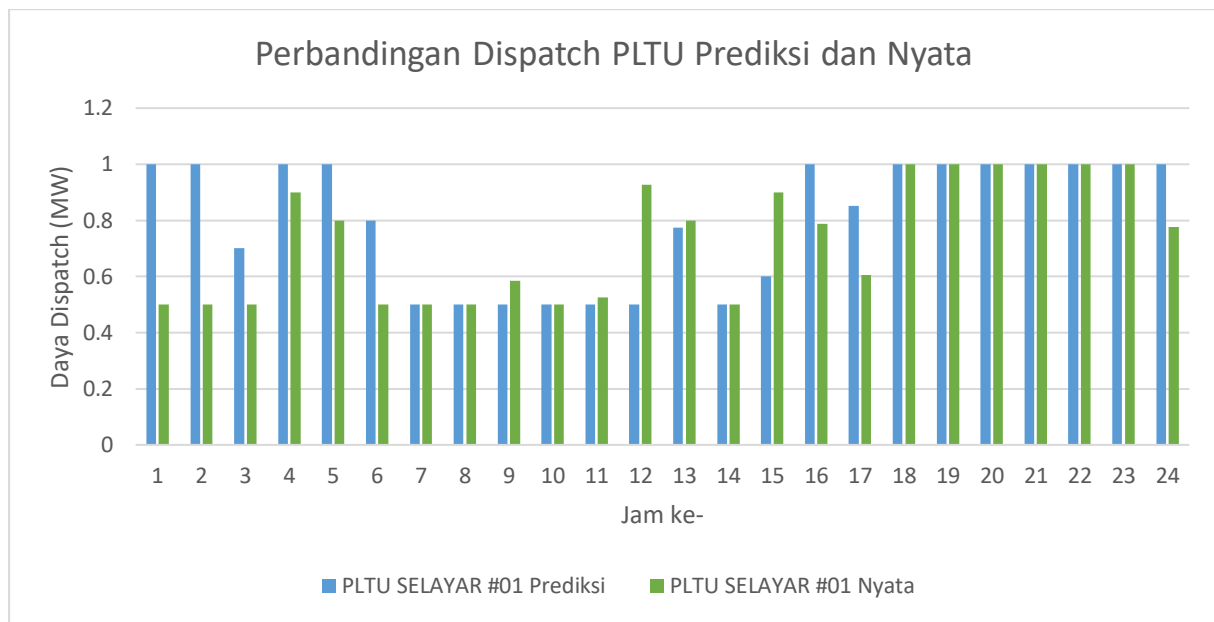
<pre> # UTT[0,i] = -GenData.iloc[i,7] # UT = np.hstack((UT, UTT)) # SA = 0 for k in range(nl): UTT = np.zeros((1, nu)) UT2 = np.zeros((1, nu)) if k == j: UTT[0,i] = -GenData.iloc[i,7] elif k == j + 1: UTT[0,i] = GenData.iloc[i,7] UT2[0,i] = -1 UT = np.hstack((UT, UTT)) UT1 = np.hstack((UT1, UT2)) UTTTT = np.vstack((UTTTTT, UT)) UTT3 = np.vstack((UTTT3, UT1)) UT = np.array([[]]) UT1 = np.array([[]]) UT6 = np.vstack((UT6, UTTTT)) UT7 = np.vstack((UT7, UTT3)) AAUT6 = np.zeros((len(KUT4), nu*nl)) </pre>	
<pre> ## BESS Matrix nbess = nu QQ = np.zeros((nl, nu*nl)) for i_bess in range(nl): print("i =",i_bess) for k_bess in range(i_bess+1): print("k =",k_bess) QQ[i_bess, (nbess*(k_bess+1)-1)] = 1 KBmin = -np.ones((nl,1)) * BESSData[2] KBmax = np.ones((nl,1)) * BESSData[3] ## SOC AWAL SOC_0 = 99 #% E_0 = (SOC_0/100) * BESSData[3] KBmin = -np.ones((nl,1)) * BESSData[2] + E_0 KBmax = np.ones((nl,1)) * BESSData[3] - E_0 </pre>	BESS
<pre> AB1 = np.hstack((AA, BB, np.zeros((len(AA), nl*nu)))) AB2 = np.hstack((DD, CC, np.zeros((len(DD), nl*nu)))) AB3 = np.hstack((UP, JJ, np.zeros((len(UP), nl*nu)))) AB4 = np.hstack((AAUT4, UT4, np.zeros((len(UT4), nl*nu)))) </pre>	Matrix A dan B inequality constrain

