



TUGAS AKHIR - TF181801

**IMPLEMENTASI *MACHINE LEARNING* DALAM
PREDIKSI CUACA MARITIM DENGAN
MENGUNAKAN ALGORITMA *ARTIFICIAL
NEURAL NETWORK LAVENBERG-
MARQUARDT***

Kartika Tri Wahyu Ramadhani (0231 15 4000 0026)

Dosen Pembimbing

Dr.Ir. Syamsul Arifin, MT

Prof. Dr. Ir. Aulia Siti Aisjah, MT

Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - TF181801

**IMPLEMENTASI *MACHINE LEARNING* DALAM
PREDIKSI CUACA MARITIM DENGAN
MENGUNAKAN ALGORITMA *ARTIFICIAL
NEURAL NETWORK LAVENBERG-
MARQUARDT***

Kartika Tri Wahyu Ramadhani (0231 15 4000 0026)

Dosen Pembimbing

Dr.Ir. Syamsul Arifin, MT

Prof. Dr. Ir. Aulia Siti Aisjah, MT

Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2019

Halaman sengaja dikosongkan



FINAL PROJECT- TF181801

**THE IMPLEMENTATION OF MACHINE
LEARNING IN MARITIME WEATHER
PREDICTION USING THE ARTIFICIAL NEURAL
NETWORK LAVENBERQ-MARQUARDT
ALGORITHM**

Kartika Tri Wahyu Ramadhani (0231 15 4000 0026)

Supervisors

Dr.Ir. Syamsul Arifin, MT

Prof. Dr. Ir. Aulia Siti Aisjah, MT

**Department of Engineering Physics
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

Halaman sengaja dikosongkan

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan dibawah ini :

Nama : Kartika Tri Wahyu Ramadhani

NRP : 0231154000026

Departemen : Teknik Fisika FTI-ITS

dengan ini menyatakan bahwa tugas akhir saya yang berjudul “Implementasi *Machine Learning* dalam Prediksi Cuaca Maritim dengan Menggunakan Algoritma *Artificial Neural Network Lavenberg-Marquard*” adalah bebas dari plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 16 Juli 2019

Yang membuat pernyataan



Kartika Tri Wahyu Ramadhani

Halaman sengaja dikosongkan

LEMBAR PENGESAHAN
IMPLEMENTASI *MACHINE LEARNING* DALAM
PREDIKSI CUACA MARITIM DENGAN
MENGGUNAKAN ALGORITMA *ARTIFICIAL NEURAL*
NETWORK LAVENBERG-MARQUARDT

TUGAS AKHIR

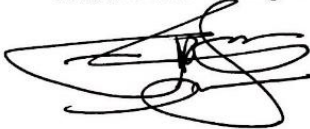
Oleh :



Kartika Tri Wahyu Ramadhani
0231154000026

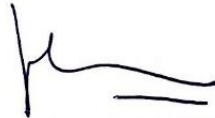
Surabaya, 20 Juli 2019
Mengetahui

Dosen Pembimbing 1



Dr. Ir. Syamsul Arifin, MT
NIP. 19630907 198903 1 004

Dosen Pembimbing 2



Prof. Dr. Ir. Aulia Siti Aisjah, MT
NIP. 19660116 198903 2 001

Menyetujui,

Kepala Departemen Teknik Fisika FTI-ITS



Agus Muhammad Hatta, ST, M.Si, Ph.D

NIP. 19780902 200312 1 002

Halaman sengaja dikosongkan

**IMPLEMENTASI MACHINE LEARNING
DALAM PREDIKSI CUACA MARITIM
DENGAN MENGGUNAKAN ALGORITMA
ARTIFICIAL NEURAL NETWORK
LAVENBERQ-MARQUARDT**



TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Syarat Memperoleh Gelar
Sarjana Teknik
pada
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

KARTIKA TRI WAHYU RAMADHANI
0231154000026

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Syamsul Arifin, M.T .....(Pembimbing 1)
2. Prof. Dr. Ir. Aulia Siti Aisjah, M.T(Pembimbing 2)
3. Dr. Suyanto, S.T, M.T .....(Ketua Penguji)
4. Moh. Kamalul Wafi, S.T, M.Sc. DIC.....(Penguji 1)

SURABAYA, 16 JULI 2019

Halaman sengaja dikosongkan

**IMPLEMENTASI *MACHINE LEARNING* DALAM
PREDIKSI CUACA MARITIM DENGAN
MENGUNAKAN ALGORITMA *ARTIFICIAL NEURAL
NETWORK LAVENBERG-MARQUARDT***

Nama Mahasiswa : Kartika Tri Wahyu Ramadhani
NRP : 0231154000026
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : 1. Dr. Ir. Syamsul Arifin, M.T
2. Prof. Dr. Ir. Aulia Siti Aisjah, M.T

Abstrak

Prediksi cuaca maritim akan sangat berpengaruh dalam berbagai aktivitas, sebagai salah satu contoh adalah aktivitas para nelayan. Perubahan cuaca maritim akan sangat bermanfaat bagi para nelayan dalam menentukan kelayakan jalur pelayaran sebagai aspek keselamatan dalam menentukan titik tangkap yang aman. Sistem prediksi saat ini, masih banyak yang menggunakan pendekatan statistika data *time series* dan menggunakan model regresi atau ARIMA dimana pendekatan tersebut masih memiliki tingkat akurasi yang rendah serta tidak dapat bekerja pada data nonlinier.

Penelitian tugas akhir ini, menerapkan *machine learning* dalam memprediksi cuaca maritim. *Machine learning* adalah salah satu aplikasi yang menjadi bagian dari *Artificial Intelligent (AI)*. Dalam penelitian ini, algoritma yang digunakan adalah algoritma *Artificial Neural Network Lavenberg-Marquardt*. Struktur algoritma *Artificial Neural Network* ini digunakan agar dapat menganalisa data dengan struktur logika yang mirip dengan bagaimana manusia mengambil keputusan. Algoritma *Artificial Neural Network* yang digunakan dalam penelitian ini adalah algoritma *Artificial Neural Network Lavenberg-Marquardt*. Dari algoritma tersebut akan dilihat performansinya dalam memprediksi cuaca maritim. Penelitian ini akan melihat performansi dari algoritma *Lavenberg Marquardt* dalam memprediksi cuaca dengan perbedaan setiap parameternya.

Dari hasil penelitian yang telah dilakukan didapatkan hasil bahwa semakin bertambahnya *hidden node* tidak membuat performa JST semakin baik. Hal ini ditunjukkan dengan nilai MSE yang semakin bertambah seiring bertambahnya *hidden node*. Sedangkan pada variasi target *error* didapatkan hasil bahwa performansi jaringan terbaik adalah ketika target *error* 0.001.

Hasil MSE pada prediksi kecepatan angin adalah sebesar 0.026385 pada arsitektur *hidden node* 10 dan target *error* 0.002, pada prediksi arah angin memiliki MSE sebesar 0.060009 pada arsitektur *hidden node* 10 dan target *error* 0.001, untuk prediksi curah hujan memiliki performansi dengan nilai MSE sebesar 0.01267 pada *hidden node* 10 dan target *error* 0.001.

Kata Kunci : *Machine Learning, Artificial Neural Network, Lavenberg-Marquadt*, Cuaca Maritim

THE IMPLEMENTATION OF MACHINE LEARNING IN MARITIME WEATHER PREDICTION USING ARTIFICIAL NEURAL NETWORK LAVENBERG-MARQUARDT ALGORITHM

Name : Kartika Tri Wahyu Ramadhani
NRP : 0231154000026
Department : Teknik Fisika FTI-ITS
Supervisor : 1. Dr. Ir. Syamsul Arifin, M.T
2. Prof. Dr. Ir. Aulia Siti Aisjah, M.T

Abstract

Maritime weather prediction will be very influential in various activities, as one example is the activity of fishermen. Maritime weather changes will be very beneficial for fishermen in determining the feasibility of shipping lanes as an aspect of safety in determining safe fishing points. Current prediction system, there are still many who use the time series data statistical approach and use a regression model or ARIMA where the approach still has a low level of accuracy and cannot work on nonlinear data.

In this final project, applying machine learning in predicting maritime weather. Machine learning is one application that is part of Artificial Intelligent (AI). In this study, the algorithm used is the Lavenberg-Marquardt Artificial Neural Network algorithm. Structure of the Artificial Neural Network algorithm is used to be able to analyze data with a logical structure similar to how humans make decisions. The Artificial Neural Network algorithm used in this study is the Artificial Neural Network Lavenberg-Marquardt algorithm. From the algorithm, it will be seen its performance in predicting maritime weather. This study will look at the performance of the Lavenberg Marquardt algorithm in predicting weather with differences in each parameter.

From the results of the research that has been done the results show that the more the increase in hidden nodes does not make the ANN performance better. This is indicated by the value

of MSE which increases with the addition of hidden nodes. Whereas the variation of target error shows that the best network performance is when the target error is 0.001.

MSE results on wind speed prediction is 0.026385 on hidden node 10 architecture and error target 0.002, in wind direction prediction has MSE of 0.060009 on hidden node 10 architecture and target error 0.001, for rainfall prediction has a performance with MSE value of 0.01267 in hidden node 10 and target error 0.001.

Keywords: *Machine Learning, Artificial Neural Network, Lavenberg-Marquadt, Maritime Weather*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena rahmat dan hikmat-Nya penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir yang berjudul “Implementasi *Machine Learning* dalam Prediksi Cuaca Maritim dengan Menggunakan *Algoritma Artificial Neural Network Lavenberq-Marquardt*”.

Penelitian ini menerapkan *Machine Learning* dalam memprediksi cuaca maritim.. *Machine learning* diperuntukkan dalam pengembangan sebuah sistem atau mesin cerdas yang dapat belajar sendiri tanpa harus di program oleh manusia secara berulang kali. Dengan kecerdasannya diharapkan *machine learning* dapat menghasilkan prediksi iklim lebih akurat berdasarkan *input* data yang telah diberikan sebelumnya.

Penulis menyadari bahwa tugas akhir ini tidak akan selesai tanpa bantuan dari berbagai pihak. Perkenankan penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Agus M. Hatta,S.T, M.Si, PhD selaku kepala departemen Teknik Fisika ITS.
2. Dr. Ir. Syamsul Arifin, M.T selaku dosen pembimbing tugas akhir, yang selalu memberikan saran dan motivasi dalam penyelesaian tugas akhir ini.
3. Prof. Dr. Ir. Aulia Siti Aisjah, M.T selaku dosen pembimbing tugas akhir, yang selalu memberikan saran dan motivasi dalam penyelesaian tugas akhir.
4. Segenap Bapak/Ibu dosen pengajar di departemen Teknik Fisika-ITS
5. Orang tua penulis Alfiyah dan Fadli yang senantiasa memberikan doa dan semangat dalam menyelesaikan tugas akhir.
6. Kakak dan saudara kembar penulis Fatikhah NRJ dan Kartika Dwi Ayu R yang senantiasa memberikan doa dan dukungannya.
7. Muhammad Khamim Assy'ari dan Rifki Kurniawan yang selalu membantu dalam pengerjaan tugas akhir serta dukungannya dalam penyelesaian tugas akhir ini.

8. Partner tugas akhir, Abiyyu, Isa, Nabila, Dina, Adel, Bima dan Wildan yang telah banyak membantu dan menemani selama pengerjaan tugas akhir.
9. Segenap keluarga besar penulis yang telah memberikan doa dan dukungannya.
10. Segenap asisten Laboratorium Rekayasa Bahan Teknik Fisika-ITS atas waktu, tempat dan kebersamaan yang diberikan.
11. Pihak BMKG Stasiun Perak II Surabaya yang telah memberikan informasi yang dibutuhkan dalam penyelesaian tugas akhir
12. Rekan-rekan mahasiswa Teknik Fisika-ITS yang senantiasa memberikan motivasi.

Semoga Allah SWT senantiasa membalas segala amal dan kebaikan pihak-pihak yang telah membantu dalam proses penyelesaian tugas akhir ini. Penulis juga menyadari bahwa pada tugas akhir ini masih terdapat banyak kekurangan. Oleh karena itu, segala bentuk kritik dan saran yang membangun akan diterima dengan senang hati. Semoga tugas akhir ini dapat bermanfaat dan berguna bagi penulis, para pembaca dan pihak-pihak yang membutuhkan.

Penulis.

DAFTAR ISI

Halaman Judul	i
Cover Page.....	iii
PERNYATAAN BEBAS PLAGIARISME.....	v
LEMBAR PENGESAHAN.....	vii
Abstrak	xi
<i>Abstract</i>.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xx
DAFTAR TABEL.....	xxiii
DAFTAR NOTASI.....	xxv
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan.....	4
1.4 Batasan Masalah	4
BAB II DASAR TEORI.....	5
2.1 Pengertian Cuaca	5
2.2 Unsur-unsur Cuaca	5
2.3 <i>Cuaca dalam Kegiatan Kelautan</i>	8
2.4 <i>Machine Learning</i>	9
2.5 <i>Artificial Neural Network</i>	11
2.6 Algoritma <i>Lavenberg-Marquardt</i>	16
2.6 Validasi Prediktor	18
2.7 Implementasi <i>Machine Learning</i> dalam sistem prediksi .	19
BAB III METODOLOGI PENELITIAN	25
3.1 Perumusan Masalah	26
3.2 Studi Literatur.....	26
3.3 Penentuan Variabel Cuaca	27
3.4 Pengumpulan Data.....	27
3.5 Pengolahan dan Analisis Data	28
3.5.1 Uji Missing Value	28
3.5.2 Uji <i>Outlier</i>	29

3.5.3 Uji Korelasi	32
3.5.4 Normalisasi.....	41
3.6 Perancangan <i>Lavenberq-Marquardt Neural Network</i>	53
3.6.1 Kecepatan Angin	54
3.6.2 Arah Angin.....	57
3.6.3 Curah Hujan	60
3.7 Pengujian prediktor.....	62
3.8 Pengujian Jaringan dengan Data <i>Real Time</i>	63
3.9 Analisis Hasil dan Pembahasan	65
3.10 Kesimpulan dan Saran	65
BAB IV HASIL dan PEMBAHASAN	67
4.1 Hasil Prediksi berdasarkan Variasi <i>Hidden node</i>	67
4.2 Hasil Prediksi berdasarkan Variasi Target <i>Error</i>	69
4.3 Pembahasan Arsitektur Terbaik dari Variasi Terbaik	75
4.4 Hasil Prediksi Cuaca Menggunakan <i>Artificial Neural Network Lavenberq-Marquardt</i>	76
4.4 Validasi Prediksi menggunakan Data <i>Realtime</i>	88
BAB V Kesimpulan dan Saran.....	103
5.1 Kesimpulan.....	103
5.2 Saran.....	103
Daftar Putaka	105
LAMPIRAN A, VARIASI TARGET <i>ERROR</i> 0.001.....	109
A.1 Curah Hujan	109
A.2 Kecepatan Angin	111
A.3 Arah Angin	114
LAMPIRAN B, VARIASI TARGET <i>ERROR</i> 0.002.....	117
B.1 Curah hujan	117
B.2 Kecepatan Angin	120
B.3 Arah Angin	123
Lampiran C, VARIASI TARGET <i>ERROR</i> 0.003	127
C.1 Curah Hujan	127
C.2 Kecepatan Angin	130
C.3 Arah Angin	133
Lampiran D, Matlab Code.....	137
D.1 Normalisasi data	137
D.2 Pelatihan JST-LM.....	138

D.3 Pengujian JST-LM.....	140
Biodata Penulis	143

Halaman sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Hubungan <i>Artificial Intelligence, Machine Learning</i> dan <i>Deep Learning</i>	10
Gambar 2.2 struktur Neuron Tiruan (Gaxiola et al., 2014)	13
Gambar 2.3 fungsi aktivasi linier	14
Gambar 2.4 fungsi aktivasi sigmoidal	14
Gambar 2.5 fungsi aktivasi hyperbolic tangen	14
Gambar 2.6 fungsi aktivasi yang lain	14
Gambar 2.7 macam-macam arsitektur Artificial Neural Network (ANN) (Aliev & Guirimov, 2014).....	15
Gambar 3.1 Diagram Alir Penelitian	25
Gambar 3.2 Presentase Data Outlier pada Data Kecepatan Angin	30
Gambar 3.3 Presentase Data Outlier pada Data Suhu	31
Gambar 3.4 Presentase Data Outlier pada Data Curah Hujan ...	31
Gambar 3.5 Persebaran data suhu sebelum normalisasi	43
Gambar 3.6 Persebaran data suhu setelah normalisasi	44
Gambar 3.7 Persebaran data kecepatan angin sebelum normalisasi	45
Gambar 3.8 Persebaran data kecepatan angin setelah normalisasi	46
Gambar 3.9 Persebaran data arah angin sebelum normalisasi ...	47
Gambar 3.10 Persebaran data arah angin setelah normalisasi	48
Gambar 3.11 Persebaran data kelembaban sebelum normalisasi	49
Gambar 3.12 Persebaran data kelembaban setelah normalisasi.	50
Gambar 3.13 Persebaran data curah hujan sebelum normalisasi	51
Gambar 3.14 Persebaran data curah hujan setelah normalisasi.	52
Gambar 3.15 Arsitektur JST Kecepatan Angin.....	55
Gambar 3.16 Arsitektur JST Arah Angin.....	58
Gambar 3.17 Arsitektur JST Curah Hujan	61
Gambar 3.18 Proses pengambilan data realtime	63
Gambar 3.19 <i>Digram blok sistem prediktor pada buoyweather</i>	64

Gambar 4.1	Hasil prediksi Kecepatan angin berdasarkan variasi <i>hidden node</i>	67
Gambar 4.2	Hasil prediksi Arah angin berdasarkan variasi <i>hidden node</i>	68
Gambar 4.3	Hasil prediksi Curah Hujan berdasarkan variasi <i>hidden node</i>	69
Gambar 4.4	Hasil Prediksi Kecepatan Angin berdasarkan variasi target <i>error</i>	71
Gambar 4.5	Hasil Prediksi Arah Angin berdasarkan variasi target <i>error</i>	73
Gambar 4.6	Hasil Prediksi Curah Hujan berdasarkan variasi target <i>error</i>	75
Gambar 4.7	Perbandingan MSE uji dan MSE latih pada prediksi kecepatan angin	77
Gambar 4.8	Grafik pelatihan JST kecepatan angin	78
Gambar 4.9	Hasil uji prediksi kecepatan angin	79
Gambar 4.10	Perbandingan MSE uji dan MSE latih pada prediksi Arah Angin	80
Gambar 4.11	Grafik pelatihan JST pada prediksi arah angin	82
Gambar 4.12	Hasil uji prediksi arah angin	83
Gambar 4.13	Perbandingan MSE uji dan MSE latih pada prediksi curah hujan	84
Gambar 4.14	Hasil pelatihan prediksi curah hujan	86
Gambar 4.15	Hasil uji prediksi curah hujan	87
Gambar 4.16	Arsitektur JST	88
Gambar 4.17	Hasil validasi realtime prediksi kecepatan angin .	93
Gambar 4.18	Hasil validasi realtime prediksi arah angin	97
Gambar 4.19	Hasil validasi realtime prediksi curah hujan	101

DAFTAR TABEL

Tabel 2. 1 Klasifikasi Curah Hujan (BMKG, 2016)	6
Tabel 2. 2 Kecepatan Angin Skala Beufort (WMO, 2008)	8
Tabel 2. 3 Macam-macam persamaan fungsi aktivasi (Yang et al., 2016)	14
Tabel 2.4 Tabel perbandingan pendekatan ARIMA dan pendekatan AI dalam sistem prediksi (Aisjah & Arifin, 2011) (Voyant et al., 2017)	20
Tabel 2. 5 Tabel hasil penelitian sebelumnya	22
Tabel 2. 6 Lanjutan tabel hasil penelitian sebelumnya.....	23
Tabel 3.1 Ukuran Presentase Missing Value dan Keputusan	29
Tabel 3.2 Jumlah Data perjam yang dioeroleh dari BMKG Perak	29
Tabel 3.3 Jumlah Data Outlier	32
Tabel 3.4 Klasifikasi ukuran korelasi berdasarkan nilai koefisien r.	34
Tabel 3.5 Hasil Uji Korelasi Terhadap Wind Speed (Kecepatan Angin)	35
Tabel 3.6 Variabel historis <i>Input</i> untuk prediksi WS(t+12)	36
Tabel 3.7 Hubungan Variabel <i>output</i> WS(t+12) dengan variabel historis <i>input</i>	36
Tabel 3.8 Hasil Uji Korelasi Terhadap Variabel Wind Direction (Arah Angin)	37
Tabel 3.9 Variabel historis <i>Input</i> untuk prediksi WD(t+12)	38
Tabel 3.10 Hubungan Variabel <i>output</i> WD(t+12) dengan variabel historis <i>input</i>	38
Tabel 3.11 Hasil Uji Korelasi Terhadap Variabel Curah Hujan.	39
Tabel 3.12 Variabel historis <i>Input</i> untuk prediksi CH(t+12).....	40
Tabel 3.13 Hubungan Variabel <i>output</i> CH(t+12) dengan variabel historis <i>input</i>	41
Tabel 3.14 Hasil MSE berdasarkan percobaan variabel kecepatan angin.....	54
Tabel 3.15 Spesifikasi Arsitektur JST untuk prediktor kecepatan angin.....	57

Tabel 3.16 Hasil MSE berdasarkan percobaan variabel arah angin	57
Tabel 3.17 Spesifikasi Arsitektur JST untuk prediktor arah angin	60
Tabel 3.18 Hasil MSE berdasarkan percobaan variabel curah hujan	60
Tabel 3.19 Spesifikasi Arsitektur JST untuk prediktor curah hujan	62
Tabel 4.1 Peformansi Prediksi Kecepatan angin berdasarkan variasi target <i>error</i>	70
Tabel 4.2 Performansi Prediksi Arah angin berdasarkan variasi target <i>error</i>	72
Tabel 4.3 Hasil Prediksi Curah Hujan berdasarkan variasi target <i>error</i>	74
Tabel 4.4 Rincian arsitektur terbaik dari variasi terbaik	76
Tabel 4.5 Bobot terbaik hasil pelatihan prediksi kecepatan angin	90
Tabel 4.6 Hasil Pengujian <i>Realtime</i> prediksi kecepatan Angin..	92
Tabel 4.7 Bobot terbaik hasil pelatihan prediksi arah angin	94
Tabel 4.8 Hasil Pengujian <i>Realtime</i> prediksi Arah Angin.....	96
Tabel 4.9 Bobot terbaik hasil pelatihan prediksi curah hujan	98
Tabel 4.10 Hasil Pengujian <i>Realtime</i> prediksi Curah Hujan	100

DAFTAR NOTASI

Notasi	Keterangan
T	Suhu rata-rata
RH	Kelembaban rata-rata
CH	Curah Hujan
WS	Kecepatan Angin
WD	Arah Angin
T	Suhu minimum
t	Hari ini
t+12	12 jam kedepan
t-12	12 jam sebelumnya
t-24	24 jam sebelumnya
t-36	36 jam sebelumnya
t-48	48 jam sebelumnya
t-60	60 jam sebelumnya
RMSE	<i>Root mean square error</i>
MAPE	<i>Mean absolute percentage error</i>
JST	Jaringan Saraf Tiruan
σ	Standar deviasi
w_{ij}	Bobot/koeffisien
z_l	Nilai keluaran JST
δ_k	Faktor unit keluaran JST
α	Laju pembelajaran
Δw	Perubahan bobot
E	<i>Sum square error</i>
SSE	<i>Sum square error</i>
$e_{p,m}$	<i>Error</i> pelatihan keluaran m saat menggunakan pola p.
$d_{p,m}$	Target keluaran JST
$o_{p,m}$	Keluaran jst
g	Gradien
H	Matriks Hessian
J	Matriks Jacobian

Notasi V_i ϕ r C_{xy} \hat{Y} **Keterangan**

Vektor mutan

Koefisien pearson

Koefisien korelasi

Kovarian variabel x dan y

Keluaran prediktor

BAB I

PENDAHULUAN

1.1 Latar belakang

Perubahan cuaca maritim akan sangat berpengaruh dalam berbagai aktivitas, sebagai salah satu contoh yaitu pada bidang penerbangan dan pelayaran. Cuaca maritim juga akan sangat mempengaruhi aktivitas para nelayan dalam menjalankan profesinya sebagai pencari ikan di tengah laut. Pada bidang penerbangan dan pelayaran, perubahan cuaca maritim akan berpengaruh dalam menentukan jalur yang aman untuk dapat digunakan, sedangkan dari sudut pandang para nelayan perubahan cuaca maritim akan sangat berpengaruh dalam menentukan kelayakan jalur pelayaran sebagai aspek keselamatan dalam menentukan titik tangkap yang aman. Maka dari itu peramalan cuaca akan sangat berguna dalam hal keselamatan pada aktivitas maritim (De Girolamo, Di Risio, Beltrami, Bellotti, & Pasquali, 2017).

Peramalan merupakan perkiraan nilai dari suatu variabel (atau himpunan variabel) pada waktu yang akan datang (Ali, Yohanna, Puwu, & Garkida, 2016). Sistem pengambilan keputusan dalam penilaian risiko kegiatan maritim dapat mempertimbangkan data kondisi cuaca disamping karakteristik kapal (Balmat, Lafont, Maifret, & Pessel, 2011). Sistem peramalan itu sendiri dapat diklasifikasikan atas *short-term forecast* pada rentang hingga 6 jam, yang dibagi lagi menjadi dua yaitu *now-casting* (0-3jam) dan *short-casting* (3-6jam) dan ada *long-casting* (hingga 6 hari) (Paulescu, Paulescu, Gravila, & Badescu, 2013). Dalam penelitian ini prediksi cuaca maritim akan dilakukan dengan media *buoyweather*. Sehingga akan lebih tepat dikatakan sebagai *short-term forecast* karena berkisar antara 3-12 jam.

Perkembangan metode yang digunakan dalam memprediksi cuaca adalah diawali dengan menggunakan pendekatan statistika data *time series* menggunakan model regresi atau ARIMA yang membutuhkan data historis tidak sedikit dengan akurasi prediksi rendah (Aisjah & Arifin, 2011), selain itu penggunaan metode

ARIMA ini lebih sesuai dalam *long-term forecasting* dan tidak dapat bekerja pada data non-linier seperti kondisi cuaca saat ini yang fluktuatif dan berubah-ubah (Voyant et al., 2017). Seiring dengan perkembangan ilmu dan teknologi saat ini mampu diciptakan metode prediksi cuaca baik pada beberapa jam atau beberapa hari kedepan dengan pendekatan baru yaitu *Artificial Intelligence (AI)*. Dalam perkembangan AI, terdapat dua konsep yang saat ini banyak digunakan dan dikembangkan yaitu *machine learning*.

Machine learning adalah salah satu aplikasi yang menjadi bagian dari *Artificial Intelligent (AI)*. *Machine learning* diperuntukkan dalam pengembangan sebuah sistem atau mesin cerdas yang dapat belajar sendiri tanpa harus diprogram oleh manusia secara berulang kali. Dengan algoritmanya, *machine learning* akan membuat proses validasi data menjadi sederhana dan tanpa perlu campur tangan manusia. Hal tersebut disebabkan karena ciri khas yang dimiliki oleh *machine learning* yaitu kecerdasan memahami pola data.

Menurut Qing Yi Feng dalam jurnalnya, menerangkan bahwa *machine learning* telah terbukti efektif dalam memprediksi energi surya untuk 30 pembangkit listrik tenaga matahari (Feng et al., 2016). Dengan cara yang sama *machine learning* juga digunakan untuk menjalankan prediksi iklim seperti prediksi event El Nino dan prediksi variasi tahunan arus Kuroshio (Qiu, 2008). Selain itu John Abbot dan Jennifer Marohasy juga memanfaatkan *machine learning* dalam membuat prediksi iklim (Abbot & Marohasy, 2017b). Telah banyak jurnal yang dihasilkan keduanya mengenai prediksi iklim meski sebagian besar merupakan studi kasus cuaca dan iklim di Australia (Abbot & Marohasy, 2017a).

Artificial Neural Network / Jaringan Saraf Tiruan (JST) adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu (Andrian & Ningsih,

2014a). Cara kerja *Neural Network* seperti cara kerja manusia, yaitu belajar melalui contoh. Lapisan-lapisan penyusun *Neural Network* dibagi menjadi 3, yaitu lapisan *input (input layer)*, lapisan tersembunyi (*hidden layer*), dan lapisan *output (ouput layer)* (Sutojo, Mulyanto, & Suhartono, 2011).

Algoritma *Levenberg-Marquardt* merupakan pengembangan algoritma *backpropagation* standar (Rahmat, Setiawan, & Purnomo, 2006). *Artificial Neural Network* memerlukan suatu pelatihan atau pembelajaran dalam memproses *input* yang diberikan. Algoritma *Levenberg-Marquardt* merupakan algoritma pelatihan yang sangat efisien untuk pelatihan jaringan dengan ukuran jaringan yang kecil hingga jaringan yang tidak terlalu besar dengan menggunakan algoritma *Backpropagation* (Nikentari, Ritha, & Wati, 2018). Berdasarkan penelitian terdahulu menunjukkan bahwa algoritma *Levenberg-Marquardt* adalah algoritma yang cepat dan memiliki konvergensi yang stabil (Yu H, 2011). Penelitian prediksi curah hujan menggunakan algoritma pelatihan *Levenberg-Marquardt* telah dilakukan di beberapa penelitian (Naik & Pathan, 2012).

Penelitian ini adalah penelitian tentang implementasi *machine learning* dalam perkiraan cuaca maritim menggunakan algoritma *Artificial Neural Network*. Algoritma yang digunakan pada penelitian ini adalah algoritma *Lavenberg-Marquardt*. Dari algoritma tersebut akan dilihat performansinya dalam memprediksi cuaca maritim. Penelitian ini akan melihat performansi dari algoritma *Lavenberg Marquardt* dalam memprediksi cuaca dengan perbedaan setiap parameternya. Hasil prediksi dengan tingkat akurasi tinggi diharapkan mampu memprediksi cuaca maritim sehingga dapat membantu kegiatan kelautan diberbagai bidang dan menciptakan keselamatan maritim di setiap kegiatan kelautan.

1.2 Perumusan Masalah

Berdasarkan latar belakang diatas maka perumusan masalah yang akan diselesaikan dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana hasil prediksi cuaca maritim menggunakan algoritma *Artificial Neural Network Levenberg-Marquardt* ?

2. Berapakah nilai parameter (*targer error* dan *hidden node*) dari algoritma *Lavenberg Marquardt* yang tepat agar mendapatkan nilai akurasi yang tinggi dalam memprediksi cuaca ?

1.3 Tujuan

Dari perumusan masalah diatas maka didapatkan tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Menganalisa hasil prediksi cuaca maritim menggunakan algoritma *Artificial Neural Network Levenberg-Marquardt* yang ditunjukkan dengan tingkat akurasi dari hasil prediksi.
2. Menganalisa perbedaan hasil akurasi hasil prediksi cuaca maritim menggunakan algoritma *Artificial Neural Network Levenberg-Marquardt* dengan mengubah parameter yang ada dalam algoritma *Lavenberg Marquardt* yaitu *hidden node* dan *taget error*.

1.4 Batasan Masalah

Berikut adalah batasan masalah yang ada pada Tugas Akhir ini :

1. Prediksi ini dilakukan pada variabel curah hujan, kecepatan angin dan arah angin dari cuaca maritim
2. Objek yang diteliti adalah cuaca yang berada di wilayah laut atau samudera tepatnya di Perairan Utara Jawa Timur
3. Prediksi ini menerapkan teknologi *machine learning* dengan menggunakan algoritma *Artificial Neural Network Levenberg-Marquardt* dengan memvariasikan paramameter *hidden layer*, *target error* dan parameter LM.
4. Program prediksi cuaca maritim dibuat dengan bantuan program MATLAB R2014b dan Toolbox-nya

BAB II DASAR TEORI

2.1 Pengertian Cuaca

Cuaca adalah seluruh kejadian di atmosfer bumi yang merupakan bagian kehidupan sehari-hari manusia di dunia. Cuaca juga merupakan keadaan yang terjadi di permukaan bumi yang dipengaruhi oleh kondisi udara, yaitu tekanan dan suhu. Cuaca di setiap planet berbeda-beda tergantung pada jaraknya dari matahari dan pergerakan gas di setiap atmosfer planet-planet tersebut, bahkan dalam suatu perkotaan juga seringkali mempunyai jenis cuaca yang berbeda dari daerah sekelilingnya (Kodoatie & Sjarief, 2010). Sedangkan menurut kamus besar bahasa Indonesia cuaca adalah keadaan udara (tentang temperatur, cahaya matahari, kelembapan, kecepatan angin dan sebagainya) pada suatu tempat tertentu dengan jangka waktu terbatas.

2.2 Unsur-unsur Cuaca

a. Curah Hujan

Hujan adalah presipitasi dalam bentuk cair. Titik-titik air hujan, berjari-jari antara 0,04-3 mm. Siklus terjadinya hujan dapat dimulai dari penyinaran matahari atau biasa disebut evaporasi. Selanjutnya, uap air yang terbawa ke atmosfer mengalami kondensasi akibat dari temperatur atmosfer yang sangat dingin dan terkumpul jadi awan. Adanya angin yang bergerak vertikal mengakibatkan awan bergumpal, sedangkan pergerakan horizontal angin akan membawa awan ke daerah yang bertekanan lebih rendah. Setelah mencapai saturasi, akan terjadi presipitasi berbentuk hujan. Hujan yang mengenai permukaan bumi akan diserap oleh tanah, sedangkan yang mengenai sungai akan dialirkan kembali ke laut dan akan mengulang siklus hidrologi (Safanah, 2008).

Curah hujan untuk wilayah tropis, misalnya Indonesia lebih tinggi dari evaporasinya, sehingga wilayahnya memiliki tutupan awan yang tinggi dan menyebabkan jumlah radiasi yang sampai permukaan rendah (Aldrian, 2008). Tren curah hujan di Indonesia hasil observasi BMKG sejak tahun 1981 hingga 2016, memiliki

tren yang bernilai positif. Di beberapa wilayah bernilai negatif dengan besaran yang bervariasi. Berikut merupakan klasifikasi curah hujan berdasarkan Badan Meteorologi, Klimatologi dan Geofisika (BMKG) dapat dilihat pada tabel dibawah ini. Untuk kuantitasnya, terbagi menjadi empat parameter yaitu ringan, sedang, lebat dan sangat lebat.

Tabel 2. 1 Klasifikasi Curah Hujan (BMKG, 2016)

Jenis Hujan	Kuantitas (mm/hari)
Ringan	5-20
Sedang	20-50
Lebat	50-100
Sangat Lebat	>100

Pada tabel 2.1 dapat dilihat pembagian klasifikasi jenis hujan berdasarkan kuantitasnya. Terdapat 4 jenis hujan yang terbagi dalam ringan dengan kuantitas 5-20 mm/hari, sedang dengan kuantitas 20-50 mm/hari, lebat dengan kuantitas 50-100 mm/hari dan sangat lebat dengan kuantitas lebih dari 100 mm/hari.

b. Kelembaban

Kelembaban adalah kandungan total uap air di udara atau banyaknya kandungan uap air di atmosfer. Udara atmosfer adalah campuran dari udara kering dan uap air. Kelembaban udara menggambarkan kandungan uap air di udara yang dapat dinyatakan sebagai kelembaban mutlak. Secara umum kelembaban (*Relative Humidity*) adalah istilah yang digunakan untuk menggambarkan jumlah uap air yang ada di udara dan dinyatakan dalam persen dari jumlah uap air maksimum dalam kondisi jenuh (Gunawan D, 2011).

Satuan kelembaban yang umum digunakan adalah RH, yaitu *Relative Humidity* atau kelembaban relatif. *Relative Humidity* adalah satuan pengukuran yang menggambarkan jumlah titik-titik air di udara pada suhu tertentu yang dibandingkan dengan jumlah maksimum titik-titik air yang dapat dikandung di udara pada suhu tersebut. RH dinyatakan dalam nilai prosentase. Udara panas dapat menyimpan titik-titik air lebih banyak daripada udara dingin.

Semakin tinggi nilai RH maka semakin tinggi terjadinya pengembunan. 100% RH berarti bahwa penambahan titik-titik air di udara akan langsung mengembun. Tingkat kelembaban yang ideal adalah 50-55% RH. 50% RH menunjukkan bahwa udara terisi setengah dari kapasitas maksimum air yang bisa ditampung di udara.

c. Suhu

Suhu atau temperatur udara adalah derajat panas dari aktivitas molekul dalam atmosfer. Alat untuk mengukur suhu atau temperatur udara atau derajat panas disebut Thermometer. Biasanya pengukuran suhu atau temperatur udara dinyatakan dalam skala Celcius (C), Reamur (R), dan Fahrenheit (F). Perubahan suhu udara di satu tempat dengan tempat lainnya bergantung pada ketinggian tempat dan letak astronomisnya (lintang). Perubahan suhu karena perbedaan ketinggian jauh lebih cepat daripada perubahan suhu karena perbedaan letak lintang. Biasanya, perubahan suhu terjadi berkisar 0,6 derajat celcius tiap kenaikan 100 m (Puspita & Yulianti, 2016).

d. Arah dan Kecepatan Angin

Angin adalah gerak udara yang sejajar dengan permukaan bumi. Udara bergerak dari daerah bertekanan tinggi ke daerah bertekanan rendah. Angin adalah besaran vektor yang mempunyai arah dan kecepatan. Arah angin dinyatakan dalam derajat. Sebagai contoh arah 360° adalah arah Utara (U), arah 22,5° adalah arah Utara Timur Laut (UTL), dan sebagainya. Kecepatan angin dinyatakan dalam satuan meter per sekon, kilometer per jam atau knot (1 knot~0,51 m/s) dan diukur menggunakan anemometer.

Kekuatan angin ditentukan oleh kecepatannya, makin cepat angin bertiup maka makin tinggi/besar kekuatannya. Pada tahun 1804 Beaufort seorang Laksamana Inggris telah membuat daftar kekuatan dan kecepatan angin yang digunakannya untuk pelayaran.

