

TUGAS AKHIR - SS234862

**MODEL PREDIKSI SUHU MAKSIMUM DAN MINIMUM
INDONESIA-*NUMERICAL WEATHER PREDICTION*
MENGUNAKAN *SUPPORT VECTOR REGRESSION* DAN
*LONG SHORT-TERM MEMORY***

HAIKAL MAULADANA ICHWANI

NRP 5003211142

Dosen Pembimbing

Dr. Sutikno, M.Si.

NIP 19710313 199702 1 001

Program Studi Sarjana Statistika

Departemen Statistika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2025



TUGAS AKHIR - SS234862

**MODEL PREDIKSI SUHU MAKSIMUM DAN MINIMUM
INDONESIA-NUMERICAL WEATHER PREDICTION
MENGUNAKAN *SUPPORT VECTOR REGRESSION* DAN
*LONG SHORT-TERM MEMORY***

HAIKAL MAULADANA ICHWANI

NRP 5003211142

Dosen Pembimbing

Dr. Sutikno, M.Si.

NIP 19710313 199702 1 001

Program Studi Sarjana Statistika

Departemen Statistika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2025



FINAL PROJECT - SS234862

**PREDICTION MODEL MAXIMUM AND MINIMUM
TEMPERATURE INDONESIAN-NUMERICAL WEATHER
PREDICTION USING SUPPORT VECTOR REGRESSION
AND LONG SHORT-TERM MEMORY**

HAIKAL MAULADANA ICHWANI

NRP 5003211142

Advisor

Dr. Sutikno, M.Si.

NIP 19710313 199702 1 001

Bachelor Program of Statistics

Department of Statistics

Faculty of Science and Data

Analytics Institut Teknologi Sepuluh

Nopember Surabaya

2025

LEMBAR PENGESAHAN

MODEL PREDIKSI SUHU MAKSIMUM DAN MINIMUM INDONESIA- NUMERICAL WEATHER PREDICTION MENGGUNAKAN *SUPPORT VECTOR REGRESSION* DAN *LONG SHORT-TERM MEMORY*

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Statistika pada
Program Studi Sarjana Statistika
Departemen Statistika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember

Oleh: **HAIKAL MAULADANA ICHWANI**
NRP 5003211142

Tanggal Ujian: 22 Juli 2025
Periode Wisuda: September 2025

Disetujui oleh:
Pembimbing:

1. Dr. Sutikno, M.Si.
NIP 19710313 199702 1 001

Penguji:

2. Shofi Andari, S.Stat., M.Si., Ph.D.
NIP 198712072014042001
3. Dr. Achmad Choiruddin, S.Si., M.Sc.
NIP 19911210 202406 1001

Kepala Departemen Statistika
Fakultas Sains dan Analitika Data



Dr. rer. pol. Dedy Dwi Prastyo, S.Si., M.Si.
NIP 19831204 200812 1 002

(Halaman ini sengaja dikosongkan)

APPROVAL SHEET

PREDICTION MODEL MINIMUM AND MAXIMUM TEMPERATURE INDONESIAN-NUMERICAL WEATHER PREDICTION USING SUPPORT VECTOR REGRESSION AND LONG SHORT-TERM MEMORY

FINAL PROJECT

Submitted to fulfill one of the requirements
For obtaining a degree Bachelor of Statistics at
Bachelor Program of Statistics
Department of Statistics
Faculty of Science and Data Analytics
Institut Teknologi Sepuluh Nopember

by: **HAIKAL MAULADANA ICHWANI**
NRP 5003211142

Exam Date: 22 July 2025
Graduation Period: September 2025

Approved by:
Advisor:

1. Dr. Sutikno, M.Si.
NIP 19710313 199702 1 001



Examiners:

2. Shofi Andari, S.Stat., M.Si., Ph.D.
NIP 198712072014042001
3. Dr. Achmad Choiruddin, S.Si., M.Sc.
NIP 19911210 202406 1001



Head of Statistics Department
Faculty of Science and Data Analytics



Dr. rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.
NIP 19831204 200812 1 002

(Halaman ini sengaja dikosongkan)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama Mahasiswa / NRP : Haikal Mauladana Ichwani / 5003211142
Departemen : Statistika
Dosen Pembimbing / NIP : Dr. Sutikno, M.Si. / 19710313 199702 1 001

Dengan ini menyatakan Tugas Akhir dengan judul "**Model Prediksi Suhu Maksimum dan Minimum Indonesia-Numerical Weather Prediction Menggunakan *Support Vector Regression* Dan *Long Short-Term Memory***" 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, 29 Juli 2025

Mengetahui,
Dosen Pembimbing



(Dr. Sutikno, M.Si.)
NIP 19710313 199702 1 001

Mahasiswa



(Haikal Mauladana Ichwani)
NRP 5003211142

(Halaman ini sengaja dikosongkan)

STATEMENT OF ORIGINALITY

The undersigned below:

Name of Student / NRP : Haikal Mauladana Ichwani / 5003211142
Department : Statistika
Advisor / NIP : Dr. Sutikno, M.Si. / 19710313 199702 1 001

Hereby declare that the Final Project with the title of "**Prediction Model Minimum And Maximum Temperature Indonesian-Numerical Weather Prediction Using Support Vector Regression and Long Short-Term Memory**" is the result of my own work, is original, and is written by following the rules 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 in Institut Teknologi Sepuluh Nopember

Surabaya, 29 July 2025

Acknowledged,
Advisor



(Dr. Sutikno, M.Si.)
NIP 19710313 199702 1 001

Student



(Haikal Mauladana Ichwani)
NRP 5003211142

(Halaman ini sengaja dikosongkan)

ABSTRAK

MODEL PREDIKSI SUHU MAKSIMUM DAN MINIMUM INDONESIA- NUMERICAL WEATHER PREDICTION MENGGUNAKAN *SUPPORT VECTOR REGRESSION* DAN *LONG SHORT-TERM MEMORY*

Nama Mahasiswa / NRP : Haikal Mauladana Ichwani / 5003211142
Departemen : Statistika FSAD-ITS
Dosen Pembimbing : Dr. Sutikno, M.Si.

Abstrak

Kebutuhan akan prakiraan cuaca yang akurat mendorong Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) telah mengembangkan model berbasis *Numerical Weather Prediction* yang dinamakan *Indonesia-Numerical Weather Prediction* (INA-NWP). Namun, model INA-NWP seringkali menunjukkan bias sistematis dalam memprediksi suhu permukaan maksimum dan minimum. Penelitian ini menerapkan teknik *statistical post-processing* berbasis *Machine Learning* (ML) untuk meminimalkan bias dengan dua metode yaitu *Support Vector Regression* (SVR) dan *Long Short-Term Memory* (LSTM). Kedua model disusun menggunakan data luaran suhu maksimum dan minimum dari INA-NWP serta data observasi di tiga stasiun meteorologi di wilayah Surabaya. Hasil penelitian menunjukkan bahwa kinerja SVR secara umum konsisten lebih baik daripada LSTM dalam memprediksi suhu maksimum dan minimum di seluruh stasiun pengamatan. Hal ini ditunjukkan dengan nilai RMSE model SVR lebih kecil dan *Improval Percentage* yang lebih besar daripada model LSTM di dua lokasi stasiun Perak I dan Juanda. Khusus untuk lokasi Sambikerep pada model suhu minimum LSTM memiliki kinerja lebih baik daripada SVR. Temuan ini menunjukkan potensi besar NWP-ML untuk mengoreksi bias sistematis dan meningkatkan ketepatan informasi cuaca di Indonesia.

Kata kunci: INA-NWP, *Long Short-Term Memory*, Prakiraan Cuaca, *Statistical post-processing*, *Support Vector Regression*.

(Halaman ini sengaja dikosongkan)

ABSTRACT

PREDICTION MODEL MINIMUM AND MAXIMUM TEMPERATURE INDONESIAN-NUMERICAL WEATHER PREDICTION USING SUPPORT VECTOR REGRESSION AND LONG SHORT-TERM MEMORY

Student Name / NRP : Haikal Mauladana Ichwani / 5003211142
Department : Statistics F-Scientics ITS
Advisor : Dr. Sutikno, M.Si.

Abstract

The demand for accurate weather forecasts has driven the Meteorological, Climatological, and Geophysical Agency (BMKG) to develop a Numerical Weather Prediction-based model known as the Indonesia-Numerical Weather Prediction (INA-NWP). However, the INA-NWP model often exhibits systematic biases in predicting surface maximum and minimum temperatures. This study applies statistical post-processing techniques based on Machine Learning (ML) to reduce such biases using two methods: Support Vector Regression (SVR) and Long Short-Term Memory (LSTM). Both models are developed using output data of maximum and minimum temperatures from INA-NWP along with observational data from three meteorological stations in the Surabaya area. The results show that SVR generally performs better than LSTM in predicting both maximum and minimum temperatures across all observation stations. This is indicated by lower RMSE values and higher Improval Percentage scores for the SVR model at two locations Perak I and Juanda. However, at the Sambikerep station, the LSTM model performs better for minimum temperature prediction than SVR. These findings demonstrate the significant potential of NWP-ML approaches in correcting systematic biases and improving the accuracy of weather information in Indonesia.

Keywords: INA-NWP, *Long Short-Term Memory*, Statistical post-processing, *Support Vector Regression*, Weather Prediction.

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul “Model Prediksi Suhu Maksimum dan Minimum Indonesia-Numerical Weather Prediction Menggunakan *Support Vector Regression* Dan *Long Short-Term Memory*” untuk memenuhi salah satu persyaratan kelulusan di Departemen Statistika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember Surabaya. Laporan Tugas Akhir ini dapat terselesaikan dengan baik atas bantuan, motivasi, dan dukungan dari berbagai pihak baik secara langsung maupun tidak langsung kepada penulis. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya serta menuntun penulis untuk terus berusaha dan bersyukur sehingga dapat menempuh pendidikan perguruan tinggi hingga mampu menyelesaikan laporan Tugas Akhir ini.
2. Bapak Gufron Ichwani dan Ibu Elok Nila Wardani selaku kedua orang tua penulis dan Chalila Ichwani selaku kakak dari penulis, yang selalu memberikan kasih sayang dan dukungan dalam berbagai bentuk, senantiasa mendoakan penulis, serta memberikan nasihat-nasihat baik, sehingga penulis selalu termotivasi untuk dapat menyelesaikan Tugas Akhir.
3. Bapak Dr. Sutikno, M.Si, selaku dosen pembimbing sekaligus dosen wali yang telah memberikan arahan, bimbingan dan motivasi dalam pengerjaan Tugas Akhir.
4. Ibu Shofi Andari, S.Stat., M.Si. Ph.D dan Bapak Dr. Achmad Choiruddin, S.Si, M.Sc, selaku dosen penguji yang telah memberikan masukan, saran dan arahan yang membangun dalam menyempurnakan laporan Tugas Akhir.
5. Bapak Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si. selaku Kepala Departemen Statistika, Ibu Dr. Wibawati, S.Si., M.Si. selaku Sekretaris Departemen Statistika, dan Ibu Shofi Andari, S.Stat., M.Si., Ph.D. selaku Kepala Program Studi S1 Statistika yang telah memberikan sarana dan prasarana selama perkuliahan.
6. Seluruh dosen Statistika ITS yang telah memberikan ilmu pengetahuan dan mendukung seluruh proses pembelajaran selama di Departemen Statistika ITS, serta seluruh staff tata usaha Departemen Statistika ITS.
7. Teman-teman mahasiswa Departemen Statistika ITS angkatan 2021 dan teman-teman Pondok Pesantren Statistika ITS yang telah kebersamaian penulis selama masa perkuliahan di Departemen Statistika ITS khususnya Shofia, Ayu, dan Elsa atas kebersamaan, semangat, dan kenangan yang tak ternilai selama ini.
8. Fiqi Hidayah yang senantiasa menemani serta memberikan motivasi, bantuan, dan dukungan yang sangat berarti dalam perjalanan ini.
9. Semua pihak yang turut membantu dalam pengerjaan Tugas Akhir yang tidak bisa penulis sebutkan satu persatu. Semoga Tugas Akhir ini dapat bermanfaat baik bagi penulis maupun pembaca. Penulis telah berusaha semaksimal mungkin untuk menyusun dan menyelesaikan Tugas Akhir ini. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun dari pembaca sehingga dapat menyempurnakan penulisan Tugas Akhir ini.

Surabaya, Juli 2025

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN	i
APPROVAL SHEET	iii
PERNYATAAN ORISINALITAS	v
STATEMENT OF ORIGINALITY	vii
ABSTRAK.....	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
DAFTAR NOTASI.....	xxiii
DAFTAR LAMPIRAN	xxv
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
BAB II	5
TINJAUAN PUSTAKA	5
2.1 <i>Numerical Weather Prediction</i>	5
2.2 <i>Statistical Post-Processing</i>	6
2.3 <i>Support Vector Machine</i>	6
2.3.1 <i>Support Vector Regression</i>	6
2.3.2 <i>Data Splitting</i>	10
2.3.3 <i>Overfitting dan Underfitting</i>	11
2.4 <i>Deep Learning</i>	12
2.5 <i>Artificial Neural Network</i>	13
2.5.1 <i>Perceptron</i>	13
2.5.2 <i>Multi Layer Perceptron</i>	14
2.6 <i>Long Short-Term Memory</i>	16

2.6.1 <i>Loss Function</i>	18
2.6.2 <i>Adam Optimizer</i>	19
2.6.3 <i>Hyperparameter</i>	20
2.6.4 <i>Random search</i>	21
2.7 <i>Robust Scaler</i>	22
2.8 <i>Root Mean Square Error</i>	23
2.9 Ukuran Pengkoreksian Bias.....	23
2.10 Suhu Udara	23
2.11 Hasil Penelitian Terdahulu	24
BAB III	27
METODE PENELITIAN	27
3.1 Sumber Data	27
3.2 Variabel Penelitian	27
3.3 Metode Analisis Data	28
3.4 Diagram Alir.....	30
BAB IV	31
ANALISIS DAN PEMBAHASAN	31
4.1 Deskripsi Suhu Udara	31
4.2 Pemodelan <i>Support Vector Regression</i>	33
4.2.1 Pemodelan <i>Support Vector Regression</i> untuk Stasiun Meteorologi Perak I	33
4.2.2 Pemodelan <i>Support Vector Regression</i> untuk Stasiun AWS GI Sambikerep	34
4.2.3 Pemodelan <i>Support Vector Regression</i> untuk Stasiun Meteorologi Juanda	34
4.3 Pemodelan Model <i>Long Short-Term Memory</i>	35
4.3.1 Pemodelan <i>Long Short-Term Memory</i> untuk Stasiun Meteorologi Perak I	36
4.3.2 Pemodelan <i>Long Short-Term Memory</i> untuk Stasiun AWS GI Sambikerep	37
4.3.3 Pemodelan <i>Long Short-Term Memory</i> untuk Stasiun Meteorologi Juanda	38
4.4 Evaluasi dan Komparasi Model.....	39
4.4.1 Pemodelan Variabel Suhu Maksimum	39
4.4.2 Pemodelan Variabel Suhu Minimum.....	40
BAB V	43
KESIMPULAN DAN SARAN	43

5.1 Kesimpulan	43
5.2 Saran	43
DAFTAR PUSTAKA.....	45
LAMPIRAN	49
BIODATA PENULIS.....	69

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Visualisasi Kondisi Cuaca Dunia Nyata dan Model NWP (Sumber : Wilks, 2006)	5
Gambar 2.2 SVR linear dengan satu dimensi	7
Gambar 2.3 Training Error dan Validation Error (Sumber: Maharana, et al., 2025)	12
Gambar 2.4 Struktur Perceptron (Sumber : Staudemeyer & Morris, 2019)	14
Gambar 2.5 Layer MLP	15
Gambar 2.6 Layer LSTM	16
Gambar 2.7 Langkah pada forget gate.....	16
Gambar 2.8 Langkah pada input gate dan tanh layer	17
Gambar 2.9 Langkah memperbarui cell state	17
Gambar 2.10 Langkah pada Output gate	18
Gambar 2.11 Perbandingan Performa Random search dan <i>Grid Search</i> (Sumber: Bergstra & Bengio, 2012)	22
Gambar 3.1 Sebaran Lokasi Stasiun Meteorologi di Surabaya	27
Gambar 3.2 Diagram Alir Pembentukan Model <i>Support Vector Regression</i> dan <i>Long Short-Term Memory</i>	30
Gambar 4.1 Pola Suhu Maksimum dan Minimum Stasiun Meteorologi Perak I Selama Periode 387 Hari	32
Gambar 4.2 Pola Suhu Maksimum dan Minimum Stasiun AWS GI Sambikerep Selama Periode 387 Hari	32
Gambar 4.3 Pola Suhu Maksimum dan Minimum Stasiun Juanda Selama Periode 387 Hari.....	32
Gambar 4.4 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model SVR di Stasiun Juanda.....	35
Gambar 4.5 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model SVR di Stasiun Juanda.....	35
Gambar 4.6 Arsitektur LSTM untuk Model Prediksi.....	36
Gambar 4.7 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model LSTM di Stasiun Juanda	38
Gambar 4.8 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model LSTM di Stasiun Juanda	39

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 Daftar Penelitian Terdahulu yang Menggunakan Dataset INA-NWP dalam Membuat Model Prakiraan	24
Tabel 2.2 Daftar Penelitian Terdahulu Menggunakan Model SVR dan LSTM dalam Membuat Model Prakiraan.....	26
Tabel 3.1 Lokasi Pengamatan Stasiun Meteorologi	27
Tabel 3.2 Variabel Respon Penelitian	27
Tabel 3.3 Parameter NWP WRF-EMS.....	28
Tabel 3.4 Struktur Data untuk Pemodelan Suhu Maksimum pada Stasiun ke-s	28
Tabel 3.5 Struktur Data untuk Pemodelan Suhu Minimum pada Stasiun ke-s	28
Tabel 3.6 Pencarian hyperparameter dan fungsi kernel optimal menggunakan random search	29
Tabel 3.7 Pencarian hyperparameter optimal menggunakan random search	30
Tabel 4.1 Nilai Rata-Rata, Standar Deviasi, Nilai Maksimum, dan Minimum dari Suhu Maksimum serta Minimum di 3 Stasiun Meteorologi.....	31
Tabel 4.2 Hyperparameter dan Fungsi Kernel Optimal untuk Pemodelan Stasiun Meteorologi Perak I.....	33
Tabel 4.3 Hyperparameter dan Fungsi Kernel Optimal untuk Pemodelan Stasiun AWS GI Sambikerep	34
Tabel 4.4 Hyperparameter dan Fungsi Kernel Optimal untuk Pemodelan Stasiun Meteorologi Juanda	34
Tabel 4.5 Hyperparameter Optimal untuk Pemodelan Stasiun Meteorologi Perak I	36
Tabel 4.6 Hyperparameter Optimal untuk Pemodelan Stasiun AWS GI Sambikerep	37
Tabel 4.7 Hyperparameter Optimal untuk Pemodelan Stasiun Meteorologi Juanda	38
Tabel 4.8 Hasil RMSE dan Improval Percentage untuk Pemodelan Suhu Maksimum	40
Tabel 4.9 Nilai RMSE dan Improval Percentage pada Model Suhu Minimum	40

(Halaman ini sengaja dikosongkan)

DAFTAR NOTASI

w^T	: vektor bobot
b	: bias
C	: parameter regularisasi yang menentukan <i>trade-off</i> antara <i>smoothness</i> dan pinalti dari kesalahan besar
$\varphi(x_i)$	: fungsi transformasi non-linier
y_i	: nilai aktual periode ke- i
ε	: parameter margin yang menentukan zona bebas <i>error</i> di sekitar fungsi estimasi
ξ_i, ξ_i^*	: variable slack
$\alpha, \alpha^*, \mu, \mu^*$	: lagrange multiplier
σ	: fungsi aktivasi <i>sigmoid</i>
f_t	: nilai yang dihasilkan pada <i>forget gate</i> ke- t
W_{xf}	: bobot dari input ke- i pada <i>forget gate</i>
W_{hf}	: bobot dari h_{t-1}
x_t	: input pada step waktu- t (nilai input suhu maksimum dan minimum)
h_{t-1}	: <i>hidden state</i> dari waktu sebelumnya
b_f	: bias dari <i>forget gate</i>
W_{xi}	: bobot dari input ke- i pada <i>input gate</i>
W_{hi}	: bobot dari h_{t-1}
b_i	: bias dari <i>input gate</i>
$\tilde{C}$	: model <i>input gate</i> , menentukan berapa banyak informasi baru diterima oleh <i>cell state</i>
$\tanh$	: fungsi aktivasi <i>hyperbolic tangent</i>
W_{xc}	: bobot dari input ke- i pada bobot dari <i>cell state</i>
W_{hc}	: bobot dari h_{t-1}
b_c	: bias dari <i>cell state</i>
O_t	: nilai yang dihasilkan pada <i>forget gate</i> ke- t
W_{xo}	: bobot dari input ke- i pada <i>output gate</i>
W_{ho}	: bobot dari h_{t-1}
b_o	: bias dari <i>output gate</i>
C_t	: <i>cell state</i>
C_{t-1}	: <i>cell state</i> dari waktu sebelumnya
$\odot$	: operasi perkalian elemen
h_t	: nilai <i>output</i> orde- t
O_t	: <i>output gate</i>
y_t	: data aktual periode- t
$\hat{y}_t$	: data prediksi periode- t
n	: jumlah data
β_1 & β_2	: <i>exponential decay rate</i>
η	: <i>learning rate</i>
m_t	: bias moment pertama
v_t	: bias moment kedua

θ_t : parameter (bobot) pada iterasi ke- t
 g_t : *gradient loss function*
 δ_i : *dropout rate*
 $Q1(X)$: kuartil pertama dari variabel X
 $Q3(X)$: kuartil ketiga dari variabel X

DAFTAR LAMPIRAN

Lampiran 1. Syntax Pemodelan LSTM	49
Lampiran 2. Syntax Pemodelan SVR	56
Lampiran 3. Dataset Suhu Maksimum	59
Lampiran 4. Dataset Suhu Minimum	60
Lampiran 5. Hasil Prediksi LSTM untuk Variabel Suhu Maksimum	61
Lampiran 6. Hasil Prediksi LSTM untuk Variabel Suhu Minimum	62
Lampiran 7. Hasil Prediksi SVR untuk Variabel Suhu Maksimum	63
Lampiran 8. Hasil Prediksi SVR untuk Variabel Suhu Minimum	64
Lampiran 9. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model LSTM di Stasiun Meteorologi Perak I	65
Lampiran 10. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model LSTM di Stasiun AWS GI Sambikerep	65
Lampiran 11. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model LSTM di Stasiun Meteorologi Perak I	65
Lampiran 12. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model LSTM di Stasiun AWS GI Sambikerep	66
Lampiran 13. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model SVR di Stasiun Perak I	66
Lampiran 14. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model SVR di Stasiun AWS GI Sambikerep	66
Lampiran 15. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model SVR di Stasiun Perak I	67
Lampiran 16. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model SVR di Stasiun AWS GI Sambikerep	67
Lampiran 17. Surat Pernyataan Data Sekunder.....	68

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan informasi cuaca yang akurat semakin penting saat ini. Untuk menyajikan informasi tersebut, BMKG telah mengembangkan sistem prakiraan cuaca berbasis numerik yang mencakup seluruh wilayah Indonesia, yang disebut *Indonesia-Numerical Weather Prediction* (INA-NWP), dengan cakupan parameter seperti suhu dan kelembapan udara (Tentang BMKG, 1841). Selama 40 tahun terakhir, prakiraan cuaca di seluruh dunia sangat diuntungkan oleh panduan yang diberikan oleh NWP ini (Durai & Bhradwaj, 2014). BMKG sendiri telah melakukan penelitian terkait prakiraan cuaca jangka pendek menggunakan NWP sejak tahun 2004. Meskipun model NWP global mampu memberikan prakiraan cuaca jangka pendek dan menengah dengan cukup baik, model ini relatif kurang terampil dalam meramalkan parameter cuaca di permukaan. Prakiraan model NWP diketahui memiliki bias sistematis dalam memperkirakan parameter permukaan, terutama suhu maksimum dan minimum. Bias ini disebabkan oleh keterbatasan dalam fisika model, ketidaksempurnaan dalam kondisi awal, serta kondisi batas yang digunakan dalam perhitungan. (Durai & Bhradwaj, 2014).