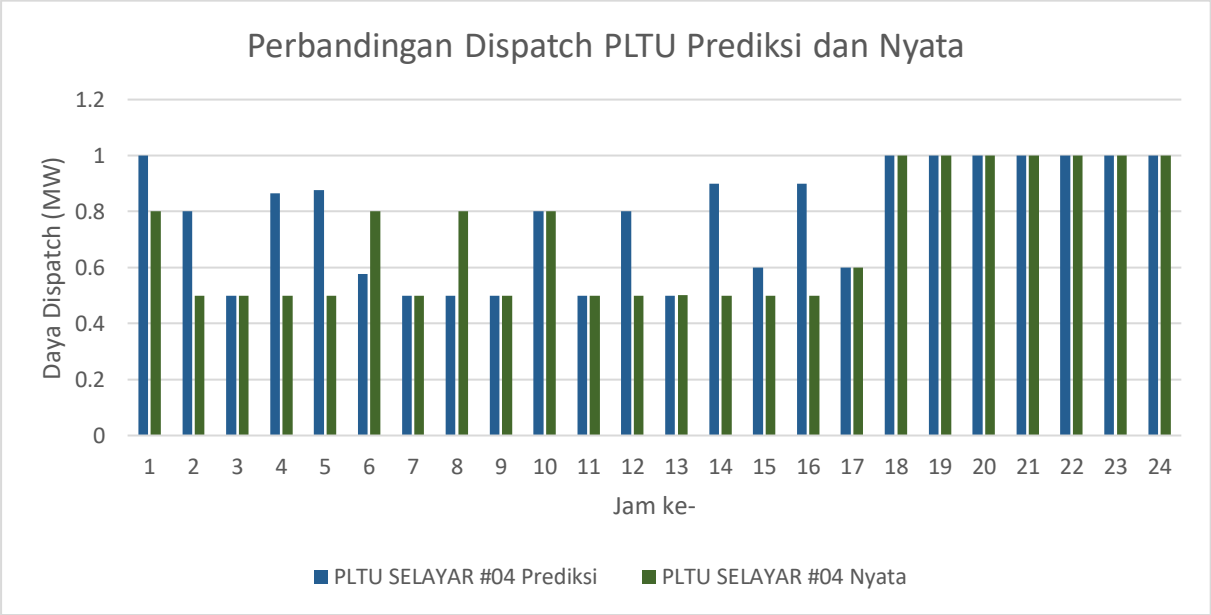
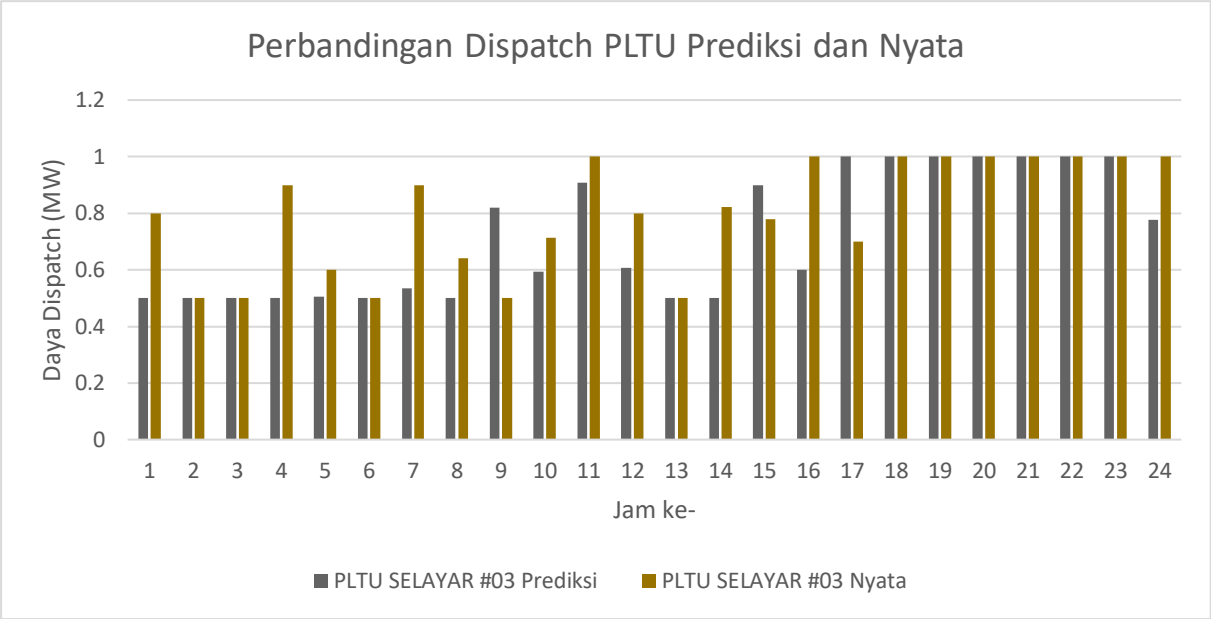
<pre> AB5 = np.hstack((AAUT5, UT5, np.zeros((len(UT5), nl*nu)))) AB6 = np.hstack((np.zeros((len(SUCVar),nl*nu)), np.zeros((len(SUCVar),nl*nu)), SUCVar)) AB7 = np.hstack((np.zeros((len(UT6), nl*nu)), UT6, UT7)) AB8 = np.hstack((QQ, np.zeros((nl, nl*nu)), np.zeros((nl, nl*nu)))) AB9 = np.hstack((-QQ, np.zeros((nl, nl*nu)), np.zeros((nl, nl*nu)))) A = np.vstack((AB1, AB2, AB3, AB4, AB5, AB6, AB7, AB8, AB9)) B = np.vstack((np.zeros((2*nu*nl, 1)), np.ones((nu*nl,1)), KUP, KUT4.reshape(-1,1), KUT5.reshape(-1,1),)) B = np.vstack((np.zeros((2*nu*nl, 1)), np.ones((nu*nl,1)), KUP, KUT4.reshape(-1,1), KUT5.reshape(-1,1), np.zeros((len(SUCVar),1)), np.zeros((len(KUT7),1)), KBmin, KBmax)) </pre>	
<pre> ## Power Balance == Equality Constraints EE = np.empty((0, 0)) C = np.ones((1, nu)) #C[1,13] = 0.9 for j in range(nl): EE = block_diag(EE, C) </pre>	Equality Contrsain (Power Balance dan BESS)