Daftar tersebut dinamakan Skala Beaufort yang memiliki nilai 0 sampai 12. World Meteorological Organization (WMO) menggunakan klasifikasi kecepatan angin berdasarkan skala Beaufort. Skala beaufort adalah sistem yang menafsirkan laporan

kecepatan angin berdasarkan efek yang ditimbulkan dari kecepatan angin tersebut. Jenis klasifikasi kecepatan angin dapat dilihat pada Tabel dibawah ini. Skala Beaufort menunjukkan semakin besar angka pada skalanya maka angin berhembus semakin kencang dan bisa semakin merusak.

Tabel 2. 2 Kecepatan Angin Skala Beufort (WMO, 2008)

Skala Beaufort	Kecepatan Angin		Nama
	m/dt	knot	
0	0-1	0-3	<i>Calm</i>
1	2-3	3-6	<i>Light Air</i>
2	3-5	6-9	<i>Light Breeze</i>
3	5-7	9-12	<i>Gentle Breeze</i>
4	7-9	13-16	<i>Moderate Breeze</i>
5	9-12	17-22	<i>Fresh Breeze</i>
6	12-14	22-27	<i>Strong Breeze</i>
7	14-17	27-32	<i>Near Gale</i>
8	17-19	32-38	<i>Gale</i>
9	19-23	38-43	<i>Strong Gale</i>
10	23-26	44-51	<i>Storm</i>
11	26-31	51-58	<i>Violent Storm</i>
12	>31	>58	<i>Hurricane</i>

e. Tekanan Udara

Tekanan udara adalah suatu gaya yang timbul akibat adanya berat dari lapisan udara. Besarnya tekanan udara di setiap tempat pada suatu saat berubah-ubah. Makin tinggi suatu tempat dari permukaan laut, makin rendah tekanan udaranya. Hal ini disebabkan karena makin berkurangnya udara yang menekan. Besarnya tekanan udara diukur dengan barometer dan dinyatakan dengan milibar (mb). Tekanan udara dapat dibedakan menjadi 3 macam, yaitu tekanan udara tinggi (lebih dari 1013 mb), tekanan udara rendah (kurang dari 1013 mb), dan tekanan di permukaan laut (sama dengan 1013 mb).

2.3 Cuaca dalam Kegiatan Kelautan

Informasi meteorologi dapat meliputi data dan keterangan tentang keadaan cuaca masa lampau, keterangan tentang keadaan cuaca yang berlangsung dari waktu ke waktu hingga saat ini dan

keadaan yang diperkirakan akan terjadi di waktu kemudian. Namun karena kegiatan kelautan bermacam-macam sehingga masing-masing memerlukan informasi yang berbeda. Unsur cuaca laut dan fenomena cuaca laut yang penting bagi kegiatan kelautan pada umumnya adalah: angin dan gelombang, curah hujan, suhu udara, suhu laut serta arus laut. Pada umumnya keselamatan, keamanan dan kenyamanan adalah hal yang diharapkan dalam kegiatan kelautan. Fungsi utama dari informasi meteorologi bagi pelayaran adalah memberi petunjuk pemilihan jalan agar dapat berlayar dengan aman, nyaman, selamat sampai tujuan dan tepat waktu (Suwardjo, Haluan, Jaya, & Poernomo, 2017).

Pemilihan alur pelayaran merupakan salah satu upaya dalam pelayaran yang berkaitan dengan kondisi cuaca laut dengan maksud agar pelayaran dapat dilakukan dengan selamat, aman, nyaman, dan efisien menggunakan waktu pelayaran sehingga diperoleh penghematan biaya operasional. Unsur dan fenomena cuaca laut yang perlu dalam pemilihan alur pelayaran adalah angin dan gelombang, awan dan hujan, suhu udara dan suhu laut (Suwardjo et al., 2017).

2.4 Machine Learning

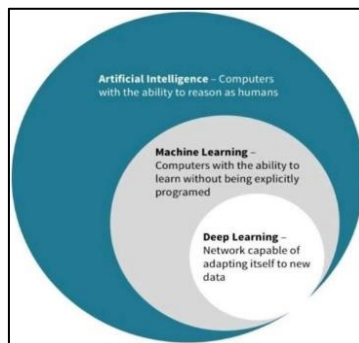
Machine Learning adalah salah satu aplikasi yang menjadi bagian dari *Artificial Intelligence*. *Machine Learning* dalam bahasa Indonesia disebut juga pembelajaran mesin. *Machine learning* diperuntukkan dalam pengembangan sebuah sistem atau mesin cerdas yang dapat belajar sendiri tanpa harus di program oleh manusia secara berulang kali. Kemampuan dan kecerdasan *machine learning* diharapkan dapat menghasilkan prediksi iklim lebih akurat berdasarkan *input* data yang telah diberikan sebelumnya.

Artificial Intelligence sendiri merupakan teknologi buatan manusia yang menggabungkan kemampuan matematika dengan proses statistika secara algoritmik. Para ahli menyebut *Artificial Intelligence* seolah memindahkan kecerdasan otak manusia ke dalam sebuah mesin sehingga mesin tersebut bisa berpikir seperti manusia. Teknologi *Artificial Intelligence* akan dapat membaca

pola sistem iklim berdasarkan data iklim yang telah tersimpan sebelumnya untuk kemudian menghasilkan prediksi cuaca ataupun iklim berdasarkan pola tersebut. Hal ini karena dalam *Artificial Intelligence* terdapat kurang lebih tiga aspek yang mendukung *Artificial Intelligence* yaitu :

1. *Fuzzy Logic*(FL), metode yang digunakan oleh mesin untuk mengadaptasi atau meniru cara makhluk hidup menyesuaikan kondisi kemudian memberikan keputusan yang tidak kaku yang hanya seperti 0 atau 1.
2. *Evolutionary Computing*(EC), sebuah pendekatan yang menggunakan skema evolusi dengan jumlah data individu yang banyak. Hasil pendekatan kemudian memberikan sebuah pengujian untuk menyeleksi individu terbaik untuk membangkitkan generasi selanjutnya. Penerapannya misalnya menggunakan kawin silang pada tanaman dan hewan.
3. *Machine Learning* (ML) atau pembelajaran mesin. Metode ini merupakan teknik yang paling banyak dikembangkan saat ini. Hal ini karena *Machine Learning* banyak digunakan untuk menggantikan atau menirukan perilaku manusia dan kemudian membuat keputusan tertentu.

Hubungan *Artificial Intelligence*, *Machine Learning* dan *Deep Learning*, dijelaskan oleh Richi Nayak (Chen & Nayak, 2011) seperti pada gambar 2.1 berikut.



Gambar 2. 1 Hubungan *Artificial Intelligence*, *Machine Learning* dan *Deep Learning*

Ilustrasi gambar 2.1 menggambarkan hubungan antara *Artificial Intelligent*, *Machine learning* dan *Deep learning*. Jika digambarkan secara diagram, wilayah kajian AI akan jauh lebih besar dibandingkan *machine learning*. Hal tersebut dapat disimpulkan bahwa *machine learning* adalah bagian dari *Artificial Intelligent* atau *machine learning* merupakan implementasi dari *Artificial Intelligent* (Primartha, 2018).

Di dalam *machine learning* terdapat algoritma yang dapat dikelompokkan dalam:

1. *Supervised learning*
2. *Unsupervised learning*
3. *Reinforcement learning*
4. *Deep learning*.

Artificial Neural Network merupakan bagian dari *machine learning* karena masuk dalam salah satu algoritma *machine learning* yaitu *Supervised learning* (Primartha, 2018). Sebagian besar praktik *machine learning* mengandalkan algoritma *supervised learning*. Algoritma tersebut menggunakan sekumpulan data untuk *training* agar menghasilkan *output* sesuai harapan. Pada *supervised learning* menggunakan sebuah algoritma untuk mempelajari *mapping function* antara *input* dengan *output*. Berbagai kemungkinan *output* sudah diketahui dan data-data yang digunakan untuk latihan (*training*) sudah diberi label dengan jawaban yang benar. *Supervised learning* dapat bermanfaat untuk memprediksi sesuatu dengan bantuan *training* dataset (Primartha, 2018).

2.5 Artificial Neural Network

Salah satu metode *Artificial Intelligence* yang banyak digunakan dalam prediksi, rekognisi, dan klasifikasi adalah jaringan syaraf tiruan buatan. Jaringan syaraf buatan merupakan sub bagian dari *machine learning*. Jaringan syaraf tiruan memiliki karakteristik seperti kemampuan untuk mengorganikan diri sendiri, beradaptasi, dan pembelajar serta *non-linearity*, *non-locality*, *non-steady* dan *non-convex* yang menjadikannya metode yang ampuh

untuk masalah yang kompleks (Yang, Chen, Wang, Li, & Li, 2016).

Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan (JST) merupakan teknik pengolahan yang terinspirasi oleh cara memproses dari sistem syaraf biologis. Konsep dasarnya pengolahan informasi terdiri dari sejumlah besar elemen pengolah atau neuron yang saling terhubung. JST ini dikonfigurasi secara spesifik untuk klarifikasi data atau pengenalan pola melalui proses belajar yang disebut training. Sistem pembelajaran melibatkan penyesuaian koneksi sinaptik antar neuron (Shaban, El-Hag, & Matveev, 2009).

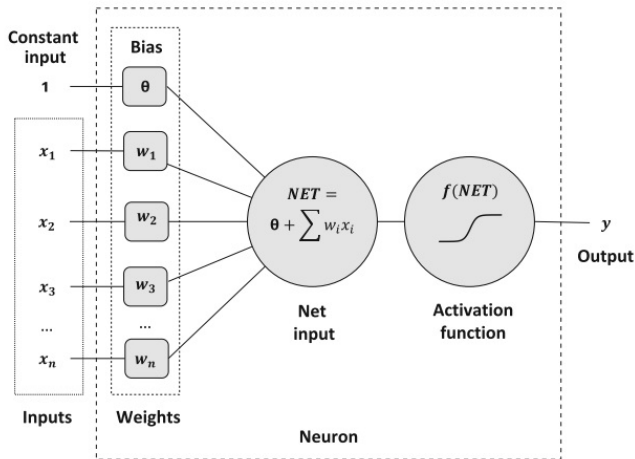
Sistem pada *Artificial Neural Network* (ANN) ini digunakan oleh organisme untuk mengolah data yang didapatkan dari lingkungan untuk mengambil keputusan lebih lanjut. Sehingga, *Artificial Neural Network* (ANN) atau Jaringan syaraf tiruan adalah model yang meniru perilaku pengolahan biologis sistem dari penerimaan data oleh ujung saraf, pemrosesan di otak, dan mengirimkan reaksi *output* ke sistem biologis lainnya (Aliev & Guirimov, 2014).

Performansi *Artificial Neural Network* dipengaruhi oleh struktur neuron dan arsitektur *neural network* (Aliev & Guirimov, 2014). Berikut adalah penjelasan mengenai struktur neuron dan arsitektur pada *Artificial Neural Network*:

a. Struktur Neuron pada *Artificial Neural Network* (ANN)

Jaringan syaraf tiruan terdiri atas banyak elemen neuron buatan, yang dikelompokkan ke dalam layer layer yang saling terhubung melalui sinapsis. Struktur ini memiliki beberapa masukan dan keluaran, yang dilatih untuk bereaksi dengan pembobotan nilai tertentu. Jaringan syaraf dituntut untuk belajar berperilaku dan seseorang seharusnya bertanggung jawab atas pengajaran atau pelatihan berdasarkan pengetahuan sebelumnya (Gaxiola, Melin, Valdez, & Castillo, 2014). Arsitektur sistem ini dibentuk dari pemroses data yang saling terhubung, yang bersifat adaptif karena memiliki algoritma yang berfungsi mengatur pembobotan dari parameter untuk mencapai performansi yang diinginkan. Sehingga, keunggulannya terletak pada

kemampuannya untuk mempelajari pola yang kemudian memodelkannya secara tepat berdasarkan data. Neuron tiruan terdiri atas beberapa bagian seperti *input*, pembobotan, penjumlahan, dan fungsi adaptor. Di bawah ini merupakan struktur dari neuron tiruan dan struktur. Di mana x_i merupakan masukan, w_i merupakan nilai pembobotan, sementara y adalah keluaran. Operasi pada neuron terjadi dengan berupa masukan yang akan mengalami pembobotan yang kemudian dijumlahkan melalui operasi penjumlahan : $\sum x_i w_{ij}$ (Gaxiola et al., 2014). Dengan batas atas dan bawah i yang telah ditentukan. Hasil penjumlahan tersebut kemudian dimasukkan ke dalam fungsi f , atau disebut sebagai fungsi transfer atau fungsi aktivasi, untuk menghasilkan nilai akhir keluaran



Gambar 2. 2 struktur Neuron Tiruan (Gaxiola et al., 2014)

Di bawah ini merupakan persamaan pada neuron tiruan (Aliev & Guirimov, 2014)

$$NET = \sum_{i=1}^n x_i w_{ij} \quad i = 1, 2, 3, \dots, n \quad (2.1)$$

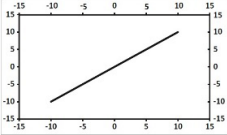
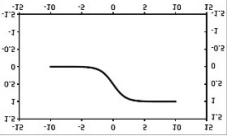
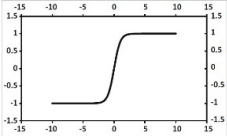
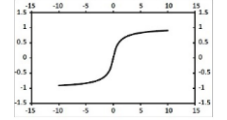
$$y_i = f(\theta + NET) \quad (2.2)$$

Fungsi $f(\cdot)$ merupakan fungsi total masukan tertimbang ke neuron. Untuk fungsi asli dari fungsi aktivasi adalah dalam bentuk fungsi *step* (McCulloch-Pitts neurons).

$$f = \begin{cases} 1, & \text{jika } NET \geq 0 \\ 0 & \text{sebaliknya} \end{cases} \quad (2.3)$$

Di mana NET merupakan total jumlah *weighter input*. Namun, bisa juga fungsi $f(\cdot)$ dalam bentuk selain fungsi *step*, seperti linier, sigmoidal, dll seperti ditunjukkan pada tabel 2.3

Tabel 2. 3 Macam-macam persamaan fungsi aktivasi (Yang et al., 2016)
Fungsi Aktivasi

Linier	$f(NET) = NET$	
Sigmoidal	$f(NET) = \frac{1}{1 + e^{-NET}}$	
Hyperbolic Tangen	$f(NET) = \frac{e^{NET} - e^{-NET}}{e^{NET} + e^{-NET}}$	
Yang lain	$f(NET) = \frac{NET}{1 + NET }$	

Gambar 2. 3

fungsi aktivasi linier

Gambar 2. 4

fungsi aktivasi sigmoidal

Gambar 2. 5

fungsi aktivasi hyperbolic tangen

Gambar 2. 6

fungsi aktivasi yang lain

Pada tahap keluaran, nilai keluaran (z_l) dari semua neuron pada *output layer* diberikan persamaan sebagai berikut (Yang et al., 2016)

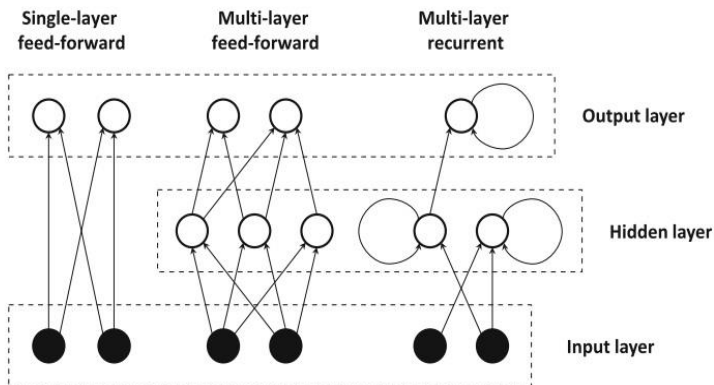
$$z_l = f_l\left(\sum_{i=1}^n y_i w_{ij}\right) \quad l=1,2,3,\dots,t \quad (2.4)$$

Dimana f_l merupakan fungsi aktivasi yang biasanya didefinisikan dalam persamaan fungsi linier. Semua pembobotan awal dipasang dengan nilai acak, yang selanjutnya akan dimodifikasi oleh *delta rule* berdasarkan data pembelajaran.

b. Arsitektur *Neural Network*

Jaringan syaraf tiruan bisa dibuat dengan arsitektur yang beragam, bisa dalam bentuk *feed forward*, *layered*, *recurrent* dan masih banyak lagi. Pada *layered* jaringan syaraf tiruan neuron diorganisasi dalam suatu lapisan. Kumpulan neuron dalam satu lapisan inilah yang kemudian memproses sinyal masukan untuk menjadi nilai keluaran yang kemudian disalurkan ke lampisan berikutnya hingga mencapai keluran akhir. Neuron pada lapisan yang sama bekerja secara parallel dan tidak berhubungan satu sama lain (Aliev & Guirimov, 2014).

Gambar 2.7 berikut ini, mengilustrasikan 3 arsitektur jaringan yaitu single-layer feed forward, multi-layer feed forward dan multi layer recurrent .



Gambar 2. 7 3 macam arsitektur *Artificial Neural Network* (ANN)
(Aliev & Guirimov, 2014)

Pada Jaringan syaraf tiruan *Multilayer* terdapat lapisan di antara lapisan masukan dan keluaran atau yang disebut *hidden layer*. *Hidden layer* biasanya dalam bentuk persamaan non-linier yang juga berfungsi sebagai fungsi aktivasi. Neuron pada *hidden layer* bisa dalam jumlah yang tak terhingga dengan jumlah lapisan yang lebih dari satu.

2.6 Algoritma *Levenberg-Marquardt*

Algoritma *Levenberg-Marquardt* merupakan pengembangan algoritma *backpropagation* standar (Rahmat et al., 2006). Algoritma pelatihan *Levenberg-Marquardt* (LM) menempatkan nilai minimum fungsi multivariate yang dapat diekspresikan sebagai jumlah dari akar fungsi nilai *real* non linear. Teknik ini bekerja dengan mengurangi fungsi performansinya di setiap iterasi (Sharma & K. Venugopalan, 2014).

Algoritma LM dianggap sebagai gabungan dari *steepest descent* dan metode *Gauss-Newton*. Algoritma *steepest descent* yang secara luas dipakai dianggap kurang efisien karena lambatnya untuk mencapai konvergensi. Hal ini dikarenakan perlunya ukuran langkah yang tepat untuk gradien menuju konvergen. Kemudian dengan menggunakan metode *Gauss-Newton*, evaluasi gradient permukaan kesalahan dapat dilakukan menggunakan fungsi turunan orde kedua sehingga menemukan ukuran langkah yang tepat dan konvergen dengan cepat. Algoritma LM sebagai gabungan dari kedua metode tersebut bekerja sebagai berikut: apabila hasil prediksi jauh dari target maka algoritma bekerja sebagai metode *steepest descent*, sedangkan apabila hasil prediksi mendekati target, ia berlaku sebagai metode *Gauss Newton* (Kamel, Mahmoud, Le Bissonnais, & Mahmoud, 2014).

Algoritma LM memberikan pendekatan pada matriks Hessian dengan parameter koefisien kombinasi. Matriks Hessian di sini memberikan perhitungan nilai perubahan bobot pada jaringan.

$$H \approx J^T J + \mu I \quad (2.5)$$

Di mana, μ adalah koefisien kombinasi (selalu bernilai positif), I adalah matriks identitas, dan J adalah matriks Jacobian.

Sehingga persamaan pembaruan bobot untuk algoritma LM yaitu

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \quad (2.6)$$

Ketika μ sangat kecil (mendekati nol) maka pembaruan bobot mendekati algoritma Gauss-Newton. Sedangkan ketika μ sangat besar, maka pembaruan bobot mendekati metode steepest descent. Dan apabila μ sangat besar dapat dianggap sebagai koefisien pembelajaran pada metode *steepest descent*.

$$\alpha = \frac{1}{\mu} \quad (2.7)$$

Proses perhitungan untuk matriks Jacobian pada algoritma LM dapat diatur sesuai dengan perhitungan backpropagation tradisional dalam algoritma orde pertama tetapi untuk setiap pola, dalam algoritma EBP, hanya satu proses backpropagation yang diperlukan, sedangkan dalam algoritma Levenberg-Marquardt proses backpropagation harus diulang. Untuk perhitungan matriks Jacobian. *Input node y* dapat dihitung dalam *forward computation*. Sementara parameter *error* δ diperoleh dalam perhitungan mundur / *Backward computation* (Wilamowski, 2011).

Proses perhitungan untuk matriks Jacobian pada algoritma LM dapat diatur sesuai dengan perhitungan backpropagation tradisional dalam algoritma orde pertama tetapi untuk setiap pola, dalam algoritma EBP, hanya satu proses backpropagation yang diperlukan, sedangkan dalam algoritma Levenberg-Marquardt proses backpropagation harus diulang. Untuk perhitungan matriks Jacobian. *Input node y* dapat dihitung dalam *forward computation*. Sementara parameter *error* δ diperoleh dalam perhitungan mundur / *Backward computation* (Wilamowski, 2011).

Proses pelatihan algoritma LM adalah sebagai berikut.

1. Evaluasi *sum square error* (SSE) menggunakan bobot awal. Penurunan persamaan algoritma sebagai berikut.

$$E(x, w) = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M e^2_{p,m} \quad (2.8)$$

dan fungsi *error* pelatihan diberikan oleh

$$e_{p,m} = d_{p,m} - o_{p,m} \quad (2.9)$$

dimana, p indeks pola, m indeks *output*, I dan j mengindikasikan

bobot, x vector *input*, w bobot, d adalah target dan o adalah *output* sebenarnya dari jaringan.

2. Lakukan pembaruan bobot berdasarkan algoritma LM
3. Evaluasi SSE dengan bobot baru
4. Jika SSE baru meningkat, maka ambil langkah (mengatur ulang vector bobot ke nilai tertentu) dan meningkatkan nilai μ dengan factor 10 atau factor lainnya. Kemudian lanjutkan ke langkah b dan perbarui lagi.
5. Jika SSE baru menurun, maka lakukan langkah (menjaga vector bobot baru sebagai yang sekarang) dan menurunkan nilai μ dengan faktor 10 atau yang lainnya.
6. Lanjutkan ke langkah b dengan bobot baru dan SSE baru yang lebih kecil dari nilai yang ditargetkan atau kondisi yang diinginkan sudah terpenuhi z

2.6 Validasi Prediktor

Suatu proses/metode dapat memberikan hasil yang konsisten sesuai dengan spesifikasi yang telah ditetapkan. Performansi prediktor ditentukan oleh besarnya presentase akurasi dan *Mean Square Error*. Besarnya presentase akurasi dinyatakan dalam persamaan berikut (Putri, 2015):

$$\% \text{ Keakuratan} = \left(\frac{\text{Jumlah prediksi benar}}{\text{Total prediksi}} \times 100\% \right) \quad (2.10)$$

Selain itu dicari pula nilai *Mean Square Error* (MSE) atau *Error* untuk mengukur performansi jaringan syaraf tiruan :

$$E = \sum_p \sum_i \frac{1}{2p} e_i^2 \quad (2.11)$$

Dimana E adalah fungsi *error* rata-rata kuadrat, e adalah selisih antara keluaran yang diinginkan dan yang dihitung, p adalah jumlah pola dalam data pelatihan, dan i adalah jumlah keluaran yang diinginkan (Santhosh, Gopika, Ghosh, & Fernandes, 2018).

Secara sederhana MSE dihitung sebagai rata-rata jumlah kuadrat dari perkiraan kesalahan

$$MSE = \frac{\sum (Y_t - T_t)^2}{n} \quad (2.12)$$

Di mana, t adalah periode waktu, n adalah jumlah yang diprediksi,

Y adalah nilai aktual yang diprediksi, dan T adalah nilai target yang dituju

Nilai MSE mengukur jumlah dispersi dari kesalahan. Semakin kecil nilai MSE semakin baik. Semakin kecil nilainya maka akan semakin akurat model peramalannya dan begitu sebaliknya. Validasi prediktor lainnya yang sering digunakan adalah *Root Mean Square Error*. Akar kuadrat dari hasil MSE dalam standar deviasi dari kesalahan atau Standar *Error* (Se) atau dapat disebut dengan *Root Mean Square Error* (RMSE). Sebagian besar peramalan mengikuti asumsi bahwa kesalahan mengikuti distribusi normal dengan rata-rata nol (yang akan diukur oleh bias) dan standar deviasi tertentu yang diperkirakan oleh Se (Klimberg, Sillup, Boyle, & Tavva, 2010).

Parameter *error* lainnya yang sering digunakan yaitu MAPE (*Mean Absolute Percentage Error*). *Mean Absolute Percentage Error* (MAPE) yang memberikan persentase akurasi hasil prediksi. Untuk menentukan nilai parameter MAPE (*Mean Absolute Percentage Error*) digunakan formula di bawah ini:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}}{x_i} \right| \quad (2.13)$$

Di mana x_i merupakan nilai aktual sementara \hat{x} merupakan nilai prediksi. Sehingga nilai parameter di atas merepresentasikan nilai peramalan. Apabila nilai parameter di atas semakin kecil menunjukkan bahwa hasil prediksi semakin baik dan akurat.

2.7 Implementasi *Machine Learning* dalam sistem prediksi

Pada penelitian ini metode yang digunakan adalah metode Artificial Neural Network. Metode tersebut merupakan salah satu metode pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen terpenting dari metode ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu (Andrian & Ningsih, 2014a). Cara kerja *Neural Network* seperti cara kerja manusia, yaitu belajar

melalui contoh. Lapisan-lapisan penyusun *Neural Network* dibagi menjadi 3, yaitu lapisan *input (input layer)*, lapisan tersembunyi (*hidden layer*), dan lapisan *output (ouput layer)*(Sutojo et al., 2011).

Metode *Arificial Neural Network* ada untuk memenuhi tujuan machine learning dalam mengembangkan sebuah sistem atau mesin cerdas yang dapat belajar sendiri tanpa harus diprogram oleh manusia secara berulang kali. Sehingga dengan adanya machine learning akan akan membuat proses validasi data menjadi sederhana dan tanpa perlu campur tangan manusia. Hal tersebut disebabkan karena ciri khas yang dimiliki oleh *machine learning* yaitu kecerdasan memahami pola data.

Sebelum adanya perkembangan metode *machine learning* yang masuk dalam perkembangan metode *Artificial Intelligent*, terdapat beberapa proses prediksi yang banyak digunakan salah satu contohnya adalah menggunakan pendekatan statistika data *time series* menggunakan model regresi atau ARIMA. Berikut pada tabel 2.4 dijelaskan perbandingan penggunaan metode ARIMA dan metode dengan pendekatan *Artificial Intelligent*.

Tabel 2. 4 Perbandingan pendekatan ARIMA dan pendekatan AI dalam sistem prediksi (Aisjah & Arifin, 2011) (Voyant et al., 2017)

Pendekatan Statistika Data <i>Time Series</i> Menggunakan Model Regresi atau ARIMA	Dengan pendekatan <i>Artificial Intelligence (AI)</i>
Membutuhkan data historis tidak sedikit dengan akurasi prediksi rendah	Mampu bekerja dengan data yang lebih sedikit dengan kinerja yang sama atau bahkan memiliki kinerja yang lebih baik
Lebih sesuai dalam <i>long-term forecasting</i> sehingga tidak dapat bekerja pada data non-linier seperti kondisi cuaca saat ini yang fluktuatif dan berubah-ubah	Mampu memprediksi (cuaca) baik pada beberapa jam atau beberapa hari kedepan

Tabel 2.4 di atas adalah tabel yang menunjukkan bahwa metode ARIMA membutuhkan data historis tidak sedikit dengan akurasi prediksi rendah. Selain itu, penggunaan metode ini lebih sesuai dalam *long-term forecasting* dan tidak dapat bekerja pada data non-linier seperti kondisi cuaca saat ini yang fluktuatif dan berubah-ubah. Berikut merupakan tabel perbedaan metode prediksi dengan pendekatan ARIMA dan pendekatan *Artificial Intelligent*.

Penelitian ini mengacu pada penelitian-penelitian sebelumnya yang telah dilakukan. Penelitian-penelitian tersebut dijadikan sumber referensi dan pembandingan dari hasil penelitian yang telah dilakukan. Tabel 2.5 dan tabel 2.6 adalah tabel yang menunjukkan perbandingan hasil dari penelitian-penelitian sebelumnya.

Tabel 2. 5 Hasil penelitian sebelumnya

Pengarang	Judul	Hasil
Made Dwi Udayana Putra, G. K. Gandhiadi, Luh Putu Ida Harini	Implementasi <i>backpropagation neural network</i> Dalam prakiraan cuaca di daerah bali selatan	<p>1. Bertambahnya jumlah neuron pada lapisan tersembunyi tidak serta merta meningkatkan atau menurunkan nilai akurasi prakiraan cuaca yang diperoleh dengan metode jaringan syaraf tiruan dengan algoritma pembelajaran <i>backpropagation</i>.</p> <p>2. Hasil prakiraan cuaca di daerah Bali Selatan menggunakan metode jaringan syaraf tiruan dengan algoritma pembelajaran <i>backpropagation</i> diperoleh tingkat akurasi prakiraan cuaca terbaik sebesar 51,6129% pada model jaringan dengan tiga buah neuron pada lapisan tersembunyi.</p>
Nola Ritha, Retantyo Wardoyo	Implementasi <i>Neural Fuzzy Inference System</i> dan Algoritma Pelatihan <i>Levenberg-Marquardt</i> untuk Prediksi Curah Hujan	<p>NFIS LM menghasilkan nilai MSE sebesar 0.0262050. Jika dibandingkan dengan model NN-Backpropagation, model NFIS-LM menunjukkan akurasi yang lebih rendah. Hal ini ditunjukkan dari MSE yang dihasilkan dimana model NN-Backpropagation menghasilkan MSE sebesar 0.0167990.</p>

Tabel 2. 6 Lanjutan hasil penelitian sebelumnya

Nola Ritha1, Martaleli Bettiza, Ariel
Dufan

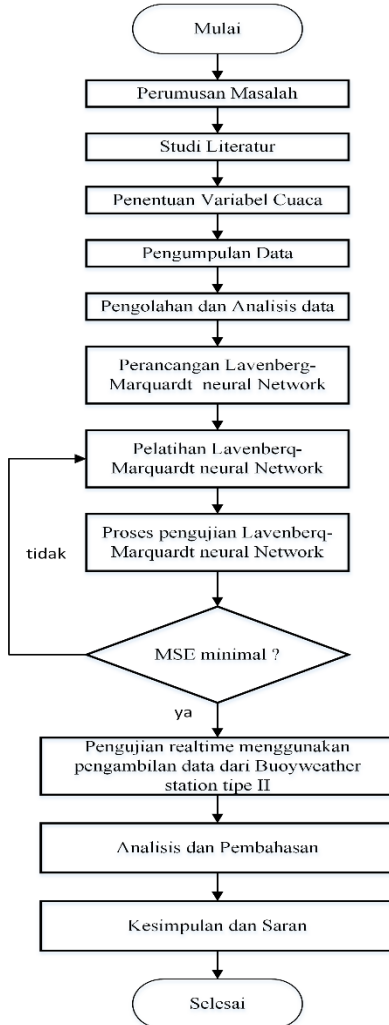
Prediksi Curah Hujan dengan
Menggunakan Algoritma Levenberg-
Marquardt dan Backpropagation

Hasil penelitian menunjukkan pemodelan dengan Algoritma Levenberg Marquardt memberikan hasil terbaik pada pemodelan data dengan jumlah *neuron hidden layer* 10, *Epoch* 100, dengan nilai mse sebesar 0.0776. Sedangkan Algoritma *Backpropagation* jumlah *neuron hidden layer* 4, *Epoch* 1.000 dengan nilai mse sebesar 0.07876. Penelitian ini menunjukkan bahwa perbandingan hasil prediksi curah hujan dengan menggunakan Algoritma *Levenberg Marquardt* menghasilkan prediksi yang lebih baik dibanding dengan Algoritma *Backpropagation*.

Halaman sengaja dikosongkan

BAB III METODOLOGI PENELITIAN

Metode penelitian ditunjukkan dalam bentuk diagram alir pada Gambar 3.1 di bawah ini.



Gambar 3. 1 Diagram Alir Penelitian

Gambar 3.1 diatas merupakan diagram alir penelitian yang menampilkan alur dari penelitian. Adapun penejalasan tiap tahapan penelitian yang akan dijelaskan secara rinci pada sub bab dibawah ini

3.1 Perumusan Masalah

Langkah pertama dalam pengerjaan tugas akhir ini adalah mengidentifikasi masalah yang akan diselesaikan dalam penelitian tugas akhir. Rumusan masalah didapatkan dari kajian penelitian sebelumnya yang telah dilakukan.

Penelitian tentang variasi target *error* pada prediktor JST telah dilakukan oleh Yudhi Andrian mengenai Analisis Variasi Jumlah *Input* dan *Hidden layer* Pada Prediksi Temperatur Kota Medan Menggunakan *Backpropagation Neural Network*. Penelitian tersebut memvariasikan target *error*, jumlah *input* dan *hiddden layer* untuk menentukan performa terbaik JST. Hasil yang didapatkan dari penelitian tersebut adalah dengan berbagai variasi target *error* menunjukkan korelasi antara besarnya target *error* dengan jumlah iterasi. Semakin besar target *error* semakin sedikit iterasi. Penelitian tersebut juga menunjukkan bahwa variasi target *error* berpengaruh terhadap performansi JST. Semakin kecil target *error* maka nilai keakurasiannya cenderung semakin baik (Andrian & Ningsih, 2014b).

Berdasarkan hasil penelitian tersebut di atas, maka pada penelitian ini dilakukan strategi dengan merubah dua parameter yaitu target *error* dan *hidden node*. Hasil performansi akibat perubahan yang dilakukan terhadap dua parameter tersebut akan digunakan dalam menentukan algoritma JST-LM terbaik.

3.2 Studi Literatur

Studi literatur merupakan suatu kegiatan yang dilakukan untuk memperdalam pengetahuan terhadap prinsip dasar dari penelitian yang dilakukan. Beberapa referensi yang relevan dengan tema penelitian yaitu hasil penelitian terdahulu dalam tema yang sama, teorema tentang JST yang diperoleh dari jurnal dan handbook serta simulink Matlab. Beberapa penelitian sebelumnya

yaitu tentang prediksi cuaca maritim melalui pendekatan kecerdasan buatan. Studi literatur ini dilakukan untuk meningkatkan pemahaman mengenai topik permasalahan. Informasi yang didapatkan oleh peneliti digunakan sebagai acuan dalam merancang suatu metode untuk menyelesaikan permasalahan dalam penelitian. Selain informasi yang didapatkan hal yang dapat diperoleh dari studi literatur adalah sebuah pembandingan dan pendukung dari penelitian yang akan dilakukan.

3.3 Penentuan Variabel Cuaca

Tahap ini adalah tahap yang dilakukan untuk menentukan variabel *inputan* dan keluaran prediktor. Variabel keluaran prediktor yang digunakan pada penelitian ini adalah arah angin, kecepatan angin dan curah hujan. Penelitian ini juga menggunakan variabel keluaran prediktor sebagai variabel *input* untuk keluaran prediktor yang berbeda. Sebagai salah satu contoh yaitu variabel curah hujan dan arah angin yang dipengaruhi oleh variabel kecepatan angin (Hardi, 2005). Variabel *input* lainnya yang digunakan adalah variabel suhu dan kelembaban.

Penentuan variabel ini juga berdasarkan sinkronisasi atas variabel yang telah ditetapkan penulis dengan ketersediaan data yang dimiliki BMKG Maritim II Perak Surabaya. Berdasarkan sinkronisasi yang telah dilakukan, didapatkan beberapa variabel yang akan digunakan pada penelitian ini yaitu curah hujan, kelembaban, suhu, arah angin dan kecepatan angin.

3.4 Pengumpulan Data

Pengumpulan data dilakukan melalui dua cara yaitu pengambilan data dari BMKG dan pengambilan data secara *realtime*. Tahap ini penulis akan mengumpulkan data pengukuran yang diperoleh dari BMKG Maritim II Perak Surabaya. Data yang dikumpulkan adalah data pengukuran pada variabel yang telah ditentukan dengan rentang tahun 2017 hingga 2019. Data 2019 yang didapatkan hingga bulan Februari. Data yang digunakan dalam penelitian ini data tiap jam. Sehingga total data tiap variabel yang digunakan dalam penelitian ini adalah 18.936. Data ini

nantinya akan menjadi data *training* yang digunakan sebagai pengembangan dan pengujian prediksi pada mesin prediktor yang telah dirancang. Pengambilan data kedua dilakukan di wilayah Selat Madura.

Pengambilan data kedua tersebut digunakan sebagai pengujian *realtime* yang dilakukan di wilayah Selat Madura. Pengambilan data ini dilakukan secara langsung menggunakan wahana apung, *buoyweather* tipe 2, yang telah dirancang untuk merekam data fisis dari cuaca maritim selama total 3 hari.

3.5 Pengolahan dan Analisis Data

Pengolahan data merupakan pra-pemrosesan data (Pratama, Permanasari, Ardiyanto, & Indrayani, 2017) yang dilakukan melalui uji missing, uji outlier, uji korelasi dan normalisasi. Uji *missing value*, uji *outlier* digunakan untuk memperbaiki data-data yang cacat. Penjelasan tentang uji tersebut dituliskan pada sub bab di bawah ini.

Tahap analisis data dilakukan untuk menentukan variabel *input* prediktor berdasarkan besarnya korelasi. Besarnya korelasi ditentukan berdasarkan persamaan 3.2.

3.5.1 Uji Missing Value

Missing value merupakan peristiwa ketidaktersediaan informasi untuk sebuah subyek (kasus). *Missing* data terjadi akibat informasi suatu objek tidak tersedia, sulit dicari, atau memang tidak ada informasi, bagian yang hilang pada kumpulan data dapat dipengaruhi oleh pembacaan sensor, transmisi data dari pengambilan data ke pengolahan data, penyimpanan data base hingga kondisi cuaca ekstrim yang menghambat pengambilan data (RencherJohn, 1998).