NWP adalah model prediksi cuaca numerik yang berbasis dinamika atau dikenal sebagai *classical statistical forecasting* (Wilks, 2006). Haryoko mengatakan NWP yang digunakan oleh BMKG terdiri atas beberapa model, antara lain *Weather Research Forecast* (WRF), *European Centre for Medium-Range Weather Forecast* (ECMWF), *National Centers for Environmental Prediction* (NCEP), ARPEGE, dan CCAM (Putra, 2023). *Indonesia Numerical Weather Prediction* (INA-NWP) adalah model regional yang dikembangkan oleh BMKG menggunakan WRF yang diasimilasikan dengan data observasi BMKG sehingga menghasilkan kondisi cuaca yang mencerminkan cuaca di seluruh wilayah Indonesia. Dalam model NWP terdapat perbedaan antara dunia nyata dan representasinya, sebab model NWP menyederhanakan dan menyeragamkan kondisi permukaan dengan merepresentasikan dunia sebagai kumpulan titik-titik *grid* di mana keluaran NWP diterapkan. Selain itu, Lokasi dan variabel yang memerlukan prakiraan mungkin tidak direpresentasikan secara eksplisit oleh model NWP. Model NWP juga bersifat deterministik, artinya satu integrasi NWP hanya mampu menghasilkan satu prakiraan untuk elemen meteorologi apa pun, mengingat serangkaian kondisi model awal (Wilks, 2006). Sehingga dapat disimpulkan model NWP bersifat bias, deterministik dan tidak bisa menjelaskan secara penuh proses stokastik cuaca. Pada sisi lain, ketidakakuratan dalam model NWP disebabkan oleh minimnya pengetahuan mengenai proses kompleks di atmosfer dan keterbatasan kemampuan komputer dalam mengeksekusi persamaan model dalam waktu tertentu (Idowu & Rautunbech, 2009). Oleh karena itu, diperlukan metode untuk meningkatkan ketepatan prediksi curah hujan dengan mengintegrasikan data INA-NWP.

Statistical post-processing adalah sebuah teknik untuk mengurangi bias dan mengukur tingkat ketidakpastian dalam prakiraan cuaca berbasis NWP (Zhang, et al., 2023). Metode yang pertama kali dikembangkan adalah “*perfect-prog*”, yang merupakan singkatan dari *perfect progonis*. Teknik *perfect-prog* tidak memperbaiki kesalahan atau bias dari NWP, melainkan menerima prakiraan variabel atmosfer dari model tersebut, dengan asumsi bahwa prakiraan tersebut akurat. Pendekatan persamaan regresi *perfect-prog* serupa dengan pengembangan regresi klasik dengan perbedaan utama yaitu pada jeda waktu (*time lag*). Persamaan klasik mempertimbangkan jeda waktu prakiraan dengan menghubungkan prediktor yang tersedia dengan nilai prediktand yang akan diamati di waktu mendatang, sebaliknya pada *perfect-frog*

persamaan untuk menentukan predictand di waktu mendatang dibuat berdasarkan nilai predictor untuk waktu mendatang (Wilks, 2006). Metode kedua yang dikembangkan yaitu *Model Output Statistics* (MOS) (Glahn & Lowry, 1972). MOS menentukan hubungan statistik antara prediktand dan variabel dari model numerik. Sama seperti pendekatan *perfect-prog*, *time lag* dalam prakiraan MOS diintegrasikan melalui NWP. Namun, berbeda dengan *perfect prog* prakiraan statistic MOS untuk prediktand besok dibuat menggunakan prakiraan dinamis NWP untuk prediktor besok yang tersedia hari ini. Selain itu, tidak seperti pendekatan *perfect-prog*, persamaan prakiraan MOS harus dibuat secara terpisah untuk setiap kasus (Wilks, 2006). Selain metode tradisional, salah satu metode *statistical post-processing* yaitu dengan menggunakan *machine learning* (ML). ML dipilih bukan tanpa sebab, kemampuan ML dalam pembelajaran mandiri yang kuat dan kemampuannya dalam menangani masalah nonlinier sudah digunakan dalam beberapa tahun terakhir (Ghazvinian, et al., 2021). Khususnya metode yang dapat mengkalibrasi model dengan menggunakan berbagai karakteristik terkait predictor sebagai variable input.

Support Vector Regression (SVR) adalah metode pembelajaran mesin berbasis *Support Vector Machine* (SVM) yang digunakan untuk menyelesaikan masalah regresi. SVR bekerja dengan menemukan *hyperplane optimal* dalam ruang fitur tinggi, memungkinkan model untuk menangani hubungan non-linear yang kompleks antara variabel *input* dan *output*. Keunggulan SVR terletak pada kemampuannya untuk mengatasi overfitting serta menghasilkan prediksi yang lebih stabil dan akurat pada data dengan pola yang tidak teratur, seperti data cuaca (Smola & Scholkopf, 2004). Sementara itu untuk membandingkan kinerja dari SVR dipilihlah *Long Short-Term Memory* (LSTM). LSTM adalah jenis jaringan saraf dalam berbasis *Recurrent Neural Network* (RNN) yang dirancang untuk menangani data dengan ketergantungan temporal jangka panjang. LSTM memiliki kemampuan untuk menyimpan informasi dalam jangka waktu yang lebih lama dibandingkan dengan RNN konvensional, sehingga cocok untuk menganalisis data cuaca yang memiliki pola musiman dan tren jangka panjang. LSTM telah terbukti efektif dalam berbagai tugas prediksi berbasis deret waktu, termasuk prakiraan cuaca (Hochreiter & Schmidhuber, 1997).

Penggunaan model *Support Vector Regression* (SVR) dan *Long Short-Term Memory* (LSTM) dalam penelitian ini dianggap cocok karena keduanya memiliki keunggulan dalam menangani permasalahan prediksi cuaca. SVR merupakan metode pembelajaran mesin yang efektif dalam menangani hubungan non-linear antara variabel cuaca dengan memanfaatkan *kernel trick* untuk menangkap pola kompleks dalam data. Sementara itu, LSTM merupakan arsitektur jaringan saraf dalam yang sangat baik dalam menangkap pola temporal jangka panjang, yang sangat relevan dalam analisis data cuaca yang memiliki dependensi waktu yang kuat. Kebaikan model prediksi dalam melakukan prakiraan dievaluasi menggunakan *Root Mean Square Error* (RMSE) dengan kriteria *improvement percentage*. Dengan melihat nilai *improvement percentage* kedua metode diharapkan dapat memperoleh informasi metode mana yang menghasilkan model prediksi yang tepat serta gambaran informasi yang akurat.

1.2 Rumusan Masalah

Berdasarkan identifikasi latar belakang diatas, rumusan masalah dalam penelitian ini adalah

1. Bagaimana kinerja *Support Vector Regression* dalam model prediksi baik untuk suhu maksimum dan untuk suhu minimum?

2. Bagaimana kinerja *Long Short-Term Memory* dalam model prediksi baik untuk suhu maksimum dan untuk suhu minimum?
3. Bagaimana perbandingan kinerja antara metode *Support Vector Regression* dan *Long Short-Term Memory* dalam pemodelan data INA-NWP untuk prediksi baik suhu maksimum dan suhu minimum dengan kriteria *percentage improvement*?

1.3 Batasan Masalah

Ruang lingkup pada penelitian ini yaitu dibatasi pada pembentukan metode *Support Vector Regression* dan *Long Short-Term Memory* di wilayah Surabaya dan sekitarnya dengan fokus 3 stasiun meteorologi yaitu Stasiun Meteorologi Perak I, AWS GI Sambikerep, dan Stasiun Meteorologi Juanda. Data INA-NWP yang digunakan pada penelitian ini adalah hasil model aplikasi *Weather Research and Forecasting Environmental Modeling System* (WRF-EMS). Variabel yang digunakan pada penelitian ini adalah suhu udara, yaitu suhu maksimum dan minimum.

1.4 Tujuan

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah sebagai berikut.

1. Mendapatkan model prediksi, baik model untuk suhu maksimum dan model untuk suhu minimum dengan metode *Support Vector Regression*.
2. Mendapatkan model prediksi, baik model untuk suhu maksimum dan model untuk suhu minimum dengan metode *Long Short-Term Memory*.
3. Membandingkan kinerja antara metode *Support Vector Regression* dan *Long Short-Term Memory* dengan kriteria *persentase improvement*.

1.5 Manfaat

Manfaat yang diharapkan pada penelitian ini adalah sebagai berikut :

1. Bagi Peneliti, dapat memberikan pengetahuan serta wawasan terkait penerapan teknik *statistical post-processing* pada bidang meteorologi, klimatologi, dan geofisika.
2. Bagi BMKG, hasil dari penelitian ini dapat dijadikan sebagai bahan acuan dalam mengoptimalkan luaran NWP dengan metode *Support Vector Regression* dan *Long Short-Term Memory* untuk pemodelan prediksi cuaca

(Halaman ini sengaja dikosongkan)

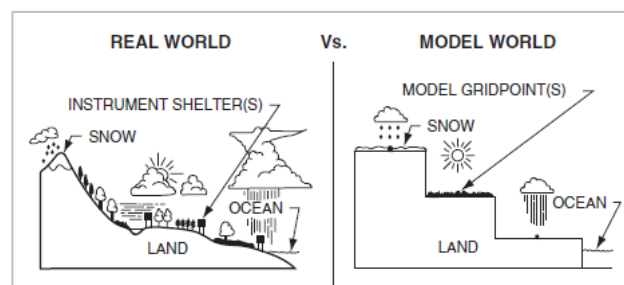
BAB II

TINJAUAN PUSTAKA

2.1 *Numerical Weather Prediction*

Numerical Weather Prediction (NWP) adalah model prediksi cuaca numerik yang berbasis dinamika atau dikenal sebagai *classical statistical forecasting* (Wilks, 2006). Dalam model NWP kondisi permukaan disederhanakan dan diseragamkan, sementara atmosfer dimodelkan sebagai kumpulan petak persegi. Karakteristik penggunaan lahan dalam model ini direpresentasikan sebagai *array* kotak-kotak *grid* yang mencakup lapisan horizontal di atmosfer vertikal (Idowu & Rautunbech, 2009). Prakiraan cuaca dapat dirumuskan dengan menyelesaikan persamaan pergerakan atmosfer. Persamaan-persamaan tersebut meliputi persamaan non-linier *time-dependent differential partial* angin, suhu, kelembaban, dan tekanan.

Pada perhitungan NWP diukur berdasarkan domain lokasi atau *grid* yang tinggi, dengan skala sebesar itu NWP akan memberikan informasi cuaca yang homogen pada daerah *grid* tersebut. Maka dari itu hasil dari prediksi cuaca dengan NWP biasanya menghasilkan bias yang besar terutama pada wilayah dengan topografi dan tutupan vegetasi yang kompleks. Gambar 2.1 memvisualisasikan adanya perbedaan kondisi cuaca riil di suatu wilayah dan kondisi cuaca di model NWP.



Gambar 2.1 Visualisasi Kondisi Cuaca Dunia Nyata dan Model NWP
(Sumber : Wilks, 2006)

Gambar 2.1 menunjukkan bahwa terdapat perbedaan penting antara kondisi cuaca riil di suatu wilayah dan representasinya dalam model NWP. Model NWP menyederhanakan kondisi permukaan dengan merepresentasikan dunia sebagai kumpulan titik *grid* tempat keluaran NWP. Seperti yang diilustrasikan, efek skala kecil seperti topografi atau perairan yang penting bagi cuaca lokal, tetapi tidak disertakan dalam permodelan NWP. Selain itu, lokasi dan variabel yang diperlukan untuk prakiraan mungkin tidak secara eksplisit direpresentasikan oleh model NWP. Sehingga dapat dikatakan bahwa model NWP bersifat deterministik dan tidak bisa secara penuh menjelaskan proses stokastik cuaca (Wilks, 2006).

INA-NWP merupakan produk NWP dari BMKG yang berfokus pada wilayah Indonesia (Putra, 2023). Perhitungan INA-NWP diukur pada resolusi *grid* yang tinggi, yang berkisar antara 3-9 km (3 km untuk pulau Jawa dan 9 km di luar pulau Jawa). Prediksi INA-NWP dilakukan selama 2 kali dalam sehari dengan *initial condition* pada pukul 00.00 dan 12.00 waktu setempat. Prediksi dari INA-NWP berlaku untuk 3 hari kedepan dengan *timestep* selama 3 jam dan untuk saat ini jumlah stasiun meteorologi dan AWS berjumlah kurang lebih 360 stasiun meteorologi dan AWS di seluruh Indonesia.

2.2 *Statistical Post-Processing*

Statistical post-processing adalah sebuah teknik untuk mengurangi bias dan mengukur tingkat ketidakpastian dalam prakiraan cuaca berbasis NWP (Zhang, et al., 2023). Secara garis besar metode post-processing dapat dibagi menjadi 2 kelompok yaitu pendekatan dengan asumsi berbasis distribusi (parametrik) dan pendekatan dengan asumsi yang tidak berbasis distribusi (nonparametrik).

Pendekatan *postprocessing* berbasis distribusi menggunakan model parametrik untuk distribusi peramalan dengan memilih keluarga distribusi probabilitas yang sesuai. Parameter distribusi peramalan kemudian dikaitkan dengan prediktor dari sistem NWP melalui regresi untuk mengoreksi kesalahan sistematis. Koefisien regresi diperkirakan dengan mengoptimalkan fungsi kehilangan seperti *Continuous Ranked Probability Score* (CRPS). Pendekatan ini menghasilkan model yang mudah diterapkan dan banyak digunakan di layanan meteorologi.

Pendekatan ini menggunakan metode pengolahan ulang (*post processing*) yang menghindari asumsi distribusi dengan membangun pendekatan terhadap distribusi prediksi. Bremnes mengusulkan metode regresi kuantil yang hanya mempertimbangkan kuantil-kuantil tertentu dari distribusi prediksi (Vannitsem, et al., 2021). Kuantil tersebut diperkirakan secara terpisah yang mengakibatkan kuantil tidak valid. Pendekatan tanpa asumsi distribusi ini memungkinkan model untuk tetap fleksibel dan adaptif terhadap data besar serta variabilitas tinggi.

Sebagian besar metodologi baru dalam post-processing prediksi cuaca kini berbasis teknik *Machine Learning* (ML). Meskipun pendekatan ML modern telah menunjukkan peningkatan yang menjanjikan dibandingkan metode tradisional, masih terdapat beberapa tantangan yang perlu diatasi. Salah satu tantangan utama adalah bagaimana memanfaatkan secara optimal struktur yang ada dalam prakiraan NWP serta kesalahan NWP itu sendiri. Pendekatan saat ini sering hanya mengandalkan sejumlah statistik ansambel dan kovariat terbatas. Namun, metode ML modern, terutama yang menggunakan jaringan saraf tiruan (neural networks), dapat dirancang untuk menangani hubungan antara sejumlah besar variabel kovariat secara lebih langsung. Salah satu tantangan lain dalam penerapan ML adalah interpretabilitas model, di mana sulit untuk memahami bagaimana model menghasilkan suatu prediksi.

2.3 *Support Vector Machine*

Support Vector Machine (SVM) adalah metode pembelajaran yang memanfaatkan ruang hipotesis berupa fungsi linear dalam ruang fitur dengan dimensi tinggi. Metode ini dilatih menggunakan algoritma berbasis teori optimasi yang didasari oleh prinsip pembelajaran statistik, dengan menerapkan *learning bias* yang relevan (Cholissodin et al, 2020). SVM juga dapat diperluas untuk menangani kasus klasifikasi dengan lebih dari dua kelas. Fungsi SVM lebih mengarah ke klasifikasi dan regresi. Pada klasifikasi SVM digunakan untuk memprediksi kelas dari data baru berdasarkan data *training*, sedangkan pada regresi SVM atau biasa dikenal dengan *Support Vector Regression* (SVR) bertujuan untuk memprediksi nilai kontinu daripada kelas diskrit.

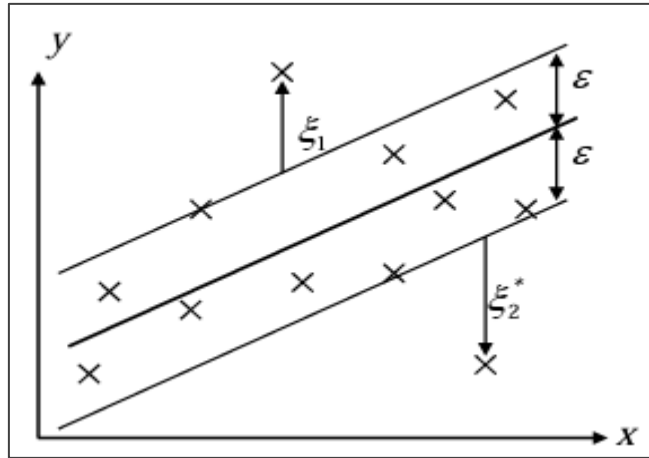
2.3.1 *Support Vector Regression*

SVR adalah metode regresi yang mengadaptasi konsep dari SVM. SVR memiliki tujuan menemukan fungsi yang paling sesuai dengan data, memberikan generalisasi yang baik sekaligus tetap tahan terhadap gangguan (noise) dengan mempertimbangkan fungsi loss ε -

insensitive yang teratur (Kocaoglu, 2024). Konsep persamaan SVR secara umum dapat didefinisikan dari perspektif geometris menggunakan satu dimensi (Awad & Khanna, 2015). Fungsi yang bernilai kontinu dapat dituliskan seperti pada Persamaan (2.1). Pada data multidimensi, ditambahkan sebuah set data latih nilai 1 ke x dan memasukkan b (bias) ke dalam vektor w (bobot) untuk menyederhanakan notasi. Diperoleh regresi multivariat pada Persamaan (2.2).

$$y = f(x) = \langle w, x \rangle + b = \sum_{j=1}^M w_j x_j + b, y, b \in \mathbb{R}, x, w \in \mathbb{R}^M \quad (2.1)$$

$$f(x) = \begin{bmatrix} w \\ b \end{bmatrix}^T \begin{bmatrix} x \\ 1 \end{bmatrix} = w^T x + b, x, w \in \mathbb{R}^{M+1} \quad (2.2)$$



Gambar 2.2 SVR linear dengan satu dimensi

SVR merumuskan masalah pendekatan fungsi ini sebagai suatu masalah optimasi yang berusaha menemukan margin kesalahan (tabung tersempit), sambil meminimalkan kesalahan prediksi. Tabung tersempit dalam konteks ini adalah dalam SVR model mencoba membungkus data dalam suatu margin kesalahan dengan lebar tertentu di sekitar kurva prediksi. Hal ini bertujuan agar sebanyak mungkin titik data jatuh dalam tabung ini, sehingga model hanya memperhitungkan data yang benar-benar menyimpang dari pola umum. Kondisi ini menghasilkan fungsi objektif dalam Persamaan (2.3), dimana $\|w\|$ adalah besar vektor normal terhadap permukaan yang sedang didekati

$$\min \frac{1}{2} \|w\|^2 \quad (2.3)$$

SVR menyelesaikan masalah dual Lagrangian dengan 2L pengali Lagrange, yaitu dua kali jumlah sampel pelatihan, dengan memanfaatkan trik *kernel* (*kernel trick*) yang memungkinkan transformasi data input secara implisit ke dalam ruang dimensi yang lebih tinggi Fungsi Destinasi Primal SVR mengurangi kompleksitas model dengan menyelesaikan Fungsi (2.3). Dengan mengadopsi hal ini bisa dijelaskan dengan memperkenalkan variabel *slack* ξ_i, ξ_i^* untuk mengukur deviasi sampel *training* yang berada di luar zona ϵ -insensitive (Fauziah, et al., 2020). Sehingga didapatkan fungsi minimisasi (2.4):

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (2.4)$$

dengan batasan :

$$y_i - w^T \varphi(x_i) - b \leq \epsilon + \xi_i$$

$$w^T \varphi(x_i) - y_i + b \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

keterangan :

w^T : vektor bobot

b : bias

$\varphi(x_i)$: fungsi transformasi non-linier

y_i : nilai aktual periode ke- i

C : parameter regularisasi yang menentukan trade-off antara smoothness dan pinalti dari kesalahan besar

ε : parameter margin yang menentukan zona bebas *error* di sekitar fungsi estimasi

ξ_i, ξ_i^* : variabel slack

Konstanta $C > 0$ menentukan *trade off* antara ketipisan fungsi f dan batas atas deviasi lebih dari ε yang masih dapat ditoleransi. Semua deviasi yang lebih besar dari akan diberi pinalti sebesar C . Masalah optimasi ini dapat diubah menjadi masalah dual. Solusi dari permasalahan tersebut dapat diekspresikan dengan *lagrange* sebagai berikut:

$$\begin{aligned} L(w, b, \xi_i, \xi_i^*, \alpha, \alpha^*, \mu, \mu^*) = & \frac{1}{2} \|w\|^2 + \left[c \sum_{i=1}^n \xi_i + \xi_i^* \right] + \\ & - \left[\sum_{i=1}^n \alpha_i (y_i - w^T \varphi(x_i) - b + \varepsilon - \xi_i) \right] + \\ & - \left[\sum_{i=1}^n \alpha_i^* (w^T \varphi(x_i) - y_i + b + \varepsilon - \xi_i^*) \right] + \\ & - \sum_{i=1}^n \mu_i \xi_i - \sum_{i=1}^n \mu_i^* \xi_i^* \end{aligned} \quad (2.5)$$

Dimana pada Persamaan (2.5) $\alpha, \alpha^*, \mu, \mu^*$ adalah *lagrange multiplier*. Solusi optimal dapat diperoleh dengan melakukan turunan parsial dari L terhadap w, b, ξ_i, ξ_i^* sehingga w dapat didefinisikan Persamaan (2.6):

$$w = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \varphi(x_i) \quad (2.6)$$

Optimal hyperplane didefinisikan Persamaan (2.7).

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \varphi^T(x_i) \varphi(x) + b \quad (2.7)$$

Solusi optimal untuk b dengan menggunakan *Karush Kuhn Tucker* adalah :

$$\alpha_i (\varepsilon + \xi_i - y_i + w^T \varphi(x_i) + b) = 0 \text{ untuk } i = 1, \dots, n \quad (2.8)$$

$$\alpha_i^* (\varepsilon + \xi_i^* + y_i - w^T \varphi(x_i) - b) = 0 \text{ untuk } i = 1, \dots, n \quad (2.9)$$

$$\mu_i \xi_i = (C - \alpha_i) \xi_i = 0 \text{ untuk } i = 1, \dots, n \quad (2.10)$$

$$\mu_i \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0 \text{ untuk } i = 1, \dots, n \quad (2.11)$$

Dari Persamaan (2.10) karena $0 < \alpha_i < C$, maka $\xi_i = 0$. Kemudian pada Persamaan (2.11) $0 < \alpha_i^* < C$, maka $\xi_i^* = 0$. Pada Persamaan (2.8) dan (2.9) akan menjadi persamaan (2.12) dan (2.13).

$$\varepsilon - y_i + w^T \varphi(x_i) + b = 0 \text{ untuk } 0 < \alpha_i < C \quad (2.12)$$

$$\varepsilon + y_i - w^T \varphi(x_i) - b = 0 \text{ untuk } 0 < \alpha_i^* < C \quad (2.13)$$

SVR menggunakan fungsi *kernel* untuk mentransformasikan input non linear ke dalam ruang fitur yang dimensinya lebih tinggi kemudian diselesaikan secara linier. Melalui fungsi pemetaan φ pada metode *kernel*, data x pada *input space* dipetakan ke *feature space* F yang berdimensi lebih tinggi sehingga $\varphi : x \rightarrow \varphi(x)$. Sering kali fungsi $\varphi(x)$ tidak tersedia dan tidak dapat dihitung namun masalah ini dapat diatasi dengan menggunakan *kernel trick*, yang memungkinkan perhitungan *dot product* $\varphi(x_i) \cdot \varphi(x_j)$. Sebagai gantinya, *dot product* tersebut digantikan oleh fungsi *kernel*, seperti yang ditunjukkan dalam Persamaan (2.14), sehingga memungkinkan operasi dalam ruang berdimensi tinggi secara efisien (Ishlah, et al., 2023). Fungsi regresi SVR didefinisikan seperti Persamaan (2.15).

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) \quad (2.14)$$

$$f(x) = \sum_{i,j=1}^n (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \quad (2.15)$$

Menurut (Yu, et al., 2006) fungsi *kernel* yang sering digunakan pada metode SVR adalah sebagai berikut:

1. *Kernel Linier*

$$K(x_i, x_j) = x_i \cdot x_j \quad (2.16)$$

2. *Kernel Polynomial*

$$K(x_i, x_j) = (x_i^T \cdot x_j + 1)^d \quad (2.17)$$

3. *Kernel Radial Basis Function*

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \quad (2.18)$$

Dimana (x_i, x_j) adalah *training set*.

