



Tugas Akhir - TF181801

**RANCANG BANGUN PREDIKTOR KETINGGIAN  
GELOMBANG AIR LAUT MENGGUNAKAN  
METODE JARINGAN SYARAF TIRUAN DENGAN  
OPTIMASI *GENETIC ALGORITHM* (GA)**

Vidiyan Nabila  
NRP. 0231154000023

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





Final Project - TF181801

# **DESIGN OF SEA WATER WAVE HEIGHT PREDICTOR USING ARTIFICIAL NEURAL NETWORK WITH GENETIC ALGORITHM (GA) OPTIMIZATION**

Vidiyan Nabila  
NRP. 0231154000023

Supervisors:  
Dr. Ir. Syamsul Arifin, MT  
Prof. Dr. Ir. Aulia Siti Aisjah, MT

DEPARTMENT OF ENGINEERING PHYSICS  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya  
2019



## PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Vidiyan Nabila  
NRP : 0231154000023  
Departemen/ Prodi : Teknik Fisika/ S1 Teknik Fisika  
Fakultas : Fakultas Teknologi Industri  
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Rancang Bangun Prediktor Ketinggian Gelombang Air Laut Menggunakan Metode Jaringan Syaraf Tiruan dengan Optimasi *Genetic Algorithm (GA)*” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 31 Juli 2019

Yang membuat pernyataan,


Vidiyan Nabila  
NRP. 0231154000023



# LEMBAR PENGESAHAN

## RANCANG BANGUN PREDIKTOR KETINGGIAN GELOMBANG AIR LAUT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN DENGAN OPTIMASI *GENETIC ALGORITHM (GA)*

### TUGAS AKHIR

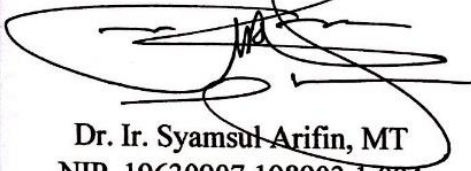
Oleh:



Vidiyan Nabila  
NRP. 0231154000023


Surabaya, 31 Juli 2019  
Mengetahui/Menyetujui

Pembimbing I



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

Pembimbing II



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

Kepala Departemen  
Teknik Fisika FTI-ITS



Agus Muhamad Isaia, ST, MSi, PhD  
NIP. 19780902 200312 1 002

DEPARTEMEN  
TEKNIK FISIKA

vii







**RANCANG BANGUN PREDIKTOR KETINGGIAN  
GELOMBANG AIR LAUT MENGGUNAKAN METODE  
JARINGAN SYARAF TIRUAN DENGAN OPTIMASI  
*GENETIC ALGORITHM (GA)***

**TUGAS AKHIR**

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

Oleh:  
Vidiyan Nabila  
NRP. 0231154000023

Disetujui oleh Tim Penguji Tugas Akhir

1. Dr. Ir. Syamsul Arifin, MT  (Pembimbing I)
2. Prof. Dr. Ir. Aulia Siti Aisjah, MT  (Pembimbing II)
3. Dr. Suyanto, ST, MT  (Ketua Penguji)
4. Iwan Cony Setiadi, ST, MT  (Penguji I)

SURABAYA  
2019



**RANCANG BANGUN PREDIKTOR KETINGGIAN  
GELOMBANG AIR LAUT MENGGUNAKAN METODE  
JARINGAN SYARAF TIRUAN DENGAN OPTIMASI  
*GENETIC ALGORITHM (GA)***

Nama : Vidiyan Nabila  
NRP : 02311540000023  
Departemen : Teknik Fisika  
Dosen Pembimbing : 1. Dr. Ir. Syamsul Arifin, MT  
2. Prof. Dr. Ir. Aulia Siti Aisjah, MT

**ABSTRAK**

Indonesia dikenal sebagai negara maritim dengan lautan yang luas sehingga banyak aktifitas setiap harinya seperti pelayaran. Banyak hal yang mempengaruhi kegiatan pelayaran diantaranya kondisi cuaca maritim. Penyebab kecelakaan laut diantaranya kondisi cuaca laut yaitu tinggi gelombang, arus laut, dan kecepatan angin. Tinggi gelombang laut dipengaruhi oleh angin serta besarnya arus laut. Penelitian tugas akhir ini dilakukan perancangan prediktor ketinggian gelombang air laut dan arus laut menggunakan Jaringan Syaraf Tiruan (JST) dengan optimasi *Genetic Algorithm (GA)* dibandingkan dengan JST-LM. Variabel masukan pada prediktor adalah kecepatan angin, arah angin, dan suhu permukaan laut. Variabel keluaran prediktor adalah tinggi gelombang laut dan kecepatan arus laut. Model JST yang digunakan ada dua yaitu JST dengan algoritma pelatihan Levenberg Marquardt (JST-LM) dan JST yang dioptimasi dengan *Genetic Algorithm (JST-GA)*. *Hidden layer* yang digunakan sebanyak satu dengan variasi *hidden node* dari 1 hingga 15. Hasil prediktor ketinggian gelombang air laut didapatkan bahwa JST-GA menghasilkan nilai RMSE lebih kecil daripada JST-LM. Arsitektur terbaik pada pengujian JST-LM adalah 4-10-1 dengan RMSE sebesar 0,059060 m sedangkan arsitektur pengujian JST-GA adalah 4-7-1 dengan nilai RMSE

sebesar 0,057556 m. Hasil prediktor kecepatan arus laut didapatkan bahwa JST-GA menghasilkan nilai RMSE lebih kecil daripada JST-LM. Arsitektur terbaik pada pengujian JST-LM adalah 4-4-1 dengan RMSE sebesar 10,93 cm/s sedangkan arsitektur pengujian JST-GA adalah 4-10-1 dengan nilai RMSE sebesar 8,66 cm/s.

**Kata Kunci:** Arus Laut, JST-GA, JST-LM, Tinggi Gelombang

**DESIGN OF SEA WATER WAVE HEIGHT PREDICTOR  
USING ARTIFICIAL NEURAL NETWORK WITH GENETIC  
ALGORITHM (GA) OPTIMIZATION**

*Name* : Vidiyan Nabila  
*NRP* : 0231154000023  
*Department* : Engineering Physics  
*Supervisors* : 1. Dr. Ir. Syamsul Arifin, MT  
2. Prof. Dr. Ir. Aulia Siti Aisjah, MT

**ABSTRACT**

*Indonesia is known as a maritime country with a vast ocean so many activities every day such as shipping. Many things that affect shipping activities include maritime weather conditions. The causes of marine accidents include sea weather conditions, namely wave height, ocean currents, and wind speed. Sea wave height is influenced by wind and the magnitude of ocean currents. The research of this final project is to design predictors of sea wave height and ocean currents using Artificial Neural Networks (ANN) with optimization of Genetic Algorithm (GA) compared to ANN-LM. Input variables on predictors are wind speed, wind direction, and sea surface temperature. Predictor output variables are sea wave height and ocean current velocity. There are two ANN models used, namely ANN with Levenberg Marquardt (ANN-LM) training algorithm and ANN algorithm optimized for Genetic Algorithm (ANN-GA). Hidden layer is used as much as one with a variety of hidden nodes from 1 to 15. The results of predictors of sea wave height found that ANN-GA produces a smaller RMSE value than ANN-LM. The best architecture for testing ANN-LM is 4-10-1 with RMSE of 0.059060 m while the testing architecture of ANN-GA is 4-7-1 with RMSE value of 0.057556 m. The results of predictors of*

*ocean current velocity were found that ANN-GA produced a smaller RMSE value than ANN-LM. The best architecture in testing ANN-LM is 4-4-1 with RMSE of 10.93 cm/s while the testing architecture of ANN-GA is 4-10-1 with RMSE value of 8.66 cm/s.*

**Keywords:** ANN-GA, ANN-LM, Current Speed, Wave Height.

## **KATA PENGANTAR**

Puji syukur kepada Allah SWT, karena atas karunia, nikmat, dan rahmat-Nya sehingga penulis diberikan kesehatan dan kelancaran dalam menyelesaikan tugas akhir dengan judul:

### **RANCANG BANGUN PREDIKTOR KETINGGIAN GELOMBANG AIR LAUT MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN DENGAN OPTIMASI *GENETIC ALGORITHM (GA)***

Penulis menyadari bahwa tugas akhir ini tidak mungkin terwujud tanpa bantuan dari berbagai pihak. Penulis menyampaikan terima kasih kepada:

1. Agus Muhamad Hatta, ST, MSi, PhD selaku Kepala Departemen Teknik Fisika ITS dan segenap bapak/ibu dosen pengajar di Departemen Teknik Fisika - ITS.
2. Dr. Ir. Syamsul Arifin, MT dan Prof. Dr. Ir. Aulia Siti Aisjah, MT selaku dosen pembimbing yang telah meluangkan waktunya membimbing, memberikan dukungan, motivasi, dan saran selama pengerjaan tugas akhir penulis.
3. Suyanto, ST, MT, Ir. Ya'umar, MT, dan Iwan Cony Setiadi, ST, MT selaku penguji pada sidang tugas akhir ini serta telah memberikan saran terhadap tugas akhir penulis.
4. Pihak BMKG Maritim Perak II Surabaya (Bapak Ari, Bapak Fajar, dan Mbak Cintya) sebagai penyuplai data sekunder dan memberikan informasi yang dibutuhkan dalam penyelesaian tugas akhir.
5. Keluarga penulis yang selalu memberikan dukungan, motivasi, dan do'a dalam kondisi apapun selama mengerjakan tugas akhir penulis.

6. Tim tugas akhir *Buoyweather*, Abiyyu, Adel, Bima, Dina, Tika, Isa, dan Wildan yang banyak membantu dan memberi support satu sama lain dalam proses pengerjaan melalui semua suka duka dari awal hingga akhir.

Semoga Allah SWT senantiasa membalas segala amal dan kebaikan pihak-pihak tersebut. Penulis juga menyadari bahwa pada tugas akhir ini masih terdapat banyak kekurangan. Oleh karena itu, segala 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

PERNYATAAN BEBAS PLAGIARISME.....	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK.....	xi
<i>ABSTRACT</i> .....	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxv
DAFTAR NOTASI.....	xxvii
DAFTAR ISTILAH.....	xxix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Gelombang Laut.....	7
2.2 Arus Laut.....	10
2.3 Jaringan Syaraf Tiruan.....	10
2.4 Algoritma Pelatihan Levenberg Marquardt.....	14
2.5 Optimasi <i>Genetic Algorithm</i> (GA).....	16
2.6 <i>Buoyweather Station Type II</i> .....	19
2.7 Validasi Prediktor.....	20
BAB III METODE PENELITIAN.....	23
3.1 Perumusan Masalah.....	24
3.2 Studi Literatur.....	24
3.3 Pengumpulan Data.....	25
3.4 Pengujian Data.....	26
3.4.1 Uji <i>Missing Data</i> .....	26
3.4.2 Uji Data <i>Outlier</i> .....	26
3.4.3 Uji Korelasi.....	27
3.4.4 Analisa Data.....	28
3.5 Perancangan Algoritma Prediktor.....	28
3.5.1 Perancangan Prediktor Ketinggian Gelombang.....	28

3.5.2	Perancangan Prediktor Kecepatan Arus Laut .....	32
3.6	Pemilihan Arsitektur Terbaik .....	35
3.6.1	Arsitektur Terbaik Prediktor Gelombang .....	35
3.6.2	Arsitektur Terbaik Prediktor Kecepatan Arus Laut..	40
3.7	Pengujian <i>Realtime</i> .....	44
3.8	Analisis Data dan Pembahasan.....	44
3.9	Kesimpulan dan Saran.....	44
<b>BAB IV</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>45</b>
4.1	Pengujian Data .....	45
4.1.1	Uji <i>Missing Data</i> .....	45
4.1.2	Uji Data <i>Outlier</i> .....	45
4.1.3	Uji Korelasi Data .....	46
4.2	Prediksi Gelombang Menggunakan JST-LM .....	49
4.3	Prediksi Kecepatan Arus Laut Menggunakan JST-LM ..	55
4.4	Prediksi Gelombang Menggunakan JST-GA .....	60
4.5	Prediksi Kecepatan Arus Laut Menggunakan JST-GA ..	66
4.6	Prediktor Ketinggian Gelombang dengan Data <i>Realtime</i>	71
4.7	Prediktor Kecepatan Arus Laut dengan Data <i>Realtime</i> ...	73
<b>BAB V</b>	<b>PENUTUP .....</b>	<b>77</b>
4.1	Kesimpulan .....	77
4.2	Saran .....	77
<b>DAFTAR RUJUKAN</b> .....		<b>79</b>
<b>LAMPIRAN</b> .....		<b>83</b>
Lampiran A.	Pelatihan JST-GA Prediktor Gelombang .....	83
Lampiran B.	Pengujian JST-GA Prediktor Gelombang.....	92
Lampiran C.	Pelatihan JST-GA Prediktor Arus Laut .....	101
Lampiran D.	Pengujian JST-GA Prediktor Arus Laut .....	110
Lampiran E.	Kode Program JST-LM .....	119
E.1	Pelatihan JST LM .....	119
E.2	Pengujian JST LM .....	122
Lampiran F.	Kode Program JST-GA.....	124
F.1	Fungsi Objektif .....	124
F.2	Optimasi <i>Genetic Algorithm</i> .....	129
F.3	Pengujian Bobot Optimasi .....	134
<b>BIODATA PENULIS</b> .....		<b>139</b>