Data *missing* dapat diisi dengan nilai *mean* dari masing-masing variabel dengan menggunakan metode imputasi *listwise*, *pairwise* dan *expected maximization*. Pada tabel 3.1 dijelaskan mengenai keputusan yang dapat dilakukan pada data yang *missing*. Tabel tersebut menjelaskan menjelaskan beberapa keputusan yang dapat diambil pada data yang *missing* sesuai dengan presentase data *missing* dari data yang ada.

Tabel 3. 1 Ukuran Presentase Missing Value dan Keputusan

Presentase Missing Value	Keputusan
<10%	<i>Missing value</i> pada variabel dapat diabaikan Kandidat variabel dapat dihapus jika
10%-50%	a. Terdapat variabel yang berkorelasi dengan variabel yang akan dihapus b. Dengan dihapusnya variabel menurunkan persentase <i>missing value</i> secara signifikan
>50%	Case (pengamatan ke-i) dihapus
<i>Missing value</i> pada variabel dependen sebaiknya dihapus untuk menghindari pengaruh buatan dari variabel independent	

Pada penelitian ini data yang digunakan adalah data perjam. Berikut adalah rician jumlah data perjam yang diperoleh dari BMKG II Perak.

Tabel 3. 2 Jumlah Data perjam yang dioeroleh dari BMKG Perak

Januari 2017-Desember 2017	8760
Januari 2018-Desember 2018	8760
Januari 2019-Februari 2019	1416
Total	18.936

Data yang diberikan oleh BMKG Perak II memiliki rentang waktu tiap 10 menit sehingga penulis mengolah nya untuk rentang waktu tiap satu jam. Sehingga dapat disimpulkan bahwa data yang didapatkan dari BMKG Perak II Surabaya tidak terdapat missing sehingga tidak perlu dilakukan perlakuan missing data. Hal ini menunjukkan bahwa pengambilan data cuaca maritim oleh BMKG berjalan secara berkala.

3.5.2 Uji *Outlier*

Data *oulier* adalah data yang secara nyata berbeda dengan data-data yang lainnya, atau dengan kata lain memiliki nilai yang jauh lebih besar ataupun jauh lebih kecil dibanding data lainnya. Keberadaan data outlier atau data yang sangat ekstrem, pada

banyak kasus akan mengganggu keseluruhan data yang dapat menyebabkan biasanya kesimpulan yang diambil. Deteksi *outlier* sudah menjadi perhatian para ahli statistik dan ilmuwan lainnya, lebih dari satu abad, penulis telah mengklaim bahwa peneliti biasanya dapat mengharapkan hingga 10% dari pengamatan untuk memiliki kesalahan dalam pengukuran atau perekaman (RencherJohn, 1998).

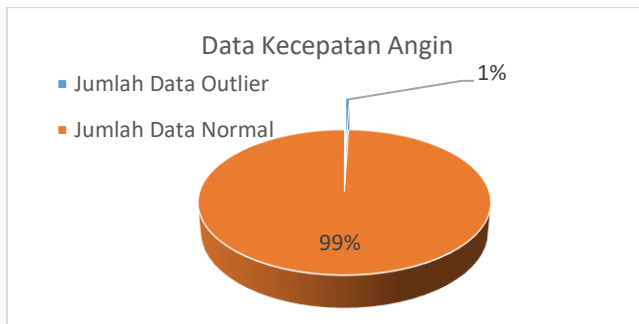
Mendeteksi *Outlier Univariate* dilakukan dengan menghitung nilai normal standar pada masing-masing observasi dengan menggunakan rumus :

$$Z_i = \frac{x_i - \mu}{\sigma} \quad (3.1)$$

Jika nilai z_i lebih besar dari angka +3 atau lebih kecil dari angka - 3, maka observasi ke- i merupakan *outlier*. Berikut merupakan hasil pengolahan data outlier tiap variabel

a. Data Kecepatan Angin

Berikut adalah presentase data outlier dari data kecepatan angin

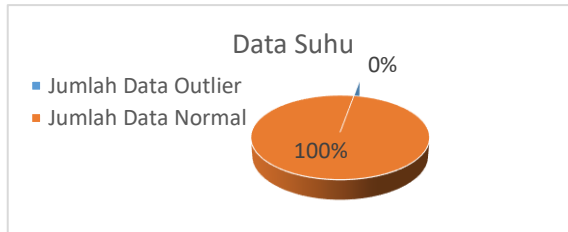


Gambar 3. 2 Presentase Data Outlier pada Data Kecepatan Angin

Jumlah data outlier pada data kecepatan angin adalah sebesar 96 dari 18.936 data atau sebesar 2% dari 100%. Sehingga data kecepatan angin masih dikategorikan pada data yang masih dapat digunakan.

b. Data Suhu

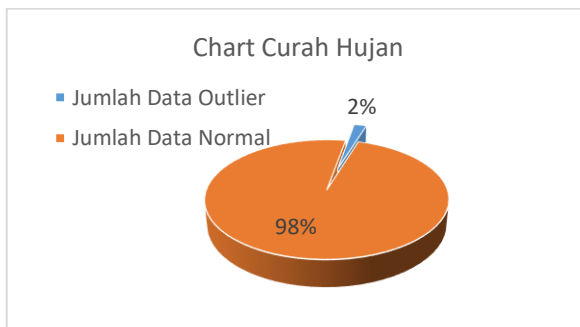
Berikut adalah presentase outlier dari data suhu



Gambar 3. 3 Presentase Data Outlier pada Data Suhu

Dari data diatas menggambarkan bahwa otlier pada data suhu 0%. Jumlah data outlier pada data suhu sebesar 1 dari jumlah data sebesar 18.936. Presentase outlier yaitu 0,0005 atau mendekati 0% dari 100%.

- c. Data Kelembaban, Tekanan Udara dan Arah Angin
Ketiga variabel ini tidak memiliki data outlier. Sehingga presentase data outlier pada ketiga variabel ini adalah 0% dari 100%.
- d. Data Curah Hujan
Berikut merupakan presentase data outlier dari Curah Hujan



Gambar 3. 4 Presentase Data Outlier pada Data Curah Hujan

Data Curah Hujan memiliki jumlah data outlier sebanyak 378. Jumlah tersebut tergolong banyak namun data tersebut masih dikategorikan dalam data yang masih dapat digunakan karena presentase data outlier hanya mencapai 2% dari data keseluruhan.

Hasil pengujian data *outlier* pada keenam variabel yang digunakan menunjukkan bahwa data masih memenuhi kaidah data ekstrim sehingga dapat digunakan sebagai variabel yang akan diprediksi maupun sebagai *input* data pada sistem prediktor. Secara keseluruhan hasil dari pengujian outlier dapat dilihat pada tabel 3.2 berikut. Dari hasil tabel dapat dilihat perbandingan jumlah data outlier dengan jumlah data normal tanpa *outlier*.

Tabel 3. 3 Jumlah Data Outlier

	Total Jumlah Data	Jumlah Data Outlier	Jumlah Data Normal
Wind Speed (m/s)		96	18840
Suhu (C) Relatif Humidity (%)		1	18935
		0	18936
Tekanan Udara (mbar)	18936	0	18936
Wind Direction (°)		0	18936
Curah Hujan (mm)		378	18558

3.5.3 Uji Korelasi

Uji korelasi pada penelitian ini dilakukan untuk menentukan variabel-variabel yang saling berpengaruh satu sama lain. Nilai korelasi ini digunakan sebagai ukuran dalam menentukan variabel masukan dari suatu prediktor. Pendekatan ini disebut dengan *space partitioning*. Menurut Guo, pendekatan ini dapat meningkatkan performa dari prediktor, khususnya jaringan saraf tiruan (Guo et al., 2013). Pendekatan ini dilakukan melalui masukan yang berkorelasi

lemah ataupun tak berkorelasi dihilangkan dari masukan semasa proses pelatihan dan dipilih masukan yang memiliki korelasi yang bernilai kuat.

Metode yang dapat digunakan dalam uji korelasi salah satunya adalah menggunakan uji korelasi pearson. Koefisien Pearson 'r' didefinisikan sebagai rasio kovarian dari dua variabel tertentu yang merepresentasikan himpunan data variabel-variabel tersebut. Di mana 'r' diformulasikan dalam formula berikut.

$$r = \frac{C_{xy}}{\sqrt{C_{xx} C_{yy}}} = \frac{C_{xy}}{\sigma_x \sigma_y} \quad (3.2)$$

Dengan x sebagai variabel 1 dengan sejumlah data N $[x_1, x_2, x_3, x_4, x_5, \dots, x_N]$ dan y adalah variabel 2 dengan sejumlah data N $[y_1, y_2, y_3, y_4, y_5, \dots, y_N]$. Penjelasan lebih rinci masing-masing variabel dijabarkan pada formula di bawah ini.

$$\bar{x} = \frac{1}{N} \sum_i x_i$$

$$\bar{y} = \frac{1}{N} \sum_i y_i$$

$$C_{xy} = \frac{1}{N-1} \sum_i (x_i - \bar{x})(y_i - \bar{y}) \quad (3.3)$$

$$C_{xx} = \sigma_x^2 = \frac{1}{N-1} \sum_i (x_i - \bar{x})^2$$

$$C_{yy} = \sigma_y^2 = \frac{1}{N-1} \sum_i (y_i - \bar{y})^2$$

Dimana

\bar{x} = rata-rata data variabel x

\bar{y} = rata-rata data variabel y

C_{xy} = kovarian variabel x dan y

C_{xx} = kovarian variabel x

C_{yy} = kovarian variabel y

Nilai korelasi pearson didefinisikan dalam interval

$$-1 \leq r \leq 1$$

Nilai tersebut menunjukkan bahwa hubungan linier antara dua variabel. Jika r bernilai positif maka menunjukkan korelasi dua variabel adalah linier positif atau naik. Jika r bernilai -1 maka korelasi antara dua variabel adalah linier negatif atau menurun. Sementara, jika r bernilai 0 maka menunjukkan bahwa kedua variabel tidak berhubungan satu sama lain. Semakin bernilai -1 atau 1 maka korelasi antara dua variabel semakin kuat.

Pendeskripsikan besar nilai koefisien Pearson, dinyatakan melalui deskripsi dari Evan (Evans, 1996) yang mendeskripsikan masing-masing interval nilai koefisien Pearson pada tabel berikut.

Tabel 3. 4 Klasifikasi ukuran korelasi berdasarkan nilai koefisien r .

Nilai koefisien r	Ukuran korelasi
0.00-0.19	Sangat Lemah
0.20-0.39	Lemah
0.40-0.59	Moderat
0.60-0.79	Kuat
0.80-1.00	Sangat Kuat

Koefisien korelasi memiliki nilai yang besar atau kecil. besar kecilnya koefisien korelasi dapat berpedoman pada Tabel 3.4 untuk memberikan interpretasi terhadap koefisien korelasi. Parameter korelasi tidak hanya ditentukan oleh nilai r saja. Namun parameter lain yang digunakan adalah signifikansi. Nilai signifikansi yang digunakan adalah maksimal $0,05$ atau nilai kepercayaan 95% .

Hasil dari uji korelasi dari masing-masing variabel yang diprediksi adalah sebagai berikut. Hasil tersebut dipilih berdasarkan nilai koefisien korelasi Pearson dari beberapa variabel masukan prediksi yang paling besar dan nilai signifikansi data

a. Uji korelasi pada variabel *Wind Speed* (Kecepatan Angin)

Tabel 3.5 menunjukkan hasil dari pengujian korelasi pada variabel *Wind Speed* (Kecepatan Angin). Pada tabel 3.5 dapat dilihat bahwa nilai korelasi paling rendah yaitu sebesar $0,0314$. Hal

ini menunjukkan bahwa kecepatan angin berkorelasi sangat lemah dengan variabel *input* tekanan udara. Hasil uji korelasi tersebut juga menunjukkan bahwa nilai korelasi paling tinggi pada variabel kecepatan angin yaitu sebesar 0.4112 dengan variabel *input* suhu. Hal ini menunjukkan bahwa kecepatan angin memiliki korelasi moderat dengan variabel suhu.

Tabel 3. 5 Hasil Uji Korelasi Terhadap *Wind Speed* (Kecepatan Angin)

Parameter	Variabel <i>Input</i>				
	Suhu	Relatif <i>Humidity</i> (%)	Tekanan Udara (mbar)	<i>Wind</i> <i>Direction</i> (<i>O</i>)	Curah Hujan (mm)
Koefisien Korelasi	0.4112	-0.1874	0.0314	-0.3456	-0.118
Deskripsi	Moderat positif	Sangat lemah negatif	Sangat lemah	Lemah negatif	Sanagt lemah negatif
Jumlah data	18816	18816	18816	18816	18816

Sehingga dapat disimpulkan bahwa dari pengujian korelasi terhadap kecepatan angin menggunakan lima variabel *input* didapatkan nilai korelasi yang tertinggi yaitu dengan variabel *input* suhu dan arah angin. Selain menggunakan nilai koefisien *r*, dalam uji korelasi dapat juga dilihat dari nilai signifikansi dari data variabel *output* dan variabel *input*.

Berdasarkan referensi, untuk prediksi kedepan diperlukan adanya pengumpulan, pemakaian data historis untuk menemukan keteraturan pola data atau hubungan dalam data berukuran besar (Mujiasih, 2011). Dari hasil pengujian korelasi menggunakan korelasi *pearson*, didapatkan variabel *input* suhu dan arah angin untuk prediksi kecepatan angin. Maka untuk prediksi kecepatan angin 12 jam kedepan dibutuhkan data historis dari kedua *input* tersebut. Berikut rincian variabel *ouput* dan variabel *input* yang digunakan dalam sistem prediksi kecepatan angin

- Variabel *output*
 $WS(t+12)$ = Prediksi kecepatan angin 12 jam kemudian
- Variabel *input*

Tabel 3. 6 Variabel historis *Input* untuk prediksi WS(t+12)

T(t-60)	Suhu 60 jam sebelumnya	WD(t-60)	Arah angin 60 jam sebelumnya
T(t-48)	Suhu 48 jam sebelumnya	WD(t-48)	Arah angin 48 jam sebelumnya
T(T-36)	Suhu 36 jam sebelumnya	WD(T-36)	Arah angin 36 jam sebelumnya
T(t-24)	Suhu 24 jam sebelumnya	WD(t-24)	Arah angin 24 jam sebelumnya
T(t-12)	Suhu 12 jam sebelumnya	WD(t-12)	Arah angin 12 jam sebelumnya
T(t)	Suhu saat ini	WD(t)	Arah angin saat ini

Dari klasifikasi variabel historis *input* pada tabel 3.6, maka berikut adalah hubungan variabel output dengan data historis *input* berdasarkan nilai signifikansi dan nilai koefisien korelasi.

Tabel 3. 7 Hubungan Variabel *output* WS(t+12) dengan variabel historis *input*

	Signif ikansi	Nilai "r"		Signifi kansi	Nilai "r"
T(t-60)	0.000	0.055	WD(t-60)	0.000	0.044
T(t-48)	0.000	0.028	WD(t-48)	0.000	0.104
T(T-36)	0.000	0.009	WD(T-36)	0.111	0.053
T(t-24)	0.000	0.135	WD(t-24)	0.000	0.056
T(t-12)	0.000	0.130	WD(t-12)	0.000	0.146
T(t)	0.000	0.253	WD(t)	0.000	0.025

Berdasarkan tabel diatas dapat dilihat bahwa nilai koefisien korelasi tertinggi antara variabel *output* WS(t+12) adalah dengan variabel *input* T(t-24), T(t) dan WD(t-12). Dilihat dari nilai signifikansi juga menunjukkan bahwa variabel output dan variabel *input* masih memiliki hubungan yang signifikan karena nilai signifikansi kurang dari 0.05

b. Uji Korelasi pada Variabel *Wind Direction* (Arah Angin)

Tabel 3.8 menunjukkan hasil dari pengujian korelasi pada variabel *Wind Direction* (Arah Angin). Pada tabel 3.8 dapat dilihat bahwa nilai korelasi paling rendah yaitu sebesar 0,013. Hal ini menunjukkan bahwa arah angin berkorelasi sangat lemah dengan variabel *input* tekanan udara. Hasil uji korelasi tersebut juga menunjukkan bahwa nilai korelasi paling tinggi pada variabel arah angin yaitu sebesar 0.3456 dengan variabel *input Wind Speed* (kecepatan angin). Hal tersebut menunjukkan bahwa variabel kecepatan angin memiliki hubungan korelasi yang lebih baik dibandingkan dengan variabel tekanan. Sehingga dapat disimpulkan bahwa dari pengujian korelasi terhadap arah angin menggunakan lima variabel *input* didapatkan nilai korelasi yang tertinggi yaitu dengan variabel *input* suhu dan *Wind Speed*.

Tabel 3. 8 Hasil Uji Korelasi Terhadap Variabel *Wind Direction* (Arah Angin)

Parameter	Variabel <i>Input</i>				
	Suhu	<i>Relatif Humidity</i> (%)	Tekanan Udara (mbar)	<i>Wind Speed</i> (m/s)	Curah Hujan (mm)
Koefisien Korelasi	0.302	0.132	0.013	0.345	0.125
Deskripsi	Lemah positif	Sangat lemah	Sangat lemah	Lemah positif	Sangat lemah
Jumlah data	18816	18816	18816	18816	18816

Dari hasil pengujian korelasi menggunakan korelasi *pearson*, didapatkan variabel *input* suhu dan kecepatan angin untuk prediksi arah angin. Maka untuk prediksi arah angin 12 jam kedepan dibutuhkan data historis dari kedua *input* tersebut. Berikut rincian variabel output dan variabel *input* yang digunakan dalam sistem prediksi arah angin

- Variabel *output*
 $WD(t+12) = \text{Prediksi arah angin 12 jam kemudian}$
- Variabel *input*

Tabel 3. 9 Variabel historis *Input* untuk prediksi WD(t+12)

T(t-60)	Suhu 60 jam sebelumnya	WS(t-60)	Kecepatan angin 60 jam sebelumnya
T(t-48)	Suhu 48 jam sebelumnya	WS(t-48)	Kecepatan angin 48 jam sebelumnya
T(T-36)	Suhu 36 jam sebelumnya	WS(T-36)	Kecepatan angin 36 jam sebelumnya
T(t-24)	Suhu 24 jam sebelumnya	WS(t-24)	Kecepatan angin 24 jam sebelumnya
T(t-12)	Suhu 12 jam sebelumnya	WS(t-12)	Kecepatan angin 12 jam sebelumnya
T(t)	Suhu saat ini	WS(t)	Kecepatan angin saat ini

Dari klasifikasi variabel historis *input* pada tabel 3.9, maka pada tabel 3.10 berikut adalah hubungan variabel output dengan data historis *input* berdasarkan nilai signifikansi dan nilai koefisien korelasi. Dari tabel 3.10 dapat dilihat bahwa hubungan korelasi paling besar yaitu pada variabel *input* T(t-12), T(t), WS(t-12) dan WS(t). Maka keempat variabel *input* tersebut yang akan digunakan dalam sistem prediksi. Selain itu jika dilihat dari nilai signifikansi, keempat variabel *input* tersebut memiliki nilai signifikansi 0.000 terhadap variabel *output* WD(t+12), sehingga hubungan variabel *output* dan *input* dapat dikatakan signifikan karena nilai signifikansi kurang dari 0.05

Tabel 3. 10 Hubungan Variabel *output* WD(t+12) dengan variabel historis *input*

	Signifika nsi	Nilai “r”		Signifi kansi	Nilai “r”
T(t-60)	0.000	0.037	WS(t-60)	0.000	0.062
T(t-48)	0.000	0.006	WS(t-48)	0.000	0.039
T(T-36)	0.000	0.088	WS(T-36)	0.000	0.112
T(t-24)	0.000	0.032	WS(t-24)	0.000	0.026
T(t-12)	0.000	0.140	WS(t-12)	0.000	0.169
T(t)	0.000	0.145	WS(t)	0.000	0.112

c. Uji Korelasi pada Variabel Curah Hujan

Berikut adalah hasil dari pengujian korelasi terhadap curah hujan. Pada tabel 3.11 menunjukkan hasil dari pengujian korelasi pada variabel Curah Hujan. Pada tabel 3.11 dapat dilihat bahwa nilai korelasi paling rendah yaitu sebesar 0,023. Hal ini menunjukkan bahwa curah hujan berkorelasi sangat lemah dengan variabel *input* tekanan udara. Hasil uji korelasi tersebut juga menunjukkan bahwa nilai korelasi paling tinggi yaitu sebesar 0.264 dan 0.204 dengan variabel *input* suhu dan kelembapan. Selain itu nilai koefisien korelasi tinggi juga ada apada variabel kecepatan angin. Sehingga dapat disimpulkan bahwa dari pengujian korelasi terhadap curah hujan menggunakan lima variabel *input* didapatkan nilai korelasi yang tertinggi yaitu dengan variabel *input* suhu, kelembapan dan kecepatan angin.

Tabel 3. 11 Hasil Uji Korelasi Terhadap Variabel Curah Hujan

Parameter	Variabel <i>Input</i>				
	Suhu	<i>Wind Direction</i> (<i>O</i>)	Tekanan Udara (mbar)	<i>Wind Speed</i> (<i>m/s</i>)	<i>Relatif Humidity</i> (%)
Koefisien Korelasi	0.264	0.118	0.023	0.1257	0.2046
Deskripsi	Lemah positif	Sangat lemah	Sangat lemah	Sangat lemah	Lemah positif
Jumlah data	18816	18816	18816	18816	18816

Dari hasil pengujian korelasi menggunakan korelasi *pearson*, didapatkan variabel *input* suhu, kelembapan dan kecepatan angin untuk prediksi curah hujan. Maka untuk prediksi curah hujan 12 jam kedepan dibutuhkan data historis dari ketiga *input* tersebut. Berikut rincian variabel output dan variabel *input* yang digunakan dalam sistem prediksi curah hujan

- Variabel *output*
CH(t+12) = Prediksi curah hujan 12 jam kemudian
- Variabel *input*

Tabel 3. 12 Variabel historis *Input* untuk prediksi CH(t+12)

T(t-60)	Suhu 60 jam sebelumnya	RH(t-60)	Kelembaban 60 jam sebelumnya	WS(t-60)	Kecepatan angin 60 jam sebelumnya
T(t-48)	Suhu 48 jam sebelumnya	RH(t-48)	Kelembaban 48 jam sebelumnya	WS(t-48)	Kecepatan angin 48 jam sebelumnya
T(T-36)	Suhu 36 jam sebelumnya	RH(T-36)	Kelembaban 36 jam sebelumnya	WS(T-36)	Kecepatan angin 36 jam sebelumnya
T(t-24)	Suhu 24 jam sebelumnya	RH(t-24)	Kelembaban 24 jam sebelumnya	WS(t-24)	Kecepatan angin 24 jam sebelumnya
T(t-12)	Suhu 12 jam sebelumnya	RH(t-12)	Kelembaban 12 jam sebelumnya	WS(t-12)	Kecepatan angin 12 jam sebelumnya
T(t)	Suhu saat ini	RH(t)	Kelembaban saat ini	WS(t)	Kecepatan angin saat ini

Dari klasifikasi variabel historis *input* pada tabel 3.12, maka berikut pada tabel 3.13 adalah hubungan variabel ouput dengan data historis *input* berdasarkan nilai signifikansi dan nilai koefisien korelasi. Dari tabel 3.13 dapat dilihat bahwa hubungan korelasi paling besar yaitu pada variabel *input* T(t-12), RH(t-12) dan WS(t-12) . Maka ketiga variabel *input* tersebut yag akan digunakan dalam sistem prediksi. Selain itu jika dilihat dari nilai signifikansi, ketiga variabel *input* tersebut memiliki nilai signifikansi 0.000 terhadap variabel *output* CH(t+12), sehingga hubungan variabel *output* dan *input* dapat dikatakan signifikan karena nilai signifikansi kurang dari 0.05

Tabel 3. 13 Hubungan Variabel *output* CH(t+12) dengan variabel historis *input*

	Signifikansi	Nilai “t”
T(t-60)	0.000	0.044
T(t-48)	0.000	0.032
T(T-36)	0.000	0.071
T(t-24)	0.000	0.039
T(t-12)	0.000	0.072
T(t)	0.002	0.020
RH(t-60)	0.000	0.113
RH(t-48)	0.000	0.082
RH(T-36)	0.000	0.127
RH(t-24)	0.000	0.084
RH(t-12)	0.000	0.131
RH(t)	0.000	0.106
WS(t-60)	0.000	0.067
WS(t-48)	0.000	0.036
WS(T-36)	0.000	0.073
WS(t-24)	0.000	0.028
WS(t-12)	0.000	0.081
WS(t)	0.000	0.051

3.5.4 Normalisasi

Langkah selanjutnya yang dilakukan sebelum masuk dalam proses pelatihan terhadap jaringan saraf tiruan adalah melakukan proses normalisasi pada setiap data variabel. Dalam normalisasi, nilai dari variabel *input* akan dikonversi menjadi nilai dengan interval 0 hingga 1. Proses normalisasi ini dilakukan agar dapat mempermudah dalam proses pelatihan jaringan syaraf tiruan (Makarynsky, 2004). Berikut adalah formula untuk melakukan normalisasi data

$$\bar{y} = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \quad (3.3)$$

Di mana \bar{y} merupakan nilai data yang telah dinormalisasi, y_i merupakan data asli hasil pengukuran, y_{\max} merupakan nilai

maksimal dari *dataset* sebelum dinormalisasi, dan y_{\min} merupakan nilai minimum sebelum data dinormalisasi.

Normalisasi dilakukan pada setiap variabel yang digunakan baik variabel *input* maupun *output*. Variabel yang digunakan dalam penelitian ini adalah kecepatan angin, arah angin, curah hujan, suhu dan kelembaban. Data yang digunakan yaitu data pada 2017 hingga Februari 2019. Total jumlah data yang digunakan yaitu 18936. Data tersebut memiliki rentang waktu perjam. Sebelum masuk proses pelatihan, data yang didapatkan dari MBKG harus terlebih dahulu dinormalisasi agar memudahkan selama proses pelatihan. Data hasil pelatihan nantinya akan memiliki rentang 0-1 setelah dilakukan normalisasi. . Sebelum dilakukan normalisasi, data setiap variabel memiliki batas atas dan batas bawah yang berbeda-beda.

Berikut pada gambar 3.5 hingga 3.14 merupakan gambar dari persebaran data sebelum dilakukan normalisasi dan setelah dilakukan normalisasi. Gambar 3.5 adalah gambar persebaran data suhu sebelum normalisasi. Gambar tersebut menunjukkan bahwa nilai maksimal pada data suhu yaitu 36°C. Selanjutnya gambar 3.7 adalah gambar yang menunjukkan persebaran data kecepatan angin sebelum dilakukan normalisasi. Gambar tersebut menunjukkan bahwa nilai maksimal data kecepatan angin yaitu 7m/s.

Gambar 3.9 menunjukkan persebaran pada data arah angin sebelum dilakukan normalisasi. Gambar tersebut menunjukkan bahwa data maksimal pada variabel arah angin yaitu 350°. Persebaran data kelembaban sebelum normalisasi ditunjukkan pada gambar 3.11. Gambar tersebut menunjukkan bahwa data kelembaban maksimal yaitu 100%. Sedangkan untuk persebaran data curah hujan sebelum dilakukan normalisasi ditunjukkan pada gambar 3.13. Gambar tersebut menunjukkan bahwa data maksimal curah hujan yaitu 80mm.

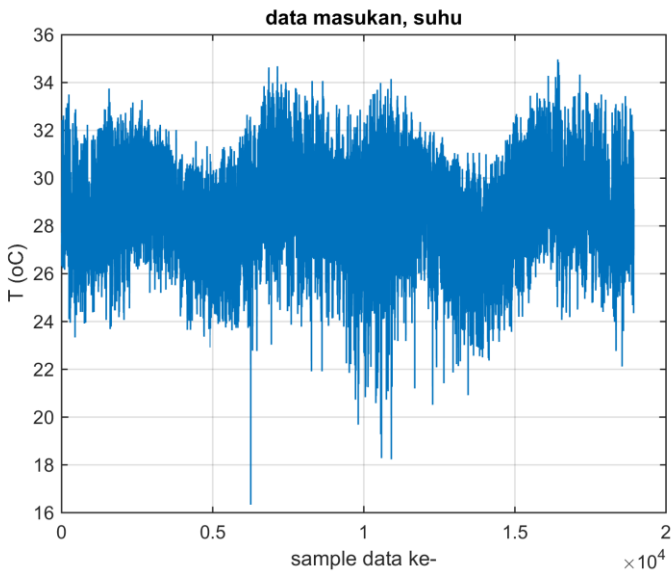
Gambar persebaran data setelah normalisasi ditunjukkan pada gambar 3.6 untuk variabel suhu, 3.8 untuk variabel kecepatan angin, 3.10 untuk variabel arah angin, 3.12 untuk variabel kelembaban dan untuk variabel curah hujan ditunjukkan pada

gambar 3.14. Semua gambar persebaran data setelah normaisasi menunjukkan bahwa rentang yang dimiliki tiap variabel yaitu 0-1.

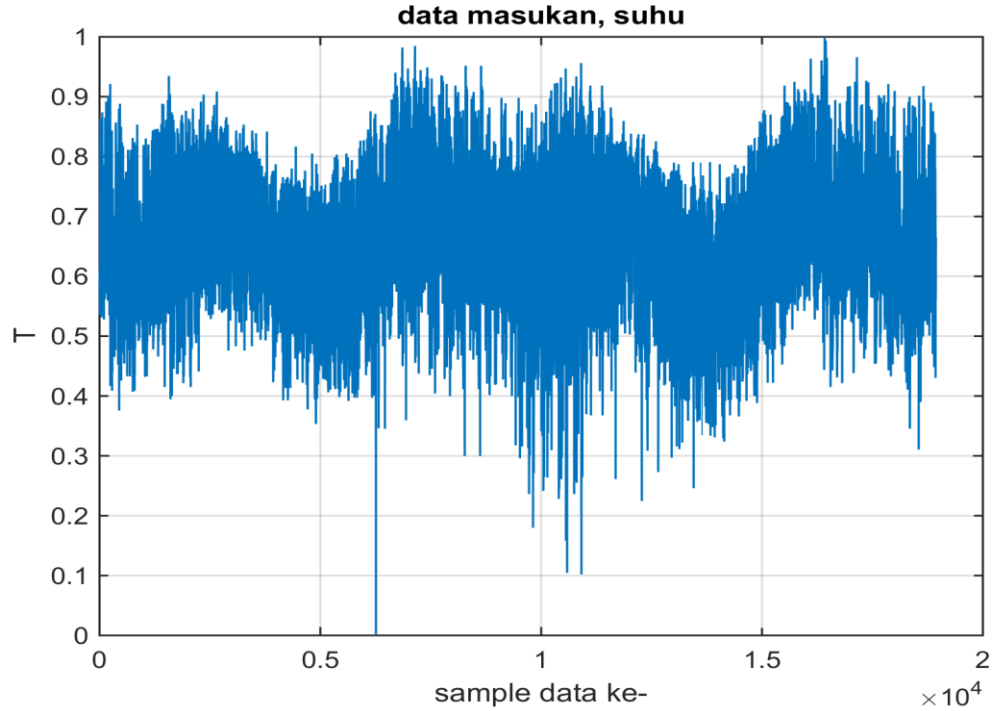
Data yang digunakan sebagai masukan maupun target dari proses training, validasi dan test JST adalah data yang telah dilakukan proses normalisasi. Kemudian, hasil dari JST yang telah melakukan ketiga proses tersebut akan didenormalisasi. Hal tersebut dilakukan untuk mendapatkan kembali nilai asli. Berikut merupakan formula untuk melakukan denormalisasi

$$y_i = \bar{y}(y_{\max} - y_{\min}) + y_{\min} \quad (3.4)$$

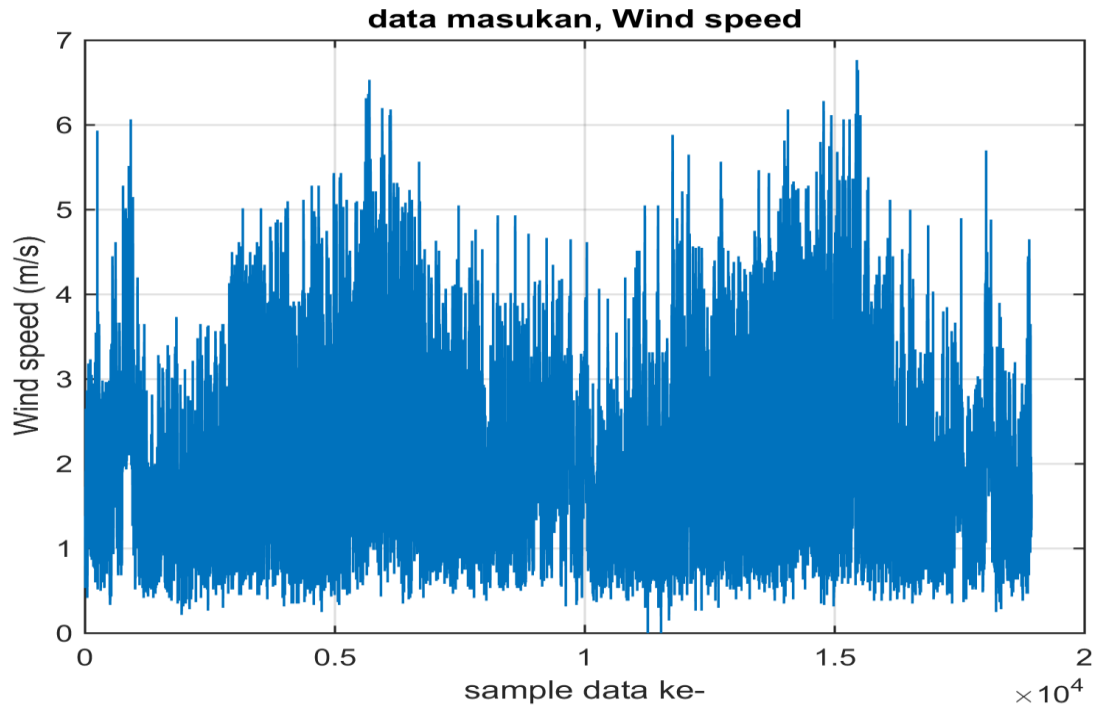
Keluaran dari denormalisasi merupakan nilai prediksi yang bernilai asli dengan rentang sesuai berdasarkan nilai fisis variabel prediksi.