Setelah mendefinisikan fungsi *kernel*, perlu diperhatikan parameter γ (*gamma*) pada Persamaan (2.18) yang merupakan *hyperparameter* penting untuk *kernel* RBF. Secara konseptual, γ mengontrol seberapa besar pengaruh yang dimiliki oleh satu *support vector* terhadap data di sekitarnya. Nilai γ yang tinggi berarti radius pengaruh dari sebuah *support vector* menjadi kecil, sehingga model akan sangat fokus pada setiap titik data individual. Hal ini dapat menghasilkan batas keputusan yang sangat kompleks dan berisiko tinggi mengalami *overfitting*, di mana model bekerja sangat baik pada *training set* namun buruk pada data baru. Sebaliknya, nilai γ yang rendah memperluas radius pengaruh, menghasilkan batas keputusan yang lebih umum. Namun, jika nilainya terlalu rendah, model bisa menjadi terlalu sederhana dan gagal menangkap pola data yang mendasarinya (*underfitting*). Dengan demikian, proses penentuan nilai γ yang optimal, seringkali dilakukan bersamaan dengan parameter regularisasi

C melalui teknik seperti *Grid Search* dan *Random search*, menjadi tahap yang esensial untuk membangun model SVR yang *robust* (Hsu, et al., 2003).

2.3.2 Data Splitting

Pembagian data menjadi beberapa kategori memiliki tujuan yaitu untuk menghindari bias dalam pemilihan model. Menurut Kumar membagi dataset menjadi tiga bagian yaitu *training set*, *testing set*, dan *validation set* dapat membantu menghindari hal tersebut (Muraina, 2022). Kumar juga menyebutkan bahwa jika ukuran dataset berada dalam rentang 100 hingga 1.000.000 sampel, maka dataset dapat dibagi dengan rasio 60:20:20, yaitu 60% untuk *training set*, 20% untuk *validation set*, dan 20% untuk *testing set*. Namun, jika dataset memiliki lebih dari 1 juta sampel, maka pembagian yang direkomendasikan adalah 98:1:1 atau 99:0.5:0.5 (Muraina, 2022). Namun dalam beberapa kasus, dataset hanya dibagi menjadi 2 yaitu *training set* dan *testing set*. Menurut Weng, *data splitting* atau *train-test split* adalah pembagian data menjadi subset untuk pelatihan model dan evaluasi secara terpisah. Sebagai contoh, dataset dengan jumlah 30.805 sampel dapat dibagi menjadi 80% untuk pelatihan dan 20% untuk validasi (Muraina, 2022).

Dalam kasus *supervised learning*, model komputasi dilatih untuk memprediksi keluaran dari suatu fungsi target yang tidak diketahui. Fungsi target ini direpresentasikan oleh dataset pelatihan terbatas T yang terdiri atas contoh-contoh pasangan input dan output yang diinginkan: $T = \{[\vec{x}_1, \vec{a}_1], \dots, [\vec{x}_n, \vec{a}_n]\}$ dimana $n > 0$ adalah jumlah pasangan input atau output yang terurut. Pada akhir proses pelatihan, model akhir harus mampu memprediksi keluaran yang benar untuk sampel input dari T , namun juga harus mampu menggeneralisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

Reitermanova (2010) menyarankan bahwa berbagai teknik pengambilan sampel statistik klasik dapat digunakan untuk membagi data. Metode - metode pengambilan sampel ini dapat dikategorikan berdasarkan prinsip, tujuan, dan kompleksitas komputasi. Berikut merupakan teknik-teknik pengambilan sampel:

1. *Simple random sampling*

Simple random sampling (SRS) merupakan metode yang paling umum, mudah untuk diterapkan, dan efisien. Untuk training set $T_{tr}: p(x \in T_{tr}) = \frac{n_{tr}}{n}, n = |T|, n_{tr} = |T_{tr}|$ setiap sampel memiliki probabilitas yang sama untuk dipilih. Sampel dipilih secara acak dengan distribusi yang seragam, keuntungan dari metode ini adalah metode ini menghasilkan bias yang rendah pada kinerja model.

2. *Trial-and-error methods*

Trial-and-error methods adalah metode yang mencoba mengatasi varians yang tinggi dari kinerja model ketika menggunakan SRS dengan mengulangi pengambilan sampel acak beberapa kali dan merata-ratakan hasilnya. Metode ini memakan banyak waktu, sehingga dikembangkan teknik yang lebih canggih dengan mencoba ruang pemisahan potensial untuk meminimalkan perbedaan statistik antara T dan subsetnya.

3. *Systematic sampling*

Systematic sampling adalah pendekatan deterministik yang dikhususkan untuk kumpulan data yang secara natural terurut seperti deret waktu. Untuk dataset yang terurut, sampel awal dipilih secara acak kemudian setelah sampel ke- k diambil untuk $k = \frac{n_{tr}}{n}, n = |T|, n_{tr}, n_{tr} = |T_{tr}|$. *Systematic sampling* merupakan metode yang efisien dan mudah diterapkan pada dataset yang tepat. Pada jenis dataset tertentu

seperti data multimedia dan sekuens gen sangat sulit untuk menemukan urutan yang tepat.

4. *Convenience sampling*

Convenience sampling merupakan metode deterministik yang efisien untuk digunakan pada dataset deret waktu. Dataset T dibagi menurut blok diskrit, misalnya interval waktu. Sama halnya dengan pengambilan sampel sistematis, pengambilan sampel *convenience sampling* hanya menguntungkan untuk dataset dengan tipe khusus - T harus terdiri dari beberapa segmen yang terdistribusi sama. Metode ini rentan terhadap tren jangka panjang dan perubahan kondisi waktu.

5. CADEX, DUPLEX

CADEX dan pengembangannya DUPLEX adalah metode yang memilih sampel berdasarkan jarak euclidean antar sampel. Metode ini dimulai dengan memilih dua sampel yang memiliki jarak terjauh satu sama lain dari dataset T , lalu secara berulang memilih sampel-sampel berikutnya yang memiliki jarak maksimum terhadap sampel-sampel yang telah dipilih sebelumnya. Metode ini memiliki kelemahan dimana metode ini tidak cocok digunakan untuk dataset berdimensi tinggi dan berukuran besar dikarenakan kompleksitas komputasi yang tinggi.

6. *Stratified sampling*

Gagasan awal dari *stratified sampling* ini adalah mengeksplorasi struktur internal dan distribusi dari dataset - T , lalu memanfaatkannya untuk membagi T dalam kelompok-kelompok sampel yang relatif homogen. *Stratified sampling* sangat menguntungkan bagi dataset yang dapat dibagi menjadi beberapa kluster. Sebaliknya, untuk dataset yang didistribusikan secara uniform akan memberikan hasil yang setara dengan metode SRS. Berbagai algoritma *clustering* dapat digunakan untuk membagi T ke dalam kluster termasuk *C-means*, *fuzzy C-means*, dan *self-organizing maps*. Namun sebagian besar algoritma tersebut sensitif terhadap pemilihan parameter awal seperti jumlah kluster yang diinginkan dan pilihan variabel input.

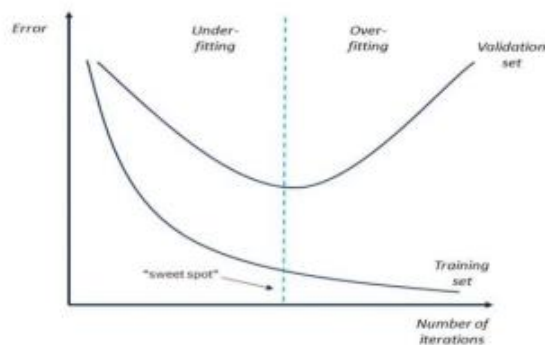
(Reitermanova, 2010)

2.3.3 *Overfitting dan Underfitting*

Dalam pengembangan model *machine learning* tantangan umum yang sering dihadapi adalah *overfitting* dan *underfitting*. Kedua permasalahan ini berkaitan langsung dengan kemampuan generalisasi model, yaitu seberapa baik model yang telah dilatih dapat melakukan prediksi akurat terhadap data yang belum pernah dilihat sebelumnya. Menurut Prajapati & Singh (2023), keduanya merupakan ancaman serius dalam pelatihan model, karena dapat menyebabkan hasil prediksi yang tidak dapat diandalkan, terutama dalam aplikasi penting seperti prediksi penyakit atau iklim.

Overfitting terjadi ketika model terlalu mempelajari noise atau fluktuasi acak pada data pelatihan, menyebabkan performa buruk pada data yang belum dilihat. Penyebab *overfitting* meliputi kurangnya data pelatihan, noise data, pelatihan berlebihan, kompleksitas model yang tinggi, dan terlalu banyak parameter, yang sering dikaitkan dengan *low bias* dan *high variance* (Maharana, et al., 2025). Selama pelatihan model, ada beberapa indikator penting yang dapat diamati. Jika *loss* validasi lebih kecil dari *loss* pelatihan, atau akurasi pelatihan lebih tinggi dari validasi, ini dapat menandakan *overfitting*. Gambar 2.3 menunjukkan *loss* pelatihan yang berhenti menurun setelah beberapa epoch menunjukkan model mulai stagnan, sedangkan *loss* pelatihan yang meningkat bisa disebabkan oleh *learning rate* yang tidak sesuai. Jika *loss*

pelatihan mendekati nol, kemungkinan model mengalami *overfitting* ekstrem. Sementara itu, *loss* validasi yang tidak stabil atau konstan menandakan bahwa model belum mampu belajar secara efektif dari data validasi.



Gambar 2.3 Training Error dan Validation Error
(Sumber: Maharana, et al., 2025)

Underfitting terjadi ketika metode yang digunakan untuk membangun model prediksi relatif sederhana dan tidak mampu mengungkap pola kompleks dari data pelatihan. Beberapa faktor yang dapat menyebabkan rendahnya kinerja model di antaranya adalah data pelatihan yang belum dibersihkan sehingga masih mengandung *noise* dan *outlier*, serta jumlah data pelatihan yang terlalu sedikit untuk mengenali pola secara menyeluruh. Selain itu, model yang terlalu sederhana dan memiliki bias yang tinggi juga berkontribusi terhadap ketidakmampuan model dalam mempelajari kompleksitas data secara efektif. Mengidentifikasi *underfitting* adalah langkah krusial dalam proses pengembangan model, karena membantu dalam memahami kapan suatu model terlalu sederhana dan tidak mampu menangkap pola dasar dalam data secara efektif. Salah satu teknik umum untuk mendeteksi *underfitting* adalah dengan menggunakan konsep *bias-variance trade-off*.

Untuk mengatasi *overfitting*, beberapa pendekatan yang dapat dilakukan antara lain menggunakan regularisasi (L1/L2) untuk menekan kompleksitas model, *dropout* untuk mengurangi ketergantungan antar neuron, serta *early stopping* untuk menghentikan pelatihan sebelum model mulai menyimpang dari generalisasi (Prajapati & Singh, 2023). Sementara itu, untuk mengatasi *underfitting* solusi yang umum digunakan adalah dengan meningkatkan kompleksitas model misalnya menambah *unit* atau *layer*, memperluas ruang pencarian *hyperparameter*, menambahkan fitur atau melakukan *feature engineering*, mengurangi regularisasi yang terlalu ketat, serta menerapkan teknik *boosting* untuk meningkatkan akurasi model yang terlalu sederhana.

2.4 Deep Learning

Deep learning (DL) merupakan cabang dari *machine learning* yang berasal dari *artificial neural network* dimana pada arsitekturnya terdiri dari banyak lapisan (*layer*) untuk memodelkan dan mempelajari pola data yang kompleks. DL telah menjadi metode komputasi yang diterapkan secara luas di berbagai bidang, seperti layanan kesehatan, pengenalan visual, analitiks teks, *cybersecurity*, dan banyak lagi (Sarker, 2021). DL memiliki beberapa contoh m seperti *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), *Long Short-Term Memory* (LSTM), dan beberapa penurunan metode lainnya.

Convolutional Neural Networks (CNN) adalah pengembangan dari DL yang didesain untuk mengelola data dua dimensi. CNN mampu mencakup beberapa lapisan konvolusi dan

pooling. Setiap lapisan dalam CNN mempertimbangkan parameter optimum untuk menghasilkan *output* yang bermakna sekaligus mengurangi kompleksitas model. CNN banyak digunakan dalam pengenalan visual, analisis gambar medis, segmentasi gambar, Pemrosesan Bahasa Alami (*Natural Language Processing/NLP*), dan masih banyak lagi.

Recurrent Neural Network (RNN) adalah jenis neural network yang dirancang untuk data deret waktu (*time-series data*). RNN menggunakan output dari langkah sebelumnya sebagai masukan sebagai masukan untuk tahap sekarang. Keluaran dari RNN bergantung pada elemen-elemen sebelumnya dalam urutan data. Namun, jaringan rekuren standar memiliki masalah *vanishing gradient*, yang membuatnya sulit untuk mempelajari urutan data yang panjang.

Long Short-Term Memory (LSTM) adalah salah satu bentuk arsitektur RNN yang menggunakan unit khusus untuk menangani *vanishing gradient*. Dalam LSTM, terdapat memori sel dalam unit LSTM yang mampu menyimpan data dalam jangka waktu yang panjang, sementara aliran informasi masuk dan keluar dari sel dikelola oleh tiga gerbang (*gates*). *Forget Gate* berfungsi untuk menentukan informasi dari kondisi sel sebelumnya yang akan diingat dan informasi mana yang akan dihapus karena dianggap tidak lagi berguna. *Input Gate* berperan dalam menentukan informasi apa yang harus masuk ke kondisi sel saat ini. Sementara itu, *Output Gate* bertugas menentukan dan mengontrol keluaran dari sel.

2.5 Artificial Neural Network

Artificial Neural Networks (ANN) adalah model komputasi yang didasarkan pada organisasi neural otak dengan parameter dinamis untuk memproses informasi secara efektif, dengan menghubungkan node input dan output secara sederhana dalam pemodelan data non-linier (Isabona, et al., 2022). Struktur ANN terdiri atas 3 lapisan yang terhubung. Lapisan pertama yaitu *output layer* terdiri dari neuron input. Neuron ini mentransmisikan data ke lapisan kedua yaitu *hidden layer*, yang kemudian mengirimkan output neuron ke *output layer* di mana ini adalah lapisan ketiga. *Neural Network* digunakan bukan tanpa sebab, kemampuannya dalam mengidentifikasi pola dan tren yang terlalu rumit untuk dikenali manusia atau metode komputasi lainnya. Karena kemampuannya dalam mengenali pola dan tren, *neural network* sangat cocok untuk model prediksi atau peramalan (Qamar & Zardari, 2023).

2.5.1 Perceptron

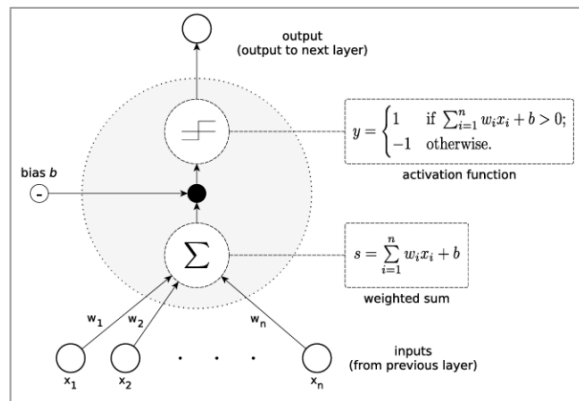
Jenis paling dasar dari *artificial neuron* adalah *perceptron*. *Perceptron* terdiri dari input internal (*bias*), input eksternal, *threshold*, dan single output eksternal. *Perceptron* menerima sebuah vektor yang terdiri dari nilai input berupa bilangan riil dimana masing-masing nilai diberi bobot menggunakan pengali. Selama fase *training*, *perceptron* mempelajari bobot berdasarkan data *training*. *Perceptron* menjumlahkan nilai input berbobot dan "aktif" jika hasilnya melebihi ambang batas tertentu. Output dari *perceptron* berupa nilai boolean, dianggap "aktif" jika outputnya adalah '1'. Nilai "nonaktif" dari *perceptron* adalah '-1', dan nilai ambang batasnya adalah '0' (Staudemeyer & Morris, 2019). *Perceptron* dapat direpresentasikan dengan Persamaan (2.19) dan (2.20):

1. Jumlah bobot

$$s = \sum_{i=1}^n w_i x_i + b \quad (2.19)$$

2. Fungsi aktivasi

$$y = \begin{cases} 1, & \text{jika } x < 0 \\ 0, & \text{jika } x \geq 0 \end{cases} \quad (2.20)$$



Gambar 2.4 Struktur *Perceptron*
(Sumber : Staudemeyer & Morris, 2019)

Algoritma training *perceptron* digunakan untuk menemukan bobot dan bias yang sesuai serta dapat mengklasifikasikan data input dengan benar. Langkah-langkah untuk algoritma tersebut adalah sebagai berikut :

1. Proses inialisasi bobot w_i dan bias b dengan nilai acak kecil atau nol.
2. Prediksi untuk setiap input , dihitung output menggunakan bobot dan bias yang sudah diperoleh dari proses inialisasi.
3. Pembaharuan bobot (2.21) dan bias (2.22) berdasarkan kesalahan prediksi. Untuk setiap pelatihan (x, y_{true}) , dimana y_{true} adalah label sebenarnya. y_{true} adalah nilai output yang diprediksi dan η adalah laju pembelajaran.

$$w_i \leftarrow w_i + \eta(y_{true} - y_{pred})x_i \quad (2. 21)$$

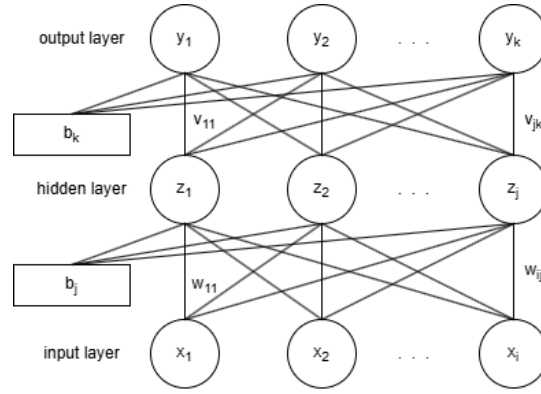
$$b \leftarrow b + \eta(y_{true} - y_{pred})x_i \quad (2. 22)$$

4. Menambahkan iterasi dengan mengulangi langkah prediksi dan pembaruan bobot untuk beberapa iterasi sampai konvergen.

2.5.2 *Multi Layer Perceptron*

Multi Layer Perceptron (MLP) adalah jenis jaringan neural network yang paling dikenal dan paling sering digunakan. Pada sebagian kasus, sinyal dalam jaringan ini ditransmisikan dalam satu arah, yaitu dari lapisan input ke lapisan output. MLP memiliki arsitektur yang dikenal sebagai *feed forward* (aliran maju) (Sarker, 2021). *Feed forward* berarti sinyal dalam jaringan ini ditransmisikan dalam satu arah, yaitu dari lapisan input ke lapisan output. MLP terdiri dari 3 layer yaitu :

1. *Input Layer* - Setiap neuron dalam layer ini mewaliki satu fitur dari data input.
2. *Hidden Layer* - Dalam layer ini terjadi pemrosesan input dari layer input ke layer berikutnya.
3. *Output Layer* - Layer ini merupakan layer terakhir yang menghasilkan output akhir dari jaringan. Jumlah neuron dalam layer ini sesuai dengan jumlah kelas atau target yang diharapkan dalam masalah klasifikasi atau regresi.



Gambar 2.5 Layer MLP

Gambar 2.5 menunjukkan MLP meneruskan input kedalam model dan mengalikannya dengan bobot w dan menambahkan bias b pada setiap layer yang akhirnya dapat digunakan untuk menemukan output model. Untuk menghitung output neuron ke- j pada *hidden layer*, digunakan Persamaan (2.23).

$$\phi_h(\sum_{i=1}^n w_{ij}x_i + b_j) = z_j \quad (2.23)$$

Di dalam perhitungan output pada *hidden layer* digunakan fungsi aktivasi ϕ_h untuk memperkenalkan non-linearitas ke dalam model yang memungkinkan jaringan untuk mempelajari dan memodelkan hubungan yang kompleks. Terdapat berbagai macam fungsi aktivasi yang dapat digunakan seperti *Sigmoid*, *Tanh*, dan ReLu.

1. Fungsi aktivasi *Sigmoid*

Sigmoid merupakan fungsi aktivasi non linier yang mengubah range nilai input menjadi antara 0 dan 1. Fungsi sigmoid ditunjukkan oleh Persamaan (2.24).

$$\sigma(x_i) = \frac{1}{1+e^{-x_i}} \quad (2.24)$$

Nilai input yang sudah tertransformasi dengan range nilai antara 0 dan 1 membuat sangat berguna untuk memperkirakan probabilitas terutama dalam tugas klasifikasi biner.

2. Fungsi aktivasi *tanh*

Fungsi tanh dapat didefinisikan sebagai rasio antara fungsi sinus hiperbolik dan cosinus hiperbolik. Fungsi ini mengubah nilai input menjadi antara -1 dan 1. Fungsi ini didefinisikan oleh Persamaan (2.25).

$$\tanh(x_i) = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} \quad (2.25)$$

3. Fungsi aktivasi ReLu

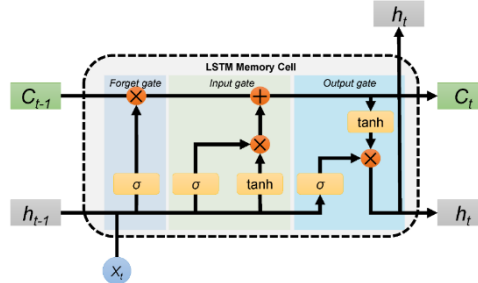
Fungsi ReLu mengubah semua nilai negatif menjadi 0 dan mempertahankan nilai positif. Fungsi ini memiliki kegunaan untuk memperkenalkan non-linearitas ke dalam sistem. Fungsi ReLu didefinisikan pada Persamaan (2.26).

$$ReLu(x_i) = \max(0, x_i) \quad (2.26)$$

Berdasarkan Persamaan (2.28), x_i merupakan input data, lalu $ReLu(x_i)$ adalah *output* yang sudah ditransformasikan.

2.6 Long Short-Term Memory

Long Short-Term Memory (LSTM) adalah jenis ANN yang digunakan untuk menangani pemodelan data deret waktu dan melakukan prediksi, LSTM memiliki kemampuan untuk menyimpan informasi dalam jangka panjang. Jaringan ini diciptakan untuk mengatasi masalah *vanishing gradient* yang ada pada RNN (Sherstinsky, 2020).

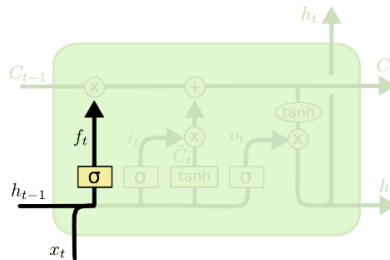


Gambar 2.6 Layer LSTM

Dapat dilihat dari Gambar 2.5 arsitektur LSTM terdapat tiga gate utama yang mengatur alur informasi, yaitu sebagai berikut

- Forget gate*, merupakan gate yang memutuskan mana yang akan dihapus dari cell
- Input gate*, merupakan gate yang memutuskan nilai dari input untuk di perbarui pada state memori
- Output gate*, merupakan gate yang memutuskan apa yang akan dihasilkan sesuai dengan input dan memori pada cell.

Berikut ini merupakan langkah-langkah perulangan dalam arsitektur LSTM :



Gambar 2.7 Langkah pada *forget gate*

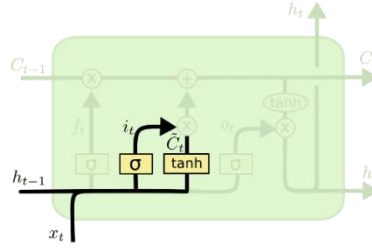
Langkah pertama dalam arsitektur LSTM adalah *forget gate*. Gate ini berfungsi sebagai pengendali berapa banyak informasi dari waktu sebelumnya yang dilupakan. Pada lapisan ini akan memproses h_{t-1} dan x_t sebagai *input* yang menghasilkan *output* berupa angka 0 dan 1 pada cell state C_{t-1} . Nilai yang mendekati 1 menunjukkan informasi akan disimpan, sedangkan nilai 0 menunjukkan informasi akan dibuang. Fungsi *forget gate* ditunjukkan oleh Persamaan (2.27).