## DAFTAR GAMBAR

Gambar 2. 1 Distribusi Gelombang Laut (Chakraborty, 2018).....	8
Gambar 2. 2 Hubungan Antar Neuron dalam Arsitektur JST (Fausset, 1994) .....	11
Gambar 2. 3 Jaringan Tipe Feed Forward (Jain, Deo, & Rajendran, 2011) .....	12
Gambar 2. 4 Fungsi Aktivasi Linier (Fausset, 1994) .....	13
Gambar 2. 5 Fungsi Aktivasi Sigmoid Bipolar (Fausset, 1994)...	13
Gambar 2. 6 Fungsi Aktivasi Sigmoid Biner (Fausset, 1994).....	14
Gambar 2. 7 Siklus Optimasi Genetic Algorithm (GA) (Man et al., 1999) .....	18
Gambar 2. 8 Buoyweather Station Type II .....	20
Gambar 3. 1 Diagram Alir Penelitian .....	23
Gambar 3. 2 Diagram Blok Prediktor .....	28
Gambar 3. 3 Diagram Alir JST-LM Prediktor Gelombang.....	29
Gambar 3. 4 Diagram Alir JST-GA Prediktor Ketinggian Gelombang .....	31
Gambar 3. 5 Diagram Alir JST-LM Prediktor Arus Laut .....	33
Gambar 3. 6 Diagram Alir JST-GA Prediktor Arus Laut .....	34
Gambar 3. 7 Arsitektur JST-LM 4-10-1 Prediktor Ketinggian Gelombang .....	36
Gambar 3. 8 Arsitektur JST-GA 4-7-1 Prediktor Ketinggian Gelombang .....	38
Gambar 3. 9 Arsitektur JST-LM 4-4-1 Prediktor Arus Laut.....	40
Gambar 3. 10 Arsitektur JST-GA 4-10-1 Prediktor Arus Laut ....	42
Gambar 4. 1 Hasil Pelatihan JST-LM Prediktor Ketinggian Gelombang Arsitektur 4-10-1 .....	52
Gambar 4. 2 Hasil Pengujian JST-LM Prediktor Ketinggian Gelombang Arsitektur 4-10-1 .....	54

Gambar 4. 3 Hasil Pelatihan JST-LM Prediktor Arus Laut Arsitektur 4-4-1 .....	57
Gambar 4. 4 Hasil Pengujian JST-LM Prediktor Arus Laut Arsitektur 4-4-1 .....	59
Gambar 4. 5 Grafik Performansi Optimasi Genetic Algorithm (GA) Prediktor Gelombang .....	60
Gambar 4. 6 Hasil Pelatihan JST-GA Prediktor Gelombang Arsitektur 4-7-1 .....	62
Gambar 4. 7 Hasil Pengujian JST-GA Prediktor Gelombang Arsitektur 4-7-1 .....	64
Gambar 4. 8 Grafik Performansi Optimasi Genetic Algorithm (GA) Prediktor Arus Laut .....	66
Gambar 4. 9 Hasil Pelatihan JST-GA Prediktor Arus Laut Arsitektur 4-10-1 .....	68
Gambar 4. 10 Hasil Pengujian JST-GA Prediktor Arus Laut Arsitektur 4-10-1 .....	70
Gambar 4. 11 Hasil Pengujian Realtime Prediktor Gelombang JST-GA 4-7-1 .....	72
Gambar 4. 12 Hasil Pengujian Realtime Prediktor Arus Laut JST- GA 4-10-1 .....	74
Gambar 4. 13 Tampilan GUI Prediktor Cuaca .....	75
Gambar A- 1 Grafik Pelatihan JST-GA Hidden Node 1 .....	83
Gambar A- 2 Grafik Pelatihan JST-GA Hidden Node 2 .....	83
Gambar A- 3 Grafik Pelatihan JST-GA Hidden Node 3 .....	84
Gambar A- 4 Grafik Pelatihan JST-GA Hidden Node 4 .....	84
Gambar A- 5 Grafik Pelatihan JST-GA Hidden Node 5 .....	85
Gambar A- 6 Grafik Pelatihan JST-GA Hidden Node 6 .....	85
Gambar A- 7 Grafik Pelatihan JST-GA Hidden Node 7 Iterasi 30 .....	86
Gambar A- 8 Grafik Pelatihan JST-GA Hidden Node 7 Iterasi 60 .....	86

Gambar A- 9 Grafik Pelatihan JST-GA Hidden Node 7 Iterasi 90 .....	87
Gambar A- 10 Grafik Pelatihan JST-GA Hidden Node 8.....	87
Gambar A- 11 Grafik Pelatihan JST-GA Hidden Node 9.....	88
Gambar A- 12 Grafik Pelatihan JST-GA Hidden Node 10.....	88
Gambar A- 13 Grafik Pelatihan JST-GA Hidden Node 11.....	89
Gambar A- 14 Grafik Pelatihan JST-GA Hidden Node 12.....	89
Gambar A- 15 Grafik Pelatihan JST-GA Hidden Node 13.....	90
Gambar A- 16 Grafik Pelatihan JST-GA Hidden Node 14.....	90
Gambar A- 17 Grafik Pelatihan JST-GA Hidden Node 15.....	91
Gambar B- 1 Grafik Pengujian JST-GA Hidden Node 1 .....	92
Gambar B- 2 Grafik Pengujian JST-GA Hidden Node 2 .....	92
Gambar B- 3 Grafik Pengujian JST-GA Hidden Node 3 .....	93
Gambar B- 4 Grafik Pengujian JST-GA Hidden Node 4 .....	93
Gambar B- 5 Grafik Pengujian JST-GA Hidden Node 5 .....	94
Gambar B- 6 Grafik Pengujian JST-GA Hidden Node 6 .....	94
Gambar B- 7 Grafik Pengujian JST-GA Hidden Node 7 Iterasi 30 .....	95
Gambar B- 8 Grafik Pengujian JST-GA Hidden Node 7 Iterasi 60 .....	95
Gambar B- 9 Grafik Pengujian JST-GA Hidden Node 7 Iterasi 90 .....	96
Gambar B- 10 Grafik Pengujian JST-GA Hidden Node 8 .....	96
Gambar B- 11 Grafik Pengujian JST-GA Hidden Node 9 .....	97
Gambar B- 12 Grafik Pengujian JST-GA Hidden Node 10 .....	97
Gambar B- 13 Grafik Pengujian JST-GA Hidden Node 11 .....	98
Gambar B- 14 Grafik Pengujian JST-GA Hidden Node 12 .....	98
Gambar B- 15 Grafik Pengujian JST-GA Hidden Node 13 .....	99
Gambar B- 16 Grafik Pengujian JST-GA Hidden Node 14 .....	99
Gambar B- 17 Grafik Pengujian JST-GA Hidden Node 15 .....	100
Gambar B- 18 Grafik Pelatihan JST-GA Hidden Node 15 .....	109

Gambar C- 1 Grafik Pelatihan JST-GA Hidden Node 1.....	101
Gambar C- 2 Grafik Pelatihan JST-GA Hidden Node 2.....	101
Gambar C- 3 Grafik Pelatihan JST-GA Hidden Node 3.....	102
Gambar C- 4 Grafik Pelatihan JST-GA Hidden Node 4.....	102
Gambar C- 5 Grafik Pelatihan JST-GA Hidden Node 5.....	103
Gambar C- 6 Grafik Pelatihan JST-GA Hidden Node 6.....	103
Gambar C- 7 Grafik Pelatihan JST-GA Hidden Node 7.....	104
Gambar C- 8 Grafik Pelatihan JST-GA Hidden Node 8.....	104
Gambar C- 9 Grafik Pelatihan JST-GA Hidden Node 9.....	105
Gambar C- 10 Grafik Pelatihan JST-GA Hidden Node 10 Iterasi 30.....	105
Gambar C- 11 Grafik Pelatihan JST-GA Hidden Node 10 iterasi 60.....	106
Gambar C- 12 Grafik Pelatihan JST-GA Hidden Node 10 iterasi 90.....	106
Gambar C- 13 Grafik Pelatihan JST-GA Hidden Node 11.....	107
Gambar C- 14 Grafik Pelatihan JST-GA Hidden Node 12.....	107
Gambar C- 15 Grafik Pelatihan JST-GA Hidden Node 14.....	108
Gambar C- 16 Grafik Pelatihan JST-GA Hidden Node 14.....	108
Gambar D- 1 Grafik Pengujian JST-GA Hidden Node 1 .....	110
Gambar D- 2 Grafik Pengujian JST-GA Hidden Node 2 .....	110
Gambar D- 3 Grafik Pengujian JST-GA Hidden Node 3 .....	111
Gambar D- 4 Grafik Pengujian JST-GA Hidden Node 4 .....	111
Gambar D- 5 Grafik Pengujian JST-GA Hidden Node 5 .....	112
Gambar D- 6 Grafik Pengujian JST-GA Hidden Node 6 .....	112
Gambar D- 7 Grafik Pengujian JST-GA Hidden Node 7 .....	113
Gambar D- 8 Grafik Pengujian JST-GA Hidden Node 8 .....	113
Gambar D- 9 Grafik Pengujian JST-GA Hidden Node 9 .....	114
Gambar D- 10 Grafik Pengujian JST-GA Hidden Node 10 Iterasi 30.....	114
Gambar D- 11 Grafik Pengujian JST-GA Hidden Node 10 iterasi 60.....	115

Gambar D- 12 Grafik Pengujian JST-GA Hidden Node 10 iterasi 90 .....	115
Gambar D- 13 Grafik Pengujian JST-GA Hidden Node 11.....	116
Gambar D- 14 Grafik Pengujian JST-GA Hidden Node 12.....	116
Gambar D- 15 Grafik Pengujian JST-GA Hidden Node 13.....	117
Gambar D- 16 Grafik Pengujian JST-GA Hidden Node 14.....	117
Gambar D- 17 Grafik Pengujian JST-GA Hidden Node 15.....	118





## DAFTAR TABEL

Tabel 2. 1 Standar Ketinggian Gelombang (Word Meteorological Organization, 1998).....	7
Tabel 2. 2 Parameter Kecepatan Angin Skala Beaufort (Word Meteorological Organization, 1998).....	9
Tabel 2. 3 Tingkat Akurasi Prediktor berdasarkan MAPE .....	21
Tabel 3. 1 Data Realtime Buoyweather .....	25
Tabel 3. 2 Nilai Interval Korelasi Terhadap Tingkat hubungannya (Sugiyono, 2007).....	27
Tabel 3. 3 Pembagian Data Pelatihan dan Validasi .....	28
Tabel 3. 4 Parameter JST-LM Prediktor Ketinggian Gelombang	30
Tabel 3. 5 Parameter JST-GA Prediktor Ketinggian Gelombang	32
Tabel 3. 6 Parameter JST-LM Prediktor Arus Laut.....	34
Tabel 3. 7 Parameter JST-GA Prediktor Arus Laut .....	35
Tabel 3. 8 Nilai Bobot pada Arsitektur 4-10-1 Metode JST-LM Prediktor Ketinggian Gelombang .....	37
Tabel 3. 9 Nilai Bobot pada Arsitektur 4-7-1 Metode JST-GA Prediktor Ketinggian Gelombang .....	39
Tabel 3. 10 Perbedaan Parameter JST-LM dan JST-GA pada Prediktor Ketinggian Gelombang.....	40
Tabel 3. 11 Nilai Bobot pada Arsitektur 4-4-1 Metode JST-LM Prediktor Arus Laut.....	41
Tabel 3. 12 Nilai Bobot pada Arsitektur 4-10-1 Metode JST-GA Prediktor Arus Laut.....	43
Tabel 3. 13 Perbedaan Parameter JST-LM dan JST-GA pada Prediktor Kecepatan Arus Laut .....	44
Tabel 4. 1 Uji Missing Data .....	45
Tabel 4. 2 Uji Data Outlier .....	46
Tabel 4. 3 Uji Korelasi Variabel Prediksi Tinggi Gelombang .....	46
Tabel 4. 4 Uji Korelasi Variabel Input JST Prediksi Ketinggian Gelombang .....	47
Tabel 4. 5 Uji Korelasi Variabel Prediksi Arus Laut .....	48

Tabel 4. 6 Uji Korelasi Variabel Input JST Prediksi Kecepatan Arus Laut .....	48
Tabel 4.7 Pengujian Fungsi Aktivasi dan Laju Pelatihan JST Prediktor Ketinggian Gelombang.....	50
Tabel 4.8 Nilai RMSE Metode JST-LM Prediktor Ketinggian Gelombang.....	51
Tabel 4. 9 Pengujian Fungsi Aktivasi dan Laju Pelatihan JST Prediktor Arus Laut.....	55
Tabel 4. 10 Nilai RMSE Metode JST-LM Prediktor Arus Laut ..	56
Tabel 4. 11 Nilai RMSE Hasil Pelatihan JST-GA Prediktor Ketinggian Gelombang .....	61
Tabel 4. 12 Nilai RMSE Hasil Pelatihan JST-GA Prediktor Arus laut.....	67
Tabel 4. 13 Data Hasil Pengujian Realtime Prediktor Gelombang .....	71
Tabel 4. 14 Data Hasil Pengujian Realtime Prediktor Arus Laut	73

## DAFTAR NOTASI

Notasi	Nama	Satuan
$v_{ij}$	Bobot Jaringan Syaraf Tiruan dari <i>input layer</i> ke <i>hidden layer</i>	-
$w_{ij}$	Bobot Jaringan Syaraf Tiruan dari <i>hidden layer</i> ke <i>ouput layer</i>	-
$W_k$	Bobot Jaringan Syaraf Tiruan seluruhnya	-
$J_k$	Matriks turunan pertama dari <i>error</i> jaringan terhadap bobot dan bias jaringan	-
$\mu$	Konstanta pembelajaran	-
$I$	Matriks identitas	-
$d_{p,m}$	Output yang diharapkan	meter atau cm/s
$e_{p,m}$	Target	meter atau cm/s
$\sigma$	Standar deviasi	-
$n$	Jumlah data	-
$net$	Output Jaringan Syaraf Tiruan berupa tinggi gelombang	meter atau cm/s
$w_i$	Bobot jaringan dari <i>input layer</i> ke <i>hidden layer</i>	-
$x_i$	Input yang digunakan pada Jaringan Syaraf Tiruan berupa arah angin dan kecepatan angin	knot atau m/s dan degree
$SSE$	<i>Sum Square Error</i> atau evaluasi pada proses pelatihan	-

<b><i>RMSE</i></b>	<i>Root Mean Square Error</i> atau parameter untuk mengevaluasi hasil prediktor	-
<b><i>MSE</i></b>	<i>Mean Square Error</i> atau parameter untuk mengevaluasi hasil prediktor	-
<b><i>MAPE</i></b>	<i>Mean Absolute Percentage Error</i> atau parameter untuk mengevaluasi hasil prediktor	%
<b><i>r</i></b>	<i>Pearson Correlation</i> atau korelasi hubungan antar variabel	-
<b><i>x'</i></b>	Data hasil normalisasi	-
<b><i>x</i></b>	Data asli	meter atau cm/s
<b><i>x<sub>max</sub></i></b>	Nilai maksimum data asli	meter atau cm/s
<b><i>x<sub>min</sub></i></b>	Nilai minimum data asli	meter atau cm/s
<b><i>gH<sub>s</sub></i></b>	Tinggi gelombang signifikan	m
<b><i>gF</i></b>	Panjang daerah yang dipengaruhi angin (daerah Fetch (F))	m
<b><i>U<sub>A</sub></i></b>	Faktor tahanan angin	m/s
<b><i>U</i></b>	Kecepatan angin	m/s

## DAFTAR ISTILAH

### ***Backpropagation***

Salah satu algoritma pembelajaran untuk memperkecil tingkat *error* dengan cara menyesuaikan bobotnya berdasarkan perbedaan output dan target yang diinginkan.