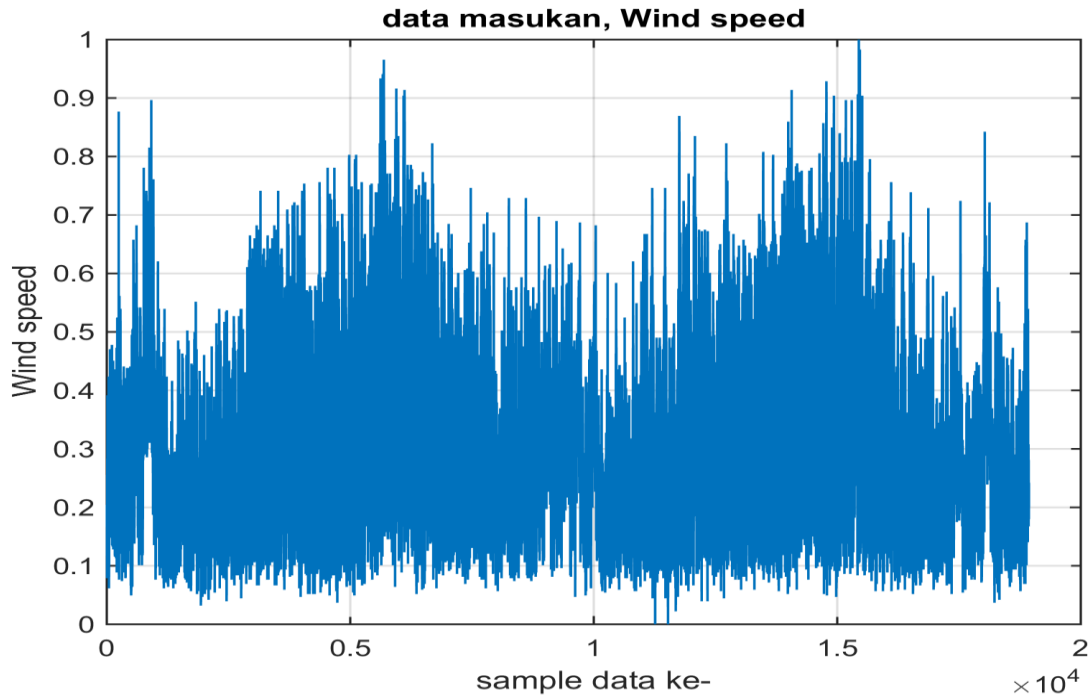
Gambar 3. 5 Persebaran data suhu sebelum normalisasi



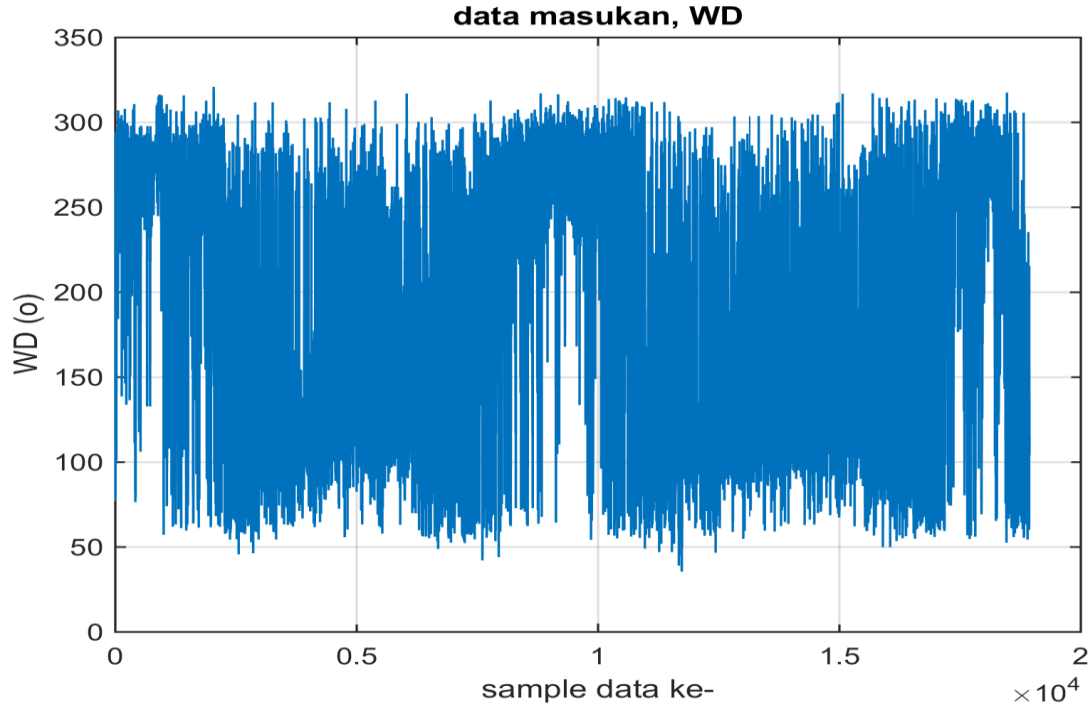
Gambar 3. 6 Persebaran data suhu setelah normalisasi



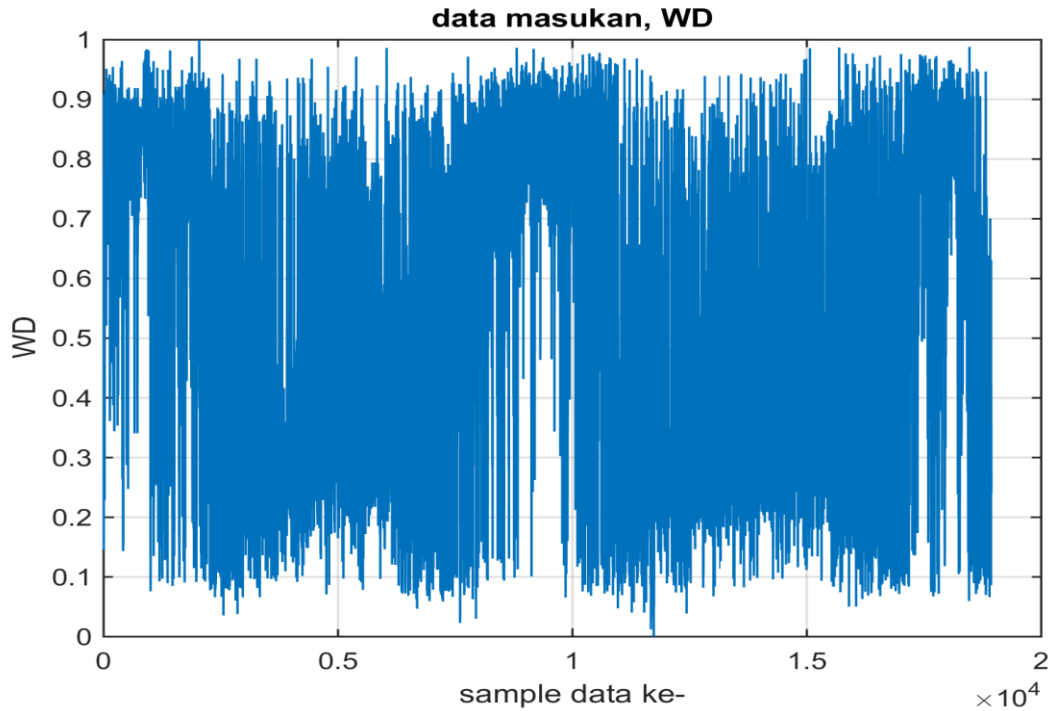
Gambar 3. 7 Persebaran data kecepatan angin sebelum normalisasi



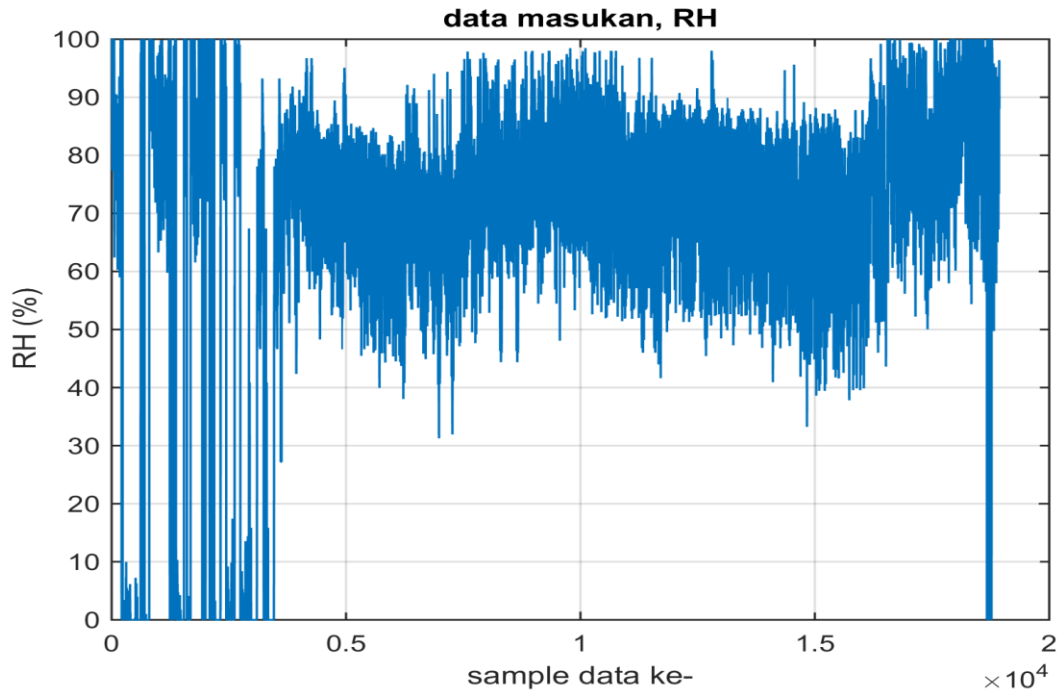
Gambar 3. 8 Persebaran data kecepatan angin setelah normalisasi



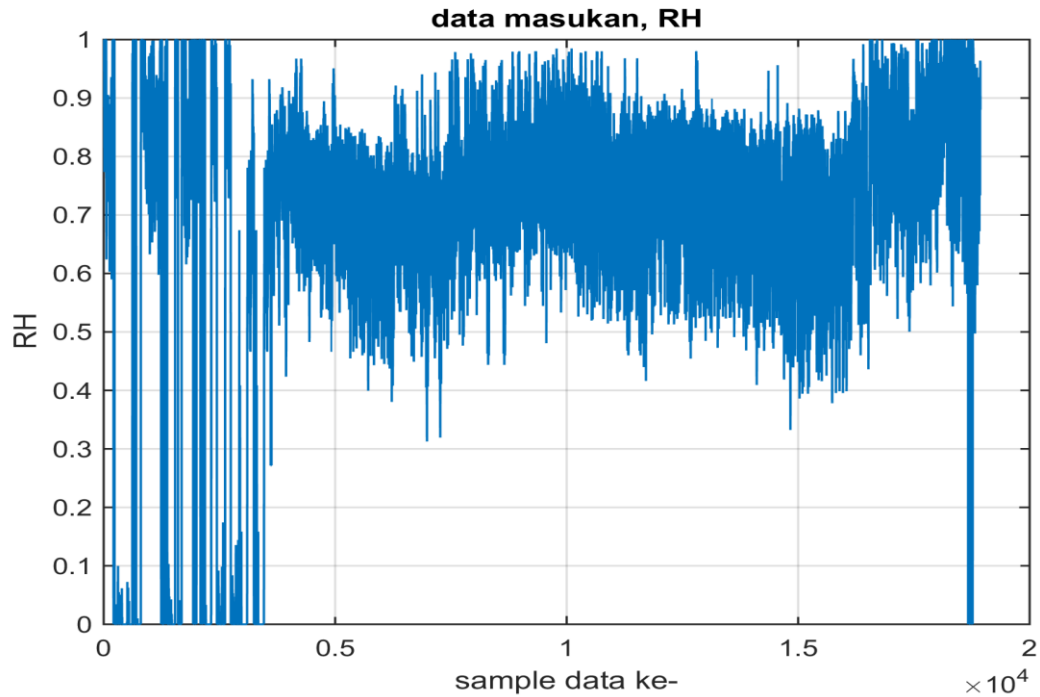
Gambar 3. 9 Persebaran data arah angin sebelum normalisasi



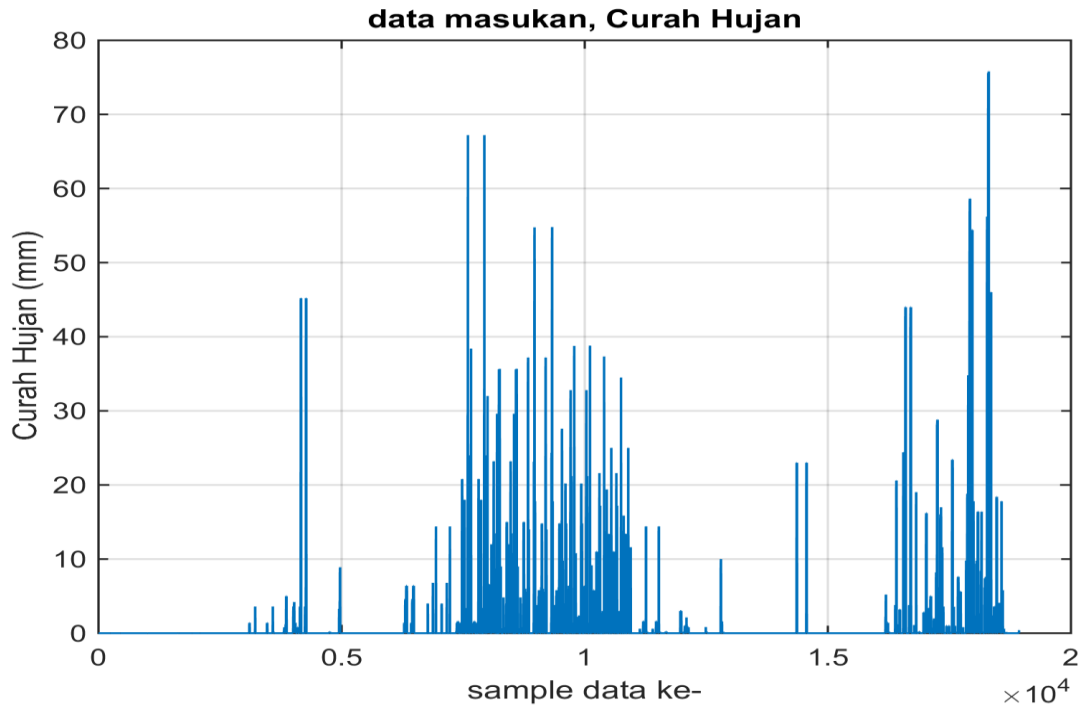
Gambar 3. 10 Persebaran data arah angn setelah normalisasi



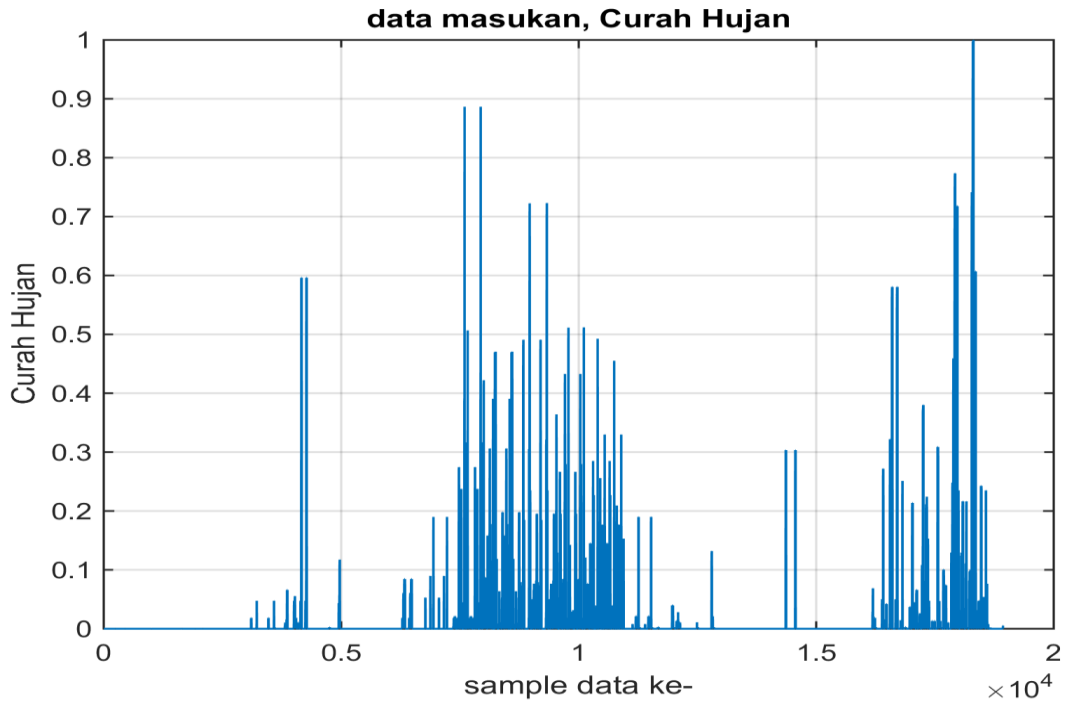
Gambar 3. 11 Persebaran data kelembaban sebelum normalisasi



Gambar 3. 12 Persebaran data kelembaban setelah normalisasi



Gambar 3. 13 Persebaran data curah hujan sebelum normalisasi



Gambar 3. 14 Persebaran data curah hujan setelah normalisasi

3.6 Perancangan *Lavenberg-Marquardt Neural Network*

Perancangan prediktor menggunakan *Lavenberg-Marquardt* ini dilakukan melalui 15 variasi arsitektur. Variasi tersebut dengan merubah jumlah *hidden node* dan jumlah *input*. *Hidden node* yang digunakan adalah 10,20,30,40 dan 50. Jumlah *input* yang digunakan sesuai dengan hasil korelasi yang telah dilakukan sebelumnya pada masing-masing keluaran prediktor yaitu arah angin, kecepatan angin dan curah hujan. Perancangan algoritma *Artificial Neural Network* ini dilakukan dengan bantuan perangkat lunak MATLAB R2014b. Arsitektur jaringan syaraf tiruan (JST) yang digunakan adalah satu *input layer*, satu *hidden layer* dan satu *output layer*. Algoritma yang digunakan pada jaringan syaraf tiruan adalah algoritma *Lavenberg-Marquardt*.

Proses prediksi menggunakan Jaringan Syaraf Tiruan ini menggunakan kumpulan data yang telah diperoleh dari BMKG Maritim II Perak yang telah dijelaskan pada sub bab 3.4. Data-data yang telah dikumpulkan dan telah diolah kemudian dibagi menjadi dua bagian yaitu sebagai data latih dan data uji. Data latih ini akan digunakan oleh algoritma JST untuk berlatih memahami pola variabel cuaca. Hal tersebut akan sangat berguna dalam menentukan tingkat akurasi prediksi. Hasil dari pelatihan algoritma tersebut akan diuji apakah sistem yang dirancang sudah menghasilkan MSE yang kecil. Perbandingan data latih dan data uji pada penelitian ini adalah 7:3 atau dalam kata lain 70% data total akan dijadikan sebagai data latih dan 30% lainnya akan dijadikan sebagai data uji.

Jumlah iterasi maksimum atau *epoch* yang digunakan pada penelitian ini adalah 1000. Pada setiap variasi *hidden node* akan dilihat performanya jika salah satu parameter algoritma *Lavenberg-Marquardt* divariasikan, salah satu parameter tersebut yaitu target *error* dengan variasi 0.001,0.002 dan 0.003. Pada penelitian yang dilakukan oleh Yudhi Andrian dengan memvariasikan target *error* pada jaringan syaraf tiruan didapatkan hasil bahwa target *error* yang berbeda akan menghasilkan jumlah iterasi yang berbeda pula (Andrian & Ningsih, 2014b). Semakin kecil target *error*, maka jumlah iterasi akan semakin besar. Dan

semakin kecil target *error* maka nilai keakurasiannya semakin baik (Andrian & Ningsih, 2014b).

Hasil pelatihan yang telah dilakukan beserta arsitektur dari jaringan syaraf tiruan pada masing-masing variasi akan dijelaskan dalam sub bab di bawah ini

3.6.1 Kecepatan Angin

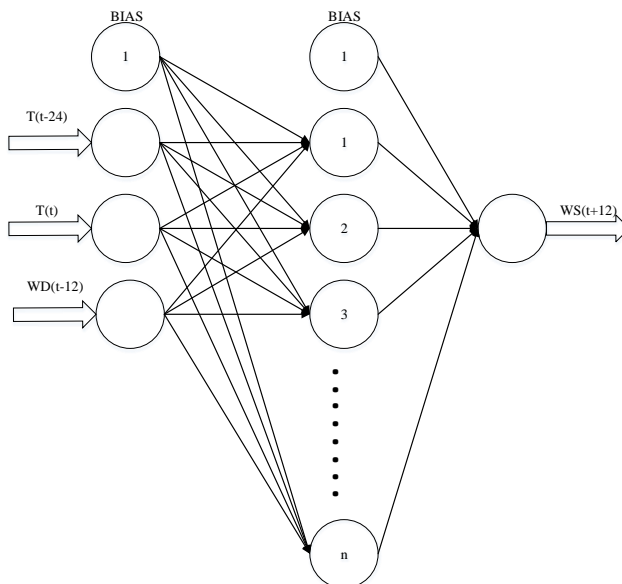
Pengukuran performa prediksi dilihat dari nilai MSE yang dihasilkan. Seperti yang telah dijelaskan sebelumnya bahwa variasi yang digunakan dalam penelitian ini adalah jumlah *hidden node* pada *hidden layer* dan salah satu parameter algoritma *Lavenberg-Marquardt* yaitu target *error*. Setiap variasi *hidden node* dan variasi target *error* memiliki performa yang berbeda. Hal tersebut ditunjukkan dari nilai MSE yang dihasilkan dari tiap variasi.

Tabel 3. 14 Hasil MSE berdasarkan percobaan variabel kecepatan angin

Target <i>error</i>	<i>Hidden node</i>	MSE
0.001	10	0.026784
	20	0.02657
	30	0.027054
	40	0.027428
	50	0.029114
0.002	10	0.026385
	20	0.026686
	30	0.026688
	40	0.027121
	50	0.02777
0.003	10	0.026862
	20	0.026753
	30	0.027309
	40	0.027165
	50	0.027926

Tabel 3.14 diatas menunjukkan bahwa model prediktor terbaik pada variabel kecepatan angin merupakan model yang memiliki *hidden node* 10 dengan nilai target *error* 0.002. Hal tersebut ditunjukkan dengan nilai MSE pada *hidden node* 10 dan target *error* 0.002 memiliki nilai MSE terkecil dibandingkan dengan nilai MSE lainnya. Performa yang didapatkan oleh model tersebut memiliki ukuran MSE 0.026385. Tabel 3.14 menunjukkan bahwa variasi nilai *target error* yang diberikan tidak berpengaruh secara signifikan pada hasil performa. Hal tersebut dapat dilihat dari selisih masing-masing performa tidak memiliki besar selisih yang signifikan.

Gambar 3.15 dibawah ini menunjukkan gambar arsitektur terbaik dari prediksi kecepatan angin



Gambar 3. 15 Arsitektur JST Kecepatan Angin

Arsitektur jaringan syaraf tiruan dari prediktor kecepatan angin terdiri atas 10 *hidden node* dengan *hidden layer* 1 lapis. Arsitektur tersebut diambil berdasarkan hasil performa yang

terbaik yaitu ditunjukkan dengan nilai MSE yang terkecil. Arsitektur prediksi kecepatan angin dapat dilihat pada gambar 3.15. Nilai n *hidden node* pada *hidden layer* bernilai 10. Dimana 10 *node* tersebut merupakan jumlah *hidden node* arsitektur prediktor yang terhubung dengan setiap *node* yang berada di *layer input* dan setiap *node* yang ada di *layer output*. Pada setiap *node* yang terhubung tersebut memiliki nilai bobot yang berbeda. Nilai bobot tersebut yang nantinya akan dimodelkan secara matematis. Sehingga, setiap *node* dikalikan nilai bobot yang kemudian hasilnya menjadi masukan dari *node* di *layer* selanjutnya

Gambar 3.15 menunjukkan bahwa pada sistem prediksi kecepatan angin 12 jam kemudian memiliki tiga variabel *input* yaitu $T(t-24)$, $T(t)$ dan $WD(t-12)$. Selain itu terdapat bias sebesar 1 yang menjadi *input* pada *hidden layer*. Semua *input* pada sistem prediksi suhu tersebut akan diproses dalam perangkat lunak Matlab baik proses pelatihan maupun uji sesuai dengan pembagian jumlah data yang telah ditentukan.

$T(t-24)$ = Suhu 24 jam sebelumnya

$T(t)$ = Suhu saat ini

$WD(t-12)$ = Arah angin 12 jam sebelumnya

Untuk mempermudah proses data saat dilatih oleh Matlab maka semua *input* tersebut dijadikan kedalam bentuk matriks $m \times n$. Dimana m menunjukkan jumlah variabel sedangkan $n=1,2,3,4,\dots,x$, dengan x adalah jumlah data untuk *training*, validasi dan uji. Berikut merupakan bentuk matriks dari *input* sebagai data training

$$[WS(t+12)] = f \begin{bmatrix} T(t-12) \\ T(t) \\ WD(t-12) \end{bmatrix} \quad (3.5)$$

Ketiga variabel *input* terdiri atas data sejumlah x kemudian dijadikan masukan dari JST untuk melalui proses *training* guna mendapatkan model prediksi.

Berdasarkan percobaan yang telah dilakukan prediktor terbaik adalah model dengan *hidden node* 10 dan target *error* 0.002. tabel 3.15 dibawah ini adalah tabel yang menunjukkan spesifikasi JST prediktor kecepatan angin terbaik

Tabel 3. 15 Spesifikasi Arsitektur JST untuk prediktor kecepatan angin

Parameter	Nilai
Jumlah <i>Hidden layer</i>	1
Jumlah <i>Hidden node</i> pada <i>Layer</i>	10
<i>Training</i>	<i>Lavenberq-Marquardt</i>
<i>Learning Rate</i>	0.1
Fungsi Aktifasi <i>Layer 1</i>	Logsig
Fungsi Aktifasi <i>Layer 2</i>	logsig
<i>Target Error</i>	0.002

3.6.2 Arah Angin

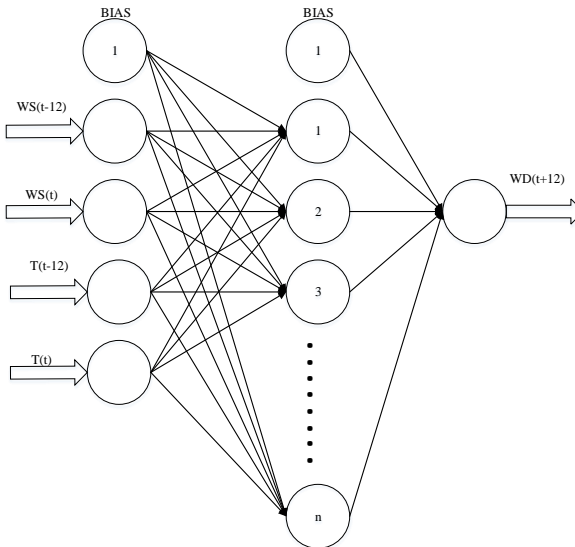
Tabel 3.16 menunjukkan hasil prediksi dari variabel arah angin dengan setiap variasi *hidden node* dan *target error*.

Tabel 3. 16 Hasil MSE berdasarkan percobaan variabel arah angin

Target <i>error</i>	<i>Hidden node</i>	MSE
0.001	10	0.060009
	20	0.060184
	30	0.061702
	40	0.063083
	50	0.063998
0.002	10	0.060419
	20	0.060786
	30	0.061188
	40	0.063041
	50	0.064276
0.003	10	0.061004
	20	0.060703
	30	0.060879
	40	0.062544
	50	0.062798

Berdasarkan hasil penelitian didapatkan variasi terbaik yaitu pada *hidden node* 10 dan *target error* 0.001. Hal tersebut ditunjukkan dengan nilai MSE pada variasi tersebut memiliki nilai yang lebih kecil dibandingkan dengan variasi lainnya yaitu sebesar 0.060009.

Gambar 3.16 dibawah ini merupakan gambar arsitektur terbaik dari prediksi arah angin



Gambar 3. 16 Arsitektur JST Arah Angin

Arsitektur jaringan syaraf tiruan dari prediktor arah angin terdiri atas 10 *hidden node* dengan *hidden layer* 1 lapis. Arsitektur tersebut diambil berdasarkan hasil performa yang terbaik yaitu ditunjukkan dengan nilai MSE yang terkecil. Arsitektur prediksi arah angin dapat dilihat pada gambar 3.16. Nilai *n hidden node* pada *hidden layer* bernilai 10. Dimana 10 *node* tersebut merupakan jumlah *hidden node* pada arsitektur prediktor. *Hidden node* tersebut akan terhubung dengan setiap *node* yang berada di *layer input* dan setiap *node* yang ada di *layer output*. Setiap *node* yang terhubung tersebut memiliki nilai bobot yang berbeda. Nilai bobot

tersebut yang nantinya akan dimodelkan secara matematis. Sehingga, setiap *node* dikalikan nilai bobot dan kemudian hasilnya menjadi masukan dari *node* di *layer* selanjutnya.

Gambar 3.16 menunjukkan bahwa untuk memprediksi arah angin 12 jam kemudian terdapat empat *input* yaitu $WS(t-12)$, $WS(t)$, $T(t-12)$ dan $T(t)$. Selain itu terdapat bias sebesar 1 yang menjadi *input* pada *hidden layer*. Semua *input* pada sistem prediksi tersebut akan diproses dalam perangkat lunak Matlab baik proses pelatihan maupun uji sesuai dengan pembagian jumlah data yang telah ditentukan.

$WS(t-12)$ = Kecepatan angin 12 jam sebelumnya
 $WS(t)$ = Kecepatan angin saat ini
 $T(t-12)$ = Suhu 12 jam sebelumnya
 $T(t)$ = Suhu saat ini

Data-data yang telah didapatkan terkait setiap variabel *input* t , diubah ke dalam bentuk matriks $m \times n$ guna memudahkan format pemrosesan data saat dilakukan proses pelatihan menggunakan MATLAB. Di mana m terdiri atas sejumlah variabel, sementara $n=1,2,3,4,\dots,x$, dengan x adalah jumlah data untuk *training*, validasi, dan uji. Berikut merupakan bentuk matriks dari masukan sebagai data *training* MATLAB.

$$[WD(t+12)] = \begin{pmatrix} WS(t-12) \\ WS \\ T(t-12) \\ T(t) \end{pmatrix} \quad (3.6)$$

Empat variabel *input* tersebut yang terdiri atas data sejumlah x yang dijadikan masukan dari JST untuk melakukan proses latih dan validasi guna mendapatkan model prediksi arah angin.

Arsitektur model JST prediktor arah angin dalam proses *training* menggunakan metode *training Lavenberg-Marquadt*. Hal ini dilakukan karena pelatihan ini merupakan metodologi pelatihan yang paling cepat mengalami konvergensi dan paling baik secara performa di antara metodologi pelatihan lain yang disediakan oleh MATLAB. Secara rinci spesifikasi JST prediktor arah angin

dengan prediktor terbaik dapat dilihat pada Tabel 3.17 di bawah ini.

Tabel 3. 17 Spesifikasi Arsitektur JST untuk prediktor arah angin

Parameter	Nilai
Jumlah <i>Hidden layer</i>	1
Jumlah <i>Hidden node</i> pada <i>Layer</i>	10
<i>Training</i>	<i>Lavenberg-Marquardt</i>
<i>Learning Rate</i>	0.1
Fungsi Aktifasi <i>Layer 1</i>	Logsig
Fungsi Aktifasi <i>Layer 2</i>	Logsig
Target <i>Error</i>	0.001

3.6.3 Curah Hujan

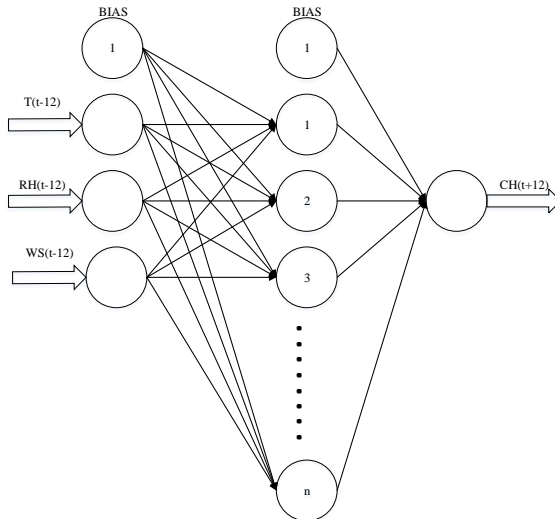
Tabel 3.18 menunjukkan hasil prediksi curah hujan tiap variasi *target error* dan variasi *hidden node*.

Tabel 3. 18 Hasil MSE berdasarkan percobaan variabel curah hujan

Target error	<i>Hidden node</i>	MSE
0.001	10	0.001267
	20	0.001595
	30	0.016287
	40	0.015519
	50	0.022266
0.002	10	0.012944
	20	0.018062
	30	0.017875
	40	0.015513
	50	0.015144
0.003	10	0.011771
	20	0.012024
	30	0.011863
	40	0.012323
	50	0.011710

Dari hasil penelitian dapat dilihat bahwa performa paling baik adalah dengan MSE 0.001267. MSE tersebut didapatkan dari variasi *target error* 0.001 dan *hidden node* 10. Hal tersebut ditunjukkan dengan nilai MSE yang paling kecil diantara variasi lain.

Arsitektur jaringan syaraf tiruan dari prediktor curah hujan terdiri atas 10 *hidden node* dengan *hidden layer* 1 lapis. Arsitektur tersebut diambil berdasarkan hasil performa yang terbaik yaitu ditunjukkan dengan nilai MSE yang terkecil. Arsitektur prediksi arah angin dapat dilihat pada gambar 3.17. Nilai *n hidden node* pada *hidden layer* bernilai 10.



Gambar 3. 17 Arsitektur JST Curah Hujan

Dari gambar 3.13 terlihat bahwa untuk memprediksi curah hujan 12 jam kemudian dibutuhkan tiga *input* yaitu $T(t-12)$, $RH(t-12)$ dan $WS(t-12)$. Selain itu terdapat bias sebesar 1 yang menjadi *input* pada *hidden layer*. Semua *input* pada sistem prediksi curah hujan tersebut akan diproses dalam perangkat lunak Matlab baik proses pelatihan maupun uji sesuai dengan pembagian jumlah data yang telah ditentukan.

$T(t-12)$ = Suhu 12 jam sebelumnya
 $RH(t-12)$ = Kelembaban saat ini
 $WS(t-12)$ = Kecepatan angin 12 jam sebelumnya

Data-data yang telah didapatkan terkait setiap variabel *input* t , diubah ke dalam bentuk matriks $m \times n$ guna memudahkan format pemrosesan data saat dilakukan proses pelatihan menggunakan MATLAB. Di mana m terdiri atas sejumlah variabel, sementara $n=1,2,3,4,\dots,x$, dengan x adalah jumlah data untuk *training*, validasi, dan uji. Berikut merupakan bentuk matriks dari masukan sebagai data *training* MATLAB.

$$[CH(t+12)] = \begin{pmatrix} T(t-12) \\ RH(t-12) \\ WS(t-12) \end{pmatrix} \quad (3.7)$$

Tiga variabel *input* tersebut yang terdiri atas data sejumlah x yang dijadikan masukan dari JST untuk melakukan proses latih dan validasi guna mendapatkan model prediksi arah angin.

Arsitektur model JST prediktor curah hujan dalam proses *training* menggunakan metode training *Lavenberg-Marquadt*. Secara rinci spesifikasi JST prediktor curah hujan dengan prediktor terbaik dapat dilihat pada Tabel 3.19 di bawah ini.

Tabel 3. 19 Spesifikasi Arsitektur JST untuk prediktor curah hujan

Parameter	Nilai
Jumlah <i>Hidden layer</i>	1
Jumlah <i>Hidden node</i> pada <i>Layer</i>	10
<i>Training</i>	<i>Lavenberg-Marquardt</i>
<i>Learning Rate</i>	0.1
Fungsi Aktifasi <i>Layer 1</i>	Logsig
Fungsi Aktifasi <i>Layer 2</i>	logsig
Target <i>Error</i>	0.002

3.7 Pengujian prediktor

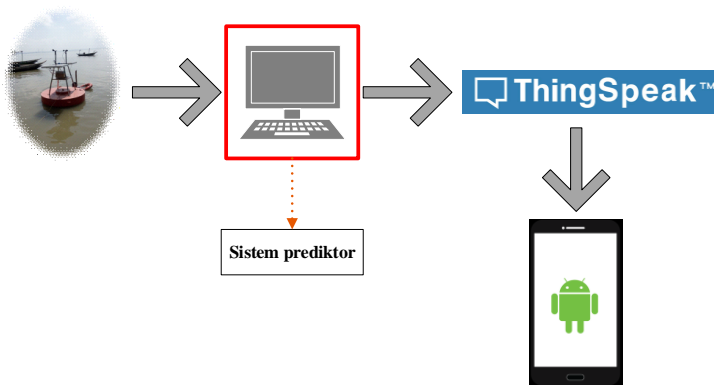
Proses prediksi pada jaringan syaraf tiruan terbagi menjadi tiga yaitu proses pelatihan, pengujian dan validasi. Ketiga proses tersebut dibedakan atas data yang digunakan. Data pelatihan dan pengujian adalah data yang didapatkan dari BMKG Maritim II

Perak dengan pembagian 70% data pelatihan dan 30% data pengujian. Sedangkan data yang digunakan sebagai proses validasi adalah data yang didapatkan dari pengambilan data secara *realtime* di Pantai Kenjeran.

Pelatihan jaringan dilakukan dengan merancang arsitektur yang terdiri dari *input layer*, *hidden layer* dan *output layer*. Jumlah variasi arsitektur dalam penelitian ini adalah sejumlah 15. Variasi arsitektur tersebut dibedakan atas jumlah *hidden node* dan target *error* yang digunakan. Setiap variasi terget *error* yang digunakan terdapat variasi *hidden node*. Hasil dari proses pelatihan adalah nilai bobot yang akan digunakan dalam proses pengujian. Bobot adalah koefisien pengali yang terletak diantara *input layer* dan *hidden layer* serta terletak diantara *hidden layer* dan *output layer*. Variasi arsitektur yang digunakan dalam proses pelatihan akan mendapatkan nilai bobot yang berbeda. Dan nilai bobot tersebut yang akan digunakan sebagai faktor pengali pada proses pengujian. Arsitektur prediktor terbaik didapatkan dari hasil pengujian dengan nilai MSE yang paling kecil. Dan arsitektur terbaik hasil pengujian tersebut akan digunakan dalam proses validasi.

3.8 Pengujian Jaringan dengan Data *Real Time*

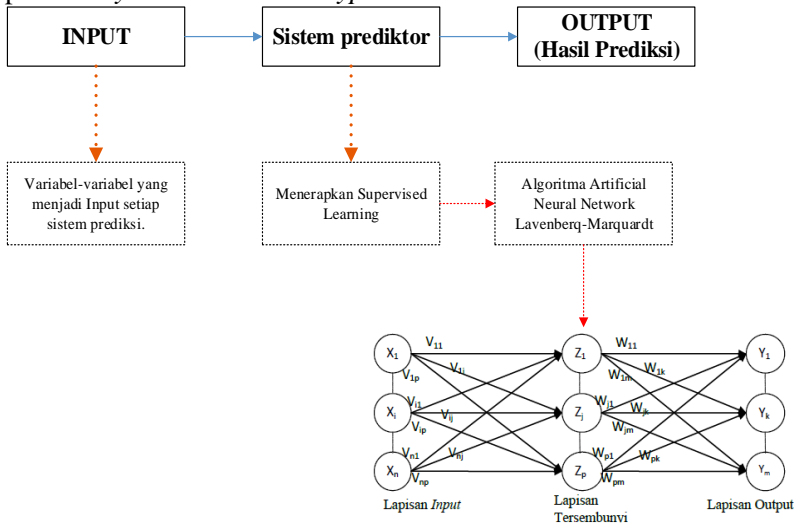
Berikut adalah proses dari pengambilan data *realtime* yang dilakukan



Gambar 3. 18 Proses pengambilan data *realtime*

Validasi dilakukan dengan menguji prediktor JST-LM dengan data hasil pengukuran menggunakan *Buoyweather Station Type II* yang dilakukan di selat Madura selama tiga hari, pada hari Senin-Selasa (10-11 Juni 2019) dan hari Minggu 14 Juni 2019. Pengambilan data *realtime* dilakukan dari pinggir pantai hingga tengah laut hingga jarak 1km . Data yang diperoleh dari tengah laut akan langsung ditransmisikan ke darat akan masuk dalam sistem database thingspeak dan kemudian akan diproses oleh prediktor. Setelah proses prediksi selesai maka hasil akan terupload kembali ke thingspeak untuk ditampilkan pada sistem monitor android. Data masuk *database* tiap 17 detik maka pengambilan data oleh sensor pun akan tercatat otomatis tiap 17 detik sehingga data tersebut diolah terlebih dahulu oleh prediktor menjadi perjam.