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (2.27)$$

keterangan :

- f_t : nilai yang dihasilkan pada *forget gate* ke- t
- σ : fungsi aktivasi *sigmoid*
- W_{xf} : bobot dari input ke- i pada *forget gate*
- W_{hf} : bobot dari h_{t-1}
- x_t : input pada step waktu- t (nilai input suhu maksimum dan minimum)
- h_{t-1} : *hidden state* dari waktu sebelumnya

b_f : bias dari *forget gate*



Gambar 2.8 Langkah pada *input gate* dan *tanh layer*

Langkah selanjutnya yaitu pada *candidate memory cell* sebagai control pembuatan pada bagian *tanh layer* yang berupa kandidat nilai baru yang akan dimasukkan ke dalam *cell state* $\tilde{C}_t$. Fungsi *candidate memory cell* ditunjukkan oleh Persamaan (2.28).

$$\tilde{C}_t = \tanh (W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (2.28)$$

keterangan :

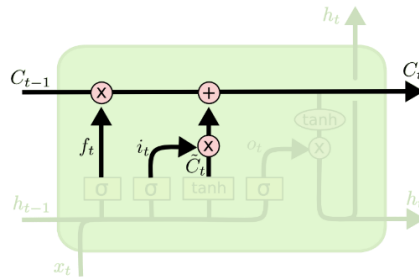
- $\tilde{C}$: model *input gate*, menentukan berapa banyak informasi baru diterima oleh *cell state*
- $\tanh$: fungsi aktivasi *hyperbolic tangent*
- W_{xc} : bobot dari input ke- i pada bobot dari *cell state*
- W_{hc} : bobot dari h_{t-1}
- h_{t-1} : *hidden state* dari waktu sebelumnya
- b_c : bias dari *cell state*

Setelah membuat kandidat dengan nilai baru, *input gate* mengatur berapa banyak informasi baru yang akan disimpan. Pada bagian lapisan *sigmoid*, *input gate* memutuskan nilai mana yang akan diperbaharui. Selanjutnya menggabungkan output dari *input gate* dan *tanh layer* untuk memperbaharui *cell state*. *Input gate* didefinisikan oleh Persamaan (2.29).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (2.29)$$

keterangan :

- σ : fungsi aktivasi *sigmoid*
- W_{xi} : bobot dari input ke- i pada *input gate*
- W_{hi} : bobot dari h_{t-1}
- h_{t-1} : *hidden state* dari waktu sebelumnya
- b_i : bias dari *input gate*



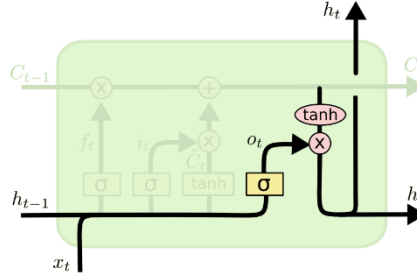
Gambar 2.9 Langkah memperbarui *cell state*

Setelah mendapatkan nilai pada *forget gate*, *input gate* dan *candidate cell state* maka dilanjutkan dengan membuat *cell state* baru C_t yang merupakan kombinasi dari Persamaan (2.27) yang dikalikan dengan *cell state* sebelumnya C_{t-1} kemudian dijumlahkan dengan hasil perkalian antara hasil yang didapat pada Persamaan (2.29) dengan hasil dari Persamaan (2.28), lalu didapatkan *cell state update* pada Persamaan (2.30).

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (2.30)$$

keterangan :

C_t : cell state
 C_{t-1} : cell state dari waktu sebelumnya
 $\odot$: operasi perkalian elemen



Gambar 2.10 Langkah pada *Output gate*

Untuk nilai pada *output gate* dapat dicari menggunakan kombinasi *input* dan *hidden state* sebelumnya serta melibatkan fungsi aktivasi *sigmoid*. *Gate* ini memutuskan apa yang menjadi output berdasarkan *cell state*. *Output gate* didefinisikan oleh Persamaan (2.31)

$$O_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (2.31)$$

keterangan:

O_t : nilai yang dihasilkan pada *forget gate* ke- t
 σ : fungsi aktivasi *sigmoid*
 W_{xo} : bobot dari input ke- i pada *output gate*
 W_{ho} : bobot dari h_{t-1}
 h_{t-1} : hidden state dari waktu sebelumnya
 b_o : bias dari *output gate*

Dengan menggunakan hasil dari Persamaan (2.31) dapat dicari nilai hidden state terbaru dengan memanfaatkan nilai dari output gate (O_t) dan hasil dari Persamaan (2.32) yang dimasukkan dalam fungsi *hyperbolic tangent*. Hasil output dari LSTM untuk *current state step* didefinisikan oleh Persamaan (2.32).

$$h_t = O_t \odot \tanh(C_t) \quad (2.32)$$

keterangan :

h_t : nilai *output* orde- t
 O_t : *output gate*
 $\odot$: operasi perkalian elemen
 $\tanh$: fungsi aktivasi *hyperbolic tangent*
 C_t : cell state

2.6.1 Loss Function

Loss function merupakan fungsi untuk mengukur seberapa baik performa yang dihasilkan model dengan membandingkan keluaran model (Masruroh, Surarso, & Warsito, 2023). Beberapa fungsi loss umum yang digunakan dalam regresi seperti *mean absolute error* (MAE), *mean square error* (MSE) dan *cross entropy*, dan lainnya. Dalam penelitian ini, model yang digunakan adalah *Long Short-Term Memory* (LSTM), sehingga fungsi kerugian MSE diterapkan untuk mengukur selisih antara output sekuens prediksi dan data observasi aktual. MSE dihitung sebagai rata-rata dari kuadrat perbedaan antara nilai yang diprediksi dan nilai

sebenarnya. Fungsi kerugian ini efektif untuk regresi dan sangat sensitif terhadap outlier, yang sesuai dengan tujuan untuk meminimalkan kesalahan prediksi pada data suhu. Secara matematis, *loss function* MSE dirumuskan oleh Persamaan (2.33):

$$\mathcal{L}(\hat{y}, y) = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (2.33)$$

Fungsi ini memberikan penalti yang lebih besar terhadap kesalahan prediksi yang lebih besar karena adanya kuadrat dari selisih. Oleh karena itu, MSE sangat sensitif terhadap outlier dan cocok digunakan ketika tujuan utama adalah meminimalkan kesalahan kuadrat rata-rata pada seluruh dataset.

2.6.2 Adam Optimizer

Optimizer adalah algoritma atau metode dalam *neural network* yang berperan penting dalam menyesuaikan parameter seperti bobot dan bias, dengan tujuan mengurangi fungsi kerugian atau meningkatkan efisiensi produksi (Mahajaya, et al., 2024). Selama proses training, bobot model diubah untuk meminimalkan *loss function*. *Loss function* mengukur seberapa akurat model dalam memprediksi data uji. Algoritma optimasi menyatukan membentuk model yang paling akurat dengan memanfaatkan bobotnya (Ramadhan & Hernadi, 2025).

Adam atau *adaptive moment optimizer* adalah algoritma optimasi yang sering digunakan dalam deep learning. Adam *optimizer* memperbarui bobot dan *learning rate* yang dapat berubah-ubah selama proses training. Adam secara efektif menggabungkan aspek-aspek terbaik dari dua optimasi sebelumnya yaitu *RMSProp* dan Momentum untuk mencapai konvergensi yang cepat dan efisien. Metode optimasi Adam diawali dengan inisiasi *learning rate* (η) dengan nilai 0.001, *exponential decay rate* (β_1 & β_2) dengan nilai yang biasa digunakan yaitu 0.9 dan 0.999, momen pertama (m_t) dan momen kedua (v_t) dengan inisiasi nilai awal 0 (Kingma & Ba, 2017). Perhitungan *gradient* menggunakan *loss function* MSE dituliskan pada persamaan (2.34)

$$g_t = \nabla \mathcal{L}(\theta_t) = \frac{\partial \mathcal{L}(\theta_t)}{\partial \theta_t} \quad (2.34)$$

Parameter θ mencakup bobot dan bias yang diperbarui untuk meminimalkan *loss function* selama pelatihan. Perhitungan *gradient* terhadap parameter menggunakan turunan berantai ditunjukkan pada Persamaan (2.35).

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_t} &= \frac{\partial}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \theta_t} \\ \frac{\partial \mathcal{L}}{\partial \theta_t} &= \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \hat{y}_i} (y_i - \hat{y}_i)^2 \cdot \frac{\partial \hat{y}_i}{\partial \theta_t} \end{aligned} \quad (2.35)$$

Turunan bagian pertama ditunjukkan Persamaan (2.36).

$$\frac{\partial}{\partial \hat{y}_i} = 2(\hat{y}_i - y_i) \quad (2.36)$$

Gradient dari fungsi loss MSE terhadap parameter θ_t , yaitu g_t , digunakan langsung sebagai masukan dalam algoritma Adam untuk melakukan pembaruan bobot. Sehingga, hasil turunan MSE pada Persamaan (2.37) menjadi dasar penting dalam proses optimasi model LSTM.

$$\frac{\partial \mathcal{L}}{\partial \theta_t} = \frac{2}{n} \sum_{i=1}^n (\hat{y}_i - y_i) \frac{\partial \hat{y}_i}{\partial \theta_t} \quad (2.37)$$

Setelah itu, nilai gradient (g_t) digunakan untuk perhitungan pembaruan parameter. Hal ini dilakukan dengan pembaruan estimasi bias momen pertama (m_t) dan momen kedua (v_t) yang secara berurutan dituliskan sebagai Persamaan (2.38) dan (2.39).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.38)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.39)$$

Estimasi bias momen sebelumnya digunakan dalam perhitungan estimasi koreksi bias momen pertama (m_t) dan momen kedua (v_t) yang ditunjukkan pada Persamaan (2.40) dan (2.41).

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.40)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.41)$$

Dengan menggunakan nilai estimasi koreksi bias momen, dapat dilakukan pembaruan parameter θ dengan menggunakan Persamaan (2.42).

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (2.42)$$

Pembaruan parameter θ melibatkan learning rate (η) yang mengatur seberapa besar perubahan model yang dapat membantu mencapai konvergen stabil dan cepat. *Learning rate* yang terlalu besar dapat menyebabkan model melewati solusi optimal, sementara *learning rate* yang terlalu kecil memperlambat konvergensi (Deswal & Dharminder, 2024).

2.6.3 Hyperparameter

Hyperparameter merupakan parameter yang tidak dapat diperoleh melalui data dan harus ditentukan sebelum proses pelatihan model dimulai. Pemilihan nilai *hyperparameter* yang optimal sangat penting untuk meningkatkan performa model dalam melakukan prediksi, proses ini disebut *hyperparameter tuning* (Satriawan & Prasetyo, 2024). Dalam penelitian ini, beberapa *hyperparameter* yang digunakan diantaranya jumlah layer, *dropout rate*, *learning rate*, dan *epoch*.

1. Dropout rate

Dropout adalah teknik yang bekerja dengan mengabaikan (drop) sejumlah unit secara acak dalam lapisan jaringan selama pelatihan, yang membantu dalam mencegah model menjadi terlalu tergantung pada unit-unit tertentu dan mempromosikan generalisasi yang lebih baik (Salehin & Kang, 2023). Secara umum, *dropout* bekerja secara acak (menyetel outputnya menjadi 0) sebagian *neuron* selama fase pelatihan. Misalkan *single linear* dalam *neural network* menghitung *output* sebagai jumlah bobot dari *input*, menggunakan fungsi aktivasi linear $f(x) = x$. Meskipun sederhana model tetap diestimasi dengan meminimalkan fungsi *loss*, salah satunya *ordinary least square* untuk mencapai hasil prediksi yang optimal.

$$L_n = \frac{1}{2} \left(t - \sum_{i=1}^n w_i I_i \right)^2 \quad (2.43)$$

$$L_D = \frac{1}{2} \left(t - \sum_{i=1}^n \delta_i w_i I_i \right)^2 \quad (2.44)$$

Persamaan (2.43) merepresentasikan fungsi *loss* yang digunakan secara umum dalam *neural network*, sementara Persamaan (2.44) berkaitan khusus dengan fungsi *loss* untuk *network dropout*. Dalam *dropout network*, Tingkat *dropout* dilambangkan dengan δ_i dimana mengikuti distribusi *Bernoulli* dengan parameter p . Hal ini membuktikan bahwa δ_i bernilai 1 dengan probabilitas p dan 0 dengan probabilitas $1 - p$. Pada *network* diatas inputan dilambangkan dengan I dan bobot yang terkait dengan inputan direpresentasikan dengan w .

2. Learning rate

Hyperparameter ini memengaruhi perubahan bobot dan bias pada model. Pemilihan nilai *learning rate* yang sesuai sangat krusial untuk memperoleh kinerja model yang maksimal karena berpengaruh langsung terhadap proses pembaruan bobot dan bias selama pelatihan. Rumus pembaruan bobot pada iterasi- t secara umum didefinisikan persamaan (2.45)

$$\theta_{t+1} = \theta - \eta \cdot \nabla \mathcal{L}(\theta_t) \quad (2.45)$$

dimana :

θ_t	: bobot pada iterasi ke- t
$\nabla \mathcal{L}(\theta_t)$	: <i>gradient fungsi loss</i> terhadap bobot
η	: <i>learning rate</i>

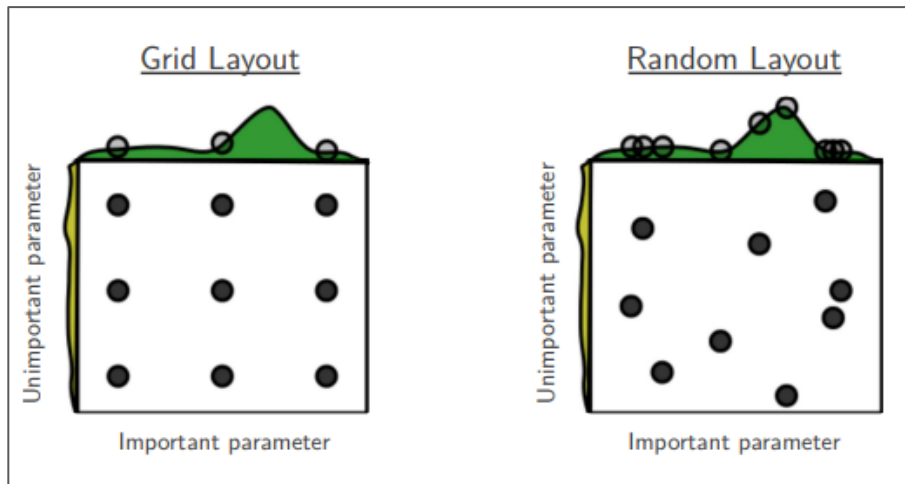
Learning rate yang terlalu besar dapat menyebabkan model melewati solusi optimal, sementara *learning rate* yang terlalu kecil memperlambat konvergensi (Deswal & Dharminder, 2024).

3. Epoch

Epoch merupakan jumlah siklus penuh seluruh data *training* yang digunakan untuk melatih sebuah model (Deswal & Dharminder, 2024). Satu *epoch* berarti seluruh data *training* telah digunakan sekali untuk memperbarui bobot model. Pelatihan dilakukan dalam ratusan hingga ribuan epoch untuk memastikan model belajar dengan baik. Jumlah *epoch* yang cukup memastikan model telah melihat seluruh data pelatihan berkali-kali untuk mempelajari pola-pola yang relevan. Namun, terlalu banyak *epoch* dapat menyebabkan *overfitting*, di mana model menghafal data pelatihan dan berkinerja buruk pada data baru yang belum pernah dilihat. Teknik seperti *early stopping* (menghentikan pelatihan ketika kinerja pada validation set mulai menurun) sering digunakan untuk mencegah *overfitting* yang disebabkan oleh terlalu banyak *epoch*.

2.6.4 Random search

Random search adalah metode optimasi *hyperparameter* yang bekerja dengan memilih nilai untuk setiap *hyperparameter* secara acak dan independen dari distribusi yang telah ditentukan sebelumnya yang mumnya distribusi seragam (Bergstra & Bengio, 2012). Metode ini dapat menangani berbagai jenis parameter, termasuk numerik, integer, dan kategorikal, serta ruang pencarian hierarkis.



Gambar 2.11 Perbandingan Performa *Random search* dan *Grid Search*
(Sumber: Bergstra & Bengio, 2012)

Gambar 2.11 menggambarkan "kegagalan *grid search*" dalam optimasi *hyperparameter* berdimensi tinggi dengan dimensionalitas efektif rendah. Berbeda dengan *grid search* yang menghabiskan komputasi pada *hyperparameter* tidak penting, *random search* lebih efisien. Metode ini memilih percobaan secara acak, membutuhkan waktu komputasi lebih singkat untuk hasil yang setara atau lebih baik. *Random search* unggul karena mampu mencari di ruang konfigurasi yang lebih luas dan fokus pada "dimensi efektif yang rendah" dari fungsi *hyperparameter* yang memengaruhi kinerja, sehingga tidak membuang-buang percobaan seperti *grid search*.

2.7 Robust Scaler

Scaling adalah suatu cara untuk mengubah rentang nilai fitur ke skala nilai data baru dengan menggunakan pendekatan tertentu yang bertujuan untuk menghindari fitur yang memiliki nilai lebih besar mendominasi fitur yang memiliki nilai lebih kecil (Pritalia, 2022). Salah satu metode *scaling* yang dapat digunakan yaitu *robust scaler*. *Robust scaler* memiliki kinerja yang baik dalam menjaga penyebaran variabel setelah proses *scaling* yang memiliki tingkat *skewness* yang tinggi dan salah satu teknik *pre-processing* yang digunakan untuk menormalkan data dengan mengurangi pengaruh *outlier* (Nugraha, et al., 2024). *Outlier* seringkali merupakan data yang tidak biasa dan mereka bisa menjadi penting karena mereka dapat mengindikasikan kesalahan dalam pengumpulan data, atau mereka bisa menjadi informasi berharga yang mengungkapkan pola atau anomali yang tidak terduga dalam data. Rumus *robust scaler* didefinisikan oleh Persamaan (2.46)

$$X_{scaled} = \frac{X - Q1(X)}{Q3(X) - Q1(X)} \quad (2.46)$$

keterangan :

X : value dari variabel X

$Q1(X)$: kuartil pertama dari variabel X

$Q3(X)$: kuartil ketiga dari variabel X

Persamaan (2.46) mengurangi setiap nilai dari variabel X dengan kuartil pertama, kemudian dibagi dengan selisih antara kuartil ketiga dan kuartil pertama. Hal ini bertujuan agar nilai-nilai yang memiliki jarak yang jauh dari kuartil pertama dan kuartil ketiga, *outlier* tidak memiliki pengaruh yang besar terhadap proses normalisasi.

2.8 Root Mean Square Error

Setelah pembuatan model prediksi menggunakan kedua metode tersebut, penting untuk mengevaluasi performa atau kinerja dari model prediksi evaluasi. Salah satu metode untuk menghitung hasil evaluasi suatu model adalah *Root Mean Square Error* (RMSE). *Root Mean Squared Error* (RMSE) merupakan metrik evaluasi yang digunakan untuk mengukur rata-rata kesalahan kuadrat antara nilai-nilai yang diobservasi (atau nilai sebenarnya) dan nilai-nilai yang diprediksi oleh model (Nugroho & Haris, 2024), dimana semakin kecil nilai RMSE maka hasil prediksi akan semakin akurat. Perhitungan RMSE dirumuskan pada Persamaan (2.47):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_t - \hat{y}_t)^2}{n}} \quad (2.47)$$

keterangan :

y_t : data aktual periode- t
 $\hat{y}_t$: data prediksi periode- t
 n : jumlah data

2.9 Ukuran Pengkoreksian Bias

NWP yang diukur berdasarkan skala lokal sering kali bias dengan keadaan cuaca skala lokal. Oleh karena itu perlu dilakukan pemrosesan secara statistika (*statistical postprocessing*) untuk mengkoreksi bias dari model NWP. Ukuran pengkoreksian bias ditunjukkan oleh ukuran *Percentage Improval* atau yang disingkat %IM (Safitri & Sutikno, 2012). %IM didefinisikan pada Persamaan (2.48):

$$\%IM = \left(\frac{RMSE_{NWP} - RMSE_{MOS}}{RMSE_{NWP}} \right) \times 100\% \quad (2.48)$$

Semakin tinggi nilai %IM maka semakin baik metode dalam mengkoreksi bias dari model NWP, sebaliknya semakin rendah nilai %IM maka dapat disimpulkan metode belum optimal dalam mengkoreksi bias model NWP.

2.10 Suhu Udara

Suhu udara adalah ukuran panas atau dinginnya udara pada suatu tempat dan waktu tertentu (Octarino, 2019). Pada tempat atau ruang lingkup yang lebih spesifik suhu atmosfer dipengaruhi oleh pancaran sinar matahari. Pancaran sinar matahari ke permukaan bumi menimbulkan adanya energi yang terdiri atas sinar ultraviolet, sinar inframerah, dan sinar cahaya. Hal inilah yang menyebabkan perbedaan suhu udara di permukaan bumi, yaitu banyaknya sinar matahari yang sampai di atmosfer bumi.

Menurut fauziah (2024) suhu udara di setiap wilayah dibagi menjadi 2 kategori, yaitu suhu udara minimum dan suhu udara maksimum. Suhu minimum merupakan suhu paling rendah pada jangka waktu tertentu dalam pengukuran, sedangkan suhu maksimum merupakan suhu paling tinggi pada jangka waktu tertentu dalam pengukuran. Menurut Utoyo faktor-faktor yang menyebabkan adanya perbedaan suhu ada 4 (Putra, 2023) yaitu sebagai berikut

1. Sudut datang sinar matahari

Sudut datang sinar matahari berpengaruh pada perbedaan suhu jika semakin tegak sudut datang maka semakin kuat penyinaran matahari hal tersebut mengakibatkan semakin tinggi suhu suatu daerah. Sebaliknya, apabila semakin miring sudut maka semakin lemah penyinaran dan suhu suatu daerah akan semakin rendah

2. Lama waktu penyinaran
Semakin lama matahari menyinari suatu daerah, maka makin tinggi pula suhu udara di suatu daerah
3. Ketinggian tempat
Semakin tinggi suatu daerah dari permukaan laut, maka rata-rata suhu udara di daerah tersebut akan rendah. Begitu juga sebaliknya, apabila ketinggian daerah tersebut rendah maka rata-rata suhu udara di daerah tersebut akan tinggi
4. Kondisi geografis wilayah
Indonesia merupakan negara kepulauan yang sebagian besar wilayahnya pasti bersinggungan dengan laut. Hal tersebut menyebabkan perbedaan suhu udara harian di seluruh wilayah Indonesia tidak terlalu tinggi.

2.11 Hasil Penelitian Terdahulu

Penelitian tentang model prediksi cuaca telah banyak dilakukan dengan berbagai metode atau teknik. Tabel 2.1 menyajikan ringkasan penelitian terdahulu tentang model prediksi cuaca menggunakan Dataset INA-NWP, dengan berbagai penggunaan parameter NWP seperti suhu udara dan kelembapan rata-rata.