### **Buoy**

Wahana apung berisi sensor-sensor pengukuran yang diteliti oleh Dr. Ir. Syamsul Arifin, M.T. Jenis wahana apung ini merupakan modifikasi dari penelitian sebelumnya sehingga dinamakan tipe II

### **Fungsi aktivasi**

Keluaran model jaringan dengan menggunakan ambang batas tertentu

### ***Genetic Algorithm (GA)***

Algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi

### ***Hidden layer***

Lapisan yang terletak di antara input dan output Jaringan Saraf Tiruan untuk meningkatkan kapabilitas komputasi

### ***Hidden node***

Unit-unit tersembunyi yang ada di hidden layer Jaringan Saraf Tiruan

### **Tinggi gelombang signifikan**

Tinggi rata-rata gelombang sepertiga tertinggi dalam spektrum gelombang



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Indonesia sebagai negara kepulauan terbesar di dunia memiliki 17.499 pulau dari Sabang hingga Merauke. Luas total wilayah Indonesia adalah 7,81 juta km<sup>2</sup> yang terdiri dari 2,01 juta km<sup>2</sup> daratan, 3,25 juta km<sup>2</sup> lautan, dan 2,55 juta km<sup>2</sup> Zona Ekonomi Eksklusif (Kementerian Kelautan dan Perikanan Republik Indonesia, 2017). Luas perairan Indonesia lebih besar dari pada luas daratan, maka dari itu disebut sebagai negara maritim. Lautan di Indonesia mempunyai peran penting dalam bidang perekonomian, transportasi, dan wisata bagi masyarakat Indonesia. Transportasi laut berperan dalam kegiatan perekonomian dan sarana transportasi untuk menjangkau dan menghubungkan pulau-pulau di wilayah nusantara (Hakim, 2011).

Peningkatan aktifitas transportasi laut mengakibatkan terjadinya kecelakaan laut. Direktorat KPLP Ditjen Hubla menyatakan bahwa dalam kurun waktu 5 tahun mulai tahun 2007 hingga 2011 sebanyak 43,6% kecelakaan laut disebabkan karena faktor alam, 28,9% disebabkan karena faktor teknis, dan 27,5% disebabkan karena faktor manusia. Kasus tenggelamnya kapal sebanyak 150 kasus dan 28 kasus kapal terbalik sepanjang tahun 2014. Kasus tenggelamnya kapal motor Marina pada tahun 2015 yang disebabkan murni karena faktor alam yaitu tingginya gelombang laut yang melebihi tinggi kapal (Saputra, 2016). Informasi mengenai kondisi cuaca maritim terkini sangatlah diperlukan, terutama informasi mengenai ketinggian gelombang yang sering menjadi penyebab terjadinya kecelakaan laut.

Ketinggian gelombang air laut dipengaruhi oleh beberapa faktor diantaranya yaitu kecepatan angin, kecepatan arus laut, dan arah angin. Cuaca yang mudah berubah secara tidak beraturan akan mengganggu sarana transportasi laut, seperti gangguan angin kencang dan gelombang tinggi akan membahayakan keselamatan pelayaran. Informasi mengenai cuaca maritim tersebut sangat penting untuk diketahui sebelum melakukan aktifitas laut. Fungsi

utama dari informasi cuaca bagi pelayaran adalah memberi petunjuk pemilihan jalan agar kapal dapat berlayar dengan aman. Kajian mengenai prediksi cuaca maritim dilakukan terus-menerus khususnya ketinggian gelombang air laut untuk mendapatkan model prediktor yang sesuai dengan kondisi perairan Indonesia.

Lembaga pemerintahan pemantau cuaca yang ada di Indonesia yaitu Badan Meteorologi Klimatologi dan Geofisika (BMKG) menggunakan metode statistik secara matematis, terdapat beberapa kelemahan salah satunya adalah tingkat keakuratan ramalan yang rendah (Sampurna, 2011). Prediktor cuaca saat ini masih menggunakan pendekatan statistika data *time series* menggunakan model regresi atau ARIMA yang membutuhkan data historis yang banyak dengan hasil akurasi prediksi yang rendah (Aisjah, 2011) (Sudarsono, 2017).

Perkembangan ilmu dan teknologi yang semakin pesat membuat manusia mampu memprediksi cuaca yang akan terjadi baik beberapa jam atau hari ke depan dengan pendekatan baru, yaitu *Artificial Intelligence* (AI). Metode *Artificial Neural Network* atau Jaringan Syaraf Tiruan merupakan salah satu dari bentuk *Artificial Intelligence* (AI) yang membutuhkan waktu komputasional tidak lama dan merupakan metode komputasi yang cocok untuk pemodelan gelombang.

Konsep dasar dari Jaringan Syaraf Tiruan (JST) adalah seperti halnya jaringan syaraf biologi manusia berupa neuron yang terhubung satu sama lain. Karakteristik yang menarik dari JST ini adalah kemampuannya untuk melakukan proses pembelajaran (*learning/training*). Proses *training* pada JST bertujuan untuk mencari bobot-bobot yang konvergen antar lapisan atau *layer* sedemikian hingga bobot yang diperoleh menghasilkan output yang diinginkan (Fausset, 1996). Permasalahan *training* pada JST dalam membentuk jaringan yang mampu mengenali pola data yang diberikan merupakan permasalahan tersendiri bagi peneliti yang akan menerapkan JST dalam suatu teknologi. Hal ini disebabkan belum ada metode dalam menerapkan struktur jaringan terbaik yang mampu mengenali pola dengan akurat tanpa melakukan *trial and error* dalam memperoleh jaringan yang optimal. Jaringan



Syaraf Tiruan *feed forward* merupakan salah satu arsitektur dari JST yang banyak digunakan dalam aplikasi di berbagai bidang. Proses *training* jaringan *feed forward* menggunakan algoritma *Backpropagation* (BP) mempunyai kelemahan dalam hal mencari bobot yang konvergen.

Penggunaan JST untuk prediksi ketinggian gelombang laut sebelumnya pernah dilakukan oleh Rizianiza pada tahun 2015 mengenai prediktor ketinggian gelombang di Perairan Sumatera-Jawa menggunakan JST dengan algoritma pelatihan *Backpropagation* (BP) didapatkan nilai RMSE sebesar 0,066 hingga 0,089 (Rizianiza & Aisjah, 2015).

Penelitian mengenai prediksi ketinggian gelombang lainnya juga pernah dilakukan Raharja pada tahun 2018 menggunakan JST dengan algoritma pelatihan *Backpropagation* (BP) menghasilkan nilai RMSE sebesar 0,065, tetapi penelitian ini memiliki kelemahan yaitu mencapai konvergensi yang lama yakni mencapai 10.000 iterasi (Raharja & Astra, 2018).

Penelitian serupa juga pernah dilakukan oleh Wellyantama pada tahun 2015 menghasilkan korelasi sebesar 0,70-0,78 (Wellyantama, 2015). Penelitian diatas dapat dikatakan bahwa algoritma pelatihan *Backpropagation* (BP) kurang maksimal dalam kecepatan mencapai konvergensi dalam proses pelatihan atau *training*.

Penelitian ini dilakukan mengenai prediksi ketinggian gelombang air laut menggunakan Jaringan Syaraf Tiruan dengan *training* atau pembelajaran menggunakan algoritma Levenberg-Marquardt (LM). Algoritma pelatihan Levenberg-Marquardt (LM) merupakan algoritma JST yang memberikan waktu konvergensi paling cepat dibandingkan dengan algoritma pelatihan yang lain, seperti Gradien Descent, Conjugate Gradient, dll (Sharma & Venugopalan, 2014).

Penelitian mengenai prediksi ketinggian gelombang juga pernah dilakukan oleh Ukhti pada tahun 2018 menggunakan JST-LM dan JST-PSO didapatkan hasil bahwa JST-LM menghasilkan RMSE lebih kecil dari JST-PSO. Nilai RMSE pada JST-LM sebesar 0,059 dan JST-PSO sebesar 0,066 (Ukhti, 2018). Penelitian

tersebut dapat dikatakan bahwa JST yang dioptimasi dengan PSO belum menghasilkan akurasi yang optimal.

Perkembangan teknologi yang semakin canggih mengenai JST yang dikombinasikan dengan optimasi *Genetic Algorithm* (GA) mampu memprediksi lebih bagus dibanding JST biasa. Hasil dari kombinasi JST dan optimasi *Genetic Algorithm* (GA) mampu mendapatkan akurasi prediksi yang lebih tepat dan lebih cepat (Zhang, Wang, Liu, Du, & Lu, 2009). Penelitian Tugas Akhir ini mengenai prediksi ketinggian gelombang dan arus laut dengan metode JST dengan optimasi *Genetic Algorithm* (GA) diharapkan mampu menghasilkan nilai RMSE yang lebih kecil dari penelitian sebelumnya.

## 1.2 Perumusan Masalah

Permasalahan yang diangkat dalam penelitian tugas akhir ini antara lain:

- a. Apakah hasil prediksi ketinggian gelombang menggunakan Jaringan Syaraf Tiruan yang telah dioptimasi dengan *Genetic Algorithm* (GA) menghasilkan nilai *Root Mean Square Error* (RMSE) yang lebih kecil daripada Jaringan Syaraf Tiruan tanpa optimasi?
- b. Apakah hasil prediksi kecepatan arus laut menggunakan Jaringan Syaraf Tiruan yang telah dioptimasi dengan *Genetic Algorithm* (GA) menghasilkan nilai *Root Mean Square Error* (RMSE) yang lebih kecil daripada Jaringan Syaraf Tiruan tanpa optimasi?
- c. Berapakah prediksi ketinggian gelombang air laut secara *real time* dengan metode Jaringan Syaraf Tiruan yang telah dioptimasi menggunakan *Genetic Algorithm* (GA)?
- d. Berapakah prediksi kecepatan arus laut secara *real time* dengan metode Jaringan Syaraf Tiruan yang telah dioptimasi menggunakan *Genetic Algorithm* (GA)?

## 1.3 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai dalam pengerjaan tugas akhir ini antara lain:

- a. Memperoleh prediktor ketinggian gelombang air laut terbaik dengan nilai *Root Mean Square Error* (RMSE) terkecil.
- b. Memperoleh prediktor kecepatan arus laut terbaik dengan nilai *Root Mean Square Error* (RMSE) terkecil.
- c. Memperoleh prediksi ketinggian gelombang air laut secara *real time* dengan metode Jaringan Syaraf Tiruan yang telah dioptimasi menggunakan *Genetic Algorithm* (GA).
- d. Memperoleh prediksi kecepatan arus laut secara *real time* dengan metode Jaringan Syaraf Tiruan yang telah dioptimasi menggunakan *Genetic Algorithm* (GA).

#### **1.4 Batasan Masalah**

Batasan masalah yang digunakan dalam pengerjaan penelitian tugas akhir ini antara lain:

- a. Variabel yang mempengaruhi ketinggian gelombang air laut adalah kecepatan angin dan arah angin sedangkan variabel yang mempengaruhi kecepatan arus laut adalah kecepatan angin, tinggi gelombang, dan suhu permukaan laut. Penggunaan variabel diatas dikarenakan adanya ketersediaan data dan tingkat korelasi antar variabel.
- b. Ketinggian gelombang dan arus laut yang diprediksi yaitu ketinggian gelombang laut signifikan dan arus laut di sekitar Selat Madura (koordinat 114,081° BT dan 7,368° LS).
- c. Metode yang digunakan untuk memprediksi ketinggian gelombang air laut dan arus laut adalah metode Jaringan Syaraf Tiruan dengan algoritma pelatihan Levenberg Marquardt (LM).
- d. Metode optimasi yang digunakan adalah *Genetic Algorithm* (GA) untuk menentukan bobot optimal yang digunakan pada Jaringan Syaraf Tiruan.

*Halaman ini sengaja dikosongkan*

## BAB II TINJAUAN PUSTAKA

### 2.1 Gelombang Laut

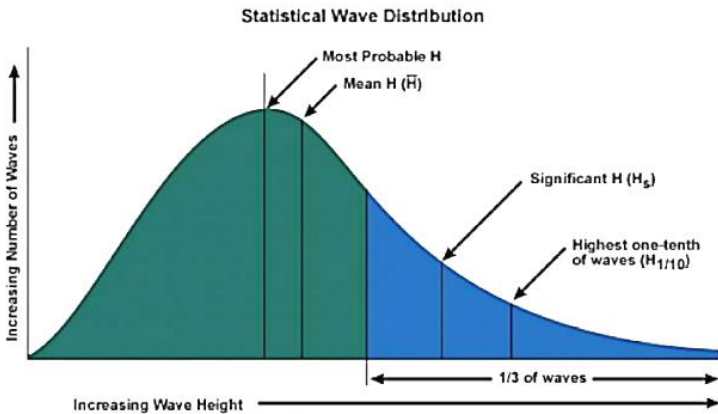
Gelombang laut adalah pergerakan naik dan turunnya air dengan arah tegak lurus permukaan air laut yang membentuk kurva/grafik sinusoidal. Gelombang di lautan dapat diklasifikasikan menjadi beberapa macam tergantung pada gaya pembangkitnya. Pembangkit gelombang laut dapat disebabkan oleh: angin (gelombang angin), gaya tarik menarik bumi bulan matahari (gelombang pasang-surut), gempa (vulkanik atau tektonik) di dasar laut (gelombang tsunami), ataupun gelombang yang disebabkan oleh gerakan kapal (Ainsworth T, 2017).