Berikut adalah diagram blok penggunaan sistem prediktor pada *Buoyweather Station Type II*



Gambar 3. 19 Digram blok sistem prediktor pada buoyweather

Gambar 3.19 menunjukkan penggunaan *machine learning* pada *buoyweather* adalah terletak pada sistem prediktor. Sistem prediktor yang digunakan pada *Buoyweather Season Type II* adalah berbasis *Supervised Learning*, dimana *Supervised Learning*

adalah bagian algoritma dari *Machine Learning*. Salah satu algoritma *supervised learning* yang digunakan adalah algoritma *Artificial Neural Network Lavenberg-Marquardt*. Pada proses Algoritma *Artificial Neural Network* terdapat arsitektur yang meliputi *input layer*, *hidden layer* dan *output layer*. Proses pelatihan dan pengujian akan menghasilkan bobot terbaik sehingga dapat menghasilkan arsitektur terbaik untuk sistem prediksi. Bobot adalah koefisien yang terletak diantara *input layer* dan *hidden layer* serta terletak diantara *hidden layer* dan *output layer*.

3.9 Analisis Hasil dan Pembahasan

Arsitektur terbaik JST digunakan untuk memprediksi kecepatan angin, arah angin dan curah hujan di suatu lokasi. Hasil yang didapat dari penelitian yang telah dilakukan dianalisis dan dilakukan pembahasan mengenai performansi penelitian dan berbagai faktor yang mempengaruhi penelitian untuk kemudian ditarik kesimpulan. Hasil analisis digunakan untuk rekomendasi ke nelayan sebagai pertimbangan kegiatan maritim.

3.10 Kesimpulan dan Saran

Kesimpulan merupakan jawaban dari rumusan masalah yang diangkat dalam penulisan tugas akhir. Jika kesimpulan masih belum menjawab rumusan masalah, maka akan dimasukkan saran sebagai acuan untuk penelitian selanjutnya atau saran terkait penelitian ini. Sehingga penelitian selanjutnya bisa lebih baik dari penelitian ini. Adanya kesimpulan dan saran diharapkan mampu mengembangkan penelitian tentang maritim agar lebih baik dan bisa diaplikasikan di masyarakat.

Halaman sengaja dikosongkan

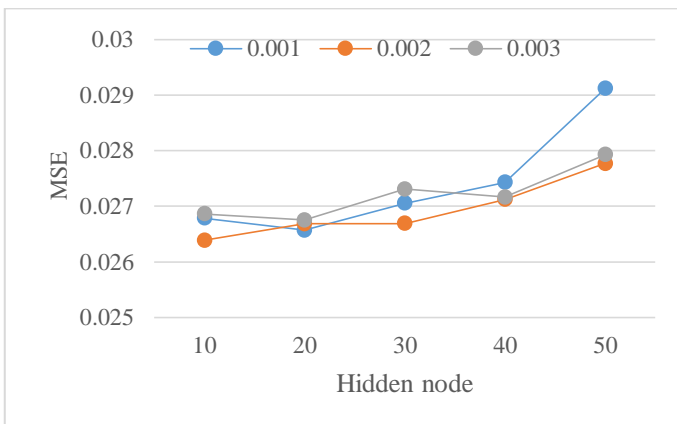
BAB IV HASIL dan PEMBAHASAN

4.1 Hasil Prediksi berdasarkan Variasi *Hidden node*

Variasi *hidden node* yang digunakan akan memiliki nilai performa yang berbeda. Nilai performa ditunjukkan dengan nilai MSE disetiap pengujian. Berikut adalah hasil prediksi yang dihasilkan dengan memvariasikan *hidden node*

a. Kecepatan Angin

Gambar 4.1 merupakan gambar yang menunjukkan perubahan MSE berdasarkan 3 variasi target *error* yaitu 0.001; 0.002 dan 0.003. Berikut adalah grafik hasil prediksi jika dilihat disetiap variasi *hidden node*



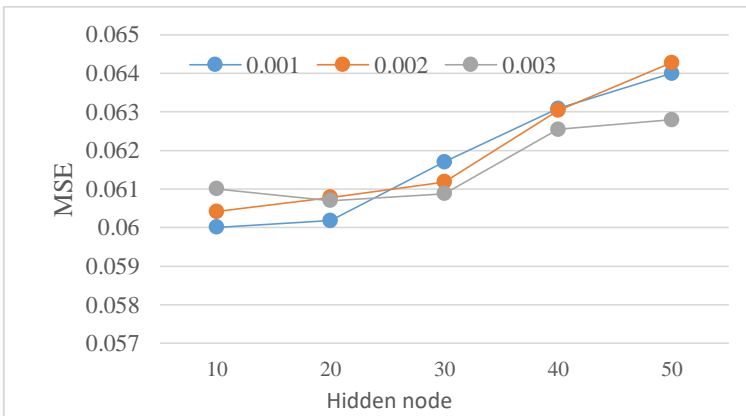
Gambar 4. 1 Hasil prediksi Kecepatan angin berdasarkan variasi *hidden node*

Dari gambar 4.1 diatas dapat dilihat bahwa setiap target *error* memiliki karakteristik performa yang berbeda jika diberikan variasi *hidden node*. Pada target *error* 0.001 dapat dilihat bahwa semakin banyak *hidden node* yang diberikan maka nilai MSE juga semakin besar. Hal ini juga ditunjukkan pada target error 0.002. Pada target *error* 0.002 dengan bertambahnya *hidden node* yang diberikan nilai MSE juga bertambah meskipun pada *hidden node*

30 mengalami penurunan, namun penurunan tersebut tidak begitu signifikan. Sedangkan pada *hidden node* 0.003 hasil MSE juga cenderung naik meskipun mengalami penurunan pada *hidden node* 40.

b. Wind Direction (Arah Angin)

Gambar 4.2 adalah grafik yang menunjukkan hasil prediksi Arah angin jika dilihat dari variasi *hidden node* di setiap variasi target *error* 0.001; 0.002 dan 0.003



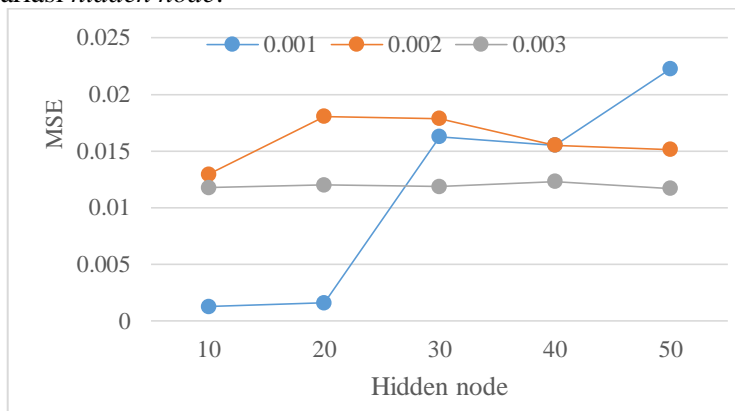
Gambar 4. 2 Hasil prediksi Arah angin berdasarkan variasi *hidden node*

Gambar 4.2 menunjukkan bahwa perubahan *hidden node* yang diberikan memberikan efek pada performa JST. Hal ini ditunjukkan dengan perubahan nilai MSE disetiap perubahan *hidden node*, hal ini juga berlaku pada semua target *error* yang divariasikan. Dapat dilihat dalam gambar diatas bahwa semakin besar nilai *hidden node* yang diberikan memberikan efek performa yang lebih rendah karena semakin besar *hidden node* yang diberikan, nilai MSE semakin bertambah pula.

c. Curah Hujan

Prediksi yang dilakukan memberikan nilai performa yang berbeda disetiap variasi *hidden node*. Pada gambar 4.3 dapat dilihat perbedaan performa disetiap variasi *hidden node*. Setiap *hidden node* yang diberikan memberikan pengaruh pada nilai MSE.

Gambar 4.3 menunjukkan karakteristik performa jika diberikan variasi *hidden node*.



Gambar 4. 3 Hasil prediksi Curah Hujan berdasarkan variasi *hidden node*

Pada target *error* 0.001 dapat dilihat bahwa semakin banyak *hidden node* yang diberikan maka nilai MSE juga semakin besar. Hal ini tidak ditunjukkan pada target *error* 0.002. Pada target *error* 0.002 dengan bertambahnya *hidden node* yang diberikan nilai MSE cenderung turun meskipun pada *hidden node* 20 mengalami peningkatan. Sedangkan pada *hidden node* 0.003 hasil MSE tidak jauh berbeda sehingga grafik yang dihasilkan cenderung datar.

4.2 Hasil Prediksi berdasarkan Variasi Target Error

Variasi target *error* pada proses prediksi JST dapat mempengaruhi iterasi pada proses prediksi menggunakan JST. Berdasarkan penelitian Yudhi Andrian target *error* semakin kecil maka iterasi semakin besar (Andrian & Ningsih, 2014b). Variasi target *error* yang digunakan pada penelitian ini adalah 0,001; 0,002 dan 0,003. Dari variasi ini akan dilihat performa dan iterasi yang dapat dicapai jika iterasi maksimum yang diberikan adalah 1000.

a. *Wind Speed* (Kecepatan Angin)

Tabel 4.1 adalah tabel yang menunjukkan hasil prediksi kecepatan angin berdasarkan variasi target *error*. Dari hasil

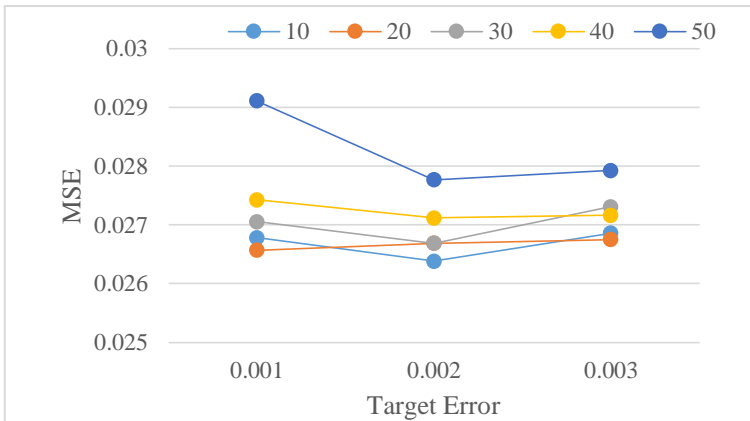
percobaan yang telah dilakukan pada prediksi kecepatan angin, setiap variasi target *error* mampu mencapai iterasi maksimum yang diberikan yaitu 1000.

Tabel 4. 1 Peformansi Prediksi Kecepatan angin berdasarkan variasi target *error*

Target <i>error</i>	<i>Hidden node</i>	MSE
0.001	10	0.026784
	20	0.026570
	30	0.027054
	40	0.027428
	50	0.029114
0.002	10	0.026385
	20	0.026686
	30	0.026688
	40	0.027121
	50	0.027770
0.003	10	0.026862
	20	0.026753
	30	0.027309
	40	0.027165
	50	0.027926

Dari hasil percobaan yang telah dilakukan menunjukkan bahwa variasi target *error* tidak memberikan pengaruh pada proses iterasi. Hal ini bisa disebabkan karena karakteristik data yang dimiliki oleh kecepatan angin. Setiap data memiliki karakteristik yang berbeda sehingga akan berpengaruh dalam proses prediksi. Tabel 4.1 menunjukkan bahwa target *error* berpengaruh pada performa dari JST. Hal ini ditunjukkan dengan berubahnya nilai MSE setiap perubahan target *error*. Berikut adalah grafik yang menunjukkan perubahan performa JST disetiap perubahan variasi target *error* yang diberikan. Performa JST dalam penelitian ini ditunjukkan

dengan nilai MSE yang dihasilkan. Gambar 4.4 menunjukkan perubahan performa yang terjadi akibat perubahan target *error* yang digunakan pada *hidden node* 10,20,30,40 dan 50.



Gambar 4. 4 Hasil Prediksi Kecepatan Angin berdasarkan variasi *target error*

Pada semua *hidden node* yang digunakan perubahan MSE dari 0.001 ke target *error* 0.002 mengalami penurunan namun meningkat kembali pada target *error* 0.003. Namun pada *hidden node* 20 berbeda, pada variasi ini performa JST mengalami kenaikan pada target *error* 0.002 dan mengalami penurunan pada target *error* 0.003. Dalam hal ini dapat dikatakan bahwa performa terbaik dari target *error* yang digunakan adalah pada target *error* 0.002.

b. *Wind Direction* (Arah Angin)

Tabel 4.2 menunjukkan hasil prediksi arah angin berdasarkan variasi target *error*. Berdasarkan hasil percobaan menunjukkan bahwa sama halnya dengan variabel kecepatan angin, hasil percobaan pada prediksi arah angin menunjukkan bahwa disetiap variasi target *error* mampu mencapai iterasi maksimum yang diberikan yaitu 1000. Namun dapat dilihat bahwa variasi target *error* yang digunakan mempengaruhi performansi JST. Hal

tersebut ditunjukkan dengan berubahnya nilai MSE disetiap variasi target *error* yang digunakan.

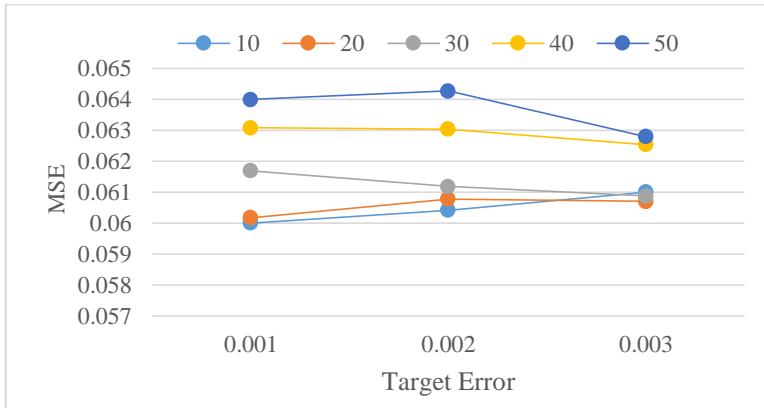
Tabel 4. 2 Performansi Prediksi Arah angin berdasarkan variasi target *error*

Target <i>error</i>	<i>Hidden node</i>	MSE
0,001	10	0.060009
	20	0.060184
	30	0.061702
	40	0.063083
	50	0.063998
0.002	10	0.060419
	20	0.060786
	30	0.061188
	40	0.063041
	50	0.064276
0.003	10	0.061004
	20	0.060703
	30	0.060879
	40	0.062544
	50	0.062798

Tabel 4.2 menunjukkan bahwa variasi target *error* yang digunakan akan berpengaruh pada performa dari JST. Berikut adalah grafik yang menunjukkan perubahan performa JST disetiap perubahan variasi target *error* yang diberikan. Performa JST dalam penelitian ini ditunjukkan dengan nilai MSE yang dihasilkan.

Gambar 4.5 adalah gambar grafik yang menunjukkan pengaruh perubahan target *error* yang digunakan terhadap performa JST. Pada gambar 4.5 dapat dilihat bagaimana perubahan performa disetiap perubahan target *error* yang digunakan. Dalam gambar

juga ditunjukkan pengaruh perubahan target *error* di setiap variasi *hidden node* yang digunakan yaitu 10,20,30,40 dan 50.



Gambar 4. 5 Hasil Prediksi Arah Angin berdasarkan variasi *target error*

Dari gambar diatas dapat dilihat bahwa perubahan target *error* mempengaruhi nilai MSE yang dihasilkan. Gambar 4.5 di atas menunjukkan bahwa variasi *hidden node* yang digunakan akan mempengaruhi perubahan MSE dari target *error* 0.001 hingga target *error* 0.003. Setiap variasi target *error* yang digunakan memiliki performansi yang berbeda di setiap variasi *hidden node* yang digunakan. *Hidden node* 10 menunjukkan semakin besar nilai target *error* nilai MSE cenderung naik. Sedangkan pada *hidden node* 20,40 dan 50 mengalami kenaikan MSE ketika target *error* 0.002 dan mengalami penurunan ketika target *error* 0.003. Dari grafik diatas dapat dilihat bahwa MSE paling kecil terletak pada target *error* 0.001 sehingga pada target *error* tersebut yang memiliki performa JST lebih baik.

c. Curah Hujan

Dalam tabel 4.3 dapat dilihat perubahan performa tiap perubahan target *error* yang diberikan. Selain itu dapat dilihat pula iterasi yang dapat dicapai tiap target *error* yang diberikan dengan iterasi maksimum yang digunakan adalah 1000.

Dari tabel 4.3 di bawah ini menunjukkan bahwa pada prediksi curah hujan, variasi target *error* 0.001 mampu mencapai iterasi maksimum yang diberikan yaitu 1000. Pada target *error* 0.002 iterasi menurun saat *hidden node* 30 hingga 50. Dan pada target *error* 0.003 iterasi juga semakin turun.

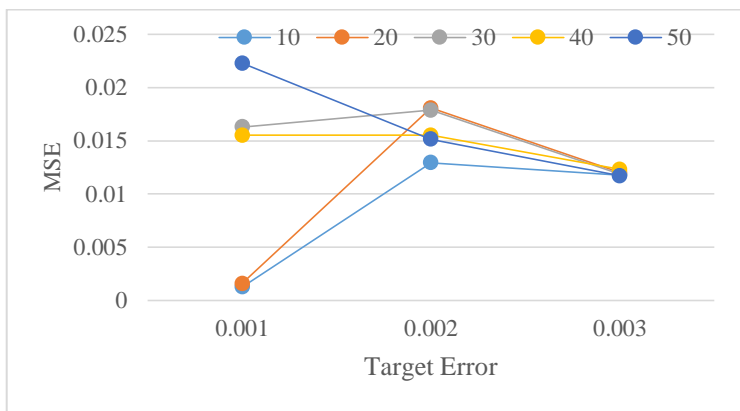
Tabel 4. 3 Hasil Prediksi Curah Hujan berdasarkan variasi target *error*

Target <i>error</i>	<i>Hidden node</i>	iterasi	RMSE
0.001	10	1000	0.001267
	20	1000	0.001595
	30	1000	0.016287
	40	1000	0.015519
	50	1000	0.022266
0.002	10	1000	0.012944
	20	1000	0.018062
	30	362	0.017875
	40	129	0.015513
	50	172	0.015144
0.003	10	6	0.011771
	20	5	0.012024
	30	4	0.011863
	40	1	0.012323
	50	4	0.011710

Dari sisi lain dapat dilihat bahwa tabel menunjukkan target *error* berpengaruh pada performa dari JST. Hal ini ditunjukkan dengan berubahnya nilai MSE setiap perubahan target *error*. Gambar 4.6 adalah grafik yang menunjukkan perubahan performa JST disetiap perubahan variasi target *error* yang diberikan.

Gambar 4.6 menunjukkan bahwa perubahan target *error* mempengaruhi nilai MSE yang dihasilkan. Hampir semua *hidden node*, nilai MSE mengalami kenaikan pada target *error* 0.002 dan

turun kembali pada target *error* 0.003. Namun tidak pada *hidden node* 50 yang mengalami penurunan hingga target *error* 0.003. Dalam grafik terlihat performa JST paling baik yaitu pada target *error* 0.001 karena nilai MSE pada target *error* tersebut bernilai kecil.



Gambar 4. 6 Hasil Prediksi Curah Hujan berdasarkan variasi *target error*

4.3 Pembahasan Arsitektur Terbaik dari Variasi Terbaik

Penelitian ini menggunakan variasi *hidden node* dan variasi target *error*. Kedua variasi terhadap tersebut menentukan performa JST terbaik. Hal tersebut dilihat dari nilai MSE setiap variasi. Tabel 4.4 menunjukkan rincian variasi terbaik dari tiap variabel yang diprediksi.

Tabel 4.4 menunjukkan bahwa dari setiap variasi target *error* yang digunakan target *error* 0.001 yang paling banyak memberikan performa JST terbaik hal tersebut dikarenakan dari tiga variabel yang diprediksi memiliki performa terbaik ketika target *error* 0.001. Performa terbaik ditunjukkan dengan nilai MSE dari jaringan yang lebih kecil dari variasi target *error* lainnya. Selain itu dari tabel 4.4 juga dapat dilihat bahwa variasi *hidden node* yang paling baik yaitu pada *hidden node* 10. Pada *hidden node* 10 menunjukkan nilai MSE yang rendah. Semakin besar nilai *hidden node* yang digunakan maka akan mengurangi performa

jaringan. Hal tersebut ditunjukkan dengan bertambahnya nilai MSE setiap bertambahnya *hidden node* yang digunakan.

Tabel 4. 4 Rincian arsitektur terbaik dari variasi terbaik

	Target Error	Hidden node	MSE
Kecepatan Angin	0.002	10	0.026385
Arah Angin	0.001	10	0.060009
Curah Hujan	0.001	10	0.01267

4.4 Hasil Prediksi Cuaca Menggunakan *Artificial Neural Network Lavenberg-Marquardt*

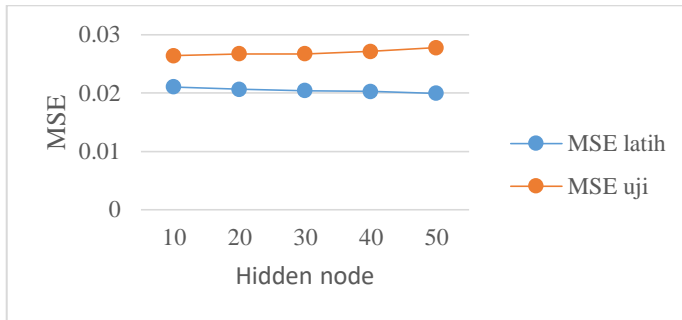
Berikut adalah hasil pelatihan dan pengujian prediksi cuaca menggunakan algoritma *Artificial Neural Network Lavenberg-Marquardt*

a. Kecepatan Angin

Input yang digunakan pada prediksi kecepatan angin adalah suhu dan arah angin. Variasi yang digunakan pada penelitian ini adalah variasi *target error* dan *hidden node*. Variasi tersebut digunakan untuk mencari performa terbaik dari sistem prediksi. Berdasarkan hasil pelatihan dan pengujian didapatkan performa terbaik yaitu pada *target error* 0.002 dan *hidden node* 10. Hal tersebut ditunjukkan dengan nilai MSE yang paling kecil dan berdasarkan keluaran jaringan yang selalu mengikuti target pada grafik pola pelatihan dan pola pengujian. Performa terbaik tersebut ditunjukkan dengan MSE uji sebesar 0.026385.

Gambar 4.7 menunjukkan perbandingan nilai MSE hasil pelatihan dan pengujian pada prediksi kecepatan angin. Gambar 4.7 menunjukkan bahwa nilai MSE uji terbaik bukan berasal dari hasil pelatihan dengan MSE yang kecil. Dari grafik dapat dilihat bahwa nilai MSE terkecil dari hasil pelatihan terletak pada *hidden node* 50. Namun pada *hidden node* 50 memiliki hasil uji dengan nilai MSE terbesar dari *hidden node* lainnya. Sehingga dari grafik

dapat terlihat bahwa pada prediksi kecepatan angin, variasi dengan performa uji terbaik yaitu pada *hidden node* 10.



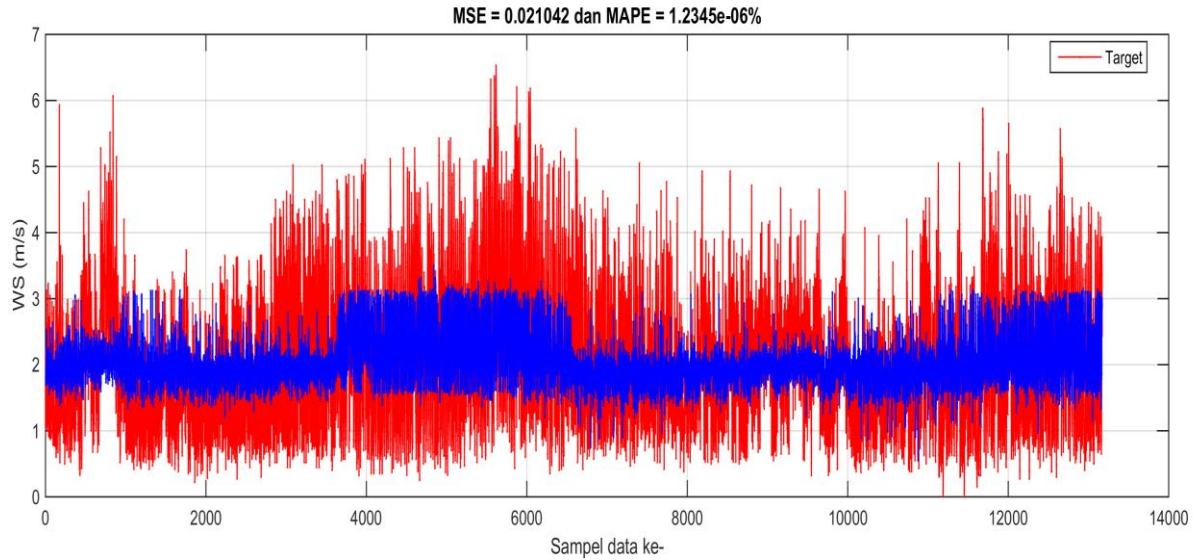
Gambar 4. 7 Perbandingan MSE uji dan MSE latih pada prediksi kecepatan angin

Grafik keluaran JST pada prediksi kecepatan angin dapat dilihat pada gambar 4.8. Grafik keluaran JST ditunjukkan dengan garis berwarna biru sedangkan target ditunjukkan dengan garis berwarna merah. Dari gambar 4.8 dapat dilihat bahwa garis keluaran JST masih mengikuti garis target.

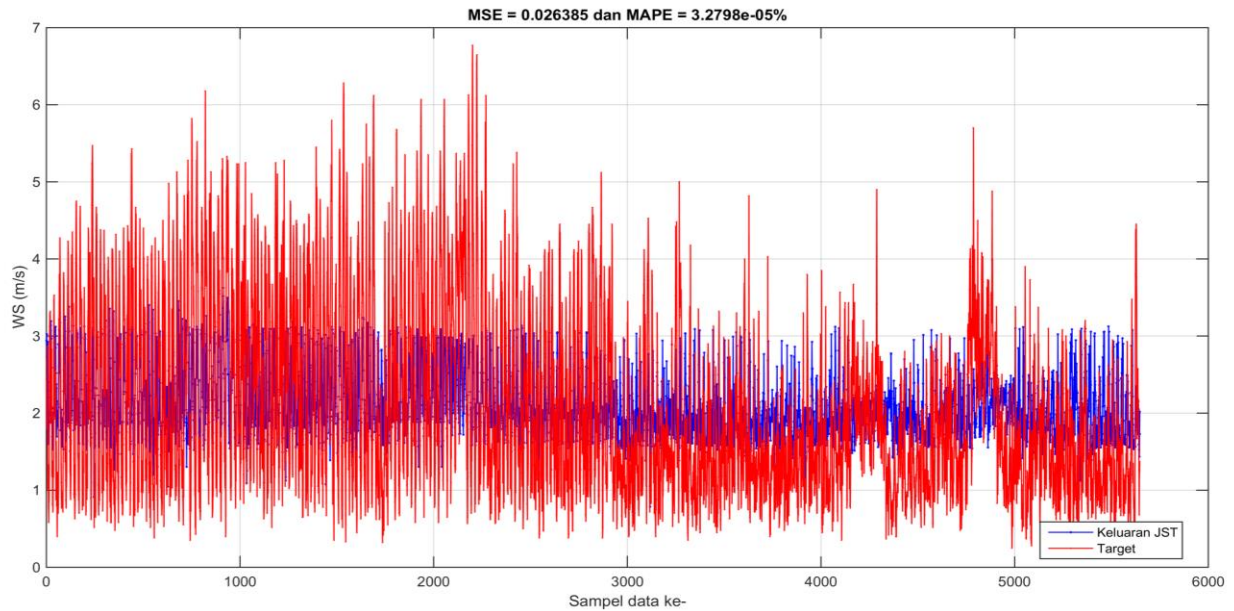
Dari grafik pelatihan JST diatas dapat dilihat bahwa garis keluaran JST masih mengikuti garis target meskipun garis keluaran JST terlihat lebih rendah dibandingkan dengan garis target. Performa pelatihan JST ditunjukkan dengan nilai MSE 0.021042, dan dengan hasil tersebut mampu menghasilkan peforma uji terbaik dengan hasil MSE 0.026385. Hasil tersebut merupakan hasil terbaik dari variasi *hidden node* lainnya pada target *error* 0.002.

Hasil uji tidak jauh berbeda dengan hasil latih. Garis keluaran JST lebih rendah dari garis target namun garis tersebut masih mengikuti garis target. Hasil MSE juga tidak memiliki selisih yang signifikan antara MSE latih dan MSE uji.

Gambar hasil pelatihan dan pengujian dapat dilihat pada gambar 4.8 dan 4.9



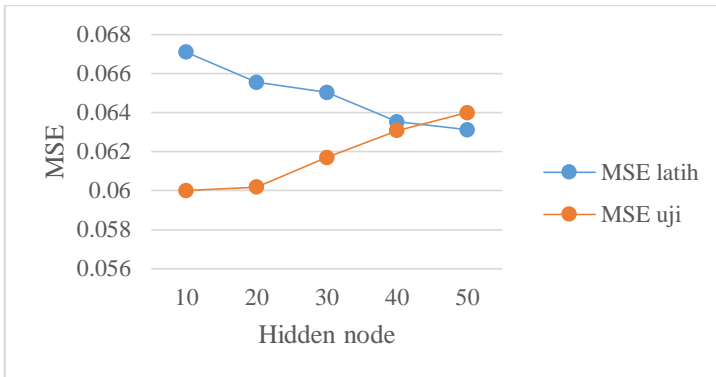
Gambar 4. 8 Grafik pelatihan JST kecepatan angin



Gambar 4.9 Hasil uji prediksi kecepatan angin

b. *Wind Direction* (Arah Angin)

Input yang digunakan pada prediksi arah angin adalah suhu dan kecepatan angin. Performa prediksi ditunjukkan dengan nilai MSE yaitu sebesar 0.060009. Performa tersebut didapatkan dari *hidden node* 10 dan target *error* 0.001. Sama halnya dengan kecepatan angin, prediksi arah angin juga memiliki performa terbaik tidak dari hasil latih dengan nilai MSE latih yang kecil. Nilai MSE latih kecil tidak memberikan hasil MSE uji yang kecil pula. Berikut grafik perbandingan MSE uji dan MSE latih pada prediksi arah angin



Gambar 4. 10 Perbandingan MSE uji dan MSE latih pada prediksi Arah Angin

Gambar 4.10 menunjukkan bahwa MSE uji terkecil tidak dari MSE latih yang paling kecil pula. Nilai MSE uji paling kecil didapatkan dari MSE latih sebesar 0.067098. Nilai MSE tersebut merupakan nilai MSE latih paling besar diantara variasi *hidden node* pada target *error* 0.001. Nilai MSE latih terkecil yaitu sebesar 0.063125. Namun pada MSE latih tersebut menghasilkan nilai MSE uji yang paling besar diantara MSE lainnya. Sehingga dalam penelitian ini untuk memprediksi arah angin arsitektur *hidden* terbaik adalah *hidden node* 10. Hal tersebut ditunjukkan dengan nilai MSE uji yang paling kecil diantara MSE uji lainnya pada variasi *hidden node*.

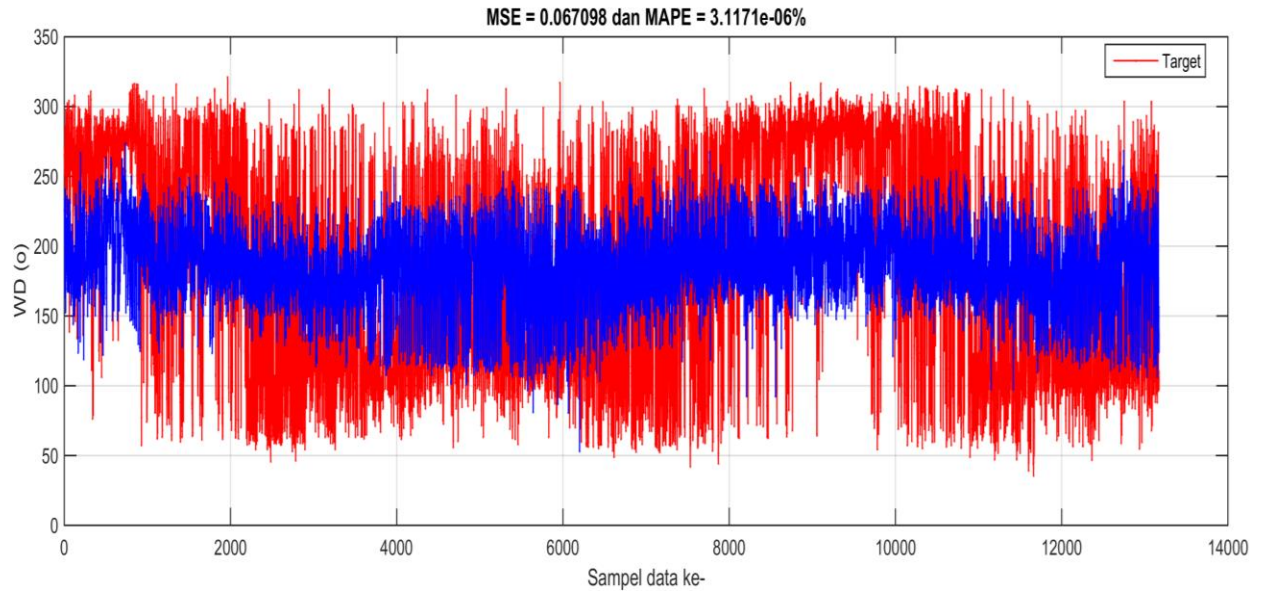
Gambar hasil pelatihan dan pengujian pada prediksi arah angin dapat dilihat pada gambar 4.11 dan gambar 4.12.

Gambar 4.11 adalah gambar yang menunjukkan grafik keluaran JST dan target dari hasil latih prediksi arah angin dengan arsitektur *hidden node* dan target *error* terbaik. Grafik keluaran JST ditunjukkan dengan garis berwarna biru sedangkan garis berwarna merah merupakan target. Dapat dilihat pada gambar 4.11 bahwa grafik keluaran JST masih mengikuti grafik keluaran target meskipun dapat dilihat bahwa grafik keluaran JST lebih rendah dibandingkan dengan grafik target. Gambar 4.11 menunjukkan bahwa MSE latih yang dihasilkan adalah sebesar 0.067098.

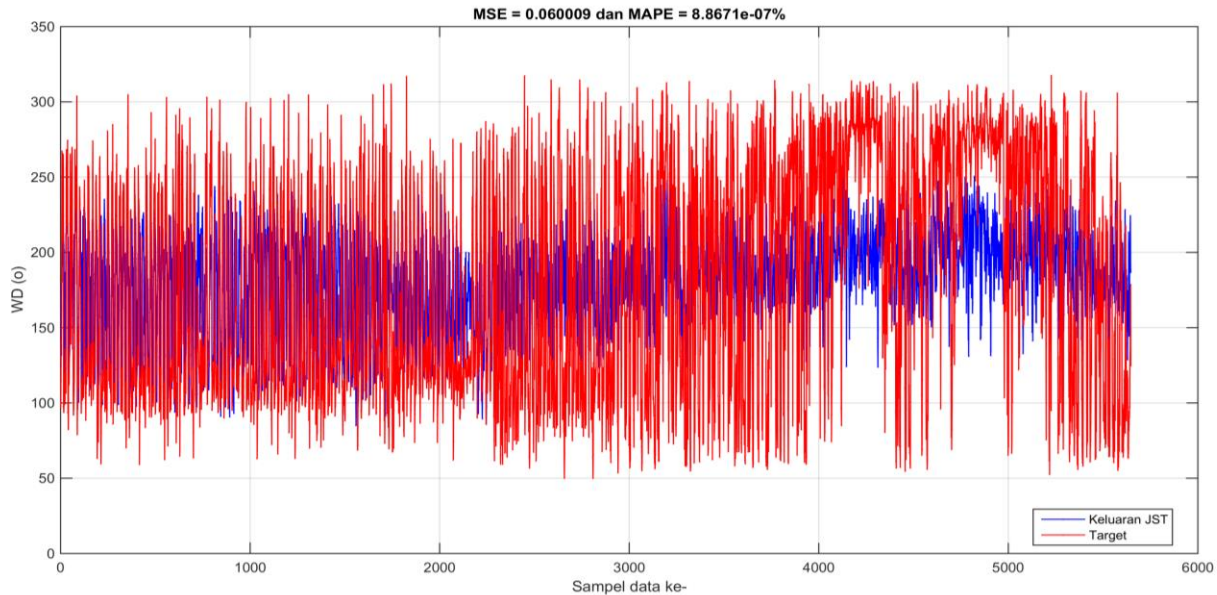
Dari hasil latih tersebut akan menghasilkan bobot yang akan digunakan dalam proses pengujian. Data yang digunakan pada proses pengujian adalah data diluar pelatihan. Sehingga dapat dilihat performa hasil pelatihan apakah masih bisa digunakan untuk prediksi arah angin dengan menggunakan data yang berbeda atau data selain data pelatihan.

Gambar 4.12 adalah gambar yang menunjukkan hasil uji prediksi arah angin. Sistem prediktor terbaik adalah sistem prediktor yang mampu menghasilkan MSE uji terkecil dari proses pelatihan.

Dari gambar 4.12 di atas terlihat bahwa nilai MSE yang dihasilkan dari hasil uji adalah 0.060009. Hasil performa tidak jauh berbeda dengan hasil uji meskipun masih memiliki selisih yang cukup signifikan namun dari grafik terlihat bahwa grafik keluaran JST mengikuti grafik target.