Tabel 2.1 Daftar Penelitian Terdahulu yang Menggunakan Dataset INA-NWP dalam Membuat Model Prakiraan

No	Peneliti	Metode dan Variabel	Temuan	Kekurangan
1.	Putra, 2023	<i>Geostatistical Ouput Pertubation</i> (Temperatur maksimum dan minimum)	GOP mampu mengurangi bias dari luaran NWP yang dapat dilihat dari penurunan nilai RMSE yang sebelumnya 4,35 menjadi 2,03 pada temperatur maksimum, sedangkan pada temperatur minimum bernilai 1,78 menjadi 1,44	Akurasi prakiraan GOP untuk temperatur maksimum cukup rendah dan prakiraan yang kurang akurat
2.	Sari, 2018	<i>Ensemble Model Output Statistics</i> (Temperatur maksimum, minimum, dan kelembapan rata-rata)	EMOS belum cukup signifikan dalam meningkatkan akurasi prediksi cuaca dibandingkan hasil prediksi model MOS terutama PLSR dan regresi stepwise.	Hasil prediksi anggota <i>ensemble</i> pada stasiun Hasanuddin dan Pattimura dibandingkan dengan data observasi, diperoleh pola data hasil prediksi setiap <i>ensemble member</i> mampu menangkap pola data observasi, namun masih

No	Peneliti	Metode dan Variabel	Temuan	Kekurangan
				menghasilkan bias yang cukup besar.
3.	Cahyoko, 2023	<i>Spatial Ensemble Model Output Statistics</i> (Suhu maksimum, suhu minimum, dan kelembapan)	SEMOs lebih baik digunakan dalam kalibrasi dan prakiraan suhu minimum dibandingkan dengan suhu maksimum	Dalam hal akurasi dan presisi prakiraan cuaca SEMOS tidak lebih baik daripada EMOS.
4.	Hakim, 2018	<i>Bayesian Model Averaging</i> (Total Rainfall, Wind)	Hasil prediksi suhu maksimum, suhu minimum, dan kelembapan menggunakan pendekatan MOS didapatkan nilai RMSE pada model PLS dan Regresi Ridge lebih baik ketimbang model PCR serta Stepwise Regression.	Dari hasil evaluasi prediksi didapatkan bahwa prediksi deterministik BMA tidak menghasilkan RMSE lebih kecil dibandingkan hasil prediksi beberapa model MOS terutama PLS, Regresi Ridge, dan stepwise regression
5.	Safitri & Sutikno, 2012	<i>Model Output Statistics dengan Pursuit Regression</i> (Suhu maksimum, suhu minimum, dan kelembapan rata-rata)	Hasil validasi model MOS menunjukkan kekonsistenan bahwa semakin banyak fungsi (m) dalam model PPR akan menaikkan nilai RMSEP dan MAPE. Nilai %IM mencapai 86%, berarti model MOS dapat mengoreksi bias mencapai 86%	Masalah dalam pemodelan MOS adalah keadaan data NWP yang berdimensi besar karena diukut dalam <i>grid</i> yang cukup luas.

Dipilihnya metode SVR dan LSTM didasarkan pada keterbatasan metode *Model Output Statistics* yang tidak mampu menangani data berdimensi besar secara optimal. MOS menggunakan pendekatan statistik sederhana yang kurang efektif dalam menangkap pola kompleks serta hubungan non-linier dalam data cuaca. Tabel 2.2 menyajikan penelitian terdahulu yang menggunakan kedua metode tersebut.

Tabel 2.2 Daftar Penelitian Terdahulu Menggunakan Model SVR dan LSTM dalam Membuat Model Prakiraan

No	Peneliti	Metode	Hasil Terkait
1.	Amanullah, et al., 2023	<i>Support Vector Regresion</i> (SVR)	Hasil downscaling pada data IMERG dengan menggunakan SVR memiliki tingkat akurasi yang baik dengan kesalahan prediksi curah hujan yang cukup kecil.
2.	Muhammad & Irawan, 2023	<i>Long Short-Term Memory</i> (LSTM)	LSTM memiliki performa yang terbaik dengan nilai RMSE dan MAE berturut-turut sebesar 0.98162 dan 0.68847, tetapi data yang dijadikan penelitian kurang panjang paling tidak lebih dari 5 tahun.
3.	Zaytar & Amrani, 2016	<i>Long Short-Term Memory</i> (LSTM)	Jaringan deep LSTM mampu meramalkan variabel cuaca di 9 kota di Maroko secara umum dengan akurasi yang baik. Hal ini dibuktikan melalui grafik perbandingan antara nilai aktual dan nilai prediksi selama 24 jam, yang menunjukkan bahwa model berhasil mengikuti pola tren data. Namun, terdapat perbedaan waktu yang cukup signifikan antara nilai prediksi dan nilai aktual pada beberapa interval tertentu. Hasil serupa juga ditunjukkan pada grafik perbandingan antara nilai aktual dan nilai prediksi untuk periode 72 jam.
4.	Siregar, 2022	<i>Support Vector Regresion</i> (SVR)	Metode SVR cukup baik untuk meramalkan curah hujan Kota medan jika diperhitungkan dengan nilai parameter yang tepat dan proses metode yang benar. Adapun nilai RMSE terbaik sebesar 0.039 dengan nilai <i>gamma</i> sebesar 0.0005, <i>complexity</i> sebesar 0.0001, dan nilai <i>epsilon</i> sebesar 1.
5.	A.Rohman, 2024	<i>Support Vector Regresion</i> (SVR) dan <i>Long Short-Term Memory</i> (LSTM)	Hasil evaluasi keakurasian antara metode SVR dan LSTM dalam kaitannya dengan prediksi curah hujan LSTM sedikit lebih unggul. Hasil penelitian masih perlu ditingkatkan mengenai kondisi geografis dan perlu adanya variabel dependen yang lain.

BAB III

METODE PENELITIAN

3.1 Sumber Data

Data yang digunakan pada penelitian ini yaitu data sekunder bersumber dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG). Data tersebut terdiri atas data observasi suhu udara (Suhu maksimum dan suhu minimum) harian pada periode 1 Maret 2021 hingga 25 Maret 2022 atau selama 387 hari. Dalam membentuk model *Support Vector Regression* dan *Long Short-Term Memory* untuk masing masing suhu, maka diperlukan data harian NWP model Indonesia *Numerical Weather Prediction* (INA-NWP) pada periode yang sesuai dengan observasi. Stasiun pengamatan yang menjadi fokus dalam penelitian ini adalah 3 stasiun meteorologi yang tersebar di Surabaya dan sekitarnya. Dipilihnya 3 stasiun meteorologi seperti pada Tabel 3.1 didasarkan pada pemilihan stasiun yang saling berjauhan secara geografis, sehingga diharapkan dapat memberikan data yang lebih representatif untuk pelatihan dan evaluasi model. Sebaran lokasi ketiga stasiun tersebut ditunjukkan pada Gambar 3.1.

Tabel 3.1 Lokasi Pengamatan Stasiun Meteorologi

No.	Stasiun Meteorologi (s)	Koordinat	
		Latitude	Longitude
1	Perak I	7° 13' 36.9762" LS	112° 42' 55.0764" BT
2	Sambikerep	7° 15' 46.9974" LS	112° 38' 35.5812" BT
3	Juanda	7° 20' 55.269" LS	112° 43' 31.4178" BT



Gambar 3.1 Sebaran Lokasi Stasiun Meteorologi di Surabaya

3.2 Variabel Penelitian

Untuk pembentukan model suhu minimum dan maksimum, masing-masing dilakukan pemodelan secara terpisah karena luaran NWP yang bersifat deterministik, yang hanya mampu menghasilkan satu prakiraan untuk setiap kondisi awal. Secara keseluruhan, akan dibangun sebanyak 12 model. Data yang digunakan pada penelitian ini meliputi observasi permukaan yang ditampilkan pada Tabel 3.2.

Tabel 3.2 Variabel Respon Penelitian

No.	Variabel	Keterangan	Satuan
1	y_1	Suhu Maksimum (T_{max})	Celcius
2	y_2	Suhu Minimum (T_{min})	Celcius

Variabel prediktor terdiri dari data *Indonesia Numerical Weather Prediction* (INA-NWP) pada model *Weather Research and Forecasting Environmental Modelling System* (WRF-EMS). Parameter NWP WRF-EMS dalam pemodelan *Support Vector Regression* dan *Long Short-Term Memory* ditampilkan pada Tabel 3.3.

Tabel 3.3 Parameter NWP WRF-EMS

No.	Variabel	Keterangan	Satuan
1	x_1	Suhu Maksimum INA-NWP (T_{maxscr})	Celcius
2	x_2	Suhu Minimum INA-NWP (T_{minscr})	Celcius

Adapun struktur data yang diolah untuk meramalkan suhu maksimum dan minimum terletak pada Tabel 3.4 dan 3.5 yang merupakan struktur data untuk stasiun meteorologi s yang berada pada koordinat lintang-bujur tertentu. Respon $y_{t,s}$ adalah observasi suhu udara pada hari ke- t pada stasiun meteorologi s dan variabel $x_{t,s}$ yang merupakan luaran/prakiraan model NWP untuk suhu udara pada *grid* terdekat atau *grid* yang bersesuaian dari stasiun meteorologi s pada hari ke- t .

Tabel 3.4 Struktur Data untuk Pemodelan Suhu Maksimum pada Stasiun ke- s

Hari ke- t	Tanggal	$y_{1,t,s}$	$x_{1,t,s}$
1	01/03/2021	$y_{1,1,s}$	$x_{1,1,s}$
2	02/03/2021	$y_{1,2,s}$	$x_{1,2,s}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
385	20/03/2022	$y_{1,385,s}$	$x_{1,385,s}$
386	21/03/2022	$y_{1,386,s}$	$x_{1,386,s}$
387	22/03/2022	$y_{1,387,s}$	$x_{1,387,s}$

Indeks s meliputi stasiun meteorologi Perak I, Sambikerep dan Juanda.

Tabel 3.5 Struktur Data untuk Pemodelan Suhu Minimum pada Stasiun ke- s

Hari ke- t	Tanggal	$y_{2,t,s}$	$x_{2,t,s}$
1	01/03/2021	$y_{2,1,s}$	$x_{2,1,s}$
2	02/03/2021	$y_{2,2,s}$	$x_{2,2,s}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
385	20/03/2022	$y_{2,385,s}$	$x_{2,385,s}$
386	21/03/2022	$y_{2,386,s}$	$x_{2,386,s}$
387	22/03/2022	$y_{2,387,s}$	$x_{2,387,s}$

3.3 Metode Analisis Data

Berikut merupakan langkah-langkah yang dilakukan untuk mencapai tujuan penelitian. Dalam mencapai tujuan penelitian dimana memodelkan prakiraan NWP dengan suhu udara digunakan langkah sebagai berikut.

1. Analisis prediksi data suhu maksimum dan suhu minimum menggunakan metode *Support Vector Regression* (SVR).

- a. Menentukan variabel untuk model SVR, variabel respon yang akan digunakan yaitu data observasi harian suhu lokal sedangkan variabel prediktor yaitu data harian INA-NWP.
- b. Membagi porsi data menjadi 90% untuk *training* dan 10% untuk *testing*.
- c. Melakukan penskalaan data untuk variabel prediktor dengan menggunakan metode *robust scaler* berdasarkan Persamaan (2.46).
- d. Membangun model SVR untuk prediksi suhu maksimum dan minimum menggunakan data *training*.
 1. Mencari *hyperparameter* terbaik dengan menggunakan *Random search*. Tabel 3.6 menunjukkan *value predefined* untuk dilakukan *random search*.

Tabel 3.6 Pencarian *hyperparameter* dan fungsi *kernel* optimal menggunakan *random search*

No	Parameter pada SVR	Nilai yang akan diuji
1	<i>C</i>	0,1, 1; 10; 100; 1000
2	<i>Epsilon</i>	0,001; 0,01; 0,1; 0,2; 0,5; 1
3	<i>Kernel</i>	'rbf', 'linear', 'poly', 'sigmoid'
4	<i>Gamma</i>	'scale'; 'auto'; 0,001; 0,01; 0,1; 1; 10

2. Menerapkan *hyperparameter* dan jenis *kernel* optimal yang ditemukan dari *random search* ke model SVR.
3. Membangun model SVR dengan *hyperparameter* dan jenis *kernel* yang telah optimal.
4. Melakukan prediksi untuk suhu maksimum dan minimum menggunakan data *testing*.
- e. Menghitung nilai evaluasi prediksi menggunakan *Root Mean Square Error* berdasarkan Persamaan (2.47).
- f. Menghitung nilai *percentage improval* berdasarkan Persamaan (2.48) untuk melihat seberapa efektif SVR dalam mengkoreksi bias model NWP.
2. Analisis prediksi data suhu maksimum dan minimum menggunakan metode *Long Short-Term Memory* (LSTM)
 - a. Menentukan variabel untuk model LSTM, variabel respon yang akan digunakan yaitu data observasi harian suhu lokal sedangkan variabel prediktor yaitu data harian INA-NWP.
 - b. Melakukan penskalaan data untuk variabel prediktor dengan menggunakan metode *robust scaler* berdasarkan Persamaan (2.46).
 - c. Membagi data secara berbasis waktu porsi data menjadi 90% untuk *training* dan 10% untuk *testing*.
 - d. Membentuk *sequences* dari data yang sudah di skalakan serta menggunakan 3 *time step* untuk 1 *sample*.
 - e. Mendefinisikan arsitektur model LSTM dengan *hyperparameter* yang akan dioptimalkan.
 1. Melakukan pelatihan model LSTM untuk setiap kombinasi *hyperparameter* yang diuji dalam *random search* menggunakan data *training* serta mengevaluasinya menggunakan data *testing*.
 2. Mencari *hyperparameter* optimal dengan menggunakan *random search* untuk meningkatkan kinerja LSTM. Tabel 3.7 menunjukkan *value predefined* untuk dilakukan *random search*.

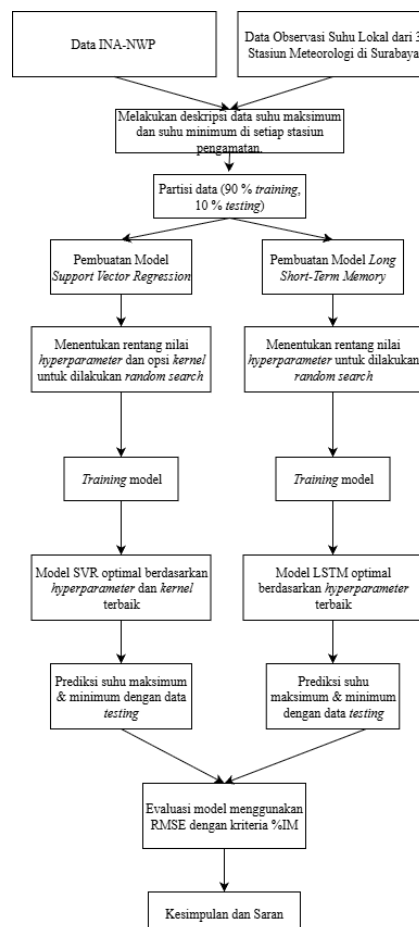
Tabel 3.7 Pencarian *hyperparameter* optimal menggunakan *random search*

No	Parameter pada LSTM	Nilai yang akan diuji
1	Fungsi Aktivasi	'Relu', 'Tanh', 'Sigmoid'
2	Units	16-256
3	Layer	1, 2
4	<i>Learning rate</i>	1×10^{-4} - 1×10^{-2}
5	<i>Dropout rate</i>	0,0 – 0,5

3. Mendapatkan model LSTM terbaik dari pencarian *random search* berdasarkan *hyperparameter* optimal.
4. Melakukan prediksi untuk suhu maksimum dan minimum menggunakan model LSTM terbaik.
- f. Menghitung nilai evaluasi prediksi menggunakan RMSE berdasarkan Persamaan (2.47).
- g. Menghitung nilai *percentage improval* berdasarkan Persamaan (2.48) untuk melihat seberapa efektif LSTM dalam mengkoreksi bias model NWP.
3. Membandingkan hasil nilai *percentage improval* antara metode SVR dengan metode LSTM.
4. Menarik kesimpulan.

3.4 Diagram Alir

Agar mempermudah pemahaman, ilustrasi mengenai metode analisis diuraikan oleh diagram alir pada Gambar 3.2.

**Gambar 3.2** Diagram Alir Pembentukan Model *Support Vector Regression* dan *Long Short-Term Memory*

BAB IV

ANALISIS DAN PEMBAHASAN

Pada bab ini dibahas karakteristik dari suhu maksimum dan minimum, sampai diperoleh model untuk memprediksi nilai suhu maksimum dan minimum dengan menggunakan metode *Support Vector Regression* serta *Long Short-Term Memory*. Selanjutnya dilakukan evaluasi model prediksi tiap metode berdasarkan nilai *Root Mean Square Error* (RMSE) untuk kemudian dibandingkan nilai tersebut dengan kriteria *Improval Percentage* (IM%).

4.1 Deskripsi Suhu Udara

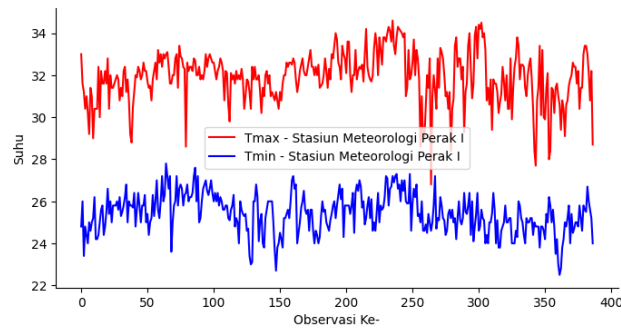
Suhu maksimum tertinggi tercatat oleh alat pengukur suhu pada Stasiun Meteorologi Juanda dalam satu hari pengamatan adalah 35,60 °C, sedangkan nilai minimum juga tercatat pada Stasiun Meteorologi Juanda sebesar 26,70 °C, seperti disajikan pada Tabel 4.1. Hal ini menunjukkan bahwa meskipun Stasiun Juanda mencatat suhu maksimum tertinggi, rentang suhu maksimum harian di sana juga cukup luas.

Tabel 4.1 Nilai Rata-Rata, Standar Deviasi, Nilai Maksimum, dan Minimum dari Suhu Maksimum serta Minimum di 3 Stasiun Meteorologi

Stasiun Meteorologi (s)	Rata-rata	Standar Deviasi	Nilai Maksimum	Nilai Minimum
---(T_{max})---				
Perak I	31,92	1,22	34,60	26,80
Sambikerep	31,71	1,36	34,84	26,80
Juanda	32,57	1,02	35,60	26,70
---(T_{min})---				
Perak I	25,44	0,93	27,80	22,50
Sambikerep	25,27	0,95	27,80	22,11
Juanda	24,77	0,98	28,10	21,00

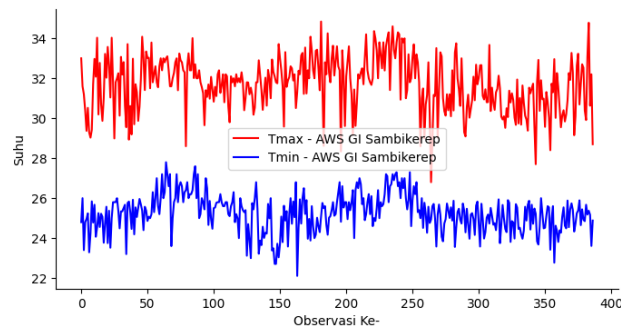
Selanjutnya pada suhu minimum tertinggi tercatat pada Stasiun Meteorologi Juanda sebesar 28,10 °C, serta nilai minimum juga tercatat pada Stasiun Meteorologi Juanda sebesar 21 °C. Ini menandakan bahwa Stasiun Juanda memiliki rentang suhu minimum harian yang signifikan. Sementara itu, rata-rata tertinggi untuk suhu maksimum tercatat pada Stasiun Meteorologi Juanda sebesar 32,57 °C yang berarti suhu maksimum rata-rata daerah sekitar pengamatan stasiun meteorologi tersebut terbilang cukup tinggi dibandingkan stasiun lainnya. Suhu setinggi ini mengindikasikan kondisi cuaca yang cenderung panas. Sementara itu, rata-rata tertinggi untuk suhu minimum justru tercatat pada Stasiun Meteorologi Perak I, yaitu 25,44 °C. Ini berarti suhu minimum rata-rata daerah sekitar pengamatan relatif lebih tinggi dibandingkan stasiun pengamatan lain. Sebaliknya, Stasiun Meteorologi Juanda justru memiliki suhu rata-rata suhu minimum terendah 24,77 °C. Hal ini menandakan perbedaan karakteristik termal antar lokasi stasiun, dimana Stasiun Juanda mengalami rentang suhu harian yang lebih besar antara siang yang panas dan malam yang lebih sejuk.

Suhu maksimum di stasiun Perak I lebih berfluktuatif daripada suhu minimum disajikan pada Gambar 4.1. Gambar tersebut menampilkan tren pergerakan serta untuk melihat pola umum dari suhu maksimum dan minimum selama periode 387 hari pengamatan yaitu 1 Maret 2021 hingga 22 Maret 2022 di ke-3 stasiun meteorologi di wilayah Surabaya dan sekitarnya.



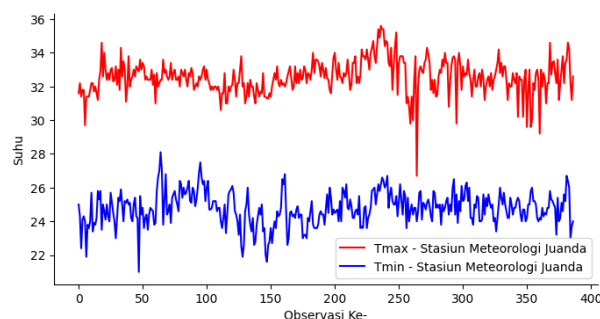
Gambar 4.1 Pola Suhu Maksimum dan Minimum Stasiun Meteorologi Perak I Selama Periode 387 Hari

Suhu maksimum berfluktuasi antara 30°C hingga 34°C dengan puncak menonjol di sekitar pengamatan ke-200 hingga ke-250 yang mencapai lebih dari 34°C dengan variasi yang lebih lebar. Sementara itu, suhu minimum terlihat lebih stabil berfluktuasi di antara 24°C hingga 27°C meskipun beberapa kali turun mendekati 23°C. Perbedaan suhu harian cenderung melebar dikarenakan rentang suhu minimum tidak naik secara proporsional.



Gambar 4.2 Pola Suhu Maksimum dan Minimum Stasiun AWS GI Sambikerep Selama Periode 387 Hari

Gambar 4.2 menunjukkan suhu maksimum secara konsisten di atas suhu minimum, dengan suhu maksimum umumnya berfluktuasi antara 30°C hingga 34°C dan beberapa puncak yang mencapai sekitar 34,5°C serta sering menunjukkan variasi yang beragam. Sementara itu, suhu minimum cenderung lebih stabil berfluktuasi antara 24°C hingga 27°C, meskipun sesekali turun hingga mendekati 22 °C. Jarak yang relatif stabil antara kedua garis mengindikasikan rentang suhu harian yang konsisten, namun fluktuasi suhu maksimum yang lebih dinamis menunjukkan bahwa kondisi siang hari dapat sangat bervariasi dan rentang suhu harian dapat berubah tergantung pada kondisi cuaca.



Gambar 4.3 Pola Suhu Maksimum dan Minimum Stasiun Juanda Selama Periode 387 Hari

Gambar 4.3 menunjukkan pola perubahan suhu maksimum dan minimum harian yang tercatat di Stasiun Meteorologi Juanda selama sekitar 387 hari pengamatan, dimana suhu

maksimum berfluktuasi antara 30°C hingga 34°C dan mencapai puncaknya mendekati 36°C sekitar pengamatan ke-250 sebelum mengalami penurunan secara signifikan. Sementara itu, suhu minimum berada dalam *range* yang lebih sempit berada diantara 22°C hingga 27°C dengan fluktuasi yang lebih acak dan beberapa penurunan suhu hingga 21°C. Jarak yang konsisten antara kedua garis mengindikasikan perbedaan suhu harian yang signifikan serta suhu maksimum memiliki variasi yang lebih menonjol dibandingkan suhu minimum

4.2 Pemodelan *Support Vector Regression*

Dalam pengembangan model *Support Vector Regression* (SVR) untuk prediksi suhu membuat model prediksi suhu, penting untuk mempertimbangkan pemilihan *kernel*, *C*, *gamma*, dan *epsilon* yang optimal guna meningkatkan akurasi model. Untuk mencapai performa yang optimal diperlukan proses pencarian *kernel* dan parameter terbaik melalui teknik *random search*. Proses pencarian *hyperparameter* dan fungsi *kernel* dilakukan dengan iterasi sebanyak 100 serta untuk memastikan evaluasi yang *robust* setiap kombinasi dievaluasi menggunakan *5-fold cross-validation*. Dengan menggunakan konfigurasi tersebut didapatkan model SVR yang mampu memprediksi suhu maksimum dan minimum secara akurat. Berikut adalah hasil dari pencarian *random search* untuk masing-masing stasiun meteorologi.