Gelombang yang paling berpengaruh dalam dunia pelayaran adalah gelombang  $H_{1/3}$ . Gelombang  $H_{1/3}$  merupakan rata-rata 33.3% gelombang tertinggi dari pencatatan gelombang.  $H_{1/3}$  ini disebut juga ketinggian gelombang signifikan, dimana  $H_{1/3}$  ini adalah variabel yang sangat berpengaruh terhadap kelayakan pelayaran (Ainsworth T, 2017). Ketinggian gelombang laut signifikan dikelompokkan menjadi 10 kategori berdasarkan standar *Word Meteorological Organization* (WMO) yang ditunjukkan pada Tabel 2.1 di bawah ini.

**Tabel 2. 1** Standar Ketinggian Gelombang (*Word Meteorological, 1998*)

Kode	Tinggi Gelombang (m)	Nama Karakteristik
0	0	<i>Glassy</i>
1	0 - 0,1	<i>Rippled</i>
2	0,1 - 0,5	<i>Smooth</i>
3	0,5 – 1,25	<i>Slight</i>
4	1,25 – 2,5	<i>Moderate</i>
5	2,5 – 4	<i>Rough</i>
6	4 – 6	<i>Very Rough</i>
7	6 – 9	<i>High</i>
8	9 - 14	<i>Very High</i>
9	>14	<i>Phenomenal</i>

Gelombang yang umumnya terjadi di laut yaitu kategori *smooth* dan *slight* sedangkan gelombang yang terjadi di pinggir laut yaitu kategori *rippled*. Semakin besar tinggi gelombang maka semakin berbahaya untuk melakukan aktifitas transportasi laut khususnya pelayaran. Distribusi gelombang laut dengan ketinggian yang berbeda akan menghasilkan grafik kurva lonceng ditunjukkan pada Gambar 2.1. Grafik menunjukkan jumlah gelombang kecil yang relatif rendah (sisi kiri grafik) dan sejumlah kecil gelombang sangat besar (sisi kanan grafik).



**Gambar 2. 1** Distribusi Gelombang Laut (Chakraborty, 2018)

Gelombang yang sering terjadi adalah gelombang yang dibangkitkan oleh angin. Angin adalah udara yang bergerak akibat adanya perbedaan tekanan udara dari tempat bertekanan tinggi ke tempat bertekanan rendah atau dari daerah bersuhu rendah ke daerah bersuhu tinggi. Klasifikasi kecepatan angin yang digunakan oleh *World Meteorological Organization* (WMO) adalah skala *beaufort*. Skala *beaufort* adalah sistem yang menafsirkan laporan kecepatan angin berdasarkan efek yang ditimbulkan dari kecepatan angin. Skala *beaufort* diciptakan oleh Sir Francis Beaufort pada tahun 1805. Skala *beaufort* dibagi menjadi 13 jenis kecepatan angin yang ditunjukkan pada Tabel 2.2.

**Tabel 2. 2** Parameter Kecepatan Angin Skala Beaufort (*Word Meteorological Organization, 1998*)

<b>Kekuatan Angin</b>	<b>Kecepatan Angin</b>		<b>Nama</b>
<b>Skala Beaufort</b>	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>

Gelombang laut akan semakin besar apabila kekuatan angin semakin besar pula. Menurut Sverdrup dan Munk terjadinya gelombang disebabkan adanya *stress* dari angin yang bekerja pada permukaan laut. Proses prediksi ketinggian gelombang laut berdasarkan data angin diperlukan tiga parameter, yaitu kecepatan angin ( $u$ ), panjang daerah yang dipengaruhi angin (daerah Fetch ( $F$ )), lamanya angin bertiup (durasi angin ( $t$ )), dan peta daerah kajian (Triatmodjo, 1999). Prediksi ketinggian gelombang bergantung pada data nilai kecepatan angin. Persamaan mengenai tinggi gelombang yang dipengaruhi oleh kecepatan angin ditunjukkan pada persamaan (2.1) dan (2.2) (Triatmodjo, 1999).

$$U_A = 0.71 U^{1.23} \quad (2.1)$$

$$gH_s = U_A^2 \times 1.6 \times 10^{-3} \left( \frac{gF}{U_A^2} \right)^{1/2} \quad (2.2)$$

## 2.2 Arus Laut

Arus laut adalah gerakan massa air dari suatu tempat (posisi) ke tempat yang lain. Energi yang menggerakkan massa air laut tersebut berasal dari matahari. Perbedaan pemanasan matahari terhadap permukaan bumi menimbulkan pula perbedaan energi yang diterima permukaan bumi. Perbedaan ini menimbulkan fenomena arus laut dan angin yang menjadi mekanisme untuk menyeimbangkan energi di seluruh muka bumi. Kedua fenomena ini juga saling berkaitan satu dengan yang lain. Angin merupakan salah satu gaya utama yang menyebabkan timbulnya arus laut selain gaya yang timbul akibat dari tidak samanya pemanasan dan pendinginan air laut. Sirkulasi dari arus laut terbagi atas dua kategori yaitu sirkulasi di permukaan laut (*surface circulation*) dan sirkulasi di dalam laut (*intermediate or deep circulation*). Arus pada sirkulasi di permukaan laut didominasi oleh arus yang ditimbulkan oleh angin sedangkan sirkulasi di dalam laut didominasi oleh arus termohalin. Arus termohalin timbul sebagai akibat adanya perbedaan densitas karena berubahnya suhu dan salinitas massa air laut.

Arus laut permukaan merupakan gerakan massa air yang disebabkan oleh angin yang berhembus di permukaan laut pada kedalaman kurang dari 200 m yang berpindah dari satu tempat yang bertekanan udara tinggi ke tempat lain yang bertekanan udara rendah yang sangat luas dan terjadi pada seluruh lautan di dunia.

## 2.3 Jaringan Syaraf Tiruan

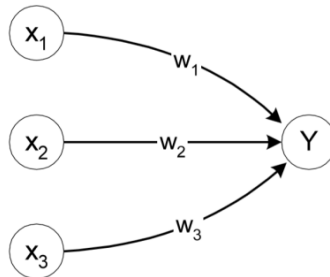
Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* merupakan salah satu metode *Artificial Intelligence* yang banyak digunakan dalam proses prediksi. Konsep dasar dari JST ini seperti jaringan syaraf biologis pada manusia yang banyak terdapat neuron yang saling terhubung satu sama lain. Sistem ini kemudian digunakan oleh organisme untuk mengolah data yang didapatkan dari lingkungan untuk mengambil keputusan lebih lanjut. Model JST 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).

JST dalam proses pengenalan pola sinyal melalui proses pembelajaran atau *training*. Sistem pembelajaran ini melibatkan penyesuaian koneksi antar neuron (Shaban dkk, 2016). JST memiliki karakteristik yaitu kemampuan beradaptasi, mengorganikkan diri sendiri, pembelajar, *non-linearity*, *non-locality*, *non-steady*, dan *non-convex* sehingga menjadikan metode ini dapat mengatasi masalah yang kompleks. Metode JST dapat digunakan untuk (Yang, Chen, Wang, Li, & Li, 2016) keperluan pemrosesan sinyal, pengendalian, pengenalan pola, pengobatan, penghasil suara, pengenalan suara, dan bisnis. Penggunaan JST ditentukan oleh tiga hal, yaitu (Fausset, 1996):

1. Pola hubungan antar neuron (arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung.
3. Fungsi aktivasi.

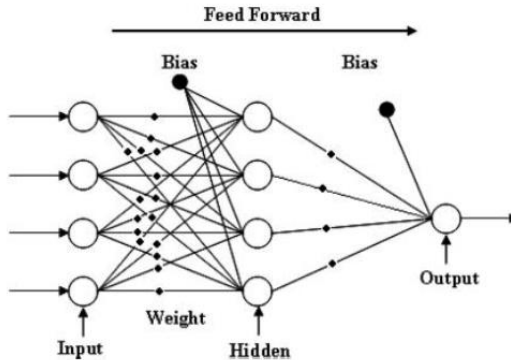


**Gambar 2. 2** Hubungan Antar Neuron dalam Arsitektur JST (Fausset, 1994)

Ketiga aspek di atas ditunjukkan pada Gambar 2.2 dimana contoh persamaan neuron  $Y$  adalah persamaan (2.3). Neuron  $Y$  merupakan hasil dari penjumlahan input neuron  $x_1$ ,  $x_2$ , dan  $x_3$  dengan bobot masing-masing  $w_1$ ,  $w_2$ , dan  $w_3$ . *Impuls* yang diterima  $Y$  mengikuti fungsi aktivasi  $y = f(\text{net})$ . Fungsi aktivasi juga dapat mengubah bobot yang terbaik.

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 \quad (2.3)$$

JST terdiri dari interkoneksi elemen komputasi yang disebut neuron. Setiap neuron menggabungkan input, menambahkan bias, dan menentukan kekuatan kombinasi kemudian mengeluarkan hasilnya sebanding dengan bobot tersebut. Aplikasi yang dibuat dalam rekayasa laut sejauh ini sebagian besar melibatkan jenis *feed forward* dari jaringan sebagai *feed back* atau *recurrent*. Jaringan *multilayer feed forward* terdiri dari lapisan input (*input layer*), satu atau lebih lapisan tersembunyi (*hidden layer*), dan lapisan output (*output layer*). Pelatihan jaringan melibatkan penentuan nilai bobot koneksi dan bias melalui proses iteratif yang berdasarkan algoritma pelatihan (Jain, Deo, & Rajendran, 2011).

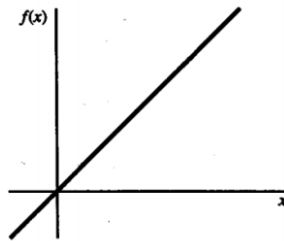


**Gambar 2. 3** Jaringan Tipe *Feed Forward* (Jain, Deo, & Rajendran, 2011)

Macam-macam fungsi aktivasi yaitu sigmoid biner (*logsig*), sigmoid bipolar (*tansig*), dan linier (*pureline*). Fungsi aktivasi linier memiliki keluaran yang sama terhadap masukannya. Persamaan fungsi aktivasi linier ditunjukkan pada persamaan (2.4) dengan turunan persamaan (2.5). Fungsi aktivasi linier dapat ditunjukkan pada Gambar 2.4 (Fausset, 1996).

$$f(x) = x \quad (2.4)$$

$$f'(x) = 1 \quad (2.5)$$

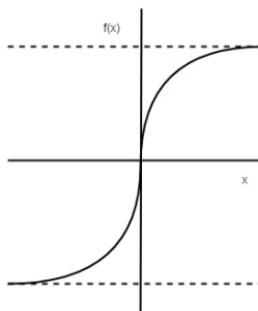


**Gambar 2. 4** Fungsi Aktivasi Linier (Fausset, 1996)

Fungsi aktivasi sigmoid bipolar memiliki kisaran -1 hingga 1. Persamaan fungsi sigmoid bipolar dapat dilihat pada persamaan (2.6) dengan turunan persamaan (2.7). Grafik sigmoid bipolar ada pada Gambar 2.5 (Fausset, 1996).

$$f(x) = \frac{2}{1+e^{-\sigma t}} - 1 \quad (2.6)$$

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)] \quad (2.7)$$



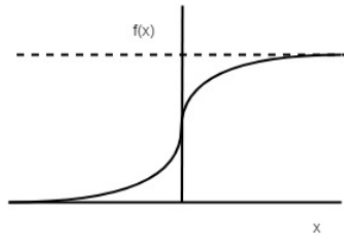
**Gambar 2. 5** Fungsi Aktivasi Sigmoid Bipolar (Fausset, 1994)

Fungsi aktivasi sigmoid biner bertindak sebagai fungsi yang mengompres fungsi input saat dibutuhkan nilai positif yang besar atau nilai negatif yang besar. Nilai positif yang besar secara asimtotik mendekati satu, sementara nilai negatif yang besar

terkompres ke nol (Fausset, 1996). Fungsi aktivasi sigmoid biner ditunjukkan pada persamaan (2.6) dengan turunan persamaan (2.7). Grafik fungsi aktivasi sigmoid biner ditunjukkan pada Gambar 2.6.

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.8)$$

$$f'(x) = f(x)(1 - f(x)) \quad (2.9)$$



**Gambar 2. 6** Fungsi Aktivasi Sigmoid Biner (*Fausset, 1996*)

#### 2.4 Algoritma Pelatihan Levenberg Marquardt

Algoritma pelatihan yang dapat digunakan untuk melakukan pelatihan JST terdapat beberapa tipe algoritma diantaranya yaitu algoritma Gradient Descent, Conjugate Gradient, dan Quasi Newton. Algoritma Levenberg Marquardt yang merupakan bagian dari tipe algoritma Quasi Newton. Algoritma pelatihan Levenberg Marquardt (LM) dianggap sebagai gabungan dari Steepest Descent dan Gauss-Newton.