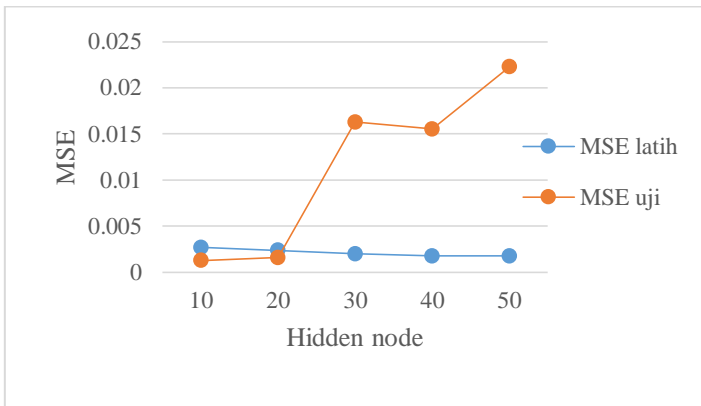
Gambar 4. 11 Grafik pelatihan JST pada prediksi arah angin



Gambar 4. 12 Hasil uji prediksi arah angin

c. Curah Hujan

Untuk prediksi curah hujan *input* yang digunakan adalah suhu, kelembaban dan kecepatan angin. Dari hasil pelatihan dan uji yang telah dilakukan, didapatkan hasil terbaik yaitu pada arsitektur *hidden node* 10 dan target *error* 0.001. Hasil performa ditunjukkan dengan nilai MSE yaitu sebesar 0.001267. Hasil MSE tersebut tidak didapatkan dari hasil latih dengan MSE yang kecil. Berikut merupakan perbandingan MSE uji dan MSE latih pada prediksi curah hujan



Gambar 4. 13 Perbandingan MSE uji dan MSE latih pada prediksi curah hujan

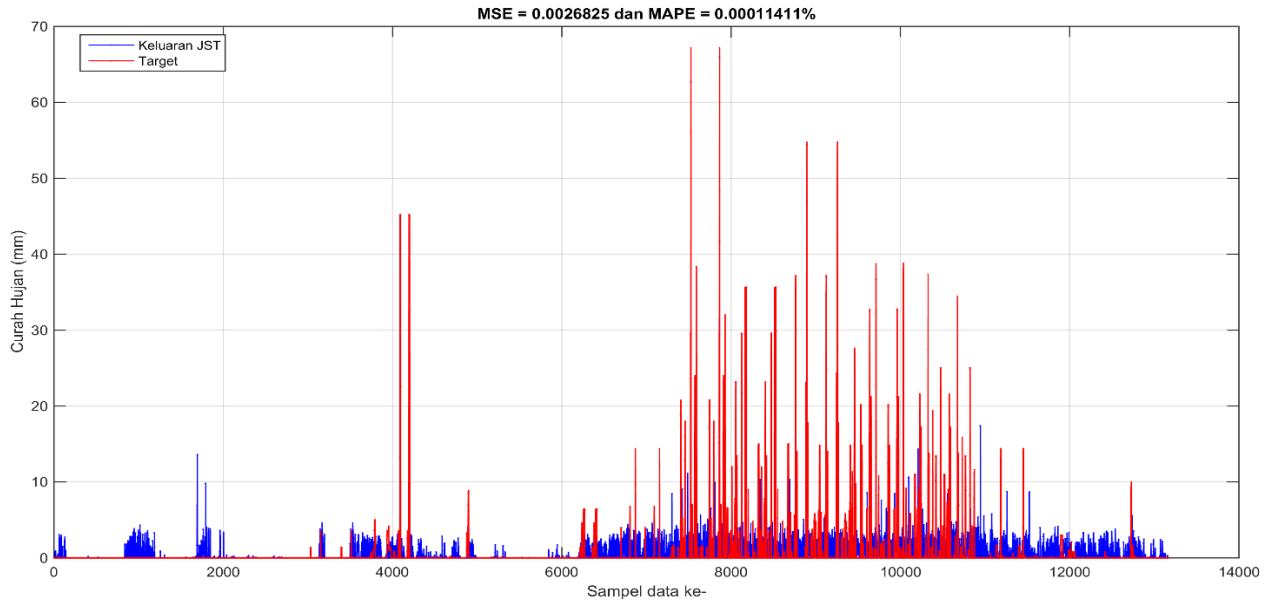
Gambar 4.13 menunjukkan perbandingan nilai MSE latih dan MSE uji pada prediksi curah hujan. Gambar 4.13 diatas menunjukkan bahwa MSE uji paling kecil dihasilkan pada *hidden node* 10. Namun pada *hidden node* tersebut memiliki nilai MSE latih paling besar diantara *hidden node* lainnya. Dalam gambar 4.13 dapat dilihat bahwa MSE uji paling kecil didapatkan dari MSE latih sebesar 0.0026825. Sedangkan dapat dilihat pada gambar 4.13 di atas menunjukkan bahwa hasil latih paling kecil terletak pada *hidden node* 50 namun pada hasil tersebut memiliki hasil uji yang paling besar. Maka dalam hal ini arstiktur *hidden node* terbaik

dalam prediksi curah hujan adalah *hidden node* 10 pada target *error* 0.001 dan dengan hasil MSE sebesar 0.001267

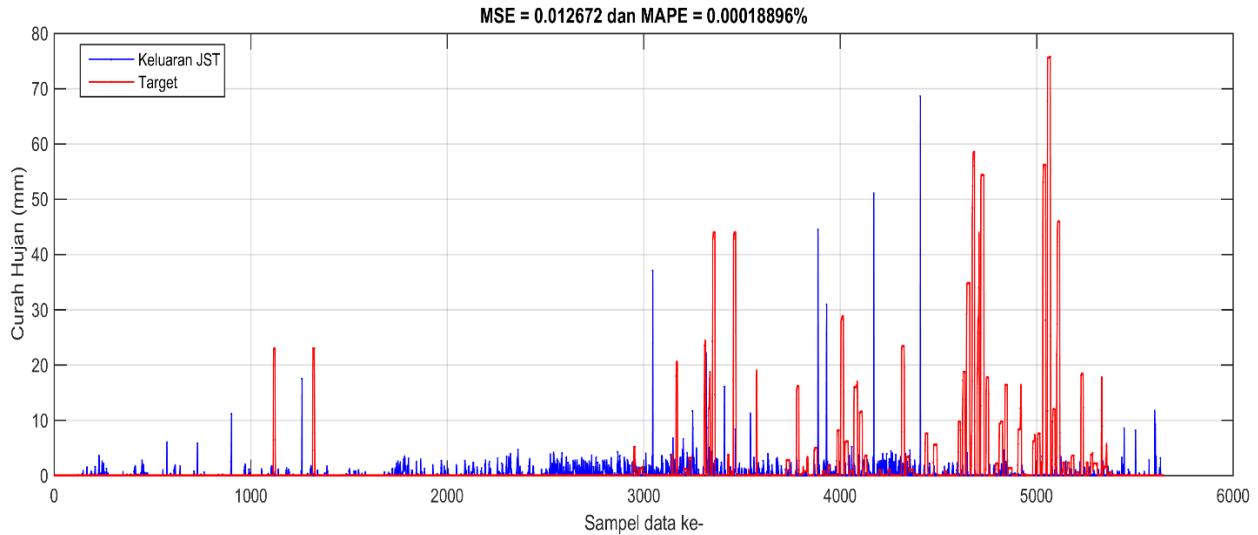
Gambar 4.14 dan gambar 4.16 merupakan gambar hasil pelatihan dan pengujian JST prediksi curah hujan dengan arsitektur *hiddden node* dan target *error*.

Gambar 4.14 menunjukkan bahwa grafik keluaran JST cukup mengikuti grafik target. Namun dapat terlihat jelas bahwa grafik keluaran JST lebih rendah dibandingkan dengan grafik target. Grafik keluaran JST ditunjukkan dengan garis berwarna biru sedangkan grafik target ditunjukkan dengan garis berwarna merah. Gambar 4.14 menunjukkan bahwa MSE hasil latih pada prediksi curah hujan yaitu sebesar 0.0026825

Gambar 4.15 menunjukkan hasil pengujian pada prediksi curah hujan. Gambar 4.15 tersebut menunjukkan bahwa gambar grafik uji lebih baik dibandingkan dengan grafik latih. Hal tersebut ditunjukkan dengan grafik keluaran JST hampir sama tinggi dengan grafik target. Gambar juga menunjukkan bahwa grafik keluaran JST mengikuti grafik terget sehingga dapat dikatakan baik dalam memprediksi karena tidak berbeda jauh dengan target. Nilai MSE pada prediksi cutrah hujan dari hasil uji terbaik yaitu sebesar 0.001267.



Gambar 4. 14 Hasil pelatihan prediksi curah hujan

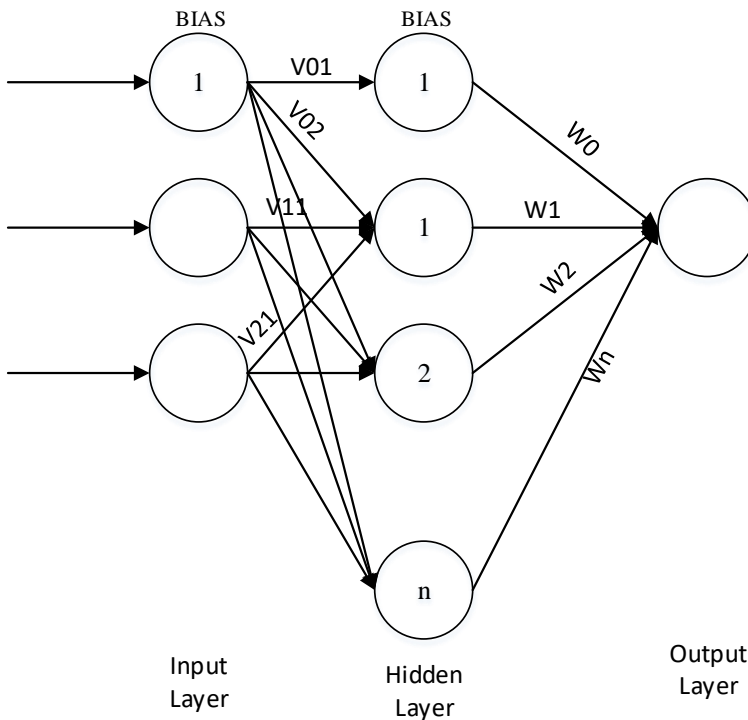


Gambar 4. 15 Hasil uji prediksi curah hujan

4.4 Validasi Prediksi menggunakan Data *Realtime*

Setelah dilakukan pelatihan dan pengujian menggunakan data dari BMKG selanjutnya adalah dilakukan validasi menggunakan data *realtime*. Pengambilan data *realtime* dilakukan menggunakan bantuan *Buoyweather* yang diletakkan di pantai Kenjeran. Hasil data *realtime* akan diolah dalam sistem prediksi dengan menggunakan arsitektur terbaik hasil pelatihan dan pengujian yang telah dilakukan. Dari hasil pelatihan dan pengujian akan didapatkan arsitektur terbaik dengan bobot terbaik. Dan nantinya bobot tersebut yang akan digunakan oleh sistem prediksi dalam mengolah data *realtime*.

Gambar 4.16 adalah gambar yang menunjukkan arsitektur JST beserta simbol bobotnya



Gambar 4. 16 Arsitektur JST

Bobot adalah koefisien yang terletak diantara *input layer* dan *hidden layer* serta terletak diantara *hidden layer* dan *output layer*. Gambar 4.16 adalah gambar yang menunjukkan arsitektur beserta bobot pada JST. Dapat dilihat pada gambar 4.16 bahwa terdapat nilai bobot yang menghubungkan tiap *layer*. Nilai bobot tersebut yang akan dijadikan koefisien pengali disetiap *inputan* yang masuk pada sistem prediktor. Dan nilai bobot tersebut akan berubah disetiap perubahan variasi yang digunakan.

Dari hasil pelatihan dan dan pengujian akan didapatkan arsitektur terbaik yang ditunjukkan dengan performansi JST atau nilai MSE pada jaringan. Dan di setiap arsitektur terbaik tersebut akan ada bobot terbaik yang dihasilkan dari proses pelatihan dan pengujian. Gambar 4.16 menjelaskan bahwa nilai bobot *input layer* ke *hidden layer* disimbolkan dengan V sedangkan simbol bobot *hidden layer* ke *output layer* adalah W. Berikut keterangan untuk nilai bobot yang digunakan dengan nilai n adalah jumlah *hidden node* yang digunakan.

$V_{01}-V_{0n}$ = bobot bias *input* ke *hidden layer*

$V_{11}-V_{1n}$ = bobot *input* pertama ke *hidden layer*

$V_{21}-V_{2n}$ = bobot *input* kedua ke *hidden layer*

W_0 = bobot bias *hidden layer* ke *output layer*

W_1-W_n = bobot *hidden node* ke *output*

Hasil validasi realtime masing-masing variabel akan dijelaskan dalam sub bab dibawah ini

a. Kecepatan Angin

Nilai bobot terbaik akan didapatkan dari arsitektur terbaik hasil pelatihan dan pengujian yang telah dilakukan. Arsitektur terbaik pada prediksi kecepatan angin yaitu pada *hidden node* 10 dan pada target *error* 0.002.

Nilai bobot yang didapatkan dari proses pelatihan akan digunakan dalam proses pengujian. Bobot terbaik terdapat pada arsitektur terbaik hasil dari pengujian dengan nilai MSE terkecil. Tabel 4.5 adalah tabel yang menunjukkan nilai bobot terbaik hasil dari pelatihan dan pengujian. V_{01} hingga V_{010} menunjukkan nilai bobot dari bias *input* ke *hidden layer*. $V_{11}-V_{110}$ menunjukkan nilai bobot dari *input* pertama ke *hidden layer*. Sedangkan $V_{21}-$

V210 dan V31-V310 berturut-turut menunjukkan nilai bobot *input* kedua dan ketiga ke *hidden layer*.

Bobot bias *hidden layer* ke *ouput* ditunjukkan dengan W0. Sedangkan W1-W10 menunjukkan nilai bobot dari *hidden node* ke *output*.

Tabel 4. 5 Bobot terbaik hasil pelatihan prediksi kecepatan angin

Bobot	Nilai	Bobot	Nilai
V01	71.094	V21	-37.207
V02	-16.175	V22	10.040
V03	-5.753	V23	0.224
V04	-5.752	V24	0.214
V05	8.199	V25	26.856
V06	16.481	V26	2.616
V07	-1.686	V27	-1.767
V08	-6.075	V28	6.091
V09	9.811	V29	28.690
V010	6.089	V210	-6.026
V11	-48.439	V31	-4.094
V12	20.006	V32	-43.516
V13	5.396	V33	13.212
V14	5.373	V34	13.306
V15	-32.254	V35	-9.801
V16	-34.338	V36	2.424
V17	19.169	V37	10.434
V18	20.992	V38	-5.250
V19	12.490	V39	3.241
V110	-20.891	V310	5.146
W0	271.904	W6	-0.817
W1	-0.457	W7	-1.185
W2	0.712	W8	-271.030
W3	314.197	W9	-0.635
W4	313.309	W10	271.882
W5	0.523		

Tabel 4.5 menunjukkan nilai bobot terbaik dari hasil pelatihan dan pengujian prediksi kecepatan angin. Nilai bobot tersebut

didapatkan dari arsitektur terbaik dalam memprediksi kecepatan angin. Dan nilai bobot tersebut yang akan digunakan untuk validasi dalam memprediksi kecepatan angin. Validasi prediksi dilakukan menggunakan data dari hasil pengambilan *realtime* di pantai Kenjeran. Hasil dari validasi menggunakan data *realtime* tersebut sama dengan hasil dari pelatihan dan pengujian yaitu berupa grafik dan disertai performa prediksi yang ditunjukkan dengan nilai MSE.

Gambar 4.17 adalah gambar yang menunjukkan hasil validasi *realtime* prediksi kecepatan angin. Gambar 4.17 dapat dilihat pada halaman 93. Dari gambar tersebut dapat terlihat bahwa index atau jumlah data yang dimasukkan dalam sistem prediksi adalah 12. Sumbu x menunjukkan waktu (jam ke-) sedangkan sumbu y menunjukkan kecepatan angin dalam satuan m/s. Grafik keluaran JST ditunjukkan pada garis berwarna biru sedangkan grafik target ditunjukkan dengan warna merah. Dari gambar dapat terlihat bahwa keluaran JST dan target memiliki nilai atau titik awal yang berbeda. Dalam gambar terlihat bahwa keluaran jst menunjukkan titik awal 2 sedangkan pada grafik target menunjukkan titik awal 4.9. Namun pola grafik antara keluaran jst dan target memiliki pola yang sama. Dari gambar juga dapat terlihat nilai MSE yaitu sebesar 1.463.

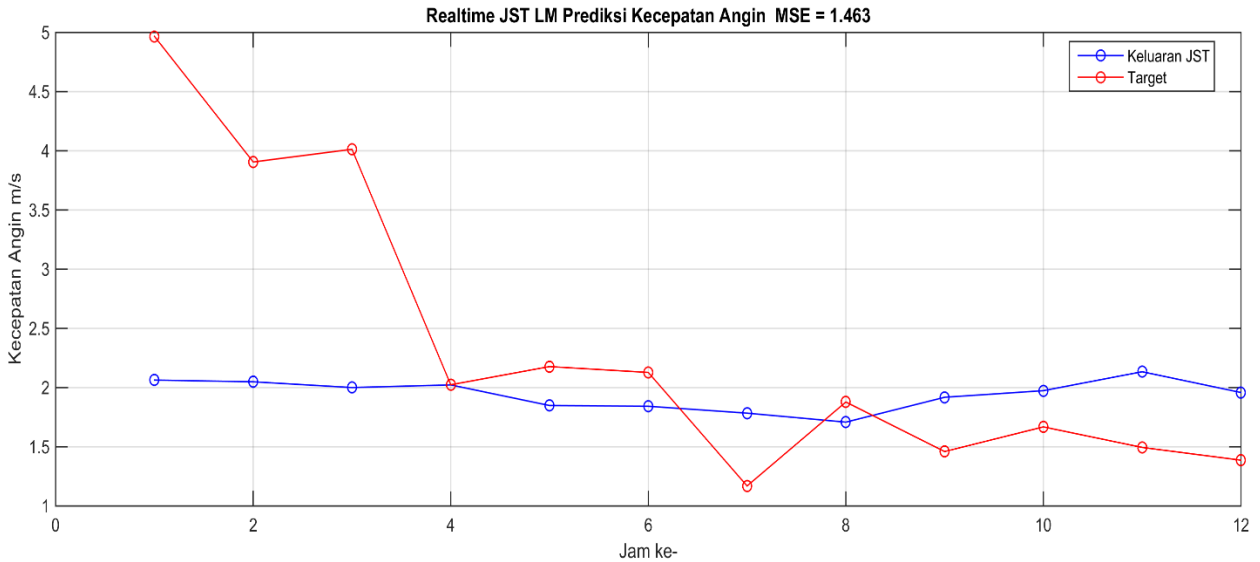
Tabel 4.6 merupakan tabel yang menunjukkan hasil pengujian *realtime* prediksi kecepatan angin. Tabel 4.6 tersebut merupakan tabel yang menunjukkan hasil pengukuran langsung oleh sensor di titik pengukuran pantai Kenjeran. Hasil pengukuran sensor yang telah ditransmisikan akan diolah sebagai *input* prediktor. Untuk prediksi kecepatan angin, *input* yang digunakan adalah suhu dan arah angin.

Hasil prediksi yang telah dilakukan menunjukkan bahwa terdapat sedikit perbedaan hasil antara kecepatan angin hasil pengukuran dan kecepatan angin hasil prediksi. Hal ini dapat dilihat dalam tabel 4.6 bahwa pada hasil pengukuran kecepatan angin dengan *input* arah angin sebesar 45° dan suhu sebesar 32.50°C yaitu 4.96 m/s sedangkan pada hasil prediksi yaitu sebesar 2.063 m/s. Terdapat nenerapa faktor yang mempengaruhi ketidaksesuaian hasil pengukuran dan hasil prediksi . Salah satunya adalah

pengaruh akurasi dari kecepatan angin dalam mengukur data sehingga data yang terukur tidak sesuai dengan kondisi yang sebenarnya.

Tabel 4. 6 Hasil Pengujian *Realtime* prediksi kecepatan Angin

Arah Angin	Suhu	Kecepatan Angin (Hasil Pengukuran)	Kecepatan angin (Hasil Prediksi)
45	32.50	4.96	2,063
44.25	32.55	3.90	2,048
66.78	33.20	4.01	2,000
99.21	33.73	2.02	2,022
78.71	34.31	2.17	1,848
113.75	35.27	2.12	1,842
114.03	35.80	1.16	1,784
59.56	34.8	1.87	1,708
96.59	34.29	1.45	1,918
76.812	33.54	1.66	1,973
105.40	33.28	1.49	2,133
52.59	33.11	1.38	1,958



Gambar 4. 17 Hasil validasi realtime prediksi kecepatan angin

b. Arah Angin

Sama halnya dengan kecepatan angin, dalam memprediksi arah angin juga dilakukan dalam tahap pelatihan dan pengujian. Setelah pengujian dilakukan tahap validasi. Dari tahap pelatihan akan didapatkan bobot terbaik dari arsitektur terbaik. Bobot tersebut yang akan digunakan untuk validasi sistem prediksi. Tabel 4.7 menunjukkan bobot terbaik dari hasil pelatihan dan pengujian prediksi arah angin. Bobot tersebut didapatkan dari arsitektur terbaik JST. Dalam prediksi arah angin arsitektur terbaik JST adalah *hidden node* 10 dan pada target *error* 0.001.

Tabel 4. 7 Bobot terbaik hasil pelatihan prediksi arah angin

Bobot	Nilai	Bobot	Nilai	Bobot	Nilai
V01	4.164	V21	0.861	V41	2.975
V02	1.136	V22	0.460	V42	-5.0161
V03	-1.941	V23	-0.279	V43	5.297
V04	6.030	V24	1.223	V44	0.825
V05	5.436	V25	1.157	V45	1.832
V06	8.746	V26	3.190	V46	2.156
V07	-588.196	V27	136.734	V47	992.20
V08	-20.535	V28	0.551	V48	4.559
V09	0.375	V29	0.656	V49	-4.716
V010	10.538	V210	-31.529	V410	-1.536
V11	-4.652	V31	-2.063	W0	150.73
V12	-1.684	V32	7.112	W1	-147.67
V13	2.117	V33	-6.130	W2	-270.46
V14	-5.151	V34	2.881	W3	-135.42
V15	5.141	V35	2.735	W4	228.27
V16	-11.690	V36	-7.419	W5	354.19
V17	146.914	V37	110.192	W6	3.985
V18	7.394	V38	20.366	W7	-0.448
V19	-1.346	V39	8.148	W8	1.383
V110	6.327	V310	-6.881	W9	136.53
				W10	0.573

V01-V010 pada tabel 4.7 menunjukkan nilai boot pada bias *input* ke *hidden layer*. V11-V110, V21-V210, V31-V310 dan V41-V410 menunjukkan nilai bobot dari *input* ke *hidden layer*. Prediksi arah angin terdapat 4 *input*, maka dari itu terdapat 4 jenis nilai bobot dari *input* ke *hidden layer*. Nilai bobot dari *hidden layer* ke *output* ditunjukkan dengan simbol W.

Validasi prediktor dilakukan menggunakan data *realtime*. Data *realtime* merupakan data hasil pengukuran langsung oleh sensor dan data tersebut nantinya akan diolah prediktor dalam memprediksi variabel. Data pengukuran langsung tersebut akan ditransmisikan dan disimpan oleh database. Data tersebut yang akan diambil prediktor sebagai validasi. Gambar 4.18 merupakan gambar yang menunjukkan hasil validasi menggunakan data *realtime* pada prediksi arah angin. Gambar 4.18 dapat dilihat pada halaman 97.

Gambar 4,18 menunjukkan bahwa index data yang diambil untuk diproses adalah 12. Grafik keluaran JST ditunjukkan dengan warna biru sedangkan grafik target ditunjukkan dengan garis warna merah. Pola grafik keluaran JST dan grafik terget terlihat berbeda. Pada grafik keluaran JST memiliki titik awal 85 sedangkan pada grafik target memiliki titik awal 45.

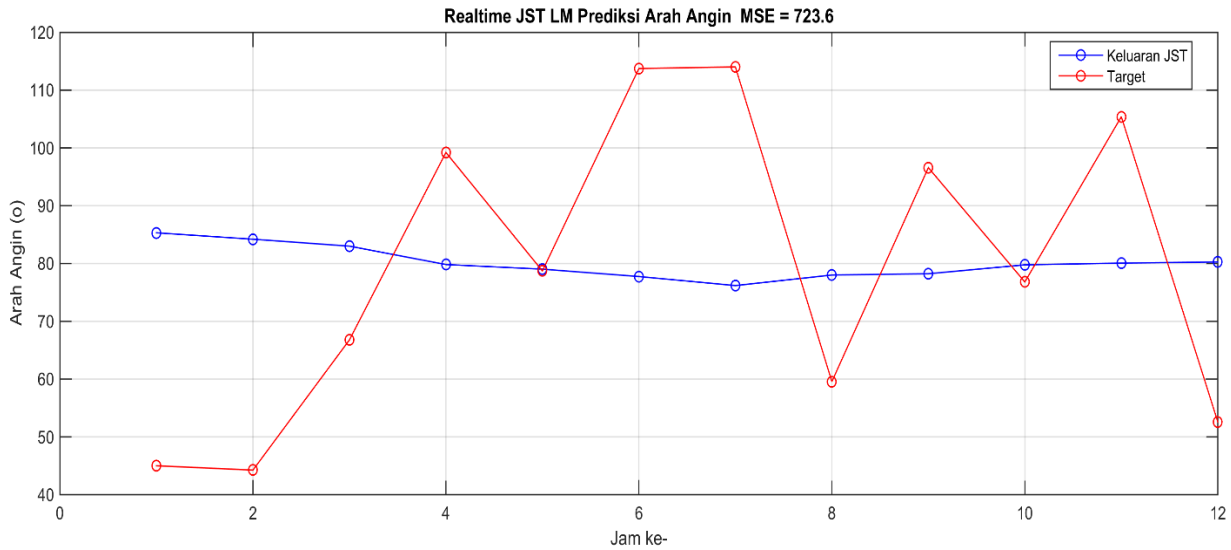
Dari gambar 4.17 dapat terlihat bahwa performa JST ditunjukkan dengan nilai MSE sebesar 723.6. Nilai *error* tersebut cukup besar disebabkan karena perbedaan antara hasil pengukuran dan hasil prediksi masih terlihat sangat jauh berbeda.

Tabel 4.8 adalah tabel yang menunjukkan hasil pengujian *Realtime* prediksi arah angin. Dari tabel 4.8 dapat dilihat hasil pengujian data *realtime* dari prediksi arah angin.

Hasil pengukuran sensor yang telah ditransmisikan akan diolah sebagai *input* prediktor. Untuk prediksi arah angin, *input* yang digunakan adalah suhu dan kecepatan angin. Dari hasil prediksi yang telah dilakukan terdapat perbedaan hasil antara arah angin hasil pengukuran dan arah angin hasil prediksi. Hal tersebut dipengaruhi oleh banyak hal. Salah satunya adalah pengaruh akurasi dari sensor arah angin dalam mengukur data sehingga data yang terukur tidak sesuai dengan kondisi yang sebenarnya.

Tabel 4. 8 Hasil Pengujian *Realtime* prediksi Arah Angin

Kecepatan Angin	Suhu	Arah Angin (Hasil Pengukuran)	Arah angin (Hasil Prediksi)
4.96	32.50	45	85,314
3.90	32.55	44.25	84,211
4.01	33.20	66.78	83,009
2.02	33.73	99.21	79,824
2.17	34.31	78.71	79,028
2.12	35.27	113.75	77,749
1.16	35.80	114.03	76,187
1.87	34.8	59.56	78,009
1.45	34.29	96.59	78,234
1.66	33.54	76.812	79,775
1.49	33.28	105.40	80,071
1.38	33.11	52.59	80,268



Gambar 4. 18 Hasil validasi realtime prediksi arah angin

c. Curah Hujan

Tabel 4.9 dibawah ini adalah tabel yang menunjukkan besarnya bobot yang digunakan pada prediksi curah hujan

Tabel 4. 9 Bobot terbaik hasil pelatihan prediksi curah hujan

Bobot	Nilai	Bobot	Nilai
V01	22.273	V21	-9.27
V02	112.67	V22	-281.75
V03	-1254.61	V23	1107.12
V04	-75.56	V24	212.35
V05	1.81	V25	5.60
V06	403.56	V26	-736.00
V07	-862.72	V27	758.28
V08	19.77	V28	-46.67
V09	5.99	V29	2.68
V010	473.82	V210	-588.84
V11	18.68	V31	0.10
V12	183.82	V32	-529.84
V13	394.18	V33	151.20
V14	-157.46	V34	443.15
V15	-5.49	V35	0.24
V16	517.72	V36	-2396.39
V17	272.20	V37	106.00
V18	41.40	V38	31.41
V19	-8.44	V39	0.280
V110	102.61	V310	238.579
W0	1037.42	W6	-73.99
W1	87.76	W7	-893.34
W2	-1245.5	W8	792.073
W3	291.53	W9	-397.01
W4	-1248.50	W10	-615.75
W5	341.20		

Hasil dari pelatihan dan pengujian adalah arsitektur terbaik dengan bobot terbaik. Bobot terbaik yang ada dalam arsitektur terbaik akan digunakan dalam proses validasi pada data realtime. Tabel 4.9 di atas adalah tabel yang menunjukkan nilainya bobot yang digunakan pada prediksi curah hujan. V01 hingga V010 pada tabel 4.9 tersebut menunjukkan nilainya bobot dari bias *input* ke

hidden layer. Sedangkan V11-V110, V21-V210 dan V31-V310 merupakan nilai bobot dari *input* ke *hidden layer*. Prediksi curah hujan terdapat tiga *input* sehingga terdapat tiga jenis bobot dalam arsitekturnya. Nilai bobot dari *hidden layer* ke *output* ditunjukkan dengan simbol W seperti yang dapat dilihat pada tabel 4.9.

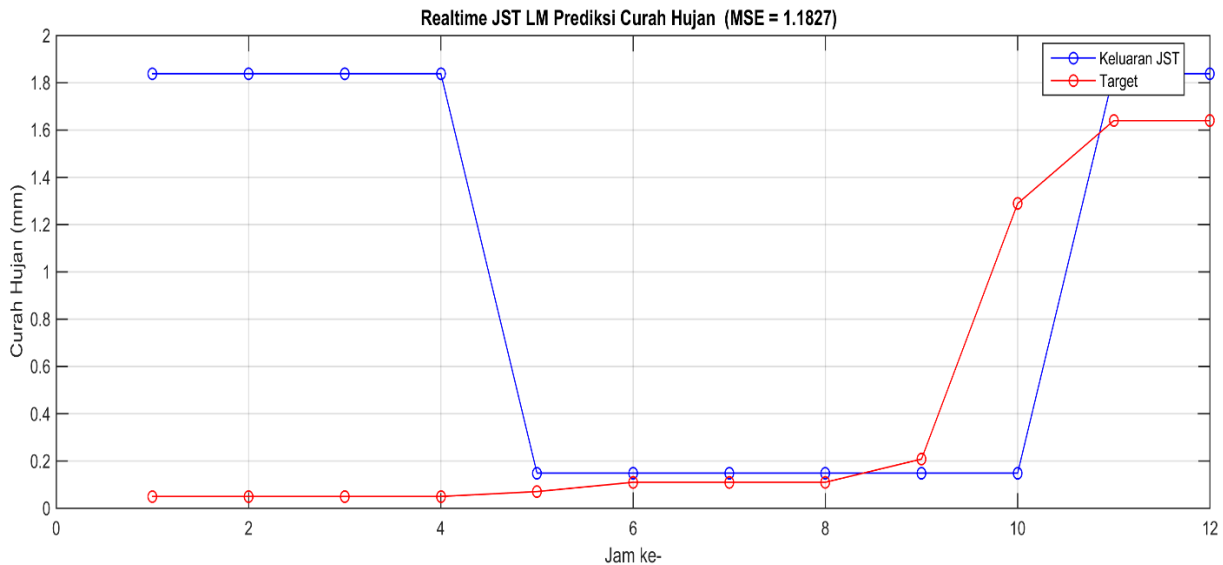
Hasil validasi *realtime* prediksi curah hujan ditunjukkan pada gambar 4.19. gambar 4.19 dapat dilihat pada halaman 101. Gambar 4.19 menunjukkan bahwa index data yang diambil dari data *realtime* adalah 12. Grafik keluaran JST ditunjukkan dengan garis berwarna biru sedangkan grafik target ditunjukkan dengan garis berwarna merah. Gambar 4.19 menunjukkan bahwa grafik keluaran JST memiliki titik awal 1.83 sedangkan grafik target memiliki titik awal 0.05. Namun dapat dilihat bahwa grafik keluaran jst dan grafik target hampir memiliki pola yang sama meskipun grafik tersebut memiliki titik awal yang berbeda. Selain itu pada gambar juga dapat dilihat hasil MSE yang dihasilkan adalah sebesar 1.1827.

Hasil validasi *realtime* prediksi curah hujan juga dapat dilihat dalam bentuk tabel. Tabel 4.10 adalah tabel yang menunjukkan hasil validasi menggunakan data *realtime*. Dari tabel 4.10 dapat dilihat hasil pengujian data *realtime* dari prediksi curah hujan. Tabel 4.10 merupakan hasil pengukuran langsung oleh sensor di titik pengukuran pantai Kenjeran. Hasil pengukuran sensor yang telah ditransmisikan akan diolah sebagai *input* prediktor. Untuk prediksi curah hujan, *input* yang digunakan adalah suhu, kelembaban dan kecepatan angin.

Dari hasil prediksi yang telah dilakukan terdapat perbedaan hasil antara curah hujan hasil pengukuran dan curah hujan hasil prediksi. Namun dapat dilihat meski terdapat perbedaan masih terdapat beberapa yang memiliki hasil yang sama. Hal ini dapat dilihat saat hasil pengukuran curah hujan menunjukkan 1.640 pada *input* kecepatan angin sebesar 1.387 m/s, kelembaban sebesar 63.125% dan suhu sebesar 32.506°C sedangkan untuk hasil prediksi curah hujan menunjukkan 1.83875.

Tabel 4. 10 Hasil Pengujian *Realtime* prediksi Curah Hujan

Kecepatan angin	Kelembaban	Suhu	Curah Hujan (Hasil Pengukuran)	Curah Hujan (Hasil Prediksi)
4.967	63.125	32.506	0.050	1.83875
3.906	62.844	32.553	0.050	1.83875
4.013	60.094	33.203	0.050	1.83875
2.024	57.719	33.738	0.050	1.83875
2.178	55.375	34.319	0.071	0.14875
2.128	51.563	35.275	0.110	0.14875
1.169	49.594	35.809	0.110	0.14875
1.878	52.469	34.800	0.110	0.14875
1.460	54.813	34.297	0.208	0.14875
1.668	57.281	33.541	1.289	0.14875
1.495	59.094	33.281	1.640	1.83875
1.387	59.656	33.119	1.640	1.83875



Gambar 4. 19 Hasil validasi realtime prediksi curah hujan

Halaman sengaja dikosongkan

BAB V

Penutup

5.1 Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan bahwa :

1. Prediksi kecepatan angin arsitektur terbaik menghasilkan nilai MSE 0.026385, prediksi arah angin menghasilkan nilai MSE 0.060009 dan pada prediksi curah hujan memiliki MSE 0.001267.
2. Arsitektur terbaik pada prediksi kecepatan angin didapatkan dari *hidden node* 10 dan target *error* 0.002, prediksi arah angin didapatkan dari *hidden node* 10 dengan target *error* 0.001. Dan pada prediksi curah hujan arsitektur terbaik adalah pada *hidden node* 10 dan target *error* 0.001.
3. Hasil validasi *realtime* menghasilkan MSE 1.463 untuk kecepatan angin. Pada prediksi arah angin menghasilkan MSE sebesar 723.6 sedangkan pada prediksi curah hujan hasil validasi *realtime* menghasilkan nilai MSE sebesar 1.1827 dan

5.2 Saran

Saran yang dapat diberikan penulis berdasarkan penelitian yang telah dilakukan adalah sebagai berikut :

1. Penambahan variasi untuk parameter-parameter didalam penggunaan metode JST-LM agar mendapatkan hasil yang lebih optimal
2. Pengambilan data *realtime* dilakukan jauh-jauh hari agar bisa mendapatkan data yang lebih banyak serta pengumpulan data dari beberapa tahun terakhir.