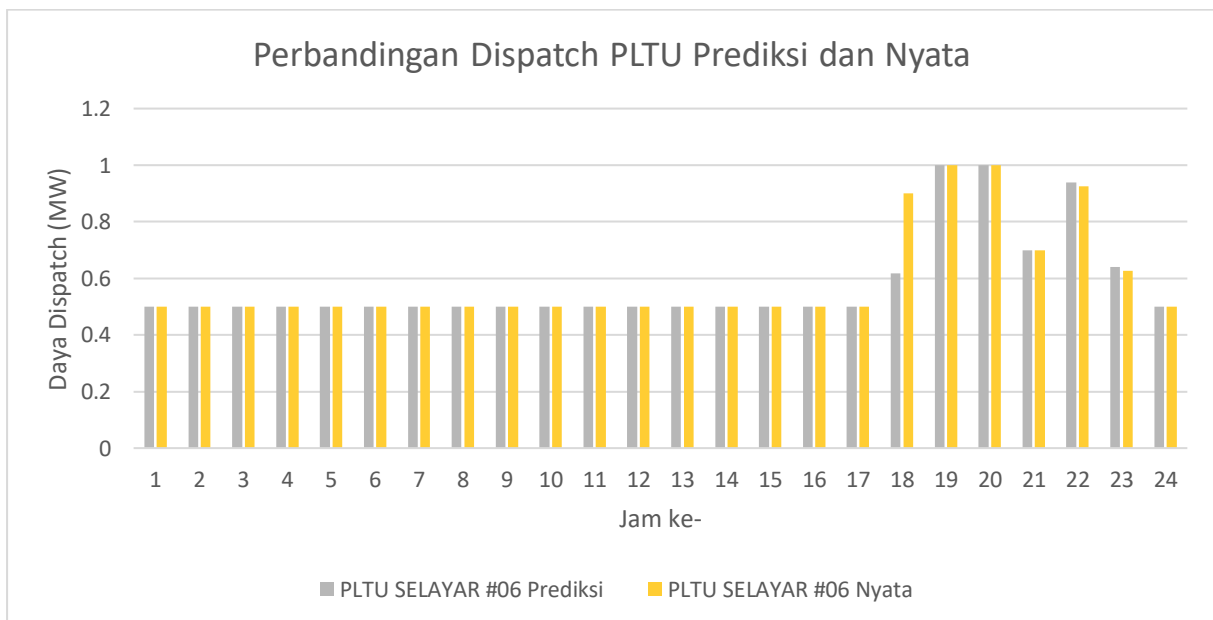
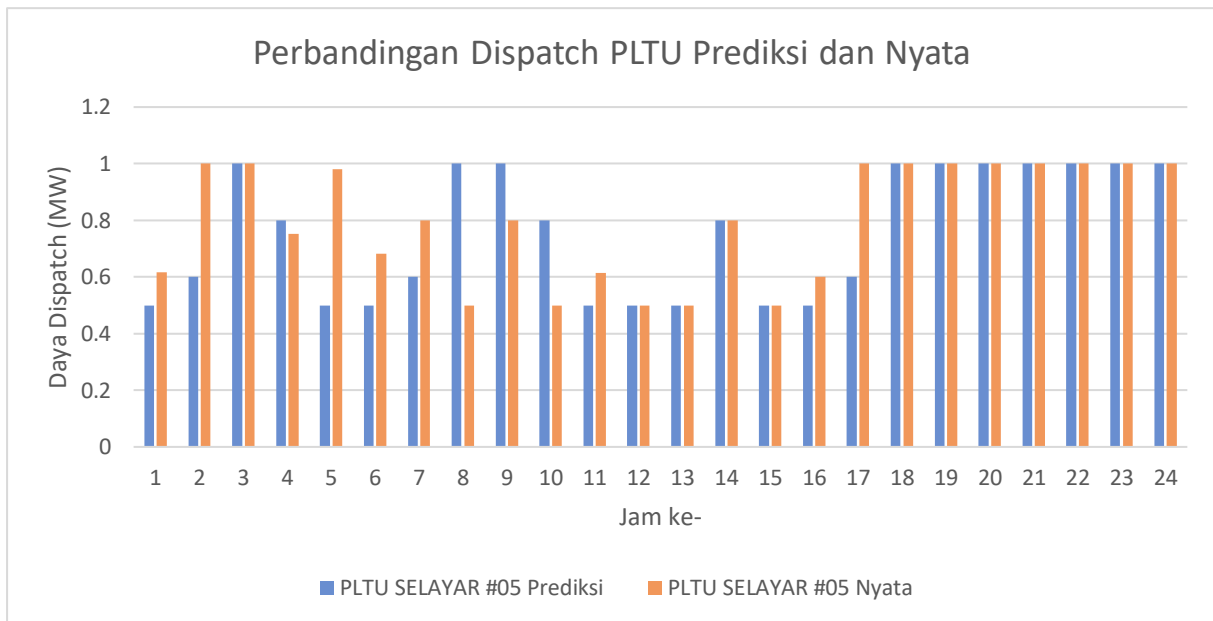
<pre> FF = EE.copy() ## Battery Equality Constrain n_days = int(nl/24) QQeq = np.empty((0, QQ.shape[1])) for j in range(n_days): BATT = QQ[(nl-1)*n_days].reshape(1, -1) QQeq = np.vstack((QQeq, BATT)) Aeq_1 = np.hstack((FF , np.zeros((len(FF), nu*nl)) , np.zeros((len(FF), nu*nl)))) Aeq_2 = np.hstack((QQeq, np.zeros((len(QQeq), nl*nu)), np.zeros((len(QQeq), nl*nu)))) Aeq = np.vstack((Aeq_1, Aeq_2)) GG = np.array([]).reshape(-1,1) for j in range(nl): GG = np.vstack((GG, np.array([[LoadData.iloc[j,1]]]))) Beq = GG </pre>	
<pre> ## Answer prep #obj function fungsi intcon = np.array(intcon) ##inequality constraint A_ineq = A_ineq B_ineq = B_ineq.flatten() #Equality Constraint Aeq = Aeq Beq = Beq.flatten() print(type(fungsi), fungsi.shape) print(type(intcon), intcon.shape) print(type(A_ineq), A_ineq.shape) print(type(B_ineq), B_ineq.shape) print(type(Aeq), Aeq.shape) print(type(Beq), Beq.shape) </pre>	Preparasi Jawaban
<pre> c = fungsi lb = -np.inf ub = np.inf bounds = Bounds(lb=lb ,ub=ub) integrality = np.hstack((np.zeros(nu*nl), np.ones(nu*nl), np.zeros(nu*nl))) ineq_constraints = LinearConstraint(A=A_ineq, lb=-np.inf, ub=B_ineq) eq_constraints = LinearConstraint(A=Aeq, lb=Beq, ub=Beq) </pre>	MILP