Algoritma Steepest Descent dianggap kurang efisien karena lambatnya untuk mencapai konvergensi, sedangkan algoritma Gauss-Newton dapat melakukan evaluasi gradien permukaan kesalahan dengan menggunakan fungsi turunan orde kedua sehingga menemukan ukuran langkah yang tepat dan konvergen dengan cepat. Algoritma pelatihan LM sebagai gabungan dari kedua metode tersebut bekerja dengan konsep apabila hasil prediksi jauh dari target maka algoritma bekerja sebagai algoritma Steepest Descent, sedangkan apabila hasil prediksi mendekati target maka akan bekerja sebagai algoritma Gauss Newton (Yaqub, Eren, & Eyupoglu, 2016).

Algoritma pelatihan LM merupakan pengembangan dari algoritma *backpropagation* standar. Penggunaan algoritma *backpropagation* dalam proses *update* bobot dan bias menggunakan *negative gradient descent* secara langsung sedangkan algoritma LM menggunakan pendekatan matriks Hessian (H) dengan parameter koefisien kombinasi. Matriks Hessian berperan memberikan perhitungan nilai perubahan bobot pada jaringan. Matriks Hessian ditunjukkan pada persamaan (2.8). Pembaruan bobot mendekati algoritma Gauss Newton apabila  $\mu$  sangat kecil (mendekati nol), sedangkan pembaruan bobot mendekati algoritma Steepest Descent apabila ketika  $\mu$  sangat besar. Pembaruan bobot pada algoritma LM ditunjukkan pada persamaan (2.10) dan (2.11).

$$H \approx J^T J + \mu I \quad (2.10)$$

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \quad (2.11)$$

Proses perhitungan untuk matriks Jacobian pada algoritma LM dapat diatur sesuai dengan perhitungan *backpropagation* tradisional dalam algoritma orde pertama tetapi untuk setiap pola hanya satu proses *backpropagation* yang diperlukan, sedangkan dalam algoritma LM proses *backpropagation* harus diulang. Perhitungan matriks Jacobian, input *node*  $y$  dapat dihitung dalam *forward computation*. Parameter *error*  $\delta$  diperoleh dalam perhitungan mundur atau *backward computation* (Yu & Wilamowski, 2011). Proses pelatihan algoritma LM melalui 6 tahap yang dituliskan dari a sampai f berikut ini:

- a. Evaluasi *sum square error* (SSE) menggunakan bobot awal. Penurunan persamaan algoritma ditunjukkan pada persamaan (2.12) dan (2.13).

$$E(x, w) = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M e^2_{p, m} \quad (2.12)$$

$$e_{p, m} = d_{p, m} - o_{p, m} \quad (2.13)$$

- b. Pembaruan bobot dilakukan berdasarkan algoritma LM.
- c. Evaluasi SSE dengan bobot baru.
- d. SSE baru meningkat, maka lakukan langkah (mengatur ulang vektor bobot ke nilai tertentu) dan meningkatkan nilai  $\mu$  dengan faktor 10 atau faktor lainnya, kemudian lanjutkan ke langkah b dan perbarui lagi.
- e. SSE baru menurun, maka lakukan langkah (menjaga vektor bobot baru sebagai yang sekarang) dan menurunkan nilai  $\mu$  dengan faktor 10 atau faktor lainnya.
- f. Proses dilanjutkan ke langkah b dengan bobot baru dan SSE baru yang lebih kecil dari nilai yang ditargetkan atau kondisi yang diinginkan sudah terpenuhi.

## 2.5 Optimasi *Genetic Algorithm* (GA)

Optimasi merupakan suatu tindakan untuk menentukan hasil terbaik dalam suatu keadaan yang diberikan. Optimasi dapat didefinisikan sebagai proses untuk menemukan keadaan yang memberikan fungsi nilai maksimal atau minimal, di mana usaha yang diperlukan dan keuntungan yang diinginkan tersebut dapat dinyatakan dalam fungsi dari variabel keputusan tertentu. Macam-macam metode optimasi diantaranya genetika, optimasi fuzzy, *particle swarm optimization*, dan optimasi simulasi *annealing*.

Optimasi *Genetic Algorithm* merupakan salah satu jenis optimasi genetika yang berkembang sangat pesat dan menghasilkan beberapa algoritma turunan diantaranya *Differential Evolutionary*, *Evolution Strategies*, *Evolutionary Programming*, *Genetic Programming*, *Grouping Genetic Algorithm* dan *Interactive Evolutionary Algorithm*. *Genetic Algorithm* (GA) diusulkan pertama kali pada tahun 1975 oleh John Henry Holland banyak digunakan di berbagai bidang ilmu pengetahuan terutama untuk menyelesaikan permasalahan optimasi (Holland, 1992). Beberapa kelebihan optimasi *Genetic Algorithm* (GA) adalah sebagai berikut:

1. Cakupan pencarian solusi yang cukup lebar memungkinkan GA menemukan solusi global optimum secara mudah. Keragaman (*diversity*) populasi awal menentukan cakupan

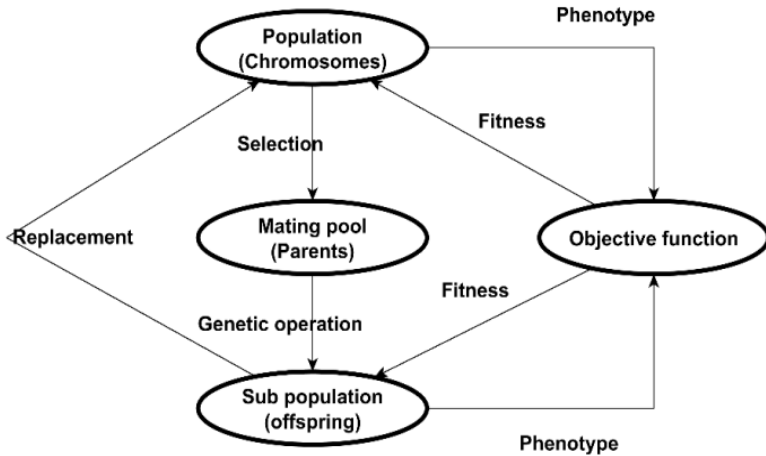
pencarian solusi, sehingga populasi awal disusun dari solusi-solusi yang cukup jauh jaraknya.

2. Kecepatan proses optimasi yang tinggi apabila dibandingkan dengan beberapa metode optimasi lain. Metode optimasi GA dapat menemukan solusi global optimum dalam waktu relatif singkat.
3. Dimensi yang tinggi walaupun pada awalnya diusulkan untuk masalah optimasi yang sederhana, pada perkembangannya GA juga mampu menyelesaikan masalah optimasi berdimensi tinggi.
4. Hasil optimasi menghasilkan stabilitas yang baik, sama halnya seperti metode optimasi heuristik lainnya, metode optimasi GA juga membutuhkan penentuan beberapa konstanta di awal iterasi. Metode optimasi GA dapat menunjukkan hasil yang sama, walaupun iterasi dimulai dengan konstanta yang berbeda.

Optimasi *Genetic Algorithm* (GA) merupakan salah satu metode optimasi berdasarkan ilmu biosains yang menerapkan sistem seleksi alam dan genetika alami. Sistem seleksi alam akan dilakukan terus pada populasi secara acak untuk mendapatkan solusi dalam menyelesaikan suatu masalah. Solusi dalam suatu populasi akan dievaluasi hingga mendapatkan solusi terbaik. Algoritma ini mensimulasikan proses evolusi sesuai jumlah generasi yang dikehendaki (Man, Tang, & Kwong, 1999).

Siklus optimasi GA ditunjukkan pada Gambar 2.7. Optimasi GA diawali dengan menginisialisasi populasi. Populasi akan terbentuk pola kromosom, jumlah gen, jumlah bit, dan jumlah variabel. Kromosom memiliki deret angka yang memiliki sifat genotip. Sifat genotip ini yang akan terjadi proses pindah silang (*crossover*) dan mutasi. Pindah silang merupakan pemotongan dua buah kromosom berdasarkan jumlah titik yang diinginkan. Hasil pemotongan kromosom digabungkan dengan pasangan lainnya, sedangkan mutasi adalah penggantian gen pada kromosom dengan gen lain dari kromosom lain. Sifat baru yang telah dilakukan decode kromosom disebut sifat fenotip. Sifat fenotip digunakan untuk mengetahui kualitas kromosom. Kromosom berevolusi terus hingga didapatkan kriteria yang sesuai atau disebut *fitness*. Proses

tersebut dilakukan berulang-ulang sampai mendapatkan *fitness* yang terbaik (K.F. Man, 2001).



**Gambar 2. 7** Siklus Optimasi *Genetic Algorithm* (GA) (Man, Tang, & Kwong, 1999)

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup (Kusumadewi & Purnomo, 2005). Terdapat 4 kondisi yang mempengaruhi proses evaluasi yaitu sebagai berikut:

1. Kemampuan organisme untuk melakukan reproduksi.
2. Keberadaan populasi organisme yang bisa melakukan reproduksi.
3. Keberadaan organisme dalam suatu populasi.
4. Perbedaan kemampuan untuk *survive*.



Optimasi GA dapat digunakan untuk beberapa permasalahan, salah satunya adalah JST. Langkah pembelajaran optimasi GA yang dikombinasikan dengan JST adalah sebagai berikut (Zhang, Wang, Liu, Du, & Lu, 2009):

1. Ukuran populasi, *crossover*, mutasi, lapisan jaringan, dan jumlah neuron ditentukan.
2. Inisialisasi populasi awal secara acak  $P = \{x_1, x_2, \dots, x_n\}$  vcv, n adalah jumlah bobot koneksi, sedangkan P terdiri dari n vektor bobot dan vektor batas ambang (*threshold*). Populasi awal dalam penelitian ini berupa nilai bobot hasil keluaran JST yang telah didapatkan berdasarkan jumlah hidden node yang digunakan.
3. Setiap bobot yang ada dilakukan perhitungan nilai fitness yaitu berupa nilai RMSE.
4. *Fitness* diurutkan dari kecil ke besar (dari kromosom terbaik sampai ke terburuk) dengan nilai RMSE terkecil.
5. Kromosom terbaik tidak diseleksi atau tetap dijaga (kromosom terbaik didapatkan dari nilai *fitness* yang terbesar), dimana bobot yang memiliki RMSE terkecil tetap dijaga dan tidak diseleksi.
6. *Crossover* dan evaluasi kelompok individu yang baru dilakukan. Bobot-bobot hasil seleksi kemudian dilakukan penyilangan untuk mendapatkan bobot baru.
7. Mutasi dan evaluasi kelompok individu yang baru dilakukan.
8. Proses dilakukan ulang menuju generasi selanjutnya dengan mengulang langkah 3-6.

Langkah di atas dilakukan secara berulang hingga mendapatkan bobot JST terbaik berdasarkan nilai *fitness* terbesar atau biasa disebut sebagai kromosom terbaik.

## 2.6 Buoyweather Station Type II

Stasiun *buoyweather* yaitu pelampung tetap yang terapung di laut dengan dilengkapi instrumen untuk mengukur sejumlah parameter meteorologi. *Buoyweather Station Type II* yang digunakan untuk pengambilan data dapat mengukur variabel cuaca antara lain suhu udara, suhu permukaan laut, kecepatan angin, arah

angin, tekanan udara, curah hujan, ketinggian gelombang, dan kelembaban udara. *Buoyweather Station Type II* dapat ditunjukkan pada Gambar 2.8.



**Gambar 2. 8** *Buoyweather Station Type II*

## 2.7 Validasi Prediktor

Validasi prediktor dilakukan dari proses pengujian jaringan prediktor yang telah dibuat. Hasil pengujian akan menunjukkan nilai performansi dari prediktor. Performansi jaringan prediktor dapat diukur dengan menggunakan nilai *Root Mean Square Error* (RMSE) dan MAPE (*Mean Absolute Percentage Error*). Perhitungan RMSE dan MAPE ditunjukkan pada persamaan (2.15) dan (2.16) (Yang, Chen, Wang, Li, & Li, 2016). Nilai RMSE digunakan sebagai parameter performansi prediktor dikarenakan dapat menggambarkan nilai *error* yang dihasilkan dari prediktor, berbeda dengan MSE yang merupakan nilai kuadrat dari *error* yang dihasilkan (Rizianiza, 2017) .

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x})^2} \quad (2.15)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}}{x_i} \right| \quad (2.16)$$

**Tabel 2. 3** Tingkat Akurasi Prediktor berdasarkan MAPE

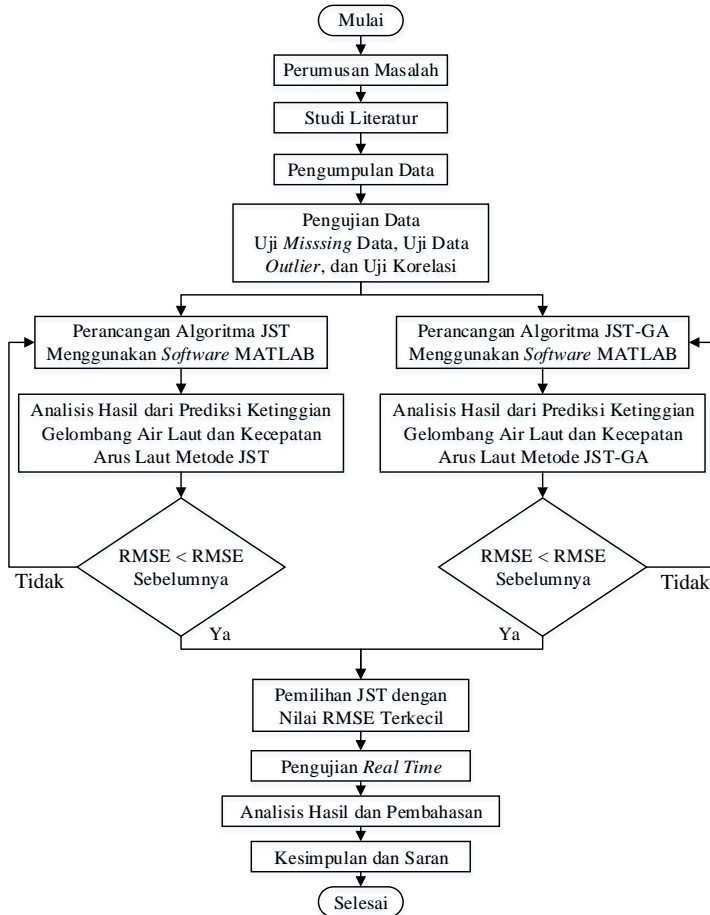
<b>MAPE</b>	<b>Tingkat Akurasi</b>
Kurang dari 10%	Sangat Akurat
11% - 20%	Baik
21% - 50%	Wajar
51% atau lebih	Tidak Akurat

Nilai parameter di atas pada Tabel 2.3 merepresentasikan nilai prediksi dari arsitektur jaringan JST yang telah dibuat. Hasil prediksi dikatakan baik dan akurat apabila nilai RMSE dan nilai MAPE semakin kecil.