Halaman sengaja dikosongkan

Daftar Rujukan

- Abbot, J., & Marohasy, J. (2017a). Forecasting extreme monthly rainfall events in regions of Queensland, Australia using artificial neural networks. *International Journal of Sustainable Development and Planning*. <https://doi.org/10.2495/SDP-V12-N7-1117-1131>
- Abbot, J., & Marohasy, J. (2017b). The application of machine learning for evaluating anthropogenic versus natural climate change. *GeoResJ*. <https://doi.org/10.1016/j.grj.2017.08.001>
- Aisjah, A. S., & Arifin, S. (2011). Maritime weather prediction using fuzzy logic in java sea. In *Proceedings of 2011 2nd International Conference on Instrumentation Control and Automation, ICA 2011*. <https://doi.org/10.1109/ICA.2011.6130157>
- Ali, D., Yohanna, M., Puwu, M. I., & Garkida, B. M. (2016). Long-term load forecast modelling using a fuzzy logic approach. *Pacific Science Review A: Natural Science and Engineering*. <https://doi.org/10.1016/j.psra.2016.09.011>
- Aliiev, R. A., & Guirimov, B. G. (2014). *Type-2 fuzzy neural networks and their applications. Type-2 Fuzzy Neural Networks and their Applications*. <https://doi.org/10.1007/978-3-319-09072-6>
- Andrian, Y., & Ningsih, E. (2014a). Prediksi Curah Hujan Di Kota Medan Menggunakan. *Seminar Nasional Informatika*. <https://doi.org/10.1126/science.165.3897.1025>
- Andrian, Y., & Ningsih, E. (2014b). Prediksi Curah Hujan Di Kota Medan Menggunakan. *Seminar Nasional Informatika*.
- Balmat, J. F., Lafont, F., Maifret, R., & Pessel, N. (2011). A decision-making system to maritime risk assessment. *Ocean Engineering*. <https://doi.org/10.1016/j.oceaneng.2010.10.012>
- BMKG. (2016). Tren Suhu. <https://doi.org/10.1016/j.biocon.2017.06.014>
- Chen, L., & Nayak, R. (2011). Social network analysis of an online dating network. In *Proceedings of the 5th International*

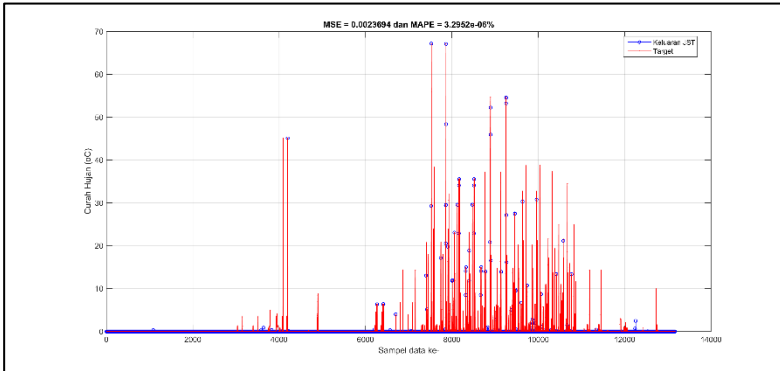
- Conference on Communities and Technologies - C&T '11.*
<https://doi.org/10.1145/2103354.2103361>
- De Girolamo, P., Di Risio, M., Beltrami, G. M., Bellotti, G., & Pasquali, D. (2017). The use of wave forecasts for maritime activities safety assessment. *Applied Ocean Research.*
<https://doi.org/10.1016/j.apor.2016.11.006>
- Evans, P. (1996). Government action, social capital and development: Reviewing the evidence on synergy. *World Development.* [https://doi.org/10.1016/0305-750X\(96\)00021-6](https://doi.org/10.1016/0305-750X(96)00021-6)
- Feng, Q. Y., Vasile, R., Segond, M., Gozolchiani, A., Wang, Y., Abel, M., ... Dijkstra, H. A. (2016). ClimateLearn: A machine-learning approach for climate prediction using network measures. *Geoscientific Model Development Discussions.* <https://doi.org/10.5194/gmd-2015-273>
- Gaxiola, F., Melin, P., Valdez, F., & Castillo, O. (2014). Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction. *Information Sciences.*
<https://doi.org/10.1016/j.ins.2013.11.006>
- Guo, S. J., Guan, S.-U., Yang, S., Li, W. F., Zhao, L. F., & Song, J. H. (2013). Input Partitioning Based on Correlation for Neural Network Learning. *Journal of Clean Energy Technologies.* <https://doi.org/10.7763/jocet.2013.v1.76>
- Kamel, K., Mahmoud, T., Le Bissonnais, Y., & Mahmoud, T. (2014). Assessment of the artificial neural networks to geomorphic modelling of sediment yield for ungauged catchments, Algeria. *Journal of Urban and Environmental Engineering.* <https://doi.org/10.4090/juee.2014.v8n2.175185>
- Klimberg, R. K., Sillup, G. P., Boyle, K. J., & Tavva, V. (2010). Forecasting performance measures - What are their practical meaning? *Advances in Business and Management Forecasting.* [https://doi.org/10.1108/S1477-4070\(2010\)0000007012](https://doi.org/10.1108/S1477-4070(2010)0000007012)
- Kodoatie, R. J., & Sjarief, R. (2010). *Tata Ruang Air. Andi Yogyakarta.* <https://doi.org/10.1111/nicc.12152>

- Makarynsky, O. (2004). Improving wave predictions with artificial neural networks. *Ocean Engineering*. <https://doi.org/10.1016/j.oceaneng.2003.05.003>
- Mujiasih, S. (2011). Utilization Of Data Mining For Weather Forecasting`. *Jurnal Meteorologi Dan Geofisika*.
- Naik, A. R., & Pathan, K. (2012). Weather Classification and Forecasting using Back Propagation Feed-forward Neural Network. *International Journal of Scientific and Research Publications*. <https://doi.org/10.1.1.301.836>
- Nikentari, N., Ritha, N., & Wati, L. (2018). Tide level forecast using grammatical evolution. In *Proceedings - 2017 International Conference on Sustainable Information Engineering and Technology, SIET 2017*. <https://doi.org/10.1109/SIET.2017.8304185>
- Paulescu, M., Paulescu, E., Gravila, P., & Badescu, V. (2013). Weather Modeling and Forecasting of PV Systems Operation. *Green Energy and Technology*. <https://doi.org/10.1007/978-1-4471-4649-0>
- Pratama, I., Permanasari, A. E., Ardiyanto, I., & Indrayani, R. (2017). A review of missing values handling methods on time-series data. In *2016 International Conference on Information Technology Systems and Innovation, ICITSI 2016 - Proceedings*. <https://doi.org/10.1109/ICITSI.2016.7858189>
- Puspita, E. S., & Yulianti, L. (2016). PERANCANGAN SISTEM PERAMALAM CUACA BERBASIS LOGIKA FUZZY. *Media Infotama*.
- Qiu, B. (2008). Kuroshio and Oyashio Currents. In *Encyclopedia of Ocean Sciences: Second Edition*. <https://doi.org/10.1016/B978-012374473-9.00350-7>
- Rahmat, R., Setiawan, R., & Purnomo, M. H. (2006). Perbandingan Algoritma Levenberg-Marquardt dengan Metoda Backpropagation pada Proses Learning Jaringan Saraf Tiruan untuk Pengenalan Pola Sinyal Elektrokardiograf. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Santhosh, T. V., Gopika, V., Ghosh, A. K., & Fernandes, B. G.

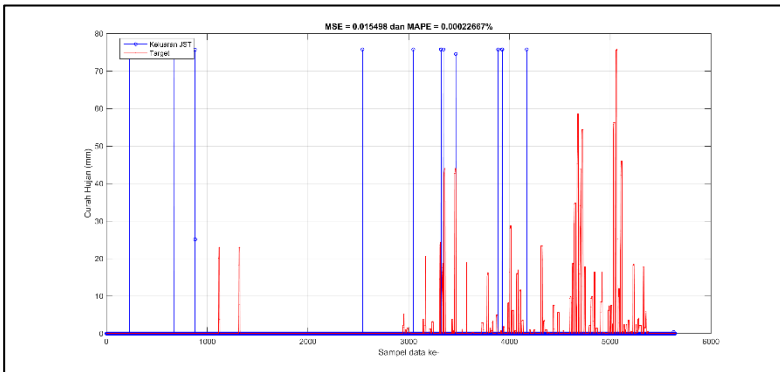
- (2018). An approach for reliability prediction of instrumentation & control cables by artificial neural networks and Weibull theory for probabilistic safety assessment of NPPs. *Reliability Engineering & System Safety*. <https://doi.org/10.1016/J.RESS.2017.10.010>
- Shaban, K., El-Hag, A., & Matveev, A. (2009). A cascade of artificial neural networks to predict transformers oil parameters. *IEEE Transactions on Dielectrics and Electrical Insulation*. <https://doi.org/10.1109/TDEI.2009.4815187>
- Sharma, B., & K. Venugopalan, P. (2014). Comparison of Neural Network Training Functions for Hematoma Classification in Brain CT Images. *IOSR Journal of Computer Engineering*. <https://doi.org/10.9790/0661-16123135>
- Sutojo, T., Mulyanto, E., & Suhartono, V. (2011). *Kecerdasaan Buatan. Journal of Chemical Information and Modeling*. <https://doi.org/http://dx.doi.org/10.1136/archdischild-2014-307384.1053>
- Suwardjo, D., Haluan, J., Jaya, I., & Poernomo, S. H. (2017). Keselamatan kapal penangkap ikan, tinjauan dari aspek regulasi nasional dan internasional. *Jurnal Teknologi Perikanan Dan Kelautan*. <https://doi.org/10.24319/jtpk.1.1-13>
- Voyant, C., Notton, G., Kalogirou, S., Nivet, M. L., Paoli, C., Motte, F., & Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*. <https://doi.org/10.1016/j.renene.2016.12.095>
- WMO. (2008). Guide to Hydrological Practices. Genoves. Word Meteorological Organization. *Hydrological Sciences Journal*. <https://doi.org/10.1080/02626667.2011.546602>
- Yang, Y., Chen, Y., Wang, Y., Li, C., & Li, L. (2016). Modelling a combined method based on ANFIS and neural network improved by DE algorithm: A case study for short-term electricity demand forecasting. *Applied Soft Computing Journal*. <https://doi.org/10.1016/j.asoc.2016.07.053>

LAMPIRAN A, VARIASI TARGET *ERROR* 0.001

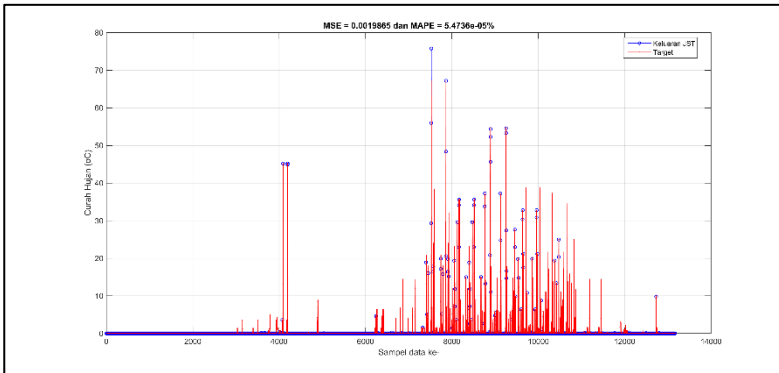
A.1 Curah Hujan



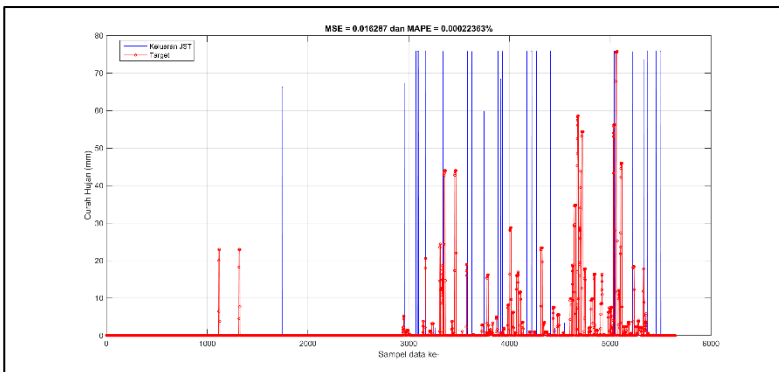
Gambar A 1 Hasil Pelatihan JST *hidden node 20*



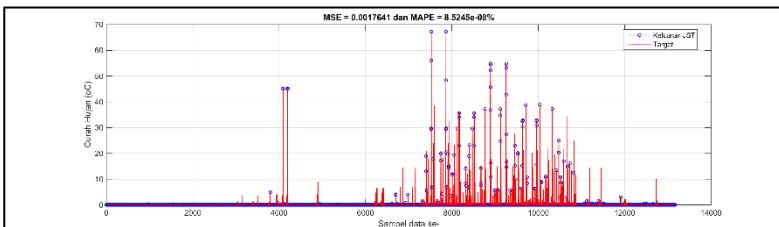
Gambar A 2 Hasil Pengujian *hidden node 20*



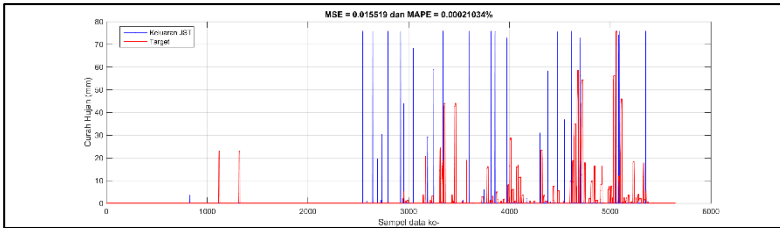
Gambar A 3 Hasil Pelatihan Hidden Node 30



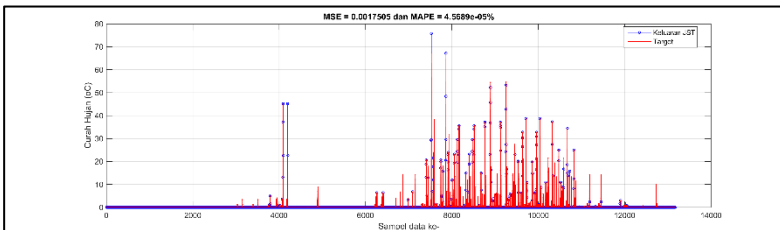
Gambar A 4 Hasil Pengujian *Hidden node 30*



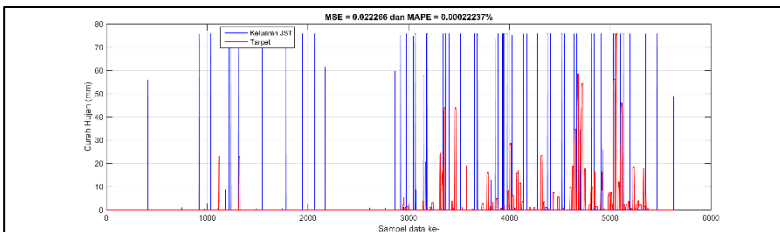
Gambar A 5 Hasil Pelatihan *Hidden node 40*



Gambar A 6 Hasil Pengujian *hidden node 40*

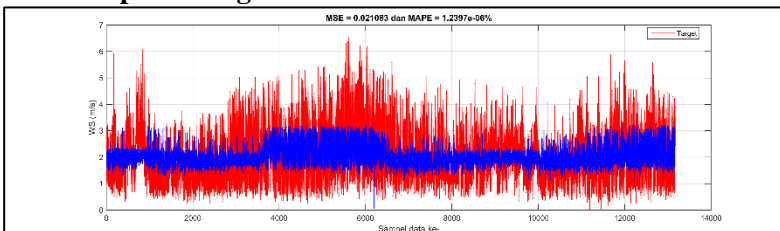


Gambar A 7 Hasil Pelatihan Hidden Node 50

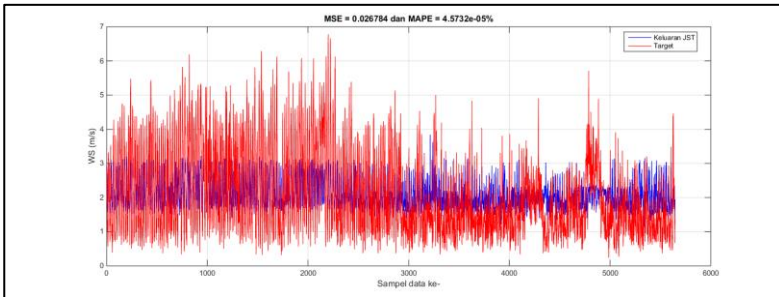


Gambar A 8 Hasil Pengujian *hidden node 50*

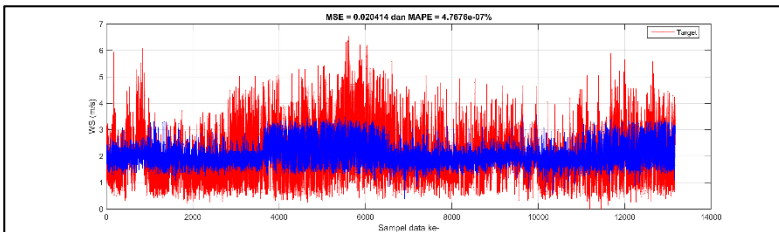
A.2 Kecepatan Angin



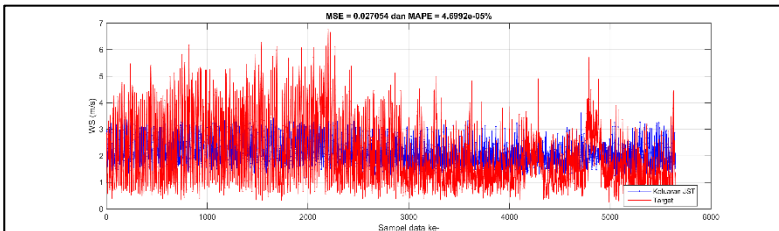
Gambar A 9 Hasil Pelatihan *Hidden node 50*



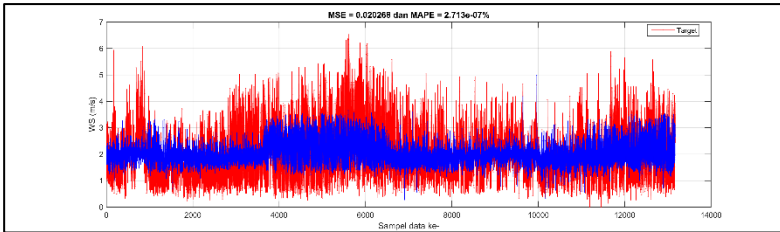
Gambar A 10 Hasil Pengujian *hidden node 10*



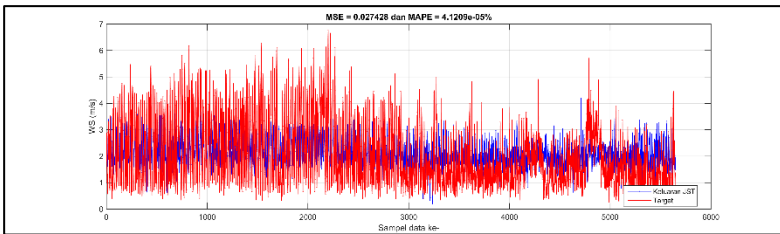
Gambar A 11 Hasil Pelatihan *hidden node 30*



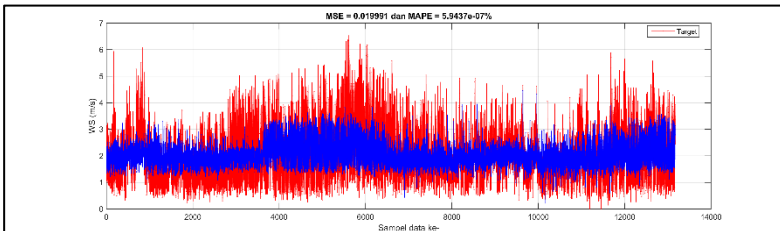
Gambar A 12 Hasil Pengujian *hidden node 30*



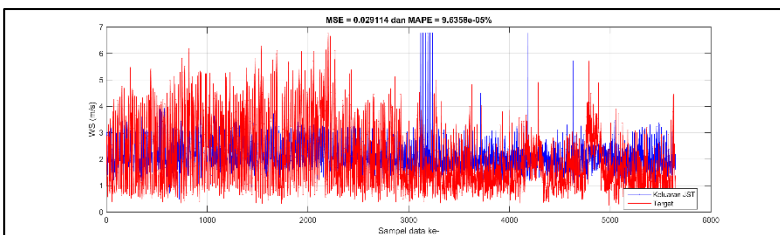
Gambar A 13 Hasil Pelatihan *hidden node 40*



Gambar A 14 Hasil Pengujian *hidden node 40*

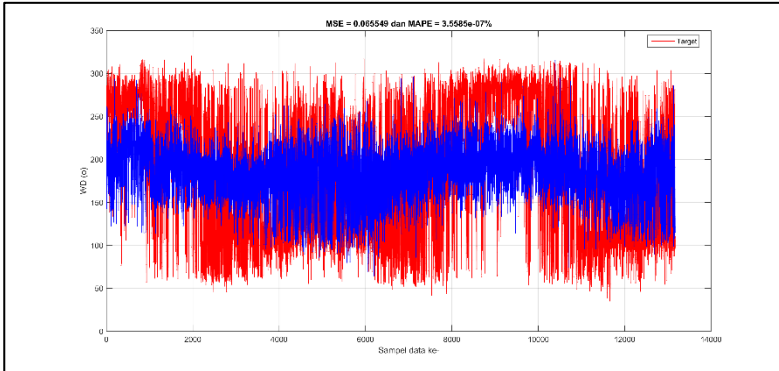


Gambar A 15 Hasil Pelatihan *hidden node 50*

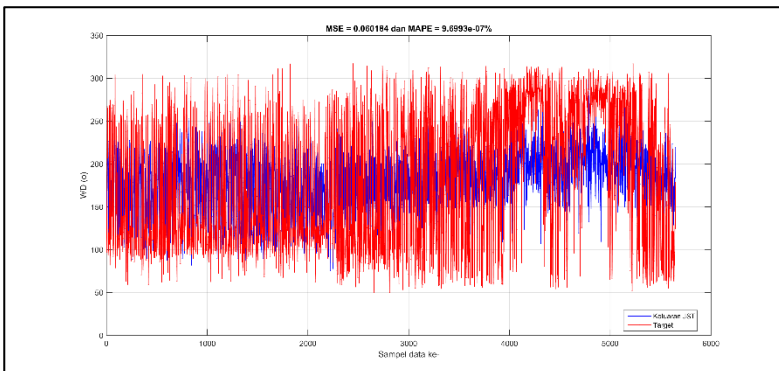


Gambar A 16 Hasil Pengujian *hidden node 50*

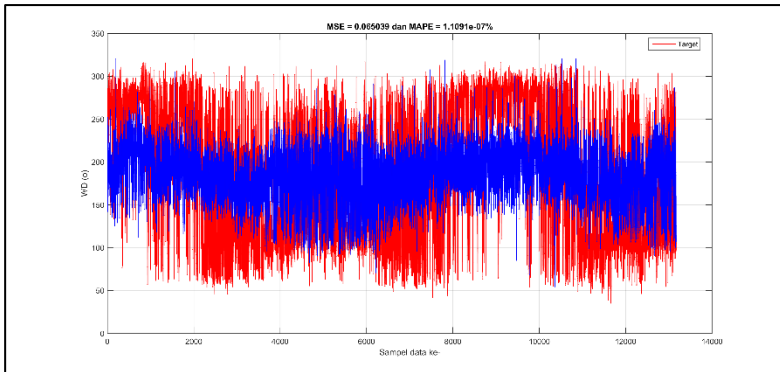
A.3 Arah Angin



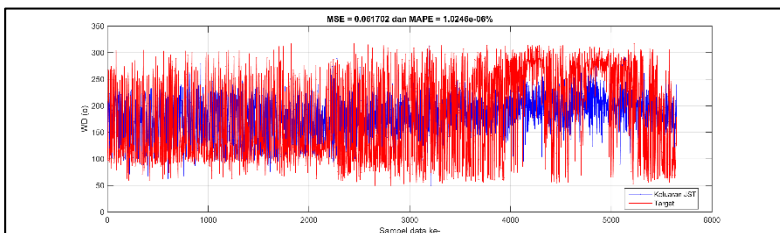
Gambar A 17 Hasil Pelatihan *hidden node 20*



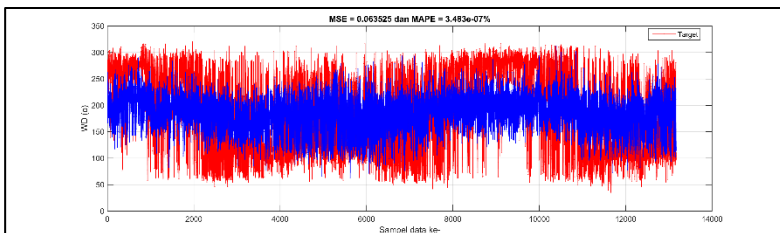
Gambar A 18 Hasil Pengujian *hidden node 20*



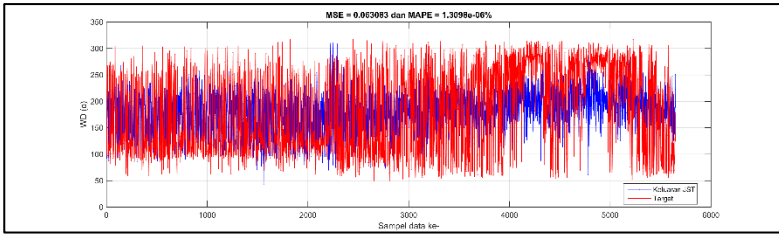
Gambar A 19 Hasil Pelatihan *hidden node 30*



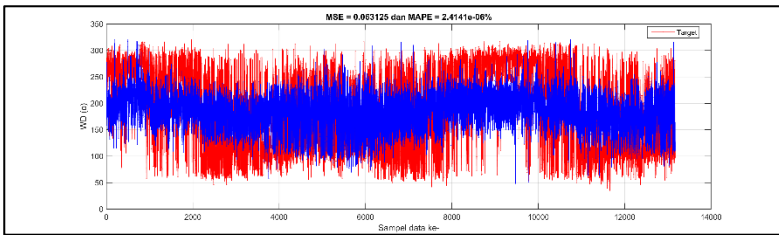
Gambar A 20 Hasil Pengujian *hidden node 30*



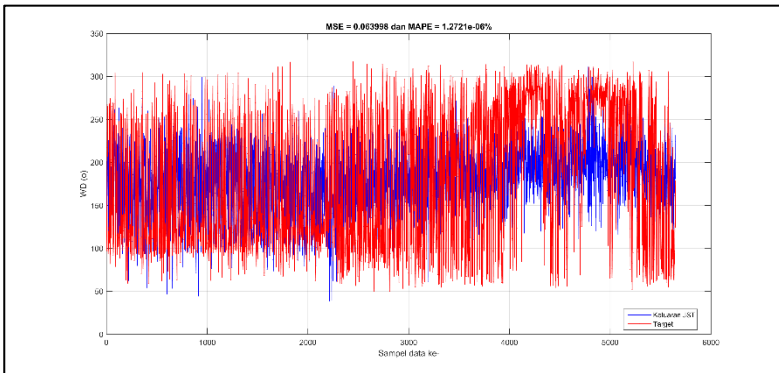
Gambar A 21 Hasil Pelatihan *hidden node 40*



Gambar A 22 Hasil Pengujian *hidden node 40*



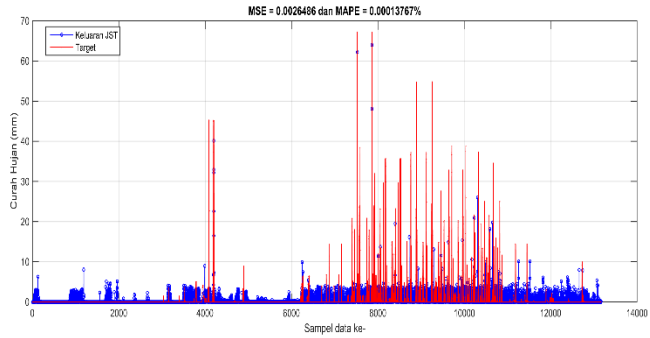
Gambar A 23 Hasil Pelatihan *hidden node 50*



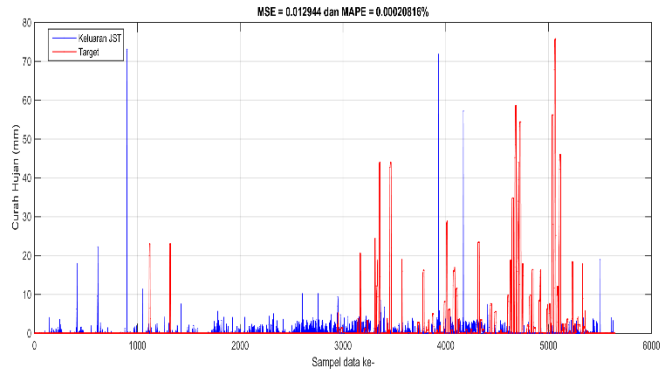
Gambar A 24 Hasil Pengujian *hidden node 50*

LAMPIRAN B, VARIASI TARGET *ERROR* 0.002

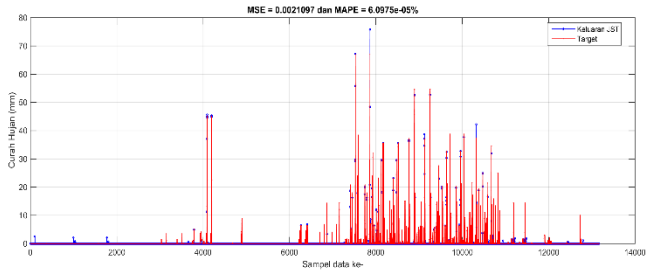
B.1 Curah hujan



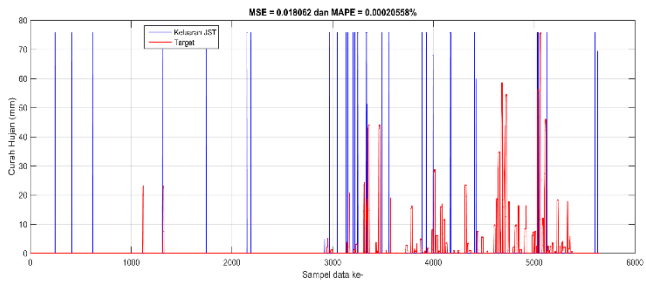
Gambar B 1 Hasil pelatihan *hidden node* 10



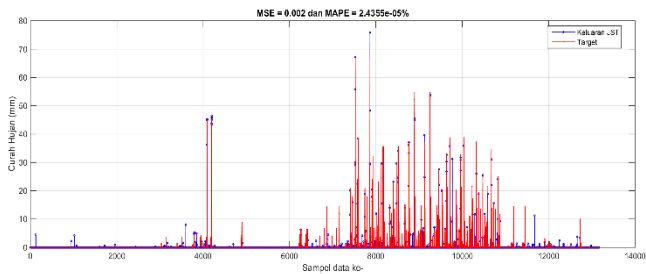
Gambar B 2 Hasil pengujian *hidden node* 10



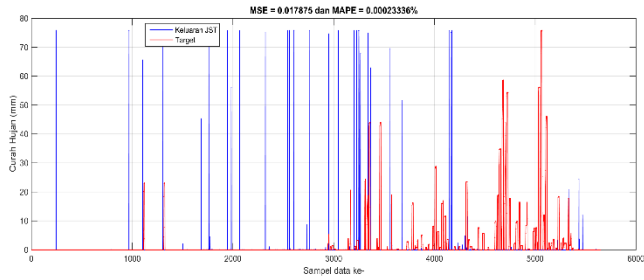
Gambar B 3 Hasil pelatihan *hidden node 20*



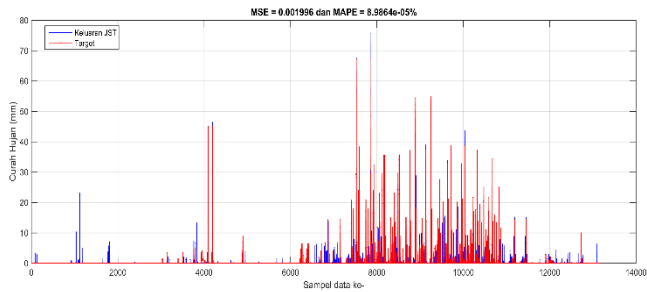
Gambar B 4 Hasil pengujian *hidden node 20*



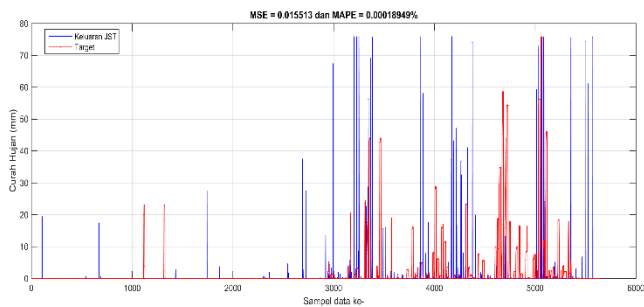
Gambar B 5 Hasil pelatihan *hidden node 30*



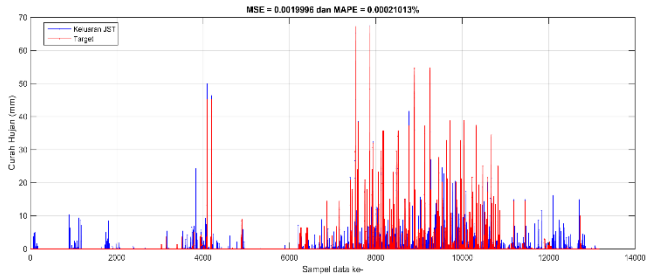
Gambar B 6 Hasil pengujian *hidden node 30*



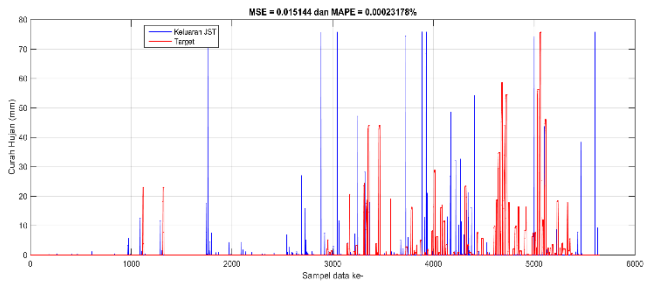
Gambar B 7 Hasil pelatihan *hidden node 40*



Gambar B 8 Hasil pengujian *hidden node 40*

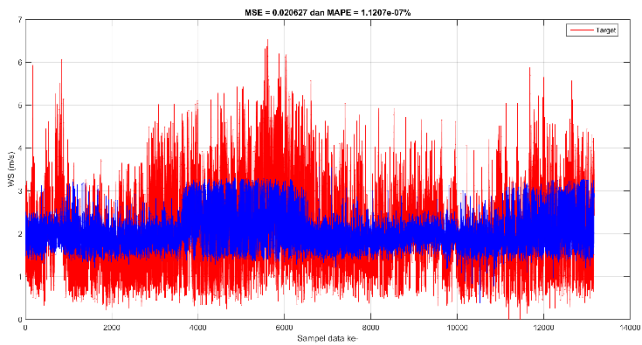


Gambar B 9 Hasil pelatihan *hidden node 50*

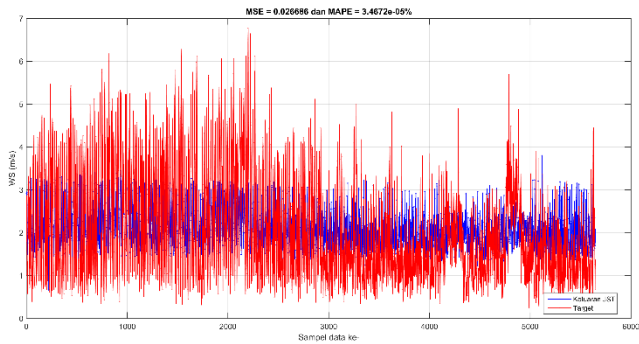


Gambar B 10 Hasil pengujian *hidden node 50*

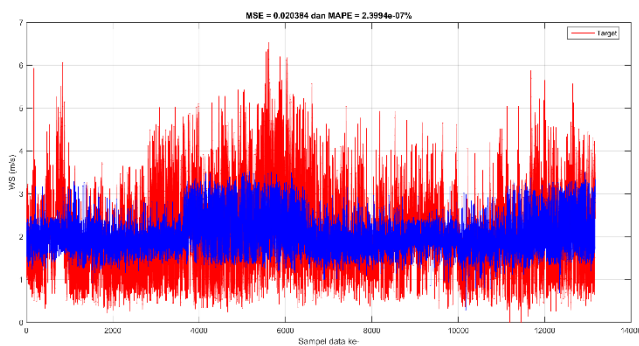
B.2 Kecepatan Angin



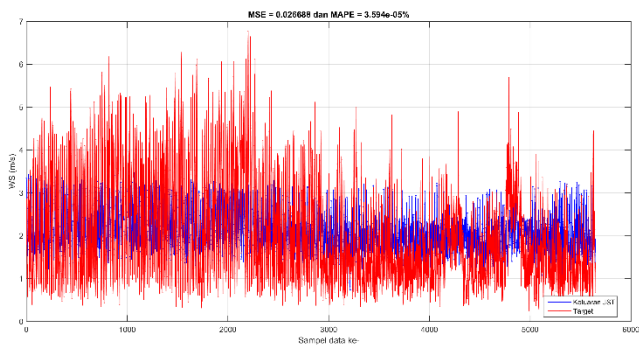
Gambar B 11 Hasil pelatihan *hidden node 20*



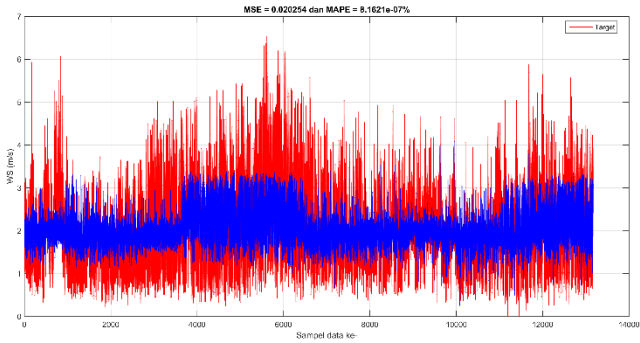
Gambar B 12 Hasil pengujian *hidden node 20*



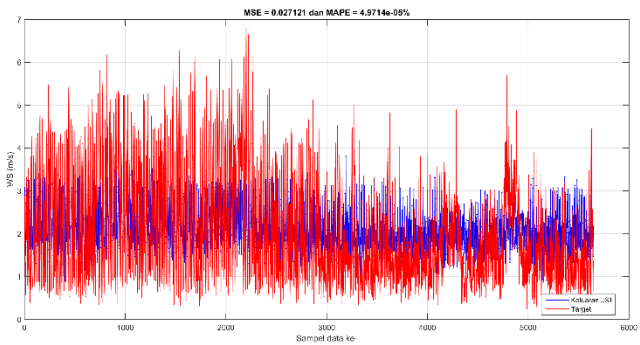
Gambar B 13 Hasil pelatihan *hidden node 30*



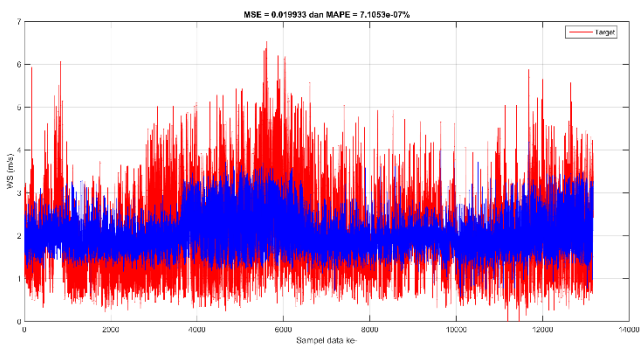
Gambar B 14 Hasil pengujian *hidden node 30*