4.2.1 Pemodelan *Support Vector Regression* untuk Stasiun Meteorologi Perak I

Pada tahap awal pemodelan prediksi untuk suhu maksimum dengan menggunakan kombinasi seperti pada Tabel 3.6, hasil prediksi menunjukkan standar deviasi yang sangat rendah dibandingkan dengan standar deviasi aktual. Kondisi ini mengindikasikan bahwa model belum mampu menangkap pola dan karakteristik variasi data yang sebenarnya, atau dengan kata lain terjadi *underfitting*. Untuk mengatasi masalah tersebut, ruang pencarian *hyperparameter* dan fungsi *kernel* kemudian dirubah menjadi; '*C*': [1, 10, 100, 1000], '*epsilon*': [1, 2, 5, 10], '*kernel*': ['rbf', 'poly'], '*gamma*': ['scale', 'auto', 1, 10]. Hasil kombinasi *hyperparameter* dan fungsi *kernel* optimal yang disajikan pada Tabel 4.2 adalah yang didapatkan dari proses *random search* menggunakan ruang pencarian yang telah diubah.

Tabel 4.2 *Hyperparameter* dan Fungsi *Kernel* Optimal untuk Pemodelan Stasiun Meteorologi Perak I

Variabel	Parameter Optimal			
	<i>C</i>	<i>Epsilon</i>	<i>Kernel</i>	<i>Gamma</i>
T_{max}	1	1	'rbf'	1
T_{min}	0,10	0,50	'rbf'	1

Secara keseluruhan, kedua model prediksi di Stasiun Meteorologi Perak I sama-sama menemukan *kernel* 'rbf' dengan nilai *gamma* yang sama yaitu 1 sebagai nilai yang optimal menunjukkan bahwa hubungan non-linier paling baik ditangkap dengan pengaruh sample yang seimbang. Namun, perbedaan nilai *C* dan *epsilon* mengindikasikan bahwa karakteristik *error* dan kebutuhan regularisasi sedikit berbeda antara prediksi suhu maksimum dan minimum di stasiun ini. Prediksi suhu minimum membutuhkan toleransi *error* yang lebih ketat tetapi dengan penalti yang lebih fleksibel untuk *error* di luar batas. Hasil prediksi di lampirkan pada Lampiran 9 dan Lampiran 11 yang menunjukkan pola yang sangat mirip dengan hasil pemodelan suhu minimum di Stasiun Juanda, yaitu prediksi yang sangat landai dan hampir datar, dengan variasi yang sangat minim. Hal ini mengindikasikan bahwa model SVR mengalami *underfitting* yang signifikan, di mana model gagal menangkap fluktuasi naik-turun dari data aktual. Dalam kasus ini, model tampak hanya mempelajari nilai rata-rata dari data historis, tanpa mampu menyesuaikan diri terhadap pola musiman atau harian. Akibatnya, prediksi yang dihasilkan

tidak akurat dalam merepresentasikan variasi aktual suhu, baik untuk suhu minimum maupun maksimum.

4.2.2 Pemodelan *Support Vector Regression* untuk Stasiun AWS GI Sambikerep

Tabel 4.3 menunjukkan prediksi suhu maksimum *hyperparameter* optimalnya adalah $C=1$, $\epsilon=1$, $\text{Kernel}='rbf'$, dan $\gamma=1$. Sementara itu, untuk prediksi suhu minimum, model SVR menemukan *hyperparameter* optimal yang sedikit berbeda $C = 0,1$, $\epsilon = 0,5$, $\text{kernel} = 'rbf'$, dan $\gamma=1$.

Tabel 4.3 *Hyperparameter* dan Fungsi *Kernel* Optimal untuk Pemodelan Stasiun AWS GI Sambikerep

Variabel	Parameter Optimal			
	C	ϵ	Kernel	γ
T_{max}	1	1	'rbf'	1
T_{min}	0,10	0,50	'rbf'	1

Kedua model menggunakan *kernel* 'rbf' dengan nilai γ yang sama, menunjukkan bahwa fungsi ini efektif dalam menangkap hubungan non-linear di stasiun ini. Namun, perbedaan nilai C dan ϵ mengindikasikan bahwa model suhu maksimum lebih ketat mengoreksi terhadap kesalahan dengan C dan ϵ yang lebih besar. Sementara model suhu minimum cenderung lebih toleran terhadap variasi data dan memberikan ruang lebih besar untuk generalisasi, sebagaimana ditunjukkan oleh nilai C yang kecil dan ϵ yang lebih rendah. Sama halnya dengan hasil prediksi SVR terhadap suhu minimum di Stasiun Juanda, pemodelan suhu minimum di Stasiun Sambikerep juga menunjukkan hasil yang cenderung datar dengan variasi yang lebih terbatas, sebagaimana ditampilkan pada Lampiran 16. Sebaliknya, pada pemodelan suhu maksimum di Stasiun Sambikerep yang ditunjukkan pada Lampiran 14 serupa dengan hasil pemodelan suhu maksimum di Stasiun Juanda model SVR mampu menangkap sebagian besar pola data.

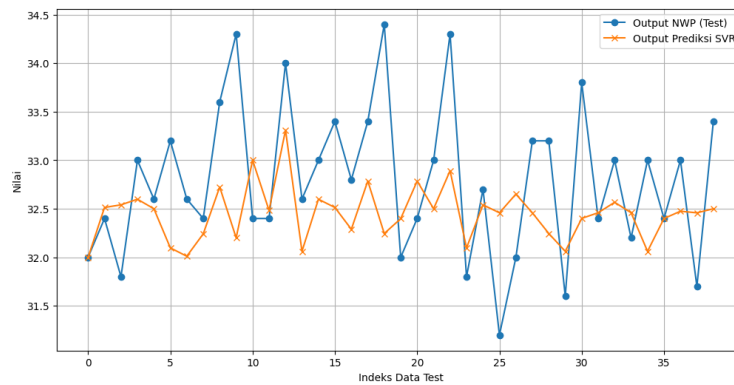
4.2.3 Pemodelan *Support Vector Regression* untuk Stasiun Meteorologi Juanda

Tabel 4.4 menunjukkan prediksi suhu maksimum model SVR mencapai kinerja optimalnya dengan C sebesar 1, ϵ sebesar 0,001, menggunakan *kernel* 'rbf', dan γ sebesar 10. Konfigurasi ini mengindikasikan model menuntut presisi sangat tinggi dengan toleransi *error* minimal. Penggunaan *kernel* 'rbf' dengan nilai γ yang besar juga membuat model lebih fokus pada pola lokal dalam data, sehingga mampu menangkap detail-detail kecil yang penting dalam prediksi suhu maksimum.

Tabel 4.4 *Hyperparameter* dan Fungsi *Kernel* Optimal untuk Pemodelan Stasiun Meteorologi Juanda

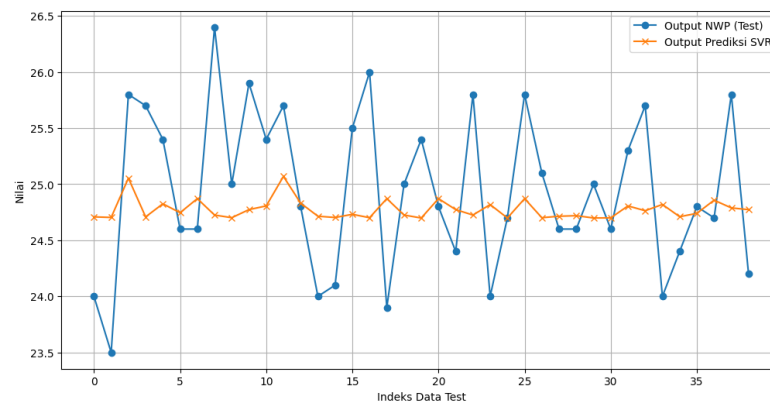
Variabel	Parameter Optimal			
	C	ϵ	Kernel	γ
T_{max}	1	0,001	'rbf'	10
T_{min}	1	0,10	'rbf'	0,10

Gambar 4.5 menunjukkan bahwa hasil prediksi SVR mampu mengikuti pola umum dari data aktual NWP. Model SVR menghasilkan pola prediksi yang jauh lebih stabil dan mendekati nilai rata-rata aktual, meskipun tidak sepenuhnya mengikuti pola dari output NWP. Kestabilan ini menandakan bahwa SVR berhasil mereduksi *noise* dan ketidakteraturan dari NWP, sekaligus memberikan estimasi suhu maksimum yang lebih akurat.



Gambar 4.4 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model SVR di Stasiun Juanda

Sementara itu, untuk prediksi suhu minimum, model SVR menemukan C sebesar 1, ϵ sebesar 0,1, kernel 'rbf', dan γ sebesar 0,1 sebagai optimal. Perbandingan menunjukkan bahwa kedua model sama-sama menggunakan kernel 'rbf' dan nilai C yang sama, namun model suhu minimum memiliki toleransi error yang lebih besar dan pengaruh sampel yang lebih luas dibandingkan suhu maksimum. Hal ini menunjukkan bahwa karakteristik pemodelan suhu maksimum membutuhkan ketelitian lebih tinggi terhadap kesalahan, sementara suhu minimum dapat lebih fleksibel dalam toleransi error -nya untuk mencapai kinerja optimal.



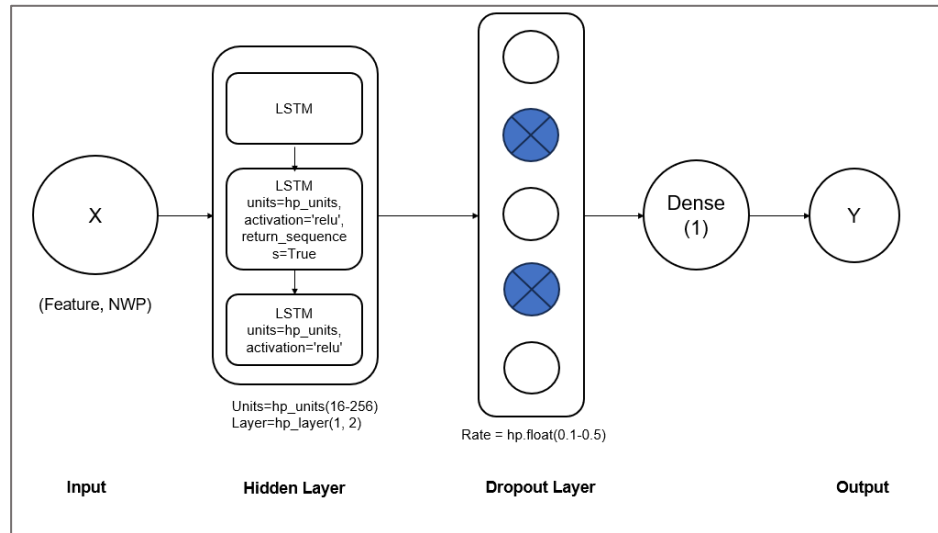
Gambar 4.5 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model SVR di Stasiun Juanda

Pola prediksi untuk suhu minimum, seperti yang ditunjukkan pada Gambar 4.5, menampilkan karakteristik yang sangat berbeda. Hasil prediksi SVR terlihat sangat landai dan hampir datar, dengan variasi yang sangat minim. Ini menunjukkan bahwa model mengalami *underfitting* yang signifikan, di mana gagal untuk menangkap fluktuasi naik-turun dari data aktual. Model pada dasarnya hanya mempelajari nilai rata-rata dari data dan tidak mampu memodelkan pola spesifik dari suhu minimum, menghasilkan prediksi yang sangat tidak akurat terhadap variasi harian.

4.3 Pemodelan Model *Long Short-Term Memory*

Pengembangan model *Long Short-Term Memory* (LSTM) untuk prediksi suhu maksimum dan minimum penting untuk mempertimbangkan pemilihan parameter yang optimal guna meningkatkan akurasi model. Sama halnya seperti SVR untuk mencapai performa yang optimal diperlukan proses pencarian parameter terbaik melalui teknik *random search*. Dengan

menggunakan teknik tersebut didapatkan model LSTM yang mampu memprediksi suhu maksimum dan minimum secara akurat. Proses pencarian *hyperparameter* dilakukan dengan *epoch* sebanyak 100 serta *trial* sebanyak 10 kali. Dalam setiap *trial*, setiap kombinasi *hyperparameter* dieksekusi sebanyak 3 kali untuk mendapatkan evaluasi kinerja yang lebih *robust*. Arsitektur LSTM yang digunakan dalam pengembangan model ditunjukkan Gambar 4.6.



Gambar 4.6 Arsitektur LSTM untuk Model Prediksi

Setelah dilakukan proses pelatihan (*training*) menggunakan arsitektur tersebut, berikut disajikan hasil pencarian parameter terbaik melalui *random search* untuk masing-masing stasiun meteorologi.

4.3.1 Pemodelan *Long Short-Term Memory* untuk Stasiun Meteorologi Perak I

Tabel 4.5 menunjukkan dua set *hyperparameter* optimal yang berbeda. Model prediksi suhu maksimum menggunakan fungsi aktivasi 'ReLU', dengan 80 units dalam 1 lapisan LSTM, *learning rate* 0,00852, dan *dropout rate* sebesar 0,2.

Tabel 4.5 *Hyperparameter* Optimal untuk Pemodelan Stasiun Meteorologi Perak I

Variabel	Fungsi Aktivasi	Units	Layer	Parameter Optimal	
				Learning rate	Dropout rate
T_{max}	'ReLU'	80	1	0,00852	0,20
T_{min}	'ReLU'	128	2	0,00127	0,10

Dengan kombinasi seperti ini model cenderung mengarah ke model yang lebih ringkas dengan *learning rate* yang relatif tinggi dan dropout sedang. Sedangkan untuk model prediksi suhu minimum menggunakan fungsi aktivasi 'ReLU', dengan 128 units dalam 2 lapisan LSTM, *learning rate* 0,00127, *dropout rate* 0,1, dan menggunakan *optimizer* yang sama yaitu 'adam'. Meskipun kedua set *hyperparameter* optimal menggunakan fungsi aktivasi 'ReLU' terdapat perbedaan signifikan dalam kompleksitas arsitektur, *learning rate*, dan *dropout rate*. Hal ini menandakan bahwa karakteristik data atau pola yang ingin diprediksi untuk kedua variabel membutuhkan kapasitas model dan strategi pelatihan yang berbeda untuk mencapai kinerja optimal. Sama halnya dengan hasil pemodelan suhu minimum di Stasiun Juanda, pada

Lampiran 9 terlihat bahwa prediksi suhu minimum oleh model LSTM di Stasiun Perak I menghasilkan pola yang relatif mendekati data aktual, meskipun sebagian prediksinya masih membentuk garis yang cukup lurus. Hal ini menunjukkan bahwa model mulai mampu menangkap tren umum suhu minimum, namun belum sepenuhnya mengikuti variasi yang lebih halus dalam data aktual. Sementara itu, sebagaimana yang telah terlampir pada Lampiran 11 prediksi suhu maksimum juga menunjukkan kecenderungan serupa, di mana garis prediksi terlihat halus namun terlalu sederhana untuk mengikuti lonjakan ekstrem dari data aktual. Perbedaan yang cukup mencolok masih terlihat, terutama pada titik-titik dengan perubahan suhu yang tajam. Secara keseluruhan pemodelan suhu minimum maupun maksimum di stasiun ini memperlihatkan pola prediksi yang serupa dengan pemodelan suhu minimum di Stasiun Juanda, namun masih terdapat keterbatasan pada model dalam menangkap detail variabilitas, terutama akibat munculnya prediksi yang terlalu datar.

4.3.2 Pemodelan *Long Short-Term Memory* untuk Stasiun AWS GI Sambikerep

Model LSTM untuk prediksi suhu maksimum mencapai kinerja optimalnya dengan menggunakan fungsi aktivasi ‘ReLU’, jumlah *units* 24 dalam satu *layer* LSTM, *learning rate* 0,00123, dan dengan *dropout rate* 0,3. Dengan Kombinasi ini menunjukkan bahwa model suhu maksimum membutuhkan kapasitas yang besar untuk satu lapisan. Sedangkan pada model suhu minimum ditemukan *hyperparameter* optimal yang berbeda dengan penggunaan fungsi aktivasi yang sama ‘ReLU’, tetapi dengan jumlah *units* yang lebih kecil yaitu 80 dalam 1 lapisan LSTM, *learning rate* yang lebih tinggi 0,00849, *dropout rate* yang lebih rendah 0,1.

Tabel 4.6 *Hyperparameter* Optimal untuk Pemodelan Stasiun AWS GI Sambikerep

Variabel	Fungsi Aktivasi	Units	Layer	Parameter Optimal	
				Learning rate	Dropout rate
T_{max}	‘ReLU’	240	1	0,00123	0,3
T_{min}	‘ReLU’	80	1	0,00849	0,1

Meskipun kedua model memiliki kesamaan dalam penggunaan fungsi aktivasi dalam 1 lapisan LSTM, terdapat perbedaan signifikan dalam jumlah *units*, *learning rate*, dan *dropout rate*. Model suhu maksimum membutuhkan kapasitas yang lebih besar dengan *dropout* yang tinggi, sementara model suhu minimum lebih kecil dengan 80 *units* dengan *learning rate* yang lebih besar dan *dropout* yang lebih rendah. Hal ini menandakan bahwa karakteristik pemodelan suhu maksimum di stasiun AWS GI Sambikerep lebih kompleks dan membutuhkan kapasitas representasi yang lebih besar dibandingkan dengan pemodelan suhu minimum. Sama halnya dengan hasil prediksi LSTM terhadap suhu maksimum di Stasiun Juanda, pemodelan suhu maksimum di Stasiun Sambikerep yang terlampir pada Lampiran 12 juga menunjukkan pola prediksi yang cenderung datar dan tidak mampu mengikuti fluktuasi ekstrem dari data aktual. Hal ini mengindikasikan bahwa model LSTM mengalami kesulitan dalam menangkap pola suhu maksimum yang kompleks. Sebaliknya, pada pemodelan suhu minimum di Stasiun Sambikerep sebagaimana terlampir pada Lampiran 10, prediksi LSTM mampu mengikuti arah dan pola pergerakan data aktual secara lebih konsisten. Pola ini serupa dengan hasil pemodelan suhu minimum di Stasiun Juanda, yang juga menunjukkan bahwa suhu minimum relatif lebih mudah dipelajari..

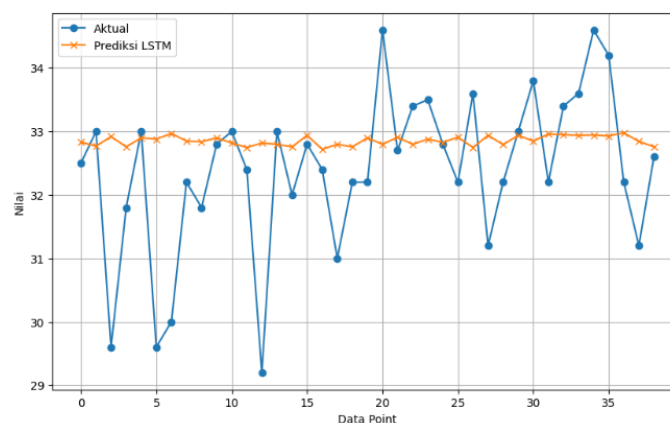
4.3.3 Pemodelan *Long Short-Term Memory* untuk Stasiun Meteorologi Juanda

Model prediksi suhu maksimum di stasiun meteorologi juanda mencapai kinerja optimalnya dengan menggunakan fungsi aktivasi 'ReLU', jumlah *units* 112 dalam 2 *layer* LSTM, *learning rate* 0,004, dengan *dropout rate* 0 dan menggunakan *optimizer* 'adam'.

Tabel 4.7 *Hyperparameter* Optimal untuk Pemodelan Stasiun Meteorologi Juanda

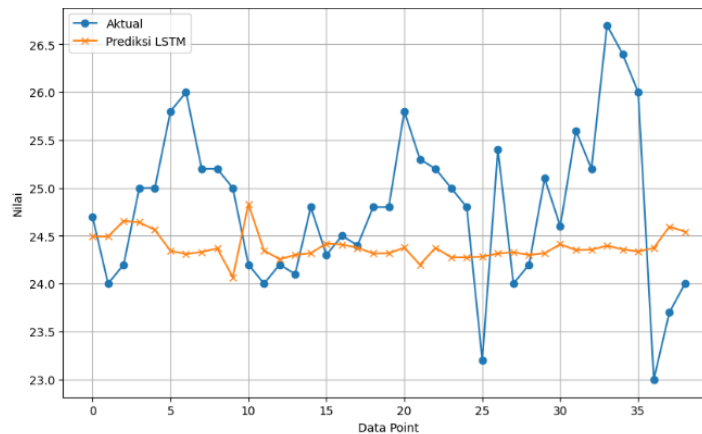
Variabel	Fungsi Aktivasi	Units	Layer	Parameter Optimal	
				<i>Learning rate</i>	<i>Dropout rate</i>
T_{max}	'ReLU'	112	2	0,004	0,0
T_{min}	'Tanh'	480	2	0,0068	0,2

Perlu dicatat bahwa untuk pemodelan suhu minimum proses pencarian *hyperparameter* dilakukan dengan konfigurasi ruang yang sedikit berbeda karena hasil prediksi awal cenderung berada pada rentang nilai tertentu, mengindikasikan model belum optimal dalam menangkap pola variasi. *Units* yang digunakan pada ruang pencarian diperbesar dari [16-256] menjadi [64-512], dan *learning rate* juga diganti yang sebelumnya $[1 \times 10^{-4} - 1 \times 10^{-2}]$ menjadi $[1 \times 10^{-5} - 1 \times 10^{-2}]$. Dari perubahan ruang pencarian tersebut, model LSTM untuk prediksi suhu minimum menemukan *hyperparameter* optimal yaitu fungsi aktivasi 'Tanh', jumlah *units* 448 dalam 2 *layer* LSTM, *learning rate* 0,0068, dan dengan *dropout rate* 0,2.



Gambar 4.7 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model LSTM di Stasiun Juanda

Gambar 4.7 menunjukkan pola hasil prediksi untuk suhu maksimum yang sangat datar dan hampir tidak memiliki variasi. LSTM gagal menangkap fluktuasi tajam dari data aktual yang merupakan indikasi dari *underfitting*. Model ini tidak berhasil mempelajari pola data yang kompleks dan hanya mampu menghasilkan prediksi yang mendekati nilai rata-rata. Model suhu minimum membutuhkan kapasitas *units* yang jauh lebih besar hal ini menandakan bahwa pola suhu minimum lebih kompleks, fungsi aktivasi yang digunakan juga berbeda penggunaan 'Tanh' pada model prediksi suhu minimum dapat membantu model dalam menangani data yang terpusat di sekitar nol yang lebih sesuai untuk karakteristik data suhu minimum. Selain itu, model suhu minimum menggunakan *dropout rate* 0,2 yang menunjukkan kebutuhan akan regularisasi tambahan untuk mencegah *overfitting* meskipun memiliki kapasitas *units* yang lebih besar.



Gambar 4.8 Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model LSTM di Stasiun Juanda

Gambar 4.8 menunjukkan pola hasil prediksi LSTM suhu minimum yang terlihat berbeda secara signifikan dengan model suhu maksimum, model untuk suhu minimum ini berhasil mengikuti arah pergerakan dan pola fluktuasi dari data aktualnya. Meskipun masih menunjukkan adanya efek perataan di mana prediksi belum mampu menangkap pola fluktuatif dari observasi, peningkatan kapasitas model dan regularisasi terbukti efektif. Secara keseluruhan, model telah mampu mempelajari pola data suhu minimum yang kompleks secara jauh lebih baik.

4.4 Evaluasi dan Komparasi Model

Untuk mengevaluasi kinerja model yang telah dioptimalkan dalam memprediksi suhu maksimum dan minimum, dilakukan perhitungan RMSE dan Improval Percentage dibandingkan dengan NWP. Evaluasi ini mencakup model *Support Vector Regression* dan *Long Short-Term Memory* untuk Stasiun Meteorologi Perak I, AWS GI Sambikerep, dan Stasiun Meteorologi Juanda.

4.4.1 Pemodelan Variabel Suhu Maksimum

Tabel 4.8 menunjukkan bahwa model SVR secara konsisten mengungguli model LSTM di ketiga stasiun pengamatan. Sebagai catatan penting, nilai RMSE NWP di ketiga stasiun terbilang cukup tinggi. Hal ini menunjukkan bahwa prediksi suhu maksimum oleh model NWP terdapat bias sistematis yang perlu dikoreksi. Model SVR berhasil menurunkan RMSE secara signifikan dari nilai NWP awal tersebut. Misalnya, di Stasiun Meteorologi Juanda, SVR menurunkan RMSE dari 5,92 menjadi hanya 0,82, mencerminkan kemampuan model dalam melakukan koreksi terhadap bias prediksi yang sangat besar. Persentase peningkatan *Improval Percentage* sebesar 83,47% menjadi yang tertinggi di antara semua stasiun dan model. Keunggulan SVR terletak pada kemampuannya beradaptasi melalui pemilihan hyperparameter. Terbukti, semua model SVR optimal untuk suhu maksimum secara konsisten memilih *kernel* rbf, yang mengindikasikan bahwa *kernel* ini sangat efektif dalam menangkap hubungan non-linear yang kompleks antara variabel prediktor dan suhu.