*Halaman ini sengaja dikosongkan*

### BAB III METODE PENELITIAN

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



Gambar 3.1 Diagram Alir Penelitian

### 3.1 Perumusan Masalah

Perumusan masalah merupakan suatu kegiatan untuk mengobservasi masalah dan potensi penelitian yang ada. Rumusan masalah ini didapatkan dari beberapa kajian penelitian yang telah dilakukan sebelumnya. Kajian mengenai penggunaan metode JST dengan algoritma pelatihan *backpropagation* mengenai prediksi ketinggian gelombang memiliki kelemahan yaitu banyaknya jumlah iterasi yang diperlukan untuk mencapai konvergensi yaitu mencapai 10.000 iterasi (Raharja & Astra, 2018). Kajian mengenai metode JST dengan algoritma pelatihan LM menghasilkan konvergensi yang cepat (Sharma & K. Venugopalan, 2014). Kajian mengenai penggunaan optimasi PSO pada JST prediksi gelombang memiliki kelemahan yaitu optimasi PSO belum optimal dan nilai RMSE bisa diperkecil lagi (Ukhti, 2018). Kajian mengenai penelitian tersebut telah dituliskan pada Bab I bagian pendahuluan

Permasalahan pada penelitian tugas akhir ini yaitu performansi dari prediktor ketinggian gelombang air laut dan kecepatan arus laut. Penelitian ini membandingkan nilai RMSE dari JST dengan algoritma pelatihan Levenberg Marquardt (JST-LM) dengan JST yang dioptimasi dengan *Genetic Algorithm* (JST-GA). Penelitian ini diharapkan dapat menghasilkan nilai RMSE yang lebih kecil dari penelitian sebelumnya.

### 3.2 Studi Literatur

Studi literatur dilakukan untuk mengembangkan ide agar lebih sistematis melalui referensi yang ada serta untuk memahami prinsip dasar dari metode yang dibuat. Studi literatur pada penelitian ini adalah mencari variabel-variabel independen yang berpengaruh pada variabel dependen yaitu ketinggian gelombang air laut dan kecepatan arus laut. Kajian studi literatur yang relevan dengan tema penelitian yaitu hasil penelitian terdahulu dalam tema yang sama yaitu prediktor ketinggian gelombang, teorema tentang gelombang, arus laut, Jaringan Syaraf Tiruan, optimasi *Genetic Algorithm* (GA), Simulink Matlab yang diperoleh dari beberapa sumber diantaranya jurnal, artikel, dan *handbook*.

### 3.3 Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan data penunjang prediksi yang akan dilakukan, baik itu data dependen maupun data independen. Pengumpulan data dilakukan melalui literatur dan lembaga-lembaga terkait yang memiliki data tersebut seperti Badan Meteorologi Klimatologi dan Geofisika (BMKG). Terdapat dua macam pengumpulan data yaitu pengumpulan data dari BMKG dan data pengambilan *real* dari Buoyweather di Pantai Kenjeran. Data yang dikumpulkan dari BMKG berupa kecepatan angin, arah angin, arus laut, dan ketinggian gelombang air laut signifikan dari kantor BMKG Maritim II Perak, Surabaya. Rentang data yang didapatkan yaitu dari Januari 2017 hingga Desember 2018. Data ketinggian gelombang dan kecepatan arus laut yang didapatkan yaitu pada koordinat 114,081° BT dan 7,368° LS. Data yang didapatkan dari BMKG yaitu sebanyak 15.053 data per jam.

Pengambilan data *realtime* Buoyweather dilakukan di Pantai Kenjeran selama tiga hari yaitu 11 Juni 2019, 15 Juni 2019, dan 18 Juni 2019. Data *realtime* dari *Buoyweather* ditunjukkan pada Tabel 3.1 di bawah ini.

**Tabel 3. 1** Data *Realtime Buoyweather*

Tanggal	Kecepatan Angin (m/s)	Arah Angin (derajat)	Suhu Permukaan Laut (°C)	Tinggi Gelombang (cm)
11 Juni 2019	4,96	45,00	45,04	9,15
	3,90	44,25	19,73	9,15
	4,01	66,78	28,84	9,17
	2,02	99,21	28,94	9,11
15 Juni 2019	2,17	78,71	28,89	9,15
	2,12	113,75	28,68	9,12
	1,16	114,03	29,02	9,06
	1,87	59,56	29,15	9,13
18 Juni 2019	1,45	96,59	28,97	10,84
	1,66	76,81	29,00	9,12
	1,49	105,40	28,34	9,13
	1,38	52,59	27,15	9,08

### 3.4 Pengujian Data

Pengujian data dilakukan dalam tiga cara yaitu uji *missing* data, uji data *outlier*, dan uji korelasi data. Penjelasan mengenai uji tersebut dituliskan pada sub bab di bawah ini.

#### 3.4.1 Uji *Missing* Data

Uji *missing* data merupakan suatu pengujian terhadap data yang tidak tersedia karena alasan tertentu. *Missing* data tersebut disebabkan karena informasi suatu objek tidak diberikan, sulit dicari, dan memang informasi tersebut tidak ada. *Missing* data tidak bermasalah jika jumlahnya hanya sedikit, misalnya jika hanya 1% data *missing* dari total data yang ada, namun jika persentasenya cukup besar maka data tidak layak untuk diproses. Batasan persentase yang dipakai pada penelitian ini adalah 1% *missing* data. Pengujian dilakukan dengan mencari *cell* mana yang tidak ada datanya atau kosong. Pencarian data yang kosong dibantu dengan *software* IBM SPSS 16.0 secara otomatis. Langkah yang harus dilakukan adalah klik *Analyze - Descriptive Statistics - Frequencies* - semua variabel dipilih - OK.

Data yang *missing* atau hilang tersebut kemudian dilakukan *imputation* data atau pengisian data menggunakan metode interpolasi linier dengan rumus pada persamaan (3.1).

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \quad (3.1)$$

#### 3.4.2 Uji Data *Outlier*

Data *outlier* merupakan data yang benar-benar berbeda dari data-data pada umumnya. Data *outlier* bisa terjadi karena ada beberapa hal yaitu kesalahan dalam pemasukan data, kesalahan pada pengambilan sampel, dan karena memang ada data-data ekstrem yang tidak bisa dihindarkan keberadaannya. Uji *outlier* dilakukan dengan membuat nilai *z* (standarisasi data). Persamaan yang digunakan untuk menstandarisasi data ditunjukkan pada persamaan (3.2). Data dapat dikatakan *outlier* apabila nilai *z* lebih besar dari angka +2,5 atau lebih kecil dari angka -2,5 (Santoso, 2010).



$$z = \frac{x - \bar{x}}{\sigma} \quad (3.2)$$

### 3.4.3 Uji Korelasi

Uji korelasi merupakan suatu pengujian data untuk mengetahui tingkat keeratan hubungan antar variabel secara linier yang dinyatakan dalam bentuk koefisien korelasi ( $r$ ). Nilai koefisien korelasi terletak di antara -1 hingga +1. Semakin mendekati nilai -1 atau +1, semakin erat hubungan antar variabel. Tanda positif menandakan bahwa korelasi antar variabel searah sedangkan tanda negatif menandakan bahwa korelasi antar variabel berlawanan. Tingkat keeratan hubungan antar variabel ditunjukkan pada Tabel 3.2. Metodologi yang digunakan dalam pengujian korelasi adalah *Pearson Correlation* yang ditunjukkan pada persamaan (3.3).

$$r = \frac{\sum(x_i - \bar{x})\sum(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}} \quad (3.3)$$

**Tabel 3. 2** Nilai Interval Korelasi Terhadap Tingkat hubungannya (Sugiyono, 2007)

<b>Interval Koefisien</b>	<b>Tingkat Hubungan</b>
0,00 – 0,199	Sangat rendah atau tidak berpengaruh
2,00 – 0,399	Rendah
0,40 – 0,599	Sedang
0,60 – 0,799	Kuat
0,80 – 1,000	Sangat kuat

Parameter besarnya nilai korelasi antar variabel tidak hanya ditentukan oleh nilai  $r$  saja. Parameter lain yang digunakan adalah nilai rata-rata signifikansi dari setiap variabel yang akan digunakan. Nilai signifikansi yang digunakan adalah maksimal 0,05 atau nilai kepercayaan sebesar 95%. Nilai signifikansi di atas 0,05 maka variabel tersebut dikatakan tidak berkorelasi atau memiliki hubungan antar variabel yang rendah sehingga tidak layak untuk digunakan sebagai variabel input.

### 3.4.4 Analisa Data

Tahap analisa data adalah tahapan untuk menentukan variabel masukan yang digunakan prediktor berdasarkan besarnya nilai korelasi. Variabel masukan yang digunakan adalah kecepatan angin, arah angin, dan suhu permukaan laut. Variabel keluaran yang digunakan adalah ketinggian gelombang dan kecepatan arus laut. Analisa data selanjutnya yaitu proses pembagian data latih dan data uji untuk JST yang akan digunakan. Penelitian ini menggunakan data latih sebesar 70% dari total data yang ada sedangkan data uji sebesar 30% dari data total yang ada. Pembagian data latih dan uji ditunjukkan pada Tabel 3.3.

**Tabel 3. 3** Pembagian Data Pelatihan dan Pengujian

Jenis Data	Rentang Data	Jumlah Data
Data Pelatihan	Januari 2017-Juli 2018	10.538
Data Pengujian	Agustus 2018-Desember 2018	4.515

### 3.5 Perancangan Algoritma Prediktor

Perancangan algoritma prediktor ketinggian gelombang air laut dan kecepatan arus laut menggunakan metode Jaringan Syaraf Tiruan (JST) yang dioptimasi dengan *Genetic Algorithm* (GA) untuk mendapatkan nilai RMSE yang minimal. Blok diagram rancangan prediktor ditunjukkan pada Gambar 3.2. Penjelasan mengenai perancangan prediktor tersebut dituliskan pada sub bab di bawah ini.



**Gambar 3. 2** Diagram Blok Prediktor

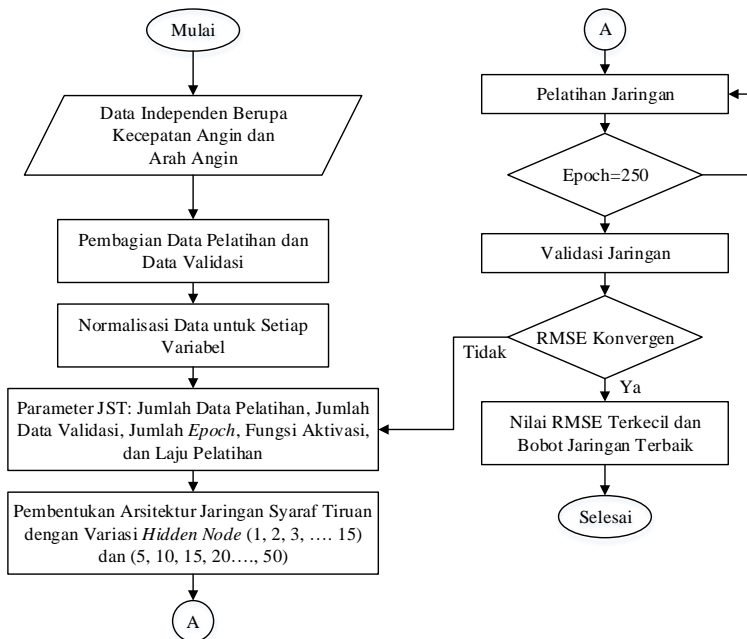
#### 3.5.1 Perancangan Prediktor Ketinggian Gelombang

Perancangan prediktor ketinggian gelombang laut menggunakan dua metode yaitu JST dengan algoritma pelatihan Levenberg Marquardt (JST-LM) dan JST dengan optimasi *Genetic Algorithm* (JST-GA). Berdasarkan blok diagram di atas, variabel

keluaran (output) berupa ketinggian gelombang dipengaruhi oleh variabel masukan (input) berupa kecepatan angin, arah angin, dan ketinggian gelombang sebelumnya berdasarkan besarnya nilai korelasi pada Tabel 4.3 dan 4.4. Penjelasan mengenai dua metode prediktor ketinggian gelombang dijelaskan di bawah ini.

#### a. Perancangan Algoritma JST-LM Prediktor Gelombang

Metode pertama yang digunakan pada prediktor ketinggian gelombang laut yaitu JST-LM. Diagram alir JST-LM prediktor ketinggian gelombang ditunjukkan pada Gambar 3.2.



**Gambar 3. 3** Diagram Alir JST-LM Prediktor Gelombang

Parameter yang digunakan pada prediktor ketinggian gelombang dengan metode JST-LM ditunjukkan pada Tabel 3.4. Arsitektur JST-LM yang digunakan yaitu dengan memvariasikan jumlah *hidden node* yaitu 1, 2, 3, 4, 5, ....., 15 serta 5, 10, 15, 20,

25, 30, 35, 40, 45, dan 50. Nilai *learning rate* serta jenis fungsi aktivasi yang digunakan pada JST-LM ini berdasarkan *trial dan error* pada Tabel 4.7. Jumlah *epoch* yang digunakan pada JST-LM ini yaitu sebanyak 250 *epoch*.