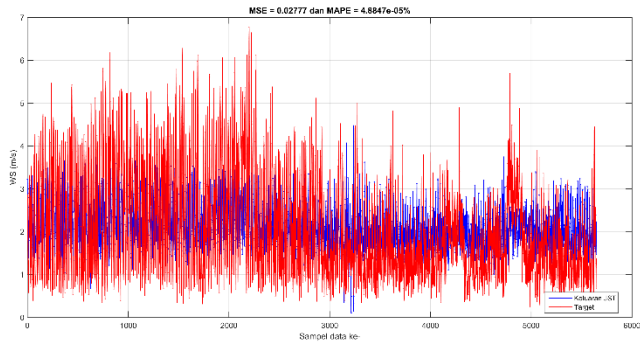
Gambar B 15 Hasil pelatihan *hidden node 40*



Gambar B 16 Hasil pengujian *hidden node 40*

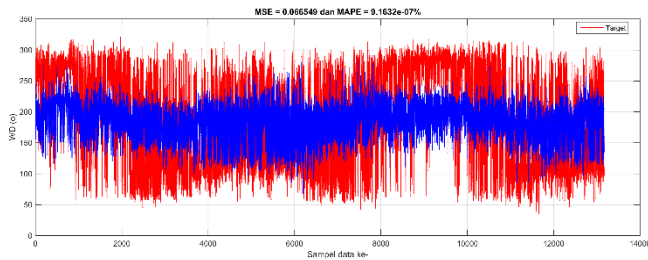


Gambar B 17 Hasil pelatihan *hidden node 50*

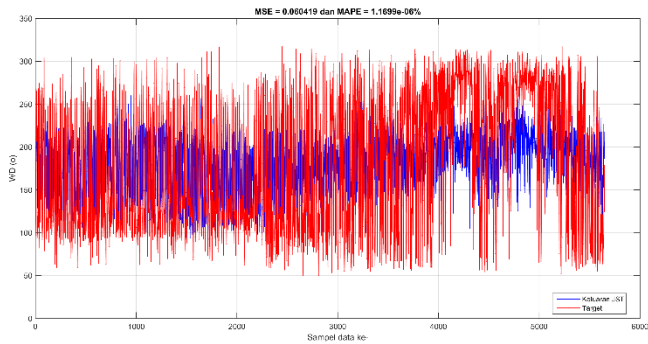


Gambar B 18 Hasil pengujian *hidden node 50*

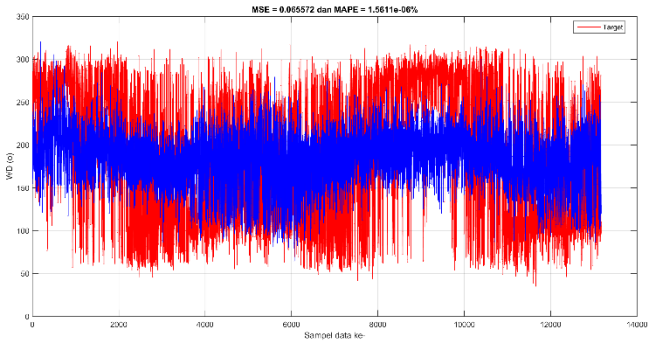
B.3 Arah Angin



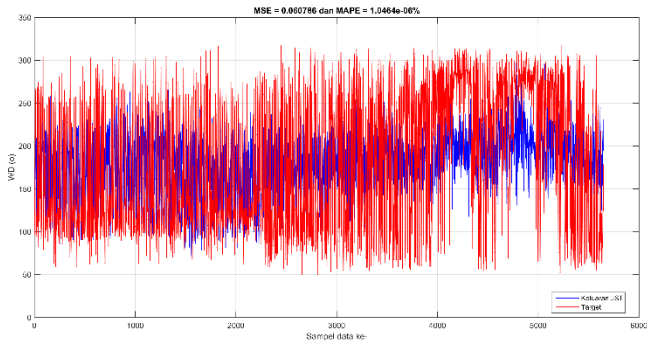
Gambar B 19 Hasil pelatihan *hidden node 10*



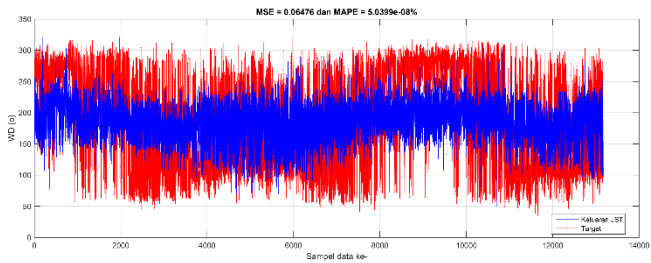
Gambar B 20 Hasil pengujian *hidden node 10*



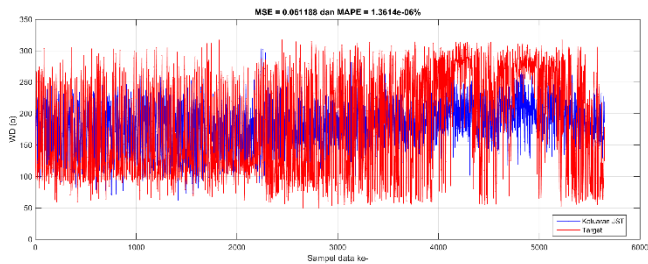
Gambar B 21 Hasil pelatihan *hidden node 20*



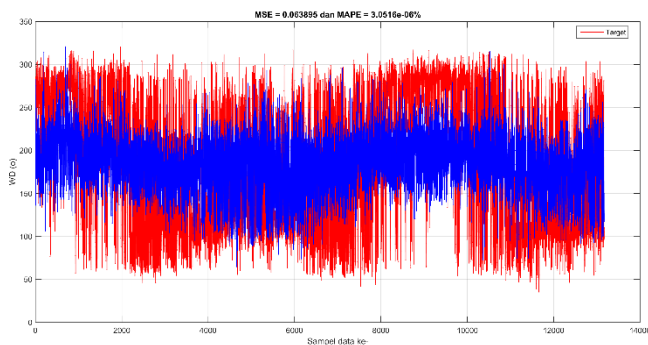
Gambar B 22 Hasil pengujian *hidden node 20*



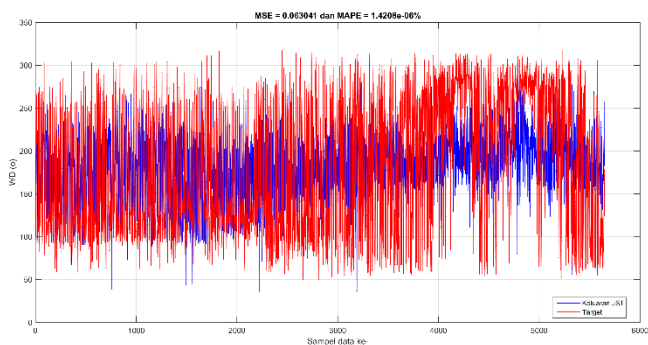
Gambar B 23 Hasil pelatihan *hidden node 30*



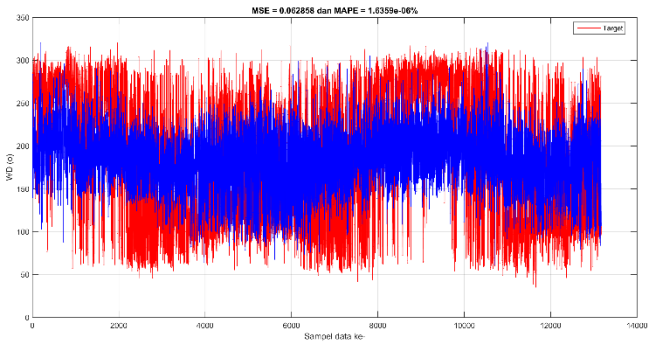
Gambar B 24 Hasil pengujian *hidden node 30*



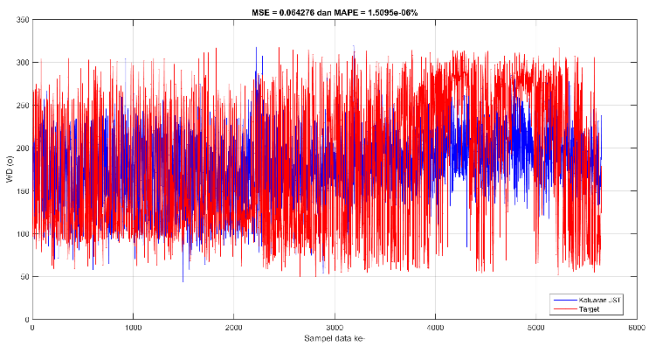
Gambar B 25 Hasil pelatihan *hidden node 40*



Gambar B 26 Hasil pengujian *hidden node 40*



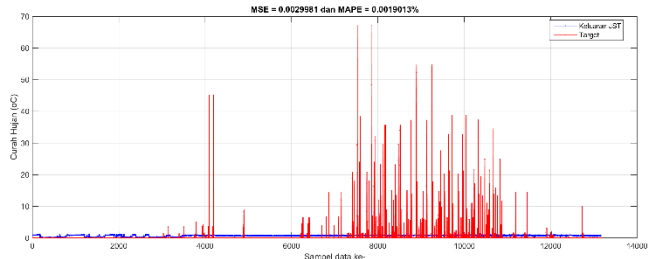
Gambar B 27 Hasil pelatihan *hidden node 50*



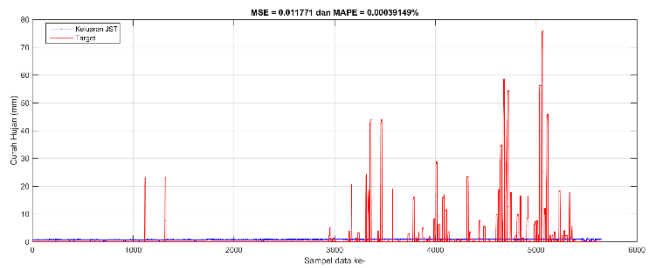
Gambar B 28 Hasil pengujian *hidden node 50*

Lampiran C, VARIASI TARGET ERROR 0.003

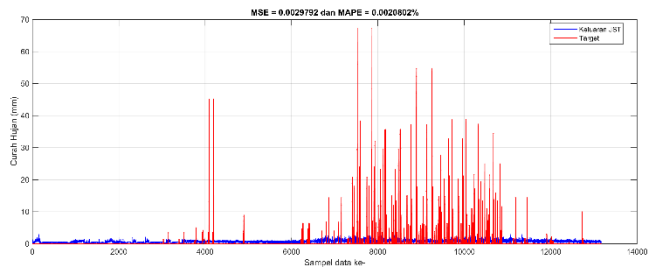
C.1 Curah Hujan



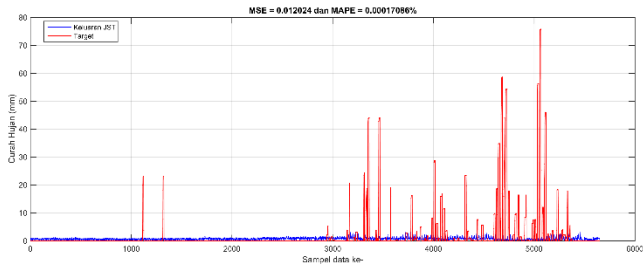
Gambar C. 1 Hasil pelatihan *hidden node 10*



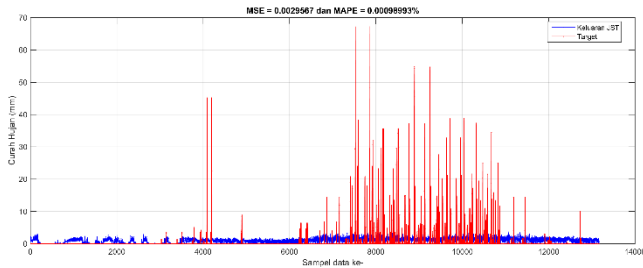
Gambar C. 2 Hasil pengujian *hidden node 10*



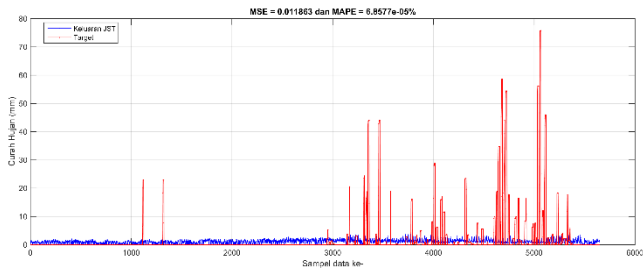
Gambar C. 3 Hasil pelatihan *hidden node 20*



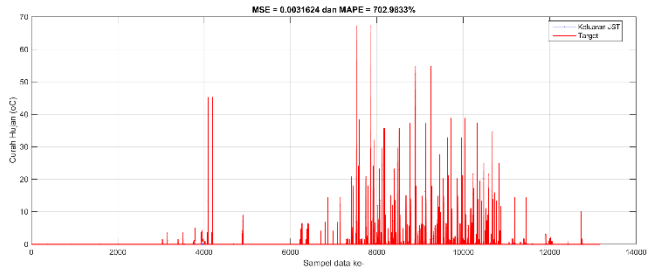
Gambar C. 4 Hasil pengujian *hidden node 20*



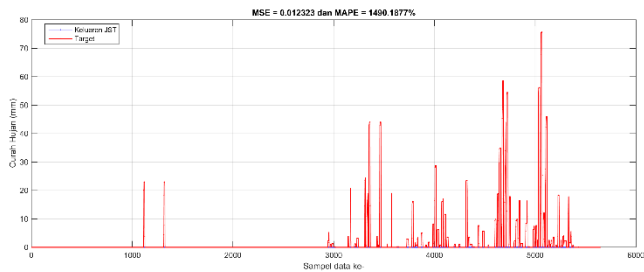
Gambar C. 5 Hasil pelatihan *hidden node 30*



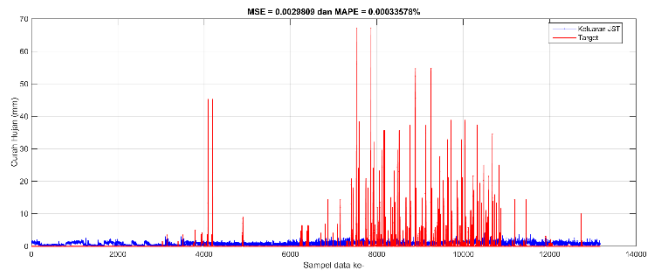
Gambar C. 6 Hasil pengujian *hidden node 30*



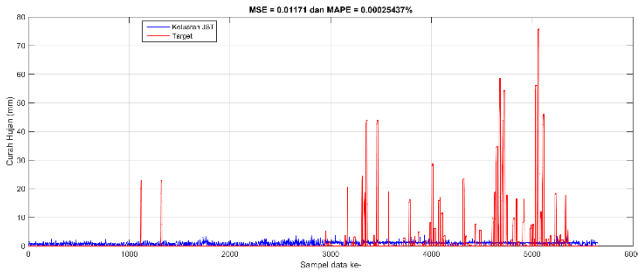
Gambar C. 7 Hasil pelatihan *hidden node 40*



Gambar C. 8 Hasil pengujian *hidden node 40*

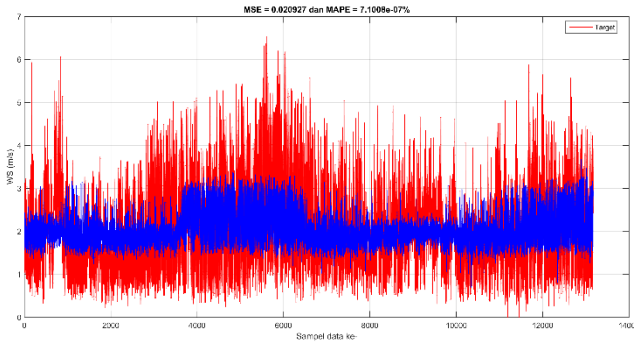


Gambar C. 9 hasil pelatiha *hidden node 50*

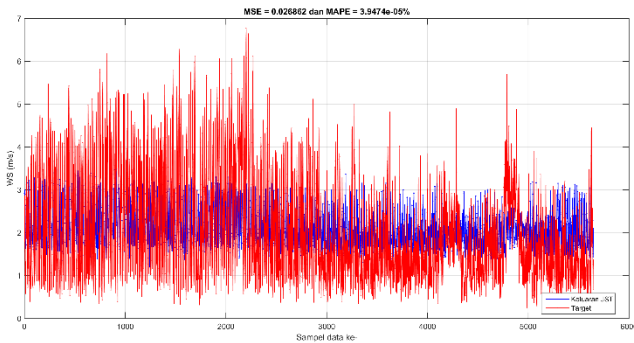


Gambar C. 10 Hasil pengujian *hidden node 50*

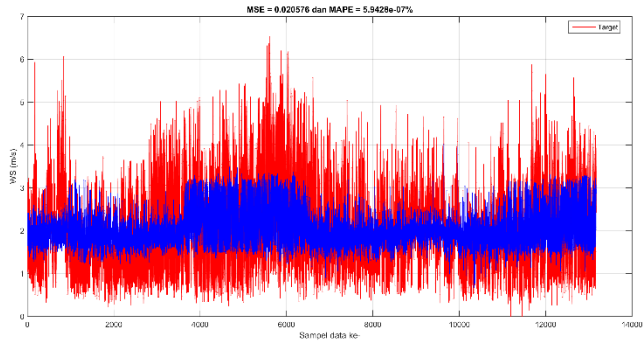
C.2 Kecepatan Angin



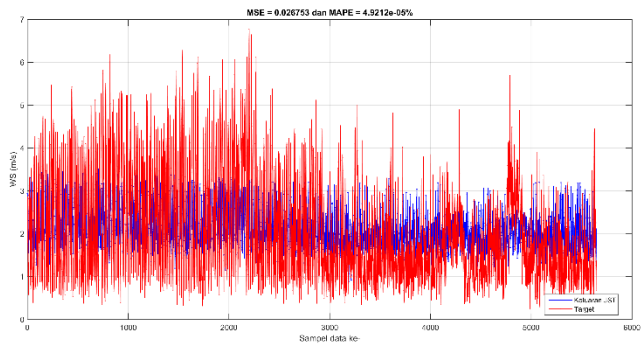
Gambar C. 11 Hasil pelatihan *hidden node 10*



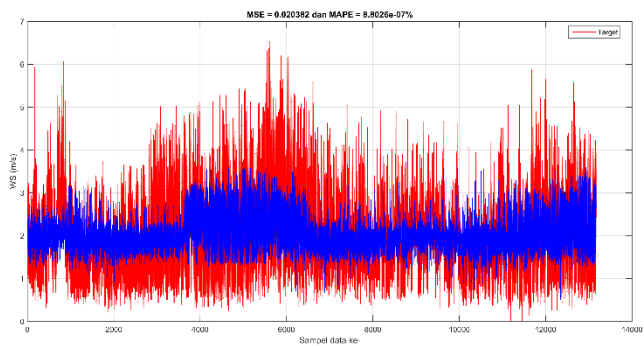
Gambar C. 12 Hasil pengujian *hidden node 10*



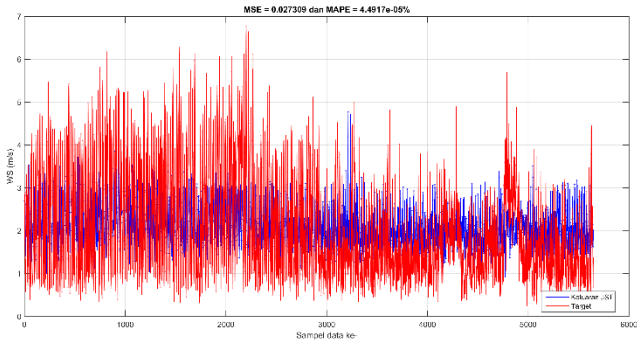
Gambar C. 13 Hasil pelatihan *hidden node 20*



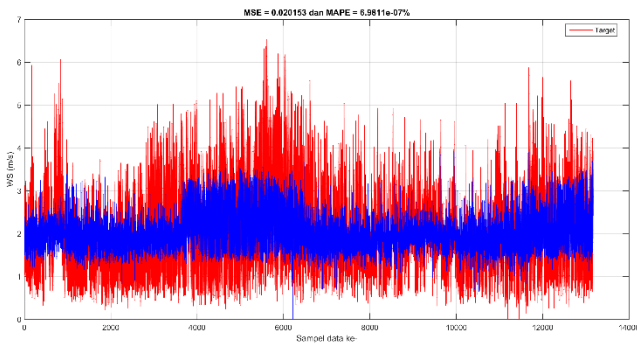
Gambar C. 14 Hasil pengujian *hidden node 20*



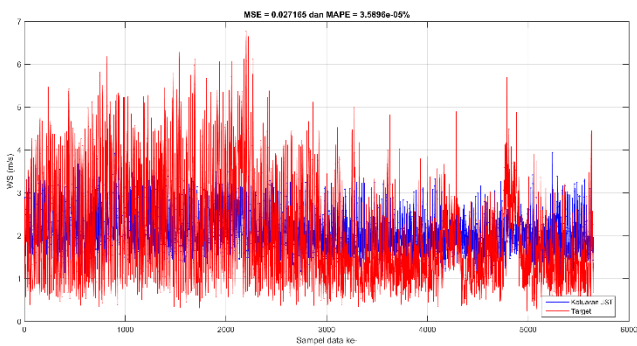
Gambar C. 15 Hasil pelatihan *hidden node 30*



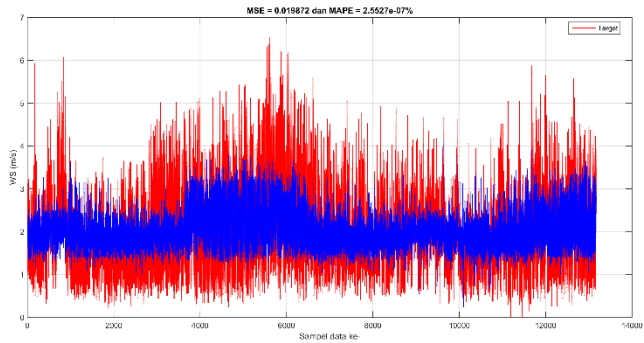
Gambar C. 16 Hasil pengujian *hidden node 30*



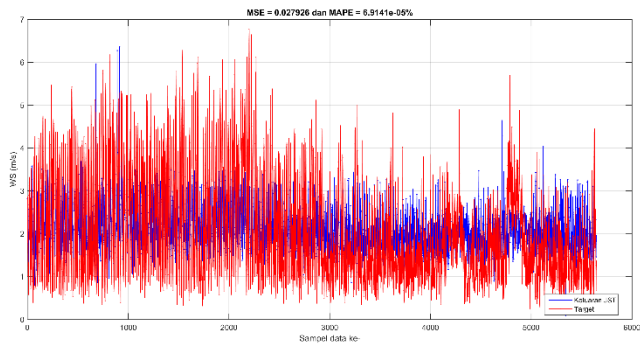
Gambar C. 17 Hasil pelatihan *hidden node 40*



Gambar C. 18 Hasil pengujian *hidden node 40*

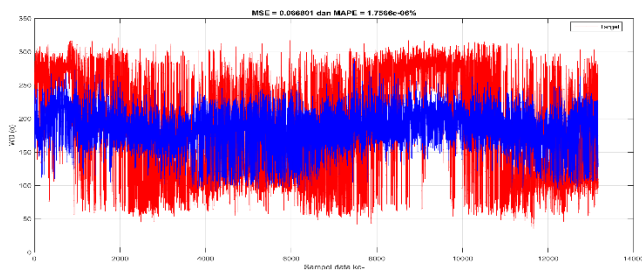


Gambar C. 19 Hasil pelatihan *hidden node* 50

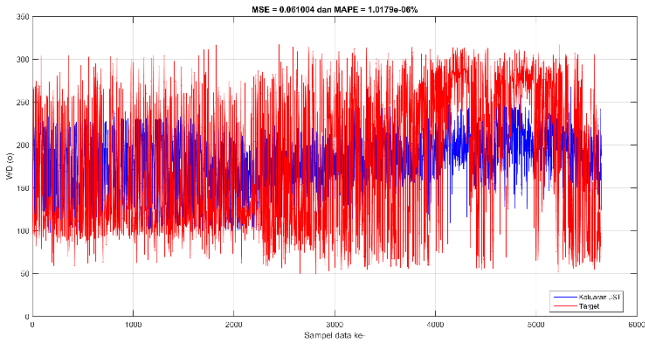


Gambar C. 20 Hasil pengujian *hidden node* 50

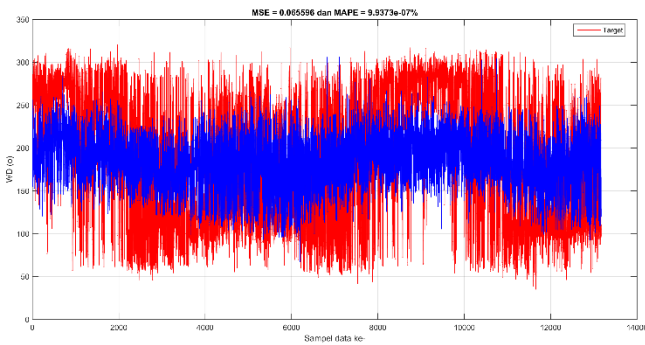
C.3 Arah Angin



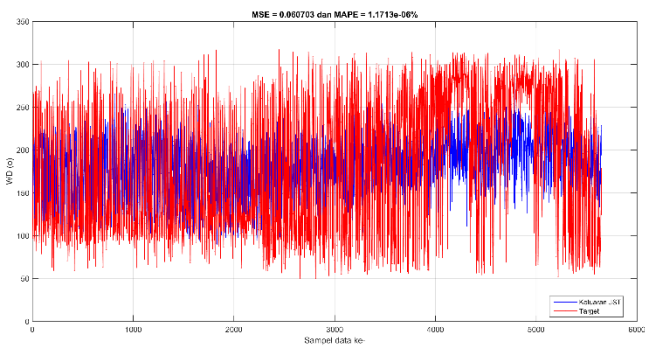
Gambar C. 21 Hasil pelatihan *hidden node* 70



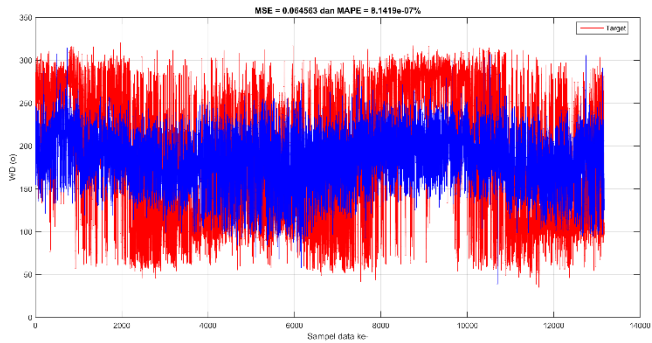
Gambar C. 22 Hasil pengujian *hidden node 10*



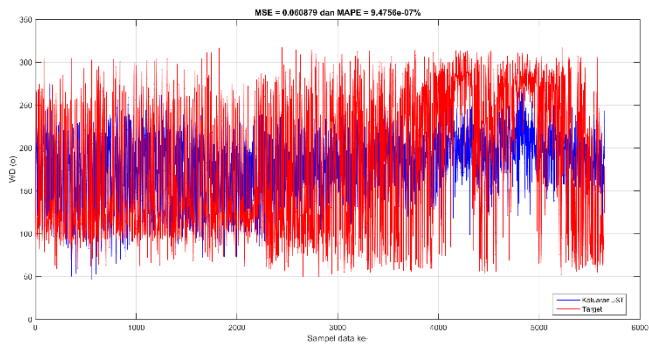
Gambar C. 23 Hasil pelatihan *hidden node 20*



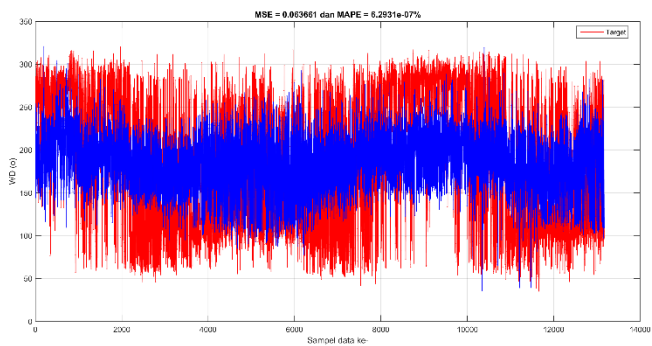
Gambar C. 24 Hasil pengujian *hidden node 20*



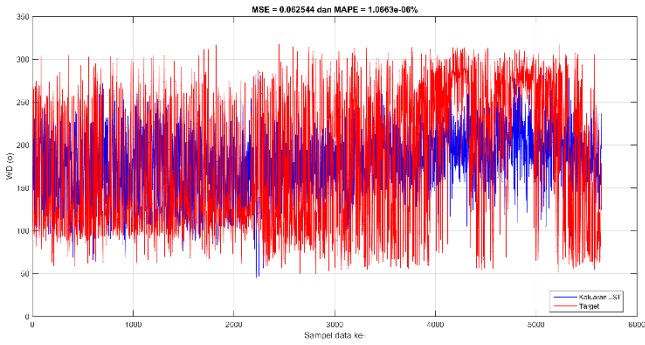
Gambar C. 25 Hasil pelatihan *hidden node 30*



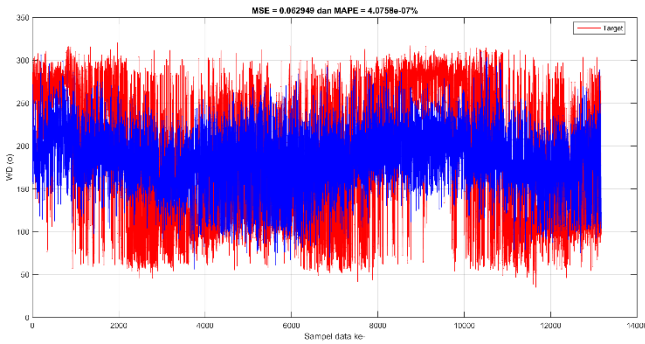
Gambar C. 26 Hasil pengujian *hidden node 30*



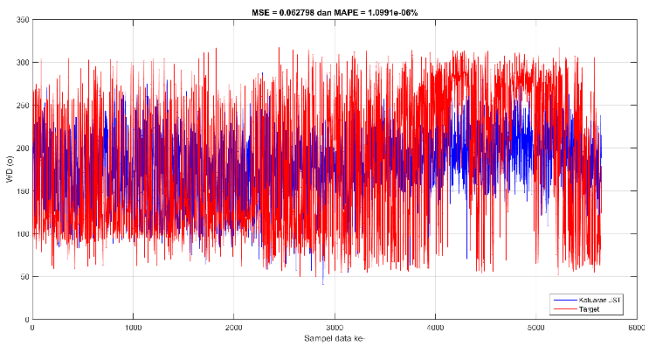
Gambar C. 27 Hasil pelatihan *hidden node 40*



Gambar C. 28 Hasil pengujian *hidden node 40*



Gambar C. 29 Hasil pelatihan *hidden node 50*



Gambar C. 30 Hasil pengujian *hidden node 50*

Lampiran D, Matlab Code

D.1 Normalisasi data

```
x1='data masukan prediktor1'(:,1);
x2='data masukan prediktor2'(:,2);
x3='data masukan prediktor3'(:,3);
y='data output prediktor'(:,4);
figure(1)
plot(x1(1:18816));
title('data masukan prediktor1');
xlabel('sample data ke-');
ylabel('data masukan prediktor1');
grid;
figure(2)
plot(x2(1:18816));
title('data masukan prediktor2');
xlabel('sample data ke-');
ylabel('data masukan prediktor2');
grid;
figure(3)
plot(x3(1:18816));
title('data masukan predikto3r');
xlabel('sample data ke-');
ylabel('data masukan prediktor3');
grid;
figure(4)
plot(y(1:18816));
title('data output prediktor');
xlabel('sample data ke-');
ylabel('data output prediktor');
grid;
%normalisasi
x1_min=min(x1);
x1_max=max(x1);
x2_min=min(x2);
x2_max=max(x2);
x3_min=min(x3);
x3_max=max(x3);
```

```

y_min=min(y);
y_max=max(y);
nor_x1=(x1-x1_min)/(x1_max-x1_min);
nor_x2=(x2-x2_min)/(x2_max-x2_min);
nor_x3=(x3-x3_min)/(x3_max-x3_min);
nor_y=(y-y_min)/(y_max-y_min);
%plot normalisasi
figure(5)
plot(nor_x1(1:18816));
title('data masukan prediktor1');
xlabel('sample data ke-');
ylabel('data masukan prediktor1');
grid;
figure(6)
plot(nor_x2(1:18816));
title('data masukan prediktor2');
xlabel('sample data ke-');
ylabel('data masukan prediktor2');
grid;
figure(7)
plot(nor_x3(1:18816));
title('data masukan prediktor3');
xlabel('sample data ke-');
ylabel('data masukan prediktor3');
grid;
figure(8)
plot(nor_y(1:18816));
title('data output prediktor');
xlabel('sample data ke-');
ylabel('data output prediktor');
grid;

```

D.2 Pelatihan JST-LM

```

interval=(1:18816);
%pemilihan data latih dan data uji
latihData=prediksi_CH;
n_latihinput=latihData(:,1:3)';
n_uji=12;

```

```

%pembuatan id data
data=[nor_x1,nor_x2,nor_x3,nor_y];
data_latih=data(1:n_latihinput,1:3)';
target_latih=data(1:n_latihinput,4)';
[m,n]=size(data_latih);
%pembuatan jst
net=newff(minmax(data_latih),[10 1],{'logsig',
'logsig'}, 'trainlm');
%jumlah node = 10
%jumlah output = 1
%logsig = fungsi aktivasi dari layer masukan ke
hidden layer
%logsig = fungsi aktivasi dari layer masukan ke
hidden layer
% Memberikan nilai untuk mempengaruhi proses
pelatihan
net.performFcn = 'mse';
net.trainParam.goal = 0.001; % target error
net.trainParam.show = 20;
net.trainParam.epochs = 1000; % jumlah iterasi
net.trainParam.mc = 0.95;
net.trainParam.lr = 0.1; % lerning rate
% Proses training
[net_keluaran,tr,Y,E] =
train(net,data_latih,target_latih)
% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hidden = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);
MAPE =
((sum(abs(nilai_error/nilai_keluaran))*100)*1/n)
;
save net.mat net_keluaran
% Hasil prediksi
hasil_latih = sim(net_keluaran,data_latih);

```

```

hasil_latih = (hasil_latih*(y_max-y_min))+y_min;
% Hasil prediksi
hasil_latih = sim(net_keluaran,data_latih);
hasil_latih = (hasil_latih*(y_max-y_min))+y_min;
% Performansi hasil prediksi
target_latih_asli = y(1:n_latihinput,1)';
figure,
plotregression(target_latih_asli,hasil_latih,'Re
gression')
figure,
plotperform(tr)
figure,
plot(hasil_latih,'bo-')
hold on
plot(target_latih_asli,'ro-')
hold off
grid on
title(strcat(['MSE = ',num2str(error_MSE),' dan
MAPE = ',num2str(MAPE),'%']))
xlabel('Sampel data ke-')
ylabel('variabel prediktor')
legend('Keluaran
JST','Target','Location','Best')

```

D.3 Pengujian JST-LM

```

% load jaringan yang sudah dibuat pada proses
pelatihan
load net.mat
% Proses membaca data uji dari excel
data_uji = data(n_latih:18816,1:3)';
target_uji = data(n_latih:18816,4)';
[m,n] = size(data_uji);
% Hasil prediksi
hasil_uji = sim(net_keluaran,data_uji);
nilai_error = hasil_uji-target_uji;
hasil_uji = (hasil_uji*(y_max-y_min))+y_min;;
% Performansi hasil prediksi
error_MSE = (1/n)*sum(nilai_error.^2);

```



```
MAPE =  
((sum(abs(nilai_error/hasil_uji)*100)*1/n));  
target_uji_asli = y(n_latih:18816,1)';  
figure,  
plot(hasil_uji,'bo-')  
hold on  
plot(target_uji_asli,'ro-')  
hold off  
grid on  
title(strcat(['MSE = ',num2str(error_MSE), ' dan  
MAPE = ',num2str(MAPE), '%']))  
xlabel('Sampel data ke-')  
ylabel('variabel prediktor')  
legend('Keluaran  
JST', 'Target', 'Location', 'Best')
```




Biodata Penulis

Penulis yang bernama Kartika Tri Wahyu Ramadhani biasa dipanggil Kartika, dilahirkan di Sidoarjo pada tanggal 10 Januari 1997 sebagai anak ketiga dari 3 bersaudara dari pasangan Fadli dan Alfiah. Penulis bertempat tinggal di Dsn. Blimbing Ds. Parerejo Rt 04 Rw 03 Kecamatan Purwodadi Kabupaten Pasuruan. Penulis telah menempuh pendidikan formal dimulai dari SD Negeri Parerejo III (2003-2009), SMP Negeri 1 Purwosari (2009-2012), SMA Negeri 1 Lawang (2012-2015), dan pada tahun 2015 penulis melanjutkan studi di Departemen Teknik Fisika ITS melalui jalur SNMPTN. Selama perkuliahan, penulis aktif mengikuti kegiatan di KM ITS. Penulis pernah bergabung dalam organisasi kemahasiswaan yaitu sebagai Staff Departemen Kesejahteraan Mahasiswa HMTF ITS periode 2016/2017, Kabinet Departemen Kesejahteraan Mahasiswa HMTF ITS periode 2018/2019. Selain itu penulis juga aktif mengikuti kepanitiaan yaitu diantaranya sebagai Anggota panitia EPC dalam acara tahunan teknik fisika EPW 2015, Koordinator Konsumsi panitia EPC dalam acara tahunan EPW 2016 dan Mentor GERIGI ITS tahun 2017. Penulis menerima segala saran dan kritik dari segala pihak yang bersifat membangun dan dapat meningkatkan manfaat Tugas Akhir ini melalui email penulis kartikadhany97@gmail.com.