Tabel 4.8 Hasil RMSE dan *Improval Percentage* untuk Pemodelan Suhu Maksimum

Stasiun Meteorologi	RMSE			<i>Improval Percentage (%)</i>	
	NWP	SVR	LSTM	SVR	LSTM
Perak I	2,88	0,95	1,03	81,03	64,30
Sambikerep	4,80	1,04	1,38	74,80	71,05
Juanda	5,92	0,82	1,23	83,47	79,29

Khususnya pada Stasiun Meteorologi Juanda yang mencatat kinerja terbaik dengan RMSE 0,82 model SVR menggunakan kombinasi dari *epsilon* yang sangat kecil (0,001) dan *gamma* yang sangat besar (10). Selain itu, nilai parameter regularisasi $C=1$ menjadi pilihan konstan untuk prediksi suhu maksimum di semua stasiun, menunjukkan tingkat penalti yang seragam terhadap kesalahan. Di sisi lain, model LSTM menunjukkan variasi arsitektur yang lebih besar dan tidak memperlihatkan pola hyperparameter yang konsisten untuk mencapai kinerja yang optimal. Meskipun fungsi aktivasi 'ReLU' dominan, tidak ada korelasi yang jelas antara jumlah *units*, *learning rate*, atau *dropout* dengan penurunan RMSE secara signifikan. Secara keseluruhan, hasil komparasi ini secara konsisten menunjukkan bahwa model SVR secara signifikan lebih unggul dalam memprediksi suhu maksimum dibandingkan dengan model LSTM di ketiga stasiun pengamatan. SVR tidak hanya menghasilkan nilai RMSE yang lebih rendah, tetapi juga memberikan persentase peningkatan yang lebih baik dalam mengoreksi bias model NWP. Kinerja terbaik dari SVR terbukti pada Stasiun Meteorologi Juanda, yang mencapai RMSE terendah dan *Improval Percentage* tertinggi.

4.4.2 Pemodelan Variabel Suhu Minimum

Tabel 4.9 menunjukkan bahwa LSTM mampu memodelkan suhu minimum lebih baik ketimbang suhu maksimum. Meskipun begitu, SVR secara umum memberikan hasil prediksi yang lebih akurat di sebagian besar stasiun. Keunggulan utama SVR terlihat pada Stasiun Meteorologi Juanda, di mana model ini mencatat nilai RMSE terendah sebesar 0.71. Kinerja unggul ini didukung oleh pemilihan hyperparameter yang spesifik $C = 1$, *epsilon* = 0.1, dan *gamma* = 0.1. Konfigurasi ini menunjukkan bahwa model SVR bersifat relatif "ketat" terhadap kesalahan $C = 1$ namun memiliki pengaruh sampel yang lebih luas *gamma* = 0.1, sehingga mampu menangkap pola suhu minimum yang lebih halus. Seperti halnya pada suhu maksimum, pemilihan *kernel* 'rbf' secara konsisten di seluruh stasiun menguatkan bahwa *kernel* ini sangat efektif dalam memodelkan hubungan non-linier pada data suhu.

Tabel 4.9 Nilai RMSE dan *Improval Percentage* pada Model Suhu Minimum

Stasiun Meteorologi	RMSE			<i>Improval Percentage (%)</i>	
	NWP	SVR	LSTM	SVR	LSTM
Perak I	2,88	0,90	1,22	66,93	57,62
Sambikerep	2,84	0,86	0,69	67,80	75,53
Juanda	2,53	0,71	0,94	71,56	62,47

Di sisi lain, model LSTM menunjukkan kecenderungan memerlukan arsitektur yang jauh lebih kompleks untuk memprediksi suhu minimum. Sebagai contoh, model optimal di Stasiun Perak I dan Juanda menggunakan konfigurasi yang kompleks, yaitu masing-masing 128 unit dengan 2 *layer*, dan 480 *unit* yang jauh lebih besar dibandingkan konfigurasi untuk suhu maksimum. Hal ini mengindikasikan bahwa pola data suhu minimum lebih sulit ditangkap oleh jaringan LSTM, sehingga diperlukan kapasitas model yang lebih besar. Menariknya, model LSTM terbaik di Juanda (RMSE 0.81) menggunakan fungsi aktivasi 'Tanh', yang sering kali

lebih sesuai untuk data yang telah dinormalisasi dan memiliki rentang terpusat di sekitar nol. Hal ini menunjukkan adaptasi khusus terhadap karakteristik data di stasiun tersebut. Secara keseluruhan, meskipun terdapat ketidakkonsistenan dalam metrik *Improval Percentage*, data RMSE pada Tabel 4.9 menunjukkan bahwa model SVR cenderung lebih unggul dalam memprediksi suhu minimum. Keunggulan ini didukung oleh pemilihan *hyperparameter* yang lebih stabil dan efisien, terutama melalui penggunaan *kernel* 'rbf' dan konfigurasi yang efektif pada stasiun Juanda. Sementara itu, model LSTM menunjukkan performa yang kompetitif, namun dengan kompleksitas arsitektur model yang lebih rumit. Hal ini menunjukkan bahwa suhu minimum merupakan tantangan yang lebih besar bagi LSTM.

(Halaman ini sengaja dikosongkan)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Model SVR terbukti sangat efektif dalam memprediksi suhu baik maksimum maupun minimum dengan performa yang secara umum konsisten mengungguli model LSTM di sebagian besar skenario pengujian. Untuk prediksi suhu maksimum, SVR menunjukkan keunggulan di semua stasiun. Kinerja puncaknya tercatat di Stasiun Meteorologi Juanda, yang menghasilkan RMSE 0.82 dan persentase peningkatan *Improval Percentage* tertinggi sebesar 83.47%. Pada prediksi suhu minimum SVR juga menunjukkan performa yang baik di mana model terbaiknya kembali tercatat di Stasiun Meteorologi Juanda dengan nilai RMSE 0.71. Hasil ini membuktikan SVR sebagai model yang lebih akurat pada penelitian ini.
2. Model LSTM mengalami kendala *underfitting* terutama pada prediksi suhu maksimum di mana model gagal menangkap pola fluktuatif pada data aktual. Untuk suhu minimum LSTM menunjukkan peningkatan kinerja setelah perluasan ruang pencarian hyperparameter. Hal ini dibuktikan untuk lokasi Sambikerep pada model suhu minimum LSTM memiliki kinerja lebih baik ketimbang SVR dengan RMSE 0,69 dan *improval percentage* 75,53%.
3. Pada seluruh stasiun pengamatan yang diuji, SVR secara umum menunjukkan nilai RMSE yang lebih rendah serta menghasilkan peningkatan *improval percentage* yang signifikan dibandingkan model LSTM. Hal ini menunjukkan bahwa model SVR lebih efektif dalam menangani bias sistematis pada keluaran model INA-NWP. Penggunaan teknik *statistical post-processing* berbasis *machine learning* terbukti mampu mengatasi keterbatasan model deterministik NWP, khususnya dalam konteks prediksi parameter permukaan seperti suhu.

5.2 Saran

Adapun saran yang dapat diberikan dari hasil penelitian ini adalah sebagai berikut:

1. Penggunaan dataset lebih panjang menggunakan data historis dengan periode pengamatan yang lebih panjang dapat membantu model menangkap pola temporal dan musiman jangka panjang yang lebih akurat.
2. Penggunaan arsitektur LSTM yang lebih kompleks dengan menambahkan *hidden layer* untuk memungkinkan model mempelajari pola dalam data.
3. Untuk meningkatkan akurasi model MOS (SVR dan LSTM) perlu ditambahkan variabel prediktor lain dari luaran INA-NWP yang berkaitan dengan proses meteorologi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- A.Rohman, P. S. (2024). *Prediksi Curah Hujan Memanfaatkan Statistical Downscaling Data Global Forecast System Menggunakan Support Vector Regression dan Long Short-Term Memory Sebagai Penunjang Keputusan Top Management*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Amanullah, A., Dananjaya, R. H., & Chrismaningwang, G. (2023). Analisis Regresi dengan Metode Support Vector Machine IMERG Downscaled di Karanganyar. *Proceeding Engineering Research Forum (pp. Vol. 3, No. 1, Juli 2023)*. Surakarta: Universitas Islam Indonesia.
- Awad, M., & Khanna, R. (2015). *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Apress Media.
- Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research* 13, 281-305.
- Cahyoko, F. D. (2023). *Spatial Ensemble Model Output Statistics Untuk Kalibrasi dan Prakiraan Cuaca Jangka Pendek*. Institut Teknologi Sepuluh Nopember : Surabaya.
- Cholissodin, I., Sutrisno, S., Soebroto, A., Hasanah, U., & Febiola, Y. (2020). *AI, Machine Learning & Deep Learning*. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.
- Deswal, V., & Dharminder, K. (2024). Hyperparameter Tuning of the LSTM model for Stock Price Prediction. *International Journal of Intelligent Systems and Applications in Engineering*, 705-712.
- Durai, V. R., & Bhradwaj, R. (2014). Evaluation of statistical bias correction methods for numerical weather prediction model forecasts of maximum and minimum temperatures. *Nat Hazards (2014)* 73:, 1229-1254.
- Fauziah , A., Angreni, D., Pradjaningsih, A., Riski, A., & Hadi, F. A. (2020). Support vector regression in statistical downscaling for rainfall forecasting. *International Journal of Scientific & Technology Research*.
- Ghazvinian, M., Zhang, Y., Seo, D. J., He, M., & Fernando, N. (2021). A novel hybrid artificial neural networkParametric scheme for postprocessing medium-range precipitation forecasts, *Adv. Water Resour.*
- Glahn, H. R., & Lowry, D. A. (1972). The Use of Model Output Statistics (MOS) in Objective Weather Forecasting. *J. Appl. Meteor*, Volume 11, pp. 1203-1211.
- Hakim, M. L. (2018). *Prediksi Cuaca Harian Dengan Bayesian Model Averaging Untuk Antisipasi Bencana Hidrometeorologi*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation* 9(8), 1735-1780.
- Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A Practical Guide to Support Vector Classification.

- Idowu, O., & Rautunbech, C. (2009). Model Output Statistics to Improve Severe Storms Prediction Over Western Sahel. *South Africa: Department of Geography, Geoinformatics and Meteorology, University of Pretoria*.
- Isabona, J., Imoize, A., Ojo, S., Karunwi, O., Kim, Y., Lee, C.-C., & Li, C.-T. (2022). Development of a Multilayer Perceptron Neural Network for Optimal Predictive Modeling in Urban Microcellular Radio Environments. *Applied Sciences*.
- Ishlah, A. W., Sudarno, & Kartika, P. (2023). Implementasi *GridsearchCV* Pada Support Vector Regression (SVR) Untuk Peramalan Harga Saham (Studi Kasus: Harga Saham PT Anabatic Technologies Tbk). *Jurnal Gaussian, Volume 12, Nomor 2, Tahun 2023*, 276-286.
- Kingma, D. P., & Ba, J. L. (2017). Adam: A Method for Sctochastic Optimization. *arXiv:1412.6980v9 [cs.LG]*.
- Kocaoglu, A. (2024). Efficient Optimization of a Support Vector Regression Model with Natural Logarithm of the Hyperbolic Cosine Loss Function for Broader Noise Distribution. *applied sciences*.
- Liashchynskiy, P., & Liashchynskiy, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*.
- Mahajaya, N. S., Ayu, P. D., & Huizen, R. R. (2024). Pengaruh Optimizer Adam, AdamW, SGD, dan LAMB terhadap Model Vision Transformer pada Klasifikasi Penyakit Paru-paru. *Prosiding Seminar Hasil Penelitian Informatika dan Komputer 2024* , 818-823.
- Maharana, K., Mishra, R., Nemade, B., Shah, D., Rana, M., Badhe, N., & Kaul, V. (2025). A Review: Data Overfitting and Underfitting Techniques. *Journal of Information Systems Engineering and Management*, 387-405.
- Masruroh, F., Surarso, B., & Warsito, B. (2023). Perbandingan Kinerja Inception-Resnet V2, Xception, Inception V3, dan Resnet50 Pada Gambar Bentuk Wajah. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 11-20.
- Mitchell, T. M. (1997). Machine Learning. *McGraw-Hill Science*.
- Muhammad, T. A., & Irawan, I. M. (2023). Implementasi Long Short-Term Memory (LSTM) untuk Prediksi Intensitas Curah Hujan (Studi Kasus: Kabupaten Malang). *Jurnal Sains dan Seni ITS, Vol. 12, No. 1*, 2337-3520.
- Muraina, I. O. (2022). Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts. *7th International Mardin Artuklu Scientific Researches Conference*, 496-504.
- Nugraha, W., Sabaruddin, R., & Murni, S. (2024). Teknik Scaling Menggunakan Robust Scaler Untuk Mengatasi Outlier Data Pada Model Prediksi Serangan Jantung. *Techno.COM, Vol. 23, No. 2*, 319-327.
- Octarino, C. (2019). Karakteristik Iklim Mikro di Ruang Publik Studi Kasus: Jalur Pedestrian Maioboro, Yogyakarta. *Jurnal Arsitektur GRID*.

- Prajapati, A. K., & Singh, U. K. (2023). An Optimal Solution to the Overfitting and Underfitting Problem of Healthcare Machine Learning Models. *Journal of Systems Engineering and Information Technology*, 77-84.
- Pritalia, G. L. (2022). Analisis Komparatif Algoritme Machine Learning pada Klasifikasi Kualitas Air Layak Minum. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, 43-55.
- Putra, F. (2023). *Prakiraan Cuaca Jangka Pendek Terkalibrasi Menggunakan Metode Geostatistical Output Pertubatuin*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Qamar, R., & Zardari, B. A. (2023). Artificial Neural Networks: An Overview. *Mesopotamian journal of Computer Science*, 130-139.
- Qona'ah, N., Sutikno, Ferawati, K., & Nirwana, M. B. (2020). Temperature Forecast Using Ridge Regression as Model Output Statistics. 383-388.
- Ramadhan, F., & Hernadi, J. (2025). Evaluasi Optimizer Adam dan RMSProp Pada Arsitektur VGG-19 Klasifikasi Ekespresi Wajah Manusia. *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 1414-1426.
- Reitermanova, Z. (2010). Data Splitting. *WDS'10 Proceedings of Contributed Papers, Part I.*, 31-36.
- Safitri, R., & Sutikno. (2012). Model Output Statistics dengan Projection Pursuit Regression untuk Meramalkan Suhu Minimum, Suhu Maksimum, dan Kelembapan. *Jurnal Sains dan Seni ITS Vol. 1, No. 1, (Sept. 2012) ISSN: 2301-928X*.
- Salehin, I., & Kang, D.-K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics*.
- Saputra, G. H., Wigena, A. H., & Sartono, B. (2019). Penggunaan Support Vector Regression Dalam Pemodelan Indeks Saham Syariah Indonesia Dengan Algoritme Grid Search. *Indonesian Journal of Statistics and Its Applications (eISSN:2599-0802)*, 148 - 160.
- Sari, D. E. (2018). *Ensemble Model Output Statistics Untuk Prediksi Suhu dan Kelembaban Jangka Pendek Terkalibrasi*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Sarker, H. A. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2:420.
- Satriawan, G. Y., & Prasetyo, B. (2024). Hyperparameter Tuning of Long Short-Term Memory Model for Clickbait Classification in News Headlines. *Recursive Journal of Informatics*, 28-36.
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*.
- Siregar, N. A. (2022). Peramalan Curah Hujan Di Kota Medan menggunakan Metode Support Vector Regression. *Journal of Informatics and Data Science (J-IDS)*.
- Smola, A. J., & Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing* 14, 199-222.

- Staudemeyer, R. C., & Morris, R. E. (2019). Understanding LSTM a tutorial into Long Short-Term Memory Recurrent Neural Networks.
- Tentang BMKG.* (1841). Retrieved from BMKG - Badan Meteorologi, Klimatologi, dan Geofisika: <https://www.bmkg.go.id/profil>
- Vannitsem, S., Bremnes, J. B., Demaeyer, J., Evans, R., Flowerdew, J., & Hemri, S. (2021). *Statistical Postprocessing for Weather Forecasts*. BAMS American Meteorological Society.
- Wigena, A. H., Djuraidah, A., & Sahriman, S. (2005). Statistical Downscaling dengan Pergeseran Waktu Berdasarkan Korelasi Silang. *Jurnal Meteorologi dan Geofisika*, Vol.16, No.1, 19-24.
- Wilks, D. S. (2006). *Statistical Methods in the Atmospheric Sciences*, 2nd ed. Boston: Elsevier.
- Yu, P. S., Chen, S. T., & Chang, I. F. (2006). Support Vector Regression for Real-Time Flood Stage Forecasting. *Journal of Hydrology* 328(3-4), 704-716.
- Zaytar, M. A., & Amrani, C. E. (2016). Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. *International Journal of Computer Applications* (0975 - 8887).
- Zhang, T., Liang, Z., Li, W., Wang, J., Hu, Y., & Li, B. (2023). Statistical post-processing of precipitation forecasts using circulation classifications and spatiotemporal deep neural networks. *Hidrology and Earth System Sciences*, 1945-1960.

LAMPIRAN

Lampiran 1. Syntax Pemodelan LSTM

Pemodelan LSTM

```
!pip install keras-tuner --upgrade

#Import Library yang dibutuhkan

import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import keras_tuner as kt

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.optimizers import Adam, RMSprop
from sklearn.preprocessing import RobustScaler
from sklearn.metrics import mean_squared_error


from google.colab import drive
drive.mount('/content/drive')


# read_csv

df = pd.read_csv("/content/drive/MyDrive/DATASET TUGAS AKHIR/Sambikerep
tmin.csv",sep=';')

df

X = df[['INA-NWP']].values
y = df['Obs'].values


# menscale variabel prediktor

scaler = RobustScaler()
```

```

X_scaled = scaler.fit_transform(X)

# time_steps 3 berarti data historis yang diambil untuk satu array ada 3
def create_sequences(data, target, time_steps=3):
    Xs, ys = [], []
    for i in range(len(data) - time_steps):
        Xs.append(data[i:(i + time_steps)]) # Mengambil time_steps data sebelumnya
        ys.append(target[i + time_steps]) # Target adalah data setelahnya
    return np.array(Xs), np.array(ys)

time_steps = 3
X_seq, y_seq = create_sequences(X_scaled, y, time_steps)

# membagi data berbasis waktu 90 persen untuk training set 10 persen untuk testing set
split_point = int(len(X_seq) * 0.9)

X_train = X_seq[:split_point]
y_train = y_seq[:split_point]
X_test = X_seq[split_point:]
y_test = y_seq[split_point:]

print(f"Shape X_train (time-split): {X_train.shape}, Shape y_train (time-split): {y_train.shape}")
print(f"Shape X_test (time-split): {X_test.shape}, Shape y_test (time-split): {y_test.shape}")

print(f"Shape X_train: {X_train.shape}, Shape y_train: {y_train.shape}")

def build_model(hp):
    model = Sequential()

```

```
# Tuning jumlah unit
```

```
hp_units = hp.Int('units', min_value=16, max_value=256, step=16) # bisa menggunakan rentang unit yang berbeda
```

```
# Tuning jumlah layer LSTM
```

```
hp_num_lstm_layers = hp.Choice('num_lstm_layers', values=[1, 2])
```

```
# Tuning learning rate
```

```
hp_learning_rate = hp.Float('learning_rate', min_value=1e-4, max_value=1e-2, sampling='log') # bisa menggunakan rentang learning rate yang berbeda
```

```
# Tuning dropout rate
```

```
hp_dropout_rate = hp.Float('dropout_rate', min_value=0.0, max_value=0.5, step=0.1) # bisa menggunakan rentang dropout rate yang berbeda
```

```
# Terdapat 2 pilihan layer untuk pemodelan
```

```
# Layer LSTM pertama
```

```
if hp_num_lstm_layers == 2:
```

```
    model.add(LSTM(units=hp_units, activation='relu', return_sequences=True,  
                   input_shape=(X_train.shape[1], X_train.shape[2])))
```

```
    model.add(tf.keras.layers.Dropout(hp_dropout_rate))
```

```
# Layer kedua
```

```
    model.add(LSTM(units=hp_units, activation='relu'))
```

```
else: hp_num_lstm_layers == 1
```

```
    model.add(LSTM(units=hp_units, activation='relu',
```

```
               input_shape=(X_train.shape[1], X_train.shape[2])))
```

```
    model.add(tf.keras.layers.Dropout(hp_dropout_rate))
```

```
model.add(Dense(1))
```

```
# Terdapat 1 optimizer
```

```
hp_optimizer = hp.Choice('optimizer', values=['adam'])
```

```
optimizer = Adam(learning_rate=hp_learning_rate)
```

```
# Untuk evaluasi training model menggunakan metrics RMSE
```

```
model.compile(optimizer=optimizer, loss='mse',  
metrics=[tf.keras.metrics.RootMeanSquaredError()])
```

```
return model
```

```
# Penggunaan randomsearch untuk pencarian parameter yang optimal
```

```
tuner = kt.RandomSearch(  
    build_model,  
    objective='val_root_mean_squared_error',  
    max_trials=10,  
    executions_per_trial=3,  
    directory='my_dir',  
    project_name='lstm_tuning_v2')
```

```
# Menggunakan early stop untuk penghentian training sebelum model mengalami  
overfitting
```

```
stop_early = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
```

```
tuner.search(X_train, y_train,  
    epochs=100,  
    validation_split=0.2,  
    callbacks=[stop_early])
```

```

validation_split=0.2,
callbacks=[stop_early])

best_model = tuner.get_best_models(num_models=1)[0]
best_hyperparams = tuner.get_best_hyperparameters(num_trials=1)[0]

print("Best Hyperparameters:")
print(best_hyperparams.values)

loss, mse = best_model.evaluate(X_test, y_test)
print(f"Test MSE: {mse:.4f}")

# melakukan prediksi menggunakan testing set
y_pred = best_model.predict(X_test)

# Buat DataFrame dari hasil prediksi
df_prediksi = pd.DataFrame(y_pred, columns=['Prediksi'])

# Simpan ke file CSV
df_prediksi.to_csv('hasil_prediksi_Sambikerep tmin.csv', index=False)

shape_before = X_test.shape # Simpan bentuk asli

# Reshape ke 2D untuk inverse_transform
X_test_2d = X_test.reshape(-1, X_test.shape[-1]) # dari (345, 3, 1) ke (1035, 1)

# Inverse transform
X_test_unscaled = scaler.inverse_transform(X_test_2d)

```

```

# Kembalikan ke bentuk semula
X_test_unscaled = X_test_unscaled.reshape(shape_before)

# Ambil nilai NWP terakhir dari setiap sequence
nwp_last_values = X_test_unscaled[:, -1, 0] # hasil shape: (samples,)
nwp_last_values

# Menghitung nilai RMSE dari model NWP
rmse_nwp = np.sqrt(mean_squared_error(y_test, nwp_last_values))

# Menghitung nilai RMSE dari model LSTM
rmse_lstm = np.sqrt(mean_squared_error(y_test, y_pred))

improvement_percentage = ((rmse_nwp - rmse_lstm) / rmse_nwp) * 100

print(f"RMSE NWP: {rmse_nwp:.4f}")
print(f"RMSE LSTM: {rmse_lstm:.4f}")
print(f"Improvement Percentage: {improvement_percentage:.2f}%")

# Untuk melihat Model Loss selama training
print("Retraining the best model to get history...")
history = best_model.fit(X_train, y_train,
                        epochs=100,
                        validation_split=0.2,
                        callbacks=[stop_early],
                        verbose=1)

# Plotting the loss
plt.figure(figsize=(14, 5))
plt.subplot(1, 2, 2)

```

```

plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()

# Plotting hasil prediksi LSTM dengan nilai observasi atau aktual
plt.figure(figsize=(10, 6))

# Plot nilai aktual
plt.plot(y_test, label='Aktual', marker='o')

# Plot nilai prediksi
plt.plot(y_pred, label='Prediksi LSTM', marker='x')

plt.title('Komparasi Nilai Aktual dan Prediksi LSTM untuk Stamet Sambikerep')
plt.xlabel('Data Point')
plt.ylabel('Nilai')
plt.legend()
plt.grid(True)
plt.show()