**Tabel 3. 4** Parameter JST-LM Prediktor Ketinggian Gelombang

No	Parameter	Nilai
1	<i>Epoch</i>	250
2	<i>Learning rate</i>	0,01
3	Fungsi Aktivasi Input ke <i>Hidden</i>	Tansig
4	Fungsi Aktivasi <i>Hidden</i> ke Output	Logsig

Proses normalisasi dilakukan sebelum dilakukan proses pelatihan pada JST. Setiap variabel dilakukan proses normalisasi pada rentang 0,1-0,9 dengan perhitungan yang ditunjukkan pada persamaan (3.3) di bawah ini.

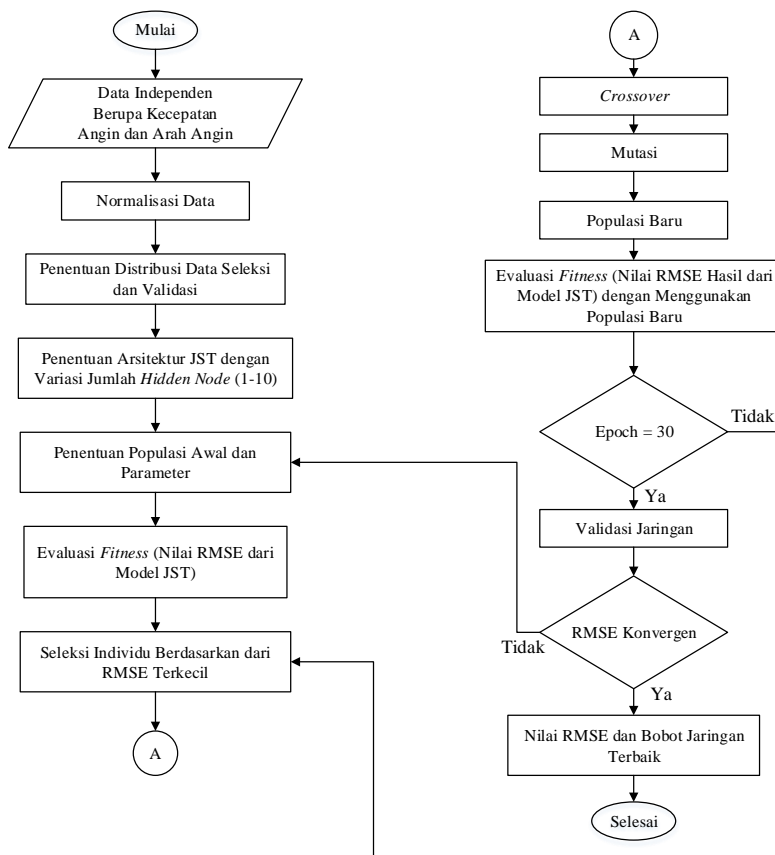
$$x' = \frac{0,8(x+1)}{(x_{max}-x_{min})} + 0,1 \quad (3.3)$$

Data prediksi hasil keluaran jaringan kemudian dikembalikan ke nilai semula dengan cara denormalisasi untuk mendapatkan nilai ketinggian gelombang yang sebenarnya. Persamaan denormalisasi ditunjukkan pada persamaan (3.4).

$$x = \left( \frac{(x'-0,1)(x_{max}-x_{min})}{0,8} \right) + x_{min} \quad (3.4)$$

#### **b. Perancangan Algoritma JST-GA Ketinggian Gelombang**

Metode kedua yang digunakan pada prediktor ketinggian gelombang laut yaitu JST-GA. Diagram alir JST-GA prediktor ketinggian gelombang ditunjukkan pada Gambar 3.4. Optimasi GA ini dilakukan untuk mengoptimalkan bobot-bobot yang dihasilkan dari JST untuk menghasilkan nilai RMSE terkecil pada setiap arsitektur yang telah dibuat. Jumlah populasi awal atau kromosom pada GA menunjukkan nilai variabel ketinggian gelombang.



**Gambar 3. 4** Diagram Alir JST-GA Prediktor Ketinggian Gelombang

Optimasi GA pada JST prediktor ketinggian gelombang diawali dengan penentuan populasi atau kromosom setiap individu berupa variabel ketinggian gelombang air laut, dimana dalam penelitian ini populasi yang digunakan sebanyak 20 dengan jumlah bit sebanyak 10. Kromosom tersebut kemudian dilakukan pemilihan kromosom terbaik (nilai tinggi gelombang prediksi yang mendekati target) berdasarkan nilai RMSE terkecil yang dihasilkan. Kromosom terbaik kemudian dilakukan proses *crossover* dengan

menyilangkan dua kromosom untuk menghasilkan kromosom baru dengan nilai RMSE yang terkecil. Penelitian ini probabilitas *crossover* yang digunakan sebesar 0,8 dan probabilitas mutasi sebesar 0,033. Kromosom baru yang dihasilkan kemudian dijadikan populasi awal untuk dilakukan pemrosesan iterasi selanjutnya.

**Tabel 3. 5** Parameter JST-GA Prediktor Ketinggian Gelombang

No	Parameter	Nilai
1	Populasi	20
2	Elits	0,8
3	<i>Epoch</i>	30
4	Probabilitas <i>crossover</i>	0,8
5	Probabilitas mutasi	0,033
6	Jumlah bit	10

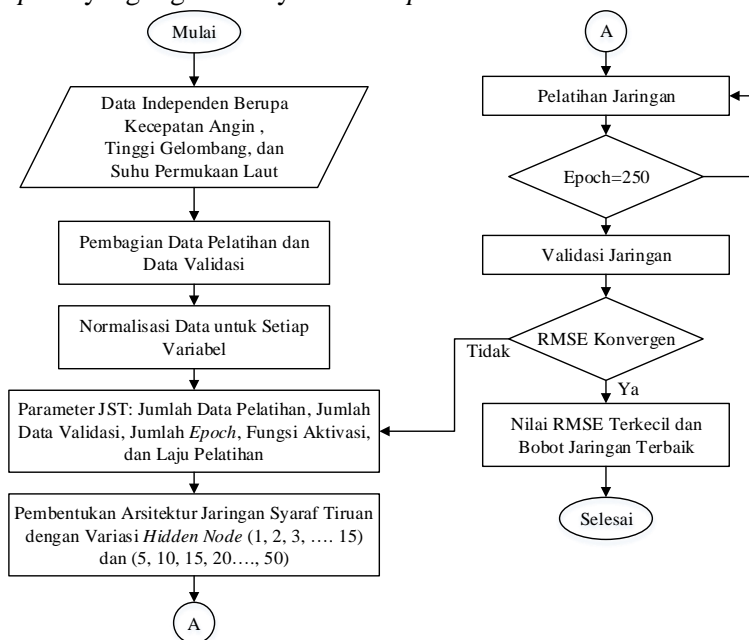
Parameter yang digunakan JST-GA ditunjukkan pada Tabel 3.5 berdasarkan acuan dari penelitian Zhang pada tahun 2009. Arsitektur JST-GA yang digunakan pada penelitian ini yaitu dengan memvariasikan jumlah *hidden node* yaitu 1, 2, 3, 4, 5, ....., 15. Jumlah *epoch* yang digunakan pada JST-GA yaitu 30 *epoch*. Perancangan JST-GA dimulai dengan mengolah data menjadi data normalisasi, kemudian menentukan parameter optimasi yang digunakan dari jumlah populasi, bit, probabilitas mutasi, dan probabilitas *crossover*. Hasil penggunaan parameter didapatkan nilai RMSE yang merupakan nilai *fitness* dari model JST.

### 3.5.2 Perancangan Prediktor Kecepatan Arus Laut

Perancangan prediktor kecepatan arus laut menggunakan dua metode yaitu JST dengan algoritma pelatihan Levenberg Marquardt (JST-LM) dan JST dengan optimasi *Genetic Algorithm* (JST-GA). Berdasarkan blok diagram Gambar 3.2, variabel keluaran berupa kecepatan arus laut dipengaruhi oleh variabel masukan berupa kecepatan angin, suhu permukaan laut, dan ketinggian gelombang berdasarkan nilai korelasi pada Tabel 4.5 dan 4.6. Penjelasan mengenai dua metode prediktor arus laut dijelaskan di bawah ini.

### a. Perancangan Algoritma JST-LM Prediktor Arus Laut

Metode pertama yang digunakan pada prediktor kecepatan arus laut yaitu JST-LM. Diagram alir JST-LM prediktor ketinggian gelombang ditunjukkan pada Gambar 3.5. Penelitian dimulai dari pengolahan data menjadi normalisasi kemudian dilakukan pemilihan parameter JST seperti jumlah *epoch* dan jenis fungsi aktivasi yang digunakan. Parameter yang digunakan pada prediktor kecepatan arus laut dengan metode JST-LM ditunjukkan pada Tabel 3.6. Arsitektur JST-LM yang digunakan yaitu dengan memvariasikan jumlah *hidden node* yaitu 1, 2, 3, 4, 5, ....., 15 serta 5, 10, 15,..... 50. Fungsi aktivasi yang digunakan yaitu logsig untuk input ke *hidden* dan logsig untuk *hidden* ke output. Jumlah *epoch* yang digunakan yaitu 250 *epoch*.



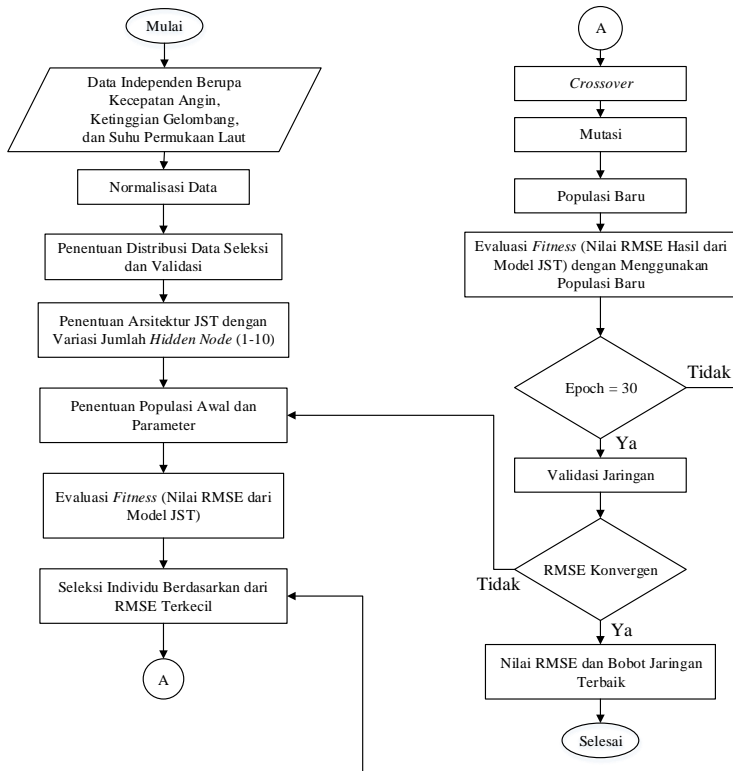
**Gambar 3.5** Diagram Alir JST-LM Prediktor Arus Laut

**Tabel 3. 6** Parameter JST-LM Prediktor Arus Laut

No	Parameter	Nilai
1	<i>Epoch</i>	250
2	<i>Learning rate</i>	0,05
3	Fungsi Aktivasi Input ke <i>Hidden</i>	Logsig
4	Fungsi Aktivasi <i>Hidden</i> ke Output	Logsig

### b. Perancangan Algoritma JST-GA Prediktor Arus Laut

Metode kedua yang digunakan pada prediktor kecepatan arus laut yaitu JST-GA. Diagram alir JST-GA prediktor kecepatan arus laut ditunjukkan pada Gambar 3.6.

**Gambar 3. 6** Diagram Alir JST-GA Prediktor Arus Laut



**Tabel 3. 7** Parameter JST-GA Prediktor Arus Laut

No	Parameter	Nilai
1	Populasi	20
2	Elits	0,8
3	<i>Epoch</i>	30
4	Probabilitas <i>crossover</i>	0,8
5	Probabilitas mutasi	0,033
6	Jumlah bit	10

Arsitektur JST-GA yang digunakan pada penelitian ini yaitu dengan memvariasikan jumlah *hidden node* yaitu 1, 2, 3, 4, 5, ....., 15. Jumlah *epoch* yang digunakan pada JST-GA ini yaitu sebanyak 30 *epoch*. Arsitektur jaringan terbaik dengan RMSE terkecil dilakukan penambahan jumlah *epoch*, yaitu 60 *epoch* dan 90 *epoch* untuk mengetahui pengaruh variasi *epoch* yang digunakan.

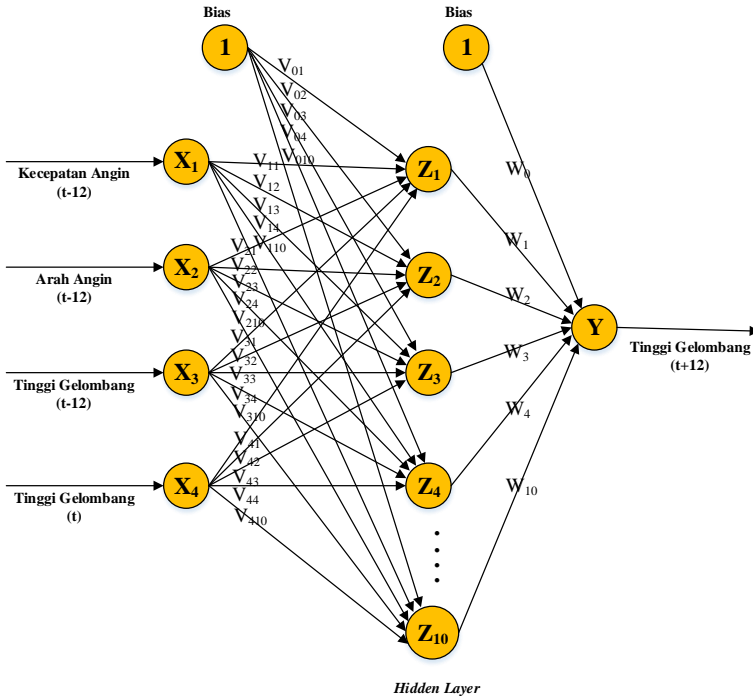
Perancangan JST-GA dimulai dengan mengolah data independen dan dependen menjadi data normalisasi, kemudian menentukan parameter optimasi yang digunakan mulai dari jumlah populasi, jumlah bit, probabilitas mutasi, dan probabilitas *crossover* yang digunakan. Hasil penggunaan parameter tersebut didapatkan nilai RMSE yang merupakan nilai *fitness* dari model JST.