<pre> constraints = [ineq_constraints, eq_constraints] res = milp(c=c, bounds=bounds, integrality=integrality, constraints=constraints) X = res.x print("EXit Value :", res.status) print("Optimal value:", res.fun) print("Optimal solution:", res.x) Jawab = X.reshape(nl*4,nu) Jawab_T = np.transpose(Jawab) Jawab_df = pd.DataFrame(Jawab) </pre>	
<pre> #Power Dispatch Gen_Dispatch = pd.DataFrame() for i in range(nu) : Gen_Dispatch[f'GEN No.{i}'] = Jawab_df.iloc[0:nl, i] Gen_Dispatch.to_excel('Output_Disaptch.xlsx', index = False, sheet_name='Dispatch_Pembangkit') ## ON/OFF STATUS GENERATOR Gen_Status = pd.DataFrame() for i in range(nu) : status = Jawab_df.iloc[nl:nl*2, i] Gen_Status[f'GEN STATUS No.{i}'] = status.reset_index(drop=True) Gen_Status.to_excel('Output_Status.xlsx', index = False, sheet_name='Status_Pembangkit') # Start-Up Varibel Generator Gen_SUV = pd.DataFrame() for i in range(nu) : suv = Jawab_df.iloc[nl*2:nl*3, i] Gen_SUV[f'GEN STATUS No.{i}'] = suv.reset_index(drop=True) Gen_SUV.to_excel('Output_SUV.xlsx', index = False, sheet_name='SUV_Pembangkit') </pre>	Excel Export