```

Lampiran 2. Syntax Pemodelan SVR

Pemodelan SVR

#Import library yang dibutuhkan

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn.svm import SVR
```

```
from sklearn.model_selection import train_test_split, RandomizedSearchCV
```

```
from sklearn.preprocessing import RobustScaler
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("D:/semester 7/file bomb/NWP ML/DATASET/Sambikerep  
tmin.csv",sep=';')
```

```
X=df[['INA-NWP']]
```

```
Y=df['Obs']
```

pembagian training set sebesar 90 persen dan test set sebesar 10 persen

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.1, random_state=42)
```

menscale variabel prediktor

```
scaler = RobustScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

```
Y.describe()
```

Menentukan parameter grid untuk random search

```
param_distributions = {
```

```
    'C': [0.1, 1, 10, 100, 1000], # Regularisasi
```

```
    'epsilon': [0.001, 0.01, 0.1, 0.2, 0.5], # Margin error toleransi
```

```
    'kernel': ['rbf', 'linear', 'poly', 'sigmoid'], # Jenis kernel
```

```

'gamma': ['scale', 'auto', 0.001, 0.01, 0.1, 1, 10] # Parameter gamma untuk kernel
}

# RandomSearchCV untuk mencari parameter terbaik

svr = SVR()

random_search = RandomizedSearchCV(svr, param_distributions, n_iter=100, cv=5,
scoring='neg_root_mean_squared_error', n_jobs=-1, random_state=42)

random_search.fit(X_train_scaled, y_train)

print("Best score (RMSE):", -random_search.best_score_)

print("Best Parameters:", random_search.best_estimator_)


# Melatih model dengan parameter terbaik

best_svr = random_search.best_estimator_


# Prediksi dengan model terbaik

y_pred = best_svr.predict(X_test_scaled)

y_pred


# Buat DataFrame dari hasil prediksi

df_prediksi = pd.DataFrame(y_pred, columns=['Prediksi'])

# Simpan ke file CSV

df_prediksi.to_csv('hasil_prediksi_Sambikerep tmin.csv', index=False)


# Hitung RMSE dari output NWP

msenwp = mean_squared_error(Y, X)

rmsenwp = np.sqrt(msenwp)


# Hitung RMSE dari prediksi SVR

mae = mean_absolute_error(y_test, y_pred)

mse = mean_squared_error(y_test, y_pred)

rmse = np.sqrt(mse)

```

```
print("RMSE SVR:", rmse)
print("RMSE NWP:", rmsenwp)

# Hitung percentage improvement
im = ((rmsenwp - rmse) / rmsenwp) * 100
print("Percentage Improvement:", im)
plt.figure(figsize=(12, 6))

plt.plot(y_test.values, label='Output NWP (Test)', marker='o') # Output NWP pada data
test

plt.plot(y_pred, label='Output Prediksi SVR', marker='x') # Output prediksi SVR

plt.xlabel('Indeks Data Test')
plt.ylabel('Nilai')
plt.title('Perbandingan Output NWP, dan Prediksi SVR')
plt.legend()
plt.grid(True)
plt.show()
```

Lampiran 3. Dataset Suhu Maksimum

No.	Observasi			NWP		
	Stasiun Meteorologi Perak I	AWS GI Sambikerep	Stasiun Meteorologi Juanda	Stasiun Meteorologi Perak I	AWS GI Sambikerep	Stasiun Meteorologi Juanda
1	33,00	33,00	31,60	28,80	27,40	27,40
2	31,60	31,60	32,20	28,20	23,10	23,10
3	31,20	31,20	31,40	26,60	26,60	26,60
4	30,40	30,40	31,80	26,20	27,00	27,00
5	31,00	29,37	31,80	26,20	27,20	27,20
6	30,20	30,52	29,70	26,40	27,90	27,90
7	29,20	29,29	31,40	24,20	27,40	27,40
8	31,40	29,03	31,40	24,80	27,20	27,20
9	31,00	29,42	31,40	29,20	27,50	27,50
10	29,00	31,71	31,80	24,00	28,00	28,00
11	30,40	32,97	32,20	24,20	24,20	24,20
12	30,40	32,08	32,20	25,20	24,80	24,80
13	30,40	34,03	31,70	25,60	25,00	25,00
14	32,40	30,19	32,00	26,70	29,00	29,00
15	30,00	32,77	31,60	25,80	29,00	29,00
16	32,20	30,71	31,20	25,20	29,20	29,20
17	31,60	29,87	32,40	25,90	27,30	27,30
18	31,60	31,01	32,90	29,20	25,60	25,60
19	32,20	33,22	34,60	29,70	27,00	27,00
20	31,60	32,03	32,60	29,30	25,10	25,10
21	32,80	33,57	34,00	28,30	29,20	29,20
22	30,40	32,65	33,00	28,70	26,00	26,00
23	32,00	31,04	32,40	26,70	30,30	30,30
24	31,40	34,03	32,80	26,10	27,80	27,80
25	31,40	31,83	32,40	25,50	29,30	29,30
26	31,60	28,99	33,00	25,60	26,00	26,00
27	31,80	31,80	33,10	28,70	26,40	26,40
28	32,00	32,00	32,30	28,10	28,80	28,80
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
386	32,20	32,20	31,20	26,20	25,70	25,70
387	28,70	28,70	32,60	30,60	25,70	25,70

Lampiran 4. Dataset Suhu Minimum

No.	Observasi			NWP		
	Stasiun Meteorologi Perak I	AWS GI Sambikerep	Stasiun Meteorologi Juanda	Stasiun Meteorologi Perak I	AWS GI Sambikerep	Stasiun Meteorologi Juanda
1	24,80	24,80	25,00	22,60	22,60	22,20
2	26,00	26,00	24,30	22,90	22,90	22,70
3	23,40	23,40	22,40	22,90	22,90	22,90
4	24,80	24,80	24,10	23,20	23,20	23,50
5	24,40	24,96	24,30	23,70	23,70	23,70
6	24,00	25,24	24,00	23,40	23,40	24,30
7	25,00	23,29	21,90	23,40	23,40	25,40
8	24,60	24,24	23,80	23,70	23,70	24,80
9	25,00	25,84	23,60	24,30	24,30	23,30
10	25,20	24,78	24,00	22,00	22,00	24,20
11	26,20	25,67	25,70	22,20	22,20	23,00
12	24,20	24,07	23,40	23,50	23,50	23,20
13	24,20	24,53	23,90	23,50	23,50	23,90
14	24,40	25,25	23,80	23,50	23,50	23,60
15	25,00	24,56	24,20	23,90	23,90	25,20
16	25,60	25,22	25,80	22,90	22,90	24,60
17	25,80	25,07	25,30	23,10	23,10	24,30
18	24,40	23,77	25,80	22,90	22,90	23,30
19	24,80	24,75	23,50	23,20	23,20	23,90
20	25,50	25,57	25,00	23,00	23,00	23,00
21	26,60	23,75	24,60	22,60	22,60	22,80
22	25,40	25,37	24,10	23,40	23,40	23,30
23	26,00	23,52	25,00	22,70	22,70	21,20
24	25,00	25,00	24,30	23,00	23,00	22,60
25	25,80	25,80	24,00	22,90	22,90	24,50
26	25,80	25,80	25,70	23,10	23,10	23,70
27	25,80	25,80	25,10	23,10	23,10	22,10
28	26,00	26,00	24,50	23,40	23,40	23,70
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
386	23,7,	23,61	23,70	21,70	24,30	21,70
387	24,00	24,88	24,00	22,20	22,70	22,20

Lampiran 5. Hasil Prediksi LSTM untuk Variabel Suhu Maksimum

No.	Stasiun Meteorologi Juanda	Stasiun Meteorologi Perak I	AWS GI Sambikerep
1	32,84	24,99	30,93
2	32,77	25,01	31,02
3	32,92	24,94	31,07
4	32,76	24,88	31,04
5	32,90	24,79	31,27
6	32,88	24,90	31,42
7	32,97	24,92	31,15
8	32,85	24,85	31,08
9	32,84	24,99	31,13
10	32,90	24,95	31,07
11	32,82	24,96	30,80
12	32,75	25,00	30,89
13	32,82	25,04	30,87
14	32,80	25,05	30,82
15	32,76	24,96	31,08
16	32,94	24,93	31,01
17	32,72	24,91	30,98
18	32,80	24,97	30,80
19	32,76	25,02	31,00
20	32,91	24,76	31,12
21	32,80	24,92	31,03
22	32,91	25,03	31,11
23	32,80	24,85	30,98
24	32,88	24,96	31,13
25	32,83	25,07	31,09
26	32,91	25,02	31,07
27	32,75	24,82	31,15
28	32,94	24,79	31,13
29	32,80	24,91	31,12
30	32,94	24,91	31,22
31	32,85	25,05	31,37
32	32,96	25,01	31,43
33	32,95	25,06	31,41
34	32,94	25,02	31,43
35	32,95	24,83	31,40
36	32,93	25,01	31,49
37	32,98	25,02	31,10
38	32,85	24,64	30,89
39	32,76	24,76	30,86

Lampiran 6. Hasil Prediksi LSTM untuk Variabel Suhu Minimum

No.	Stasiun Meteorologi Juanda	Stasiun Meteorologi Perak I	AWS GI Sambikerep
1	24,50	25,70	24,93
2	24,50	25,70	24,95
3	24,66	25,72	24,89
4	24,65	25,73	24,82
5	24,57	25,44	24,72
6	24,35	25,69	24,86
7	24,31	25,70	24,85
8	24,34	25,40	24,77
9	24,37	25,72	24,93
10	24,07	25,72	24,89
11	24,84	25,64	24,91
12	24,35	25,69	24,94
13	24,26	25,67	24,97
14	24,31	25,62	24,97
15	24,32	25,64	24,88
16	24,42	25,65	24,94
17	24,41	25,71	24,87
18	24,38	25,65	24,90
19	24,32	25,48	24,92
20	24,32	25,65	24,82
21	24,38	25,77	25,20
22	24,20	25,63	24,94
23	24,38	25,72	24,97
24	24,28	25,69	24,89
25	24,28	25,60	24,98
26	24,29	25,58	24,87
27	24,32	25,63	24,90
28	24,33	25,74	24,97
29	24,31	25,62	25,05
30	24,32	25,54	24,80
31	24,42	25,96	25,24
32	24,36	25,70	24,94
33	24,36	25,61	24,98
34	24,40	25,63	24,90
35	24,36	25,73	25,00
36	24,34	25,61	24,91
37	24,38	25,51	24,85
38	24,60	25,76	24,77
39	24,55	25,94	25,17

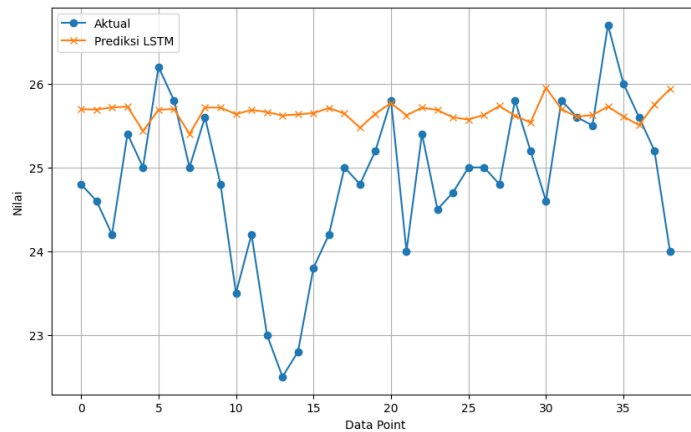
Lampiran 7. Hasil Prediksi SVR untuk Variabel Suhu Maksimum

No.	Stasiun Meteorologi Juanda	Stasiun Meteorologi Perak I	AWS GI Sambikerep
1	31,99	32,12	31,32
2	32,14	32,84	31,49
3	32,41	32,25	31,45
4	32,99	32,95	31,94
5	32,01	32,79	31,50
6	32,98	32,49	31,28
7	32,12	32,49	31,76
8	32,43	32,75	31,90
9	32,25	31,77	31,26
10	32,02	32,12	31,60
11	33,01	31,89	32,06
12	32,83	31,75	32,78
13	33,11	32,66	32,35
14	32,63	31,76	31,94
15	32,99	31,11	31,94
16	32,14	31,89	31,49
17	32,90	31,44	31,32
18	32,86	32,14	32,04
19	32,46	31,44	31,10
20	32,99	32,36	31,67
21	32,86	31,81	32,04
22	32,01	32,12	31,5
23	32,93	31,01	32,56
24	32,98	31,48	31,28
25	32,41	31,44	31,45
26	32,59	31,79	31,62
27	32,52	32,51	31,17
28	32,59	32,59	31,62
29	32,46	32,38	31,10
30	32,63	31,77	31,94
31	32,02	31,79	31,64
32	32,60	32,14	32,20
33	32,68	31,81	31,10
34	32,60	32,03	32,20
35	32,63	32,36	31,94
36	32,01	32,49	31,30
37	32,76	31,01	31,05
38	32,59	32,64	31,62
39	32,01	32,12	31,50

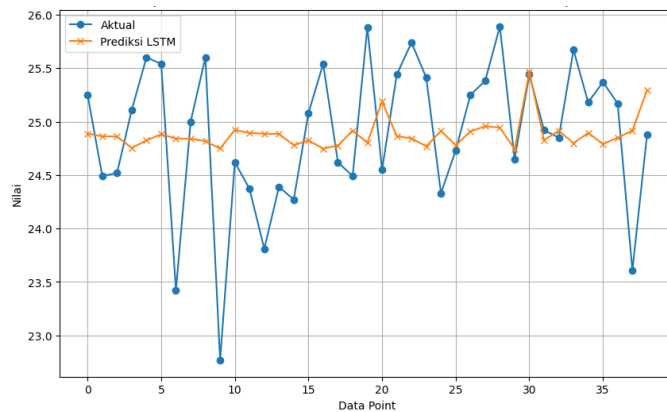
Lampiran 8. Hasil Prediksi SVR untuk Variabel Suhu Minimum

No.	Stasiun Meteorologi Juanda	Stasiun Meteorologi Perak I	AWS GI Sambikerep
1	24,09	25,48	25,71
2	24,04	25,07	25,19
3	25,54	25,09	25,18
4	24,09	25,98	25,13
5	24,26	25,88	25,10
6	24,46	25,10	25,20
7	24,72	25,07	25,19
8	24,26	25,09	25,18
9	24,02	25,95	25,01
10	24,74	25,10	25,20
11	24,07	25,07	25,19
12	25,70	25,10	25,20
13	24,31	25,99	25,79
14	24,15	25,07	25,19
15	24,04	25,02	25,58
16	24,32	25,07	25,19
17	24,02	25,02	25,68
18	24,72	25,08	25,21
19	24,26	25,95	25,05
20	24,99	25,69	25,63
21	24,72	25,48	25,71
22	24,74	25,56	25,65
23	24,26	25,98	25,06
24	24,19	25,01	25,33
25	24,99	25,07	25,19
26	24,72	25,69	25,63
27	24,02	25,09	25,18
28	24,15	25,08	25,21
29	24,20	25,07	25,51
30	24,99	25,10	25,49
31	25,00	25,10	25,49
32	24,07	25,48	25,71
33	24,64	25,23	25,40
34	24,20	25,06	25,92
35	24,11	25,10	25,19
36	24,41	25,10	25,49
37	24,58	25,96	25,03
38	24,89	25,10	25,19
39	24,74	26,00	25,50

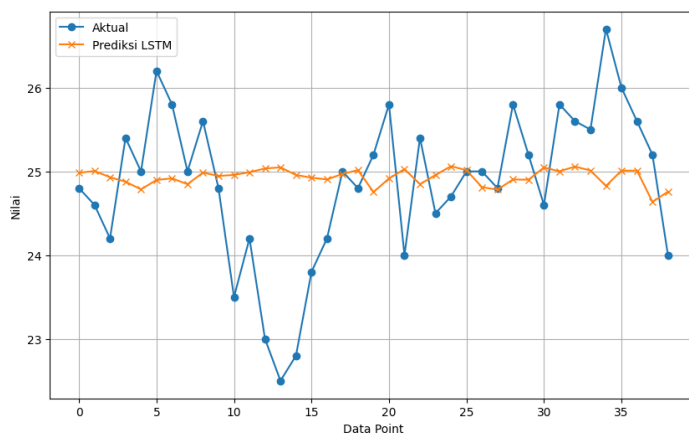
Lampiran 9. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model LSTM di Stasiun Meteorologi Perak I



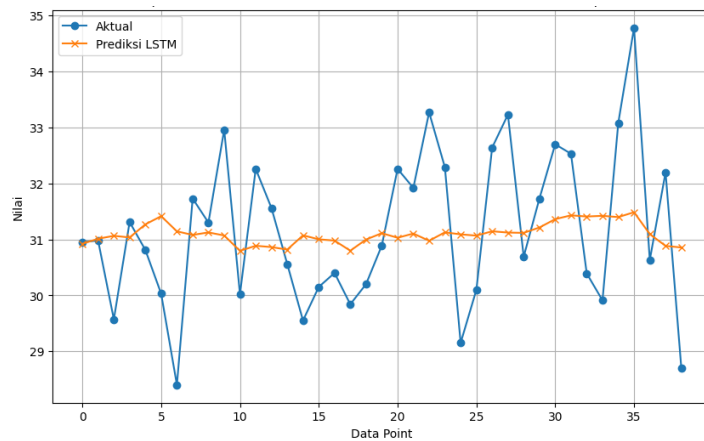
Lampiran 10. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model LSTM di Stasiun AWS GI Sambikerep



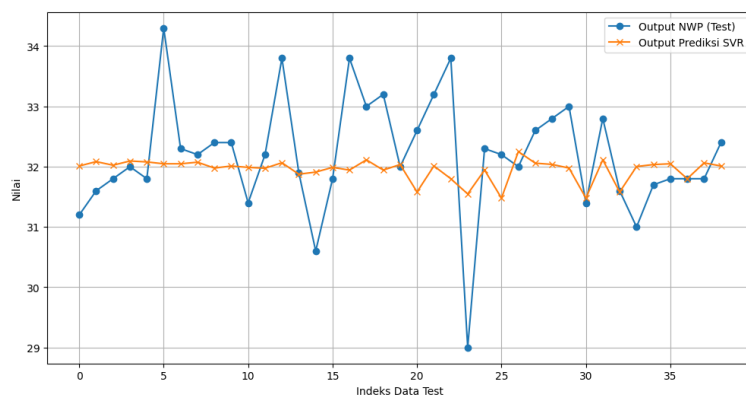
Lampiran 11. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model LSTM di Stasiun Meteorologi Perak I



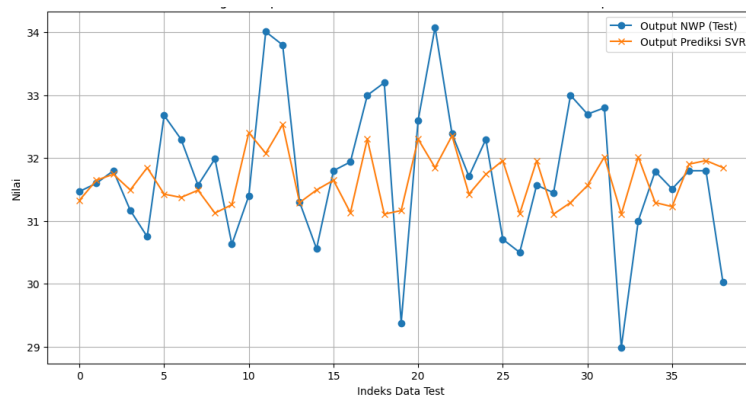
Lampiran 12. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model LSTM di Stasiun AWS GI Sambikerep



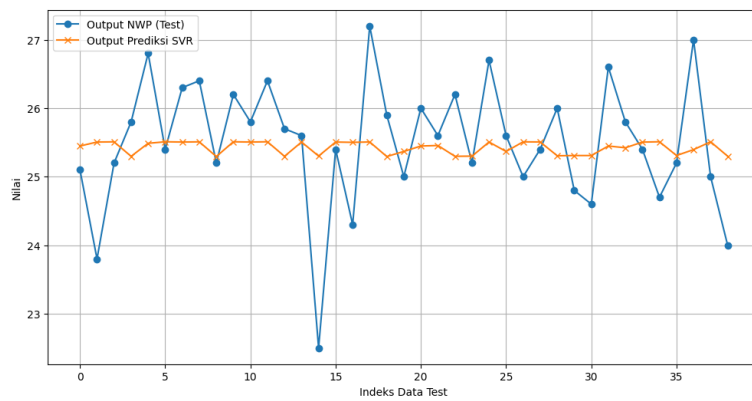
Lampiran 13. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model SVR di Stasiun Perak I



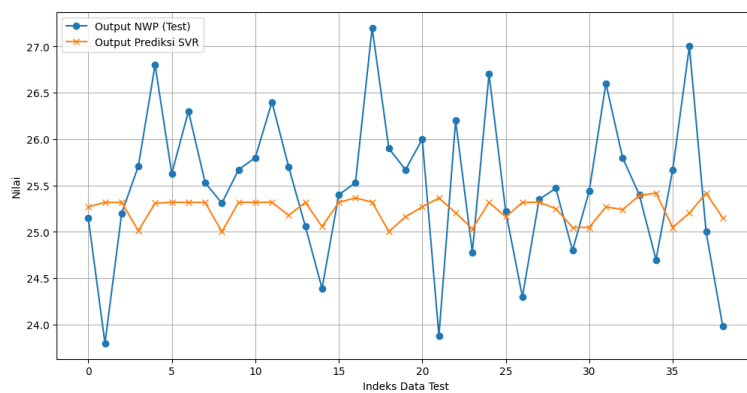
Lampiran 14. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Maksimum oleh Model SVR di Stasiun AWS GI Sambikerep



Lampiran 15. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model SVR di Stasiun Perak I



Lampiran 16. Perbandingan Nilai Aktual dan Hasil Prediksi Suhu Minimum oleh Model SVR di Stasiun AWS GI Sambikerep



Lampiran 17. Surat Pernyataan Data Sekunder

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FSAD ITS:

Nama : Haikal Mauladana Ichwani

NRP : 5003211142

menyatakan bahwa data yang digunakan dalam Tugas Akhir/~~Thesis~~ ini merupakan data sekunder yang diambil dari ~~penelitian~~ ~~buku~~ ~~Tugas Akhir~~ ~~Thesis~~ publikasi lainnya yaitu:

Sumber : Badan Meteorologi, Klimatologi, dan Geofisika

Keterangan :

- Data Observasi Suhu Udara Harian dan Data Luaran INA-NWP

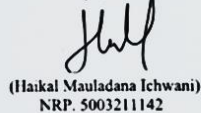
Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tugas Akhir



(Dr. Sutikno, M.Si.)
NIP. 19710313 199702 1 001

Surabaya, 21 Juli 2025



(Haikal Mauladana Ichwani)
NRP. 5003211142

BIODATA PENULIS



Haikal Mauladana Ichwani, merupakan anak ke-dua dari dua bersaudara yang lahir di Kota Patria Blitar pada tanggal 13 Mei 2003. Penulis menempuh pendidikan sekolah menengah atas di SMAN 1 Talun yang kemudian melanjutkan pendidikannya dengan menempuh pendidikan program S1 Departemen Statistika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember Surabaya.

Selama menempuh pendidikannya di ITS, penulis sangat aktif mengikuti kegiatan baik kegiatan akademik maupun non akademik. Kegiatan non-akademik yang pernah diikuti oleh penulis diantaranya yaitu kegiatan keorganisasian pada lingkup departemen yaitu sebagai staff HRD PSt-HIMASTA ITS serta wakil manajer SPV PSt-HIMASTA ITS. Penulis memiliki berbagai perminatan diantaranya optimasi database, *data engineering*, *machine learning* dan *deep learning*, serta berbagai pengaplikasiannya.

Penulisan Tugas Akhir ini merupakan hasil implementasi minat-minat penulis khususnya pada bidang pengaplikasian *machine learning* pada pemodelan prediksi suhu, dengan harapan penulisan tugas akhir ini dapat membantu serta mempermudah perkembangan dalam prediksi suhu dengan mengaplikasikan teknologi yang sedang relevan saat ini. Untuk saran, kritik, maupun masukan, penulis dapat dihubungi melalui email (haikal10mauladana@gmail.com).