Perbandingan Prediksi dispatch tiap PLTU dan yang Nyata pada tiap Jam







	DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS INSTITUT TEKNOLOGI SEPULUH NOPEMBER	Nama Dokumen	:	FORM PERNYATAAN PENGGUNAAN AI
		No. Dokumen	:	001/ST_AI/DTE/V/2025
		Halaman	:	1 dari 2

PERNYATAAN KODE ETIK PENGGUNAAN GENERATIVE AI

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

Saya yang bertanda tangan di bawah ini:

I the undersigned:

Nama Lengkap / *Full name* : Nathanael Dylan Wanasida
 NRP / *Student ID* : 5022201080
 Program Studi / *Study Program* : Sarjana Teknik Elektro
 Judul / *Title* :
 Pembimbing / *Supervisors* : 1. Dr.Eng. Rony Seto Wibowo, S.T., M.T
 2. Dedet Candra Riawan, S.T., M.Eng., Ph.D

Menyatakan dalam pengerjaan Tugas Akhir/Tesis/Disertasi* dengan judul tersebut di atas:

State that during the finalization of my final project/thesis/dissertation with the title above:*

No.	Pernyataan / <i>Statement</i>	Checklist (☒)
1	Hanya menggunakan Generative AI sebagai alat bantu untuk memperbaiki tata bahasa pada Tugas Akhir / Thesis / Disertasi saya. Ini tidak digunakan untuk membuat keseluruhan isi pekerjaan saya. <i>Use Generative AI only as a tool to improve the readability or language of the text in my Final Project / Thesis / Dissertation*. It is not used to generate a complete text of my work.</i>	✓
2	Telah memeriksa dan/atau memperbaiki seluruh bagian dari pekerjaan saya yang dibantu oleh Generative AI agar sesuai dengan baku mutu penulisan karya ilmiah <i>Has checked and/or refined all parts of my work that is assisted by Generative AI in such a way that it complies with the standard of academic writing</i>	✓
3	Tidak menggunakan Generative AI untuk pembuatan data primer, grafik dan/atau tabel pada pekerjaan saya. <i>Do not use Generative AI to generate primary data, figure and/or table in my works.</i>	✓

	DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS INSTITUT TEKNOLOGI SEPULUH NOPEMBER	Nama Dokumen	:	FORM PERNYATAAN PENGGUNAAN AI
		No. Dokumen	:	001/ST_AI/DTE/V/2025
		Halaman	:	2 dari 2

No.	Pernyataan / <i>Statement</i>	Checklist (☒)
4	Penulis telah memberikan atribusi/pengakuan terhadap alat AI yang digunakan, secara rinci pada suatu bagian pada lampiran. <i>The author has acknowledged the use of Generative AI in any part of the work in the specific appendix page.</i>	✓
5	Buku tugas akhir / Tesis / Disertasi* telah mematuhi Baku Mutu Akademik ITS terkait keaslian karya dan etika penulisan. <i>The final project/thesis/dissertation* has complied with the Academic Standard of ITS related to the originality of scientific work and scientific writing code of conduct</i>	✓
6	Penulis memastikan tidak ada plagiarisme, termasuk yang berasal dari penggunaan AI. <i>The authors has ensured that there is plagiarism issue in the work, including any part that is generated by AI.</i>	✓

Catatan/note: * Pilih salah satu yang sesuai

Tempat/ <i>Place</i>	: Apartemen Educity Tower Harvard Unit 3010 Kalisari, Mulyorejo, Surabaya, Jawa Timur 60112
Tanggal/ <i>Date</i>	: 15 Juni 2026
Nama Lengkap/ Full Name	: Nathanael Dylan Wanasida
Tanda tangan/ Signature	: 

BIODATA PENULIS



Penulis dilahirkan di Madiun, 07 April 2024, merupakan anak pertama dari 3 bersaudara. Setelah lulus dari SMAN tahun 2020, Penulis meneruskan pendidikan di Institut Teknologi Sepuluh Nopember diterima di Departemen Teknik ELEKTRO FTEIC - ITS pada tahun 2020 dan terdaftar dengan NRP 5022201080.

Selama di Institut Teknologi Sepuluh Nopember Penulis sempat aktif dan bekerja di STP OTOMOTIF ITS sebagai teknisi yang terlibat dalam pembuatan bis listrik.