### 3.6 Pemilihan Arsitektur Terbaik

Pemilihan arsitektur jaringan terbaik didapatkan setelah proses pelatihan dan pengujian berdasarkan rancangan yang telah dijelaskan pada sub bab 3.5 di atas. Arsitektur terbaik pada prediktor ketinggian gelombang dan kecepatan arus laut dijelaskan pada sub bab di bawah ini.

#### 3.6.1 Arsitektur Terbaik Prediktor Gelombang

Arsitektur terbaik pada prediktor ketinggian gelombang dengan metode JST-LM dengan RMSE terkecil yaitu arsitektur 4-10-1 (4 *input layer*, 1 *hidden layer* dengan 10 *hidden node* dan 1 *output layer*). Arsitektur terbaik prediktor ketinggian gelombang metode JST-LM ditunjukkan pada Gambar 3.7.



**Gambar 3. 7** Arsitektur JST-LM 4-10-1 Prediktor Ketinggian Gelombang

Arsitektur JST pada Gambar 3.7 di atas apabila dituliskan dalam bentuk fungsi matematis prediktor ketinggian gelombang ditunjukkan pada persamaan (3.5) di bawah ini.

$$WH(t + 12) = f(WS(t - 12), WD(t - 12), WH(t - 12), WH(t)) \quad (3.5)$$

$WH(t + 12)$  = ketinggian gelombang 12 jam ke depan

$WS(t - 12)$  = kecepatan angin 12 jam sebelumnya

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

$WH(t - 12)$  = ketinggian gelombang 12 jam sebelumnya

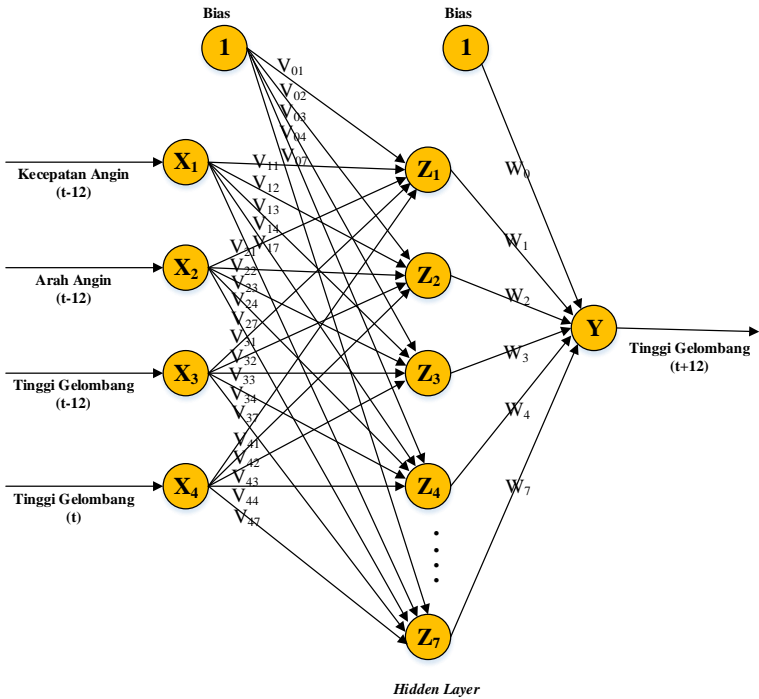
$WH(t)$  = ketinggian gelombang saat ini

*Hidden layer* sebanyak 1 dan *hidden node* sebanyak 10. Variabel  $V_{ij}$  merupakan bobot dari input ke *hidden* sedangkan  $W_x$  merupakan bobot dari *hidden* ke output. Nilai  $i$  yaitu *input layer* bernilai 1, 2, 3, 4 dan 0 untuk nilai bias. Nilai  $j$  yaitu variasi *hidden node* bernilai 1, 2, 3, 4, 5, 6, 7, 8, 9, dan 10. Nilai  $x$  yaitu variasi *hidden node* bernilai 1, 2, 3, 4, ..., 10 dan 0 untuk nilai bias. Metode JST-LM dengan *hidden node* 10 ini menghasilkan nilai bobot sebanyak 61. Nilai bobot keseluruhan JST-LM ini ditunjukkan pada Tabel 3.8.

**Tabel 3. 8** Nilai Bobot pada Arsitektur 4-10-1 Metode JST-LM  
Prediktor Ketinggian Gelombang

<b>Bobot</b>	<b>Nilai</b>	<b>Bobot</b>	<b>Nilai</b>	<b>Bobot</b>	<b>Nilai</b>
$V_{11}$	4,2041	$V_{31}$	2,5297	$V_{01}$	-8,2423
$V_{12}$	7,2533	$V_{32}$	-2,7308	$V_{02}$	-3,3479
$V_{13}$	-11,5889	$V_{33}$	0,8839	$V_{03}$	8,6235
$V_{14}$	17,0867	$V_{34}$	12,1225	$V_{04}$	-14,2759
$V_{15}$	0,3163	$V_{35}$	0,1251	$V_{05}$	0,6616
$V_{16}$	-13,0003	$V_{36}$	6,8062	$V_{06}$	3,1963
$V_{17}$	-0,7000	$V_{37}$	3,6453	$V_{07}$	-12,0766
$V_{18}$	5,1156	$V_{38}$	-4,3218	$V_{08}$	-2,1943
$V_{19}$	-2,1260	$V_{39}$	6,8602	$V_{09}$	-3,3152
$V_{110}$	5,5907	$V_{310}$	0,5259	$V_{010}$	2,0757
$V_{21}$	-2,1867	$V_{41}$	7,8146	$W_1$	5,2635
$V_{22}$	5,0665	$V_{42}$	-7,4799	$W_2$	0,3994
$V_{23}$	5,2766	$V_{43}$	-4,0546	$W_3$	1,3236
$V_{24}$	0,9433	$V_{44}$	-10,5555	$W_4$	0,3598
$V_{25}$	0,2470	$V_{45}$	-1,0381	$W_5$	-9,0293
$V_{26}$	-10,3127	$V_{46}$	5,4014	$W_6$	-0,1835
$V_{27}$	1,2076	$V_{47}$	7,4579	$W_7$	-2,2637
$V_{28}$	7,6573	$V_{48}$	4,2076	$W_8$	-0,3926
$V_{29}$	7,2138	$V_{49}$	6,4034	$W_9$	0,1011
$V_{210}$	5,2308	$V_{410}$	5,3361	$W_{10}$	3,0944
$W_0$	1,9906				

Arsitektur terbaik pada prediktor ketinggian gelombang dengan metode JST-GA dengan RMSE terkecil yaitu arsitektur 4-7-1 (4 *input layer*, 1 *hidden layer* dengan 7 *hidden node* dan 1 *output layer*). Arsitektur terbaik prediktor ketinggian gelombang metode JST-GA ditunjukkan pada Gambar 3.8. Metode JST-GA dengan *hidden node* 7 menghasilkan nilai bobot sebanyak 43. Nilai bobot keseluruhan JST-GA ini ditunjukkan pada Tabel 3.9. Variabel  $V_{ij}$  merupakan bobot dari input ke *hidden* sedangkan  $W_x$  merupakan bobot dari *hidden* ke output.



**Gambar 3. 8** Arsitektur JST-GA 4-7-1 Prediktor Ketinggian Gelombang

Bobot ini didapatkan dari proses pelatihan optimasi GA yang menghasilkan nilai *fitness* terbaik berdasarkan nilai RMSE

terkecil. *Epoch* yang digunakan sebanyak 30 *epoch* dan lama waktu proses optimasi ini selama kurang lebih 45 menit. Lama proses optimasi ini tergantung jumlah *hidden node* yang digunakan, semakin banyak *hidden node* yang digunakan maka proses optimasi yang berjalan semakin lama. Lama proses optimasi GA ini juga dipengaruhi oleh jumlah variabel atau bobot yang dioptimasi serta jumlah *epoch* yang digunakan. Setiap penambahan 30 *epoch* pada optimasi GA memerlukan waktu dua kali lipat lebih lama, tetapi penambahan *epoch* belum tentu menghasilkan RMSE lebih kecil.

**Tabel 3. 9** Nilai Bobot pada Arsitektur 4-7-1 Metode JST-GA  
Prediktor Ketinggian Gelombang

<b>Bobot</b>	<b>Nilai</b>	<b>Bobot</b>	<b>Nilai</b>	<b>Bobot</b>	<b>Nilai</b>
V <sub>11</sub>	0,6579	V <sub>32</sub>	-0,4154	V <sub>03</sub>	-0,6657
V <sub>12</sub>	0,6794	V <sub>33</sub>	0,1085	V <sub>04</sub>	0,6168
V <sub>13</sub>	0,1711	V <sub>34</sub>	0,3236	V <sub>05</sub>	-0,4174
V <sub>14</sub>	-0,5679	V <sub>35</sub>	-0,6129	V <sub>06</sub>	-0,1222
V <sub>15</sub>	0,6266	V <sub>36</sub>	0,9707	V <sub>07</sub>	-0,8201
V <sub>16</sub>	0,3236	V <sub>37</sub>	-0,4995	W <sub>1</sub>	0,2923
V <sub>17</sub>	0,1535	V <sub>41</sub>	0,6774	W <sub>2</sub>	-0,9609
V <sub>21</sub>	-0,1105	V <sub>42</sub>	-0,9003	W <sub>3</sub>	0,1437
V <sub>22</sub>	-0,9374	V <sub>43</sub>	0,4213	W <sub>4</sub>	-0,7439
V <sub>23</sub>	0,4252	V <sub>44</sub>	-0,7498	W <sub>5</sub>	0,1593
V <sub>24</sub>	-0,8495	V <sub>45</sub>	-0,1281	W <sub>6</sub>	-0,3118
V <sub>25</sub>	-0,0987	V <sub>46</sub>	-0,9022	W <sub>7</sub>	0,4878
V <sub>26</sub>	0,6755	V <sub>47</sub>	-0,7048	W <sub>0</sub>	0,0225
V <sub>27</sub>	0,5699	V <sub>01</sub>	0,8045		
V <sub>31</sub>	0,8475	V <sub>02</sub>	0,9472		

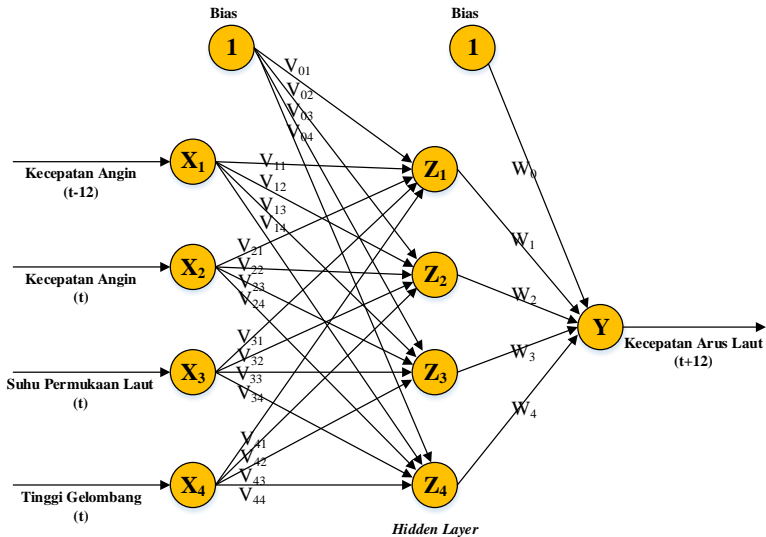
Perbedaan nilai parameter (*epoch*, jenis fungsi aktivasi, dan *hidden node*) yang digunakan pada JST-LM dan JST-GA pada prediktor ketinggian gelombang ditunjukkan pada Tabel 3.10. Nilai parameter tersebut digunakan berdasarkan *trial and error* pada pemilihan laju pelatihan dan jenis fungsi aktivasi. Jumlah *epoch* yang digunakan berdasarkan *trial and error* pada arsitektur yang dibuat hingga mendapatkan nilai yang konvergen.

**Tabel 3. 10** Perbedaan Parameter JST-LM dan JST-GA pada Prediktor Ketinggian Gelombang

Prediktor Ketinggian Gelombang		
Parameter	JST-LM	JST-GA
Epoch	250	30
Fs Aktivasi Input ke <i>Hidden</i>	Tansig	Tansig
Fs Aktivasi <i>Hidden</i> ke Output	Logsig	Logsig
Waktu <i>running</i>	5 menit	35 menit
Jumlah <i>Hidden Node</i> Terbaik	10 <i>hidden node</i>	7 <i>hidden node</i>

**3.6.2 Arsitektur Terbaik Prediktor Kecepatan Arus Laut**

Arsitektur terbaik pada prediktor kecepatan arus laut dengan metode JST-LM dengan RMSE terkecil yaitu arsitektur 4-4-1 (4 *input layer*, 1 *hidden layer* dengan 4 *hidden node* dan 1 *output layer*). Arsitektur terbaik prediktor kecepatan arus laut metode JST-LM ditunjukkan pada Gambar 3.7.



**Gambar 3. 9** Arsitektur JST-LM 4-4-1 Prediktor Arus Laut

Arsitektur JST-LM prediktor kecepatan arus laut 4-4-1 pada Gambar 3.9 apabila dituliskan dalam bentuk fungsi matematis prediktor kecepatan arus laut ditunjukkan pada persamaan (3.6) di bawah ini.

$$CS(t + 12) = f(WS(t - 12), WS(t), SST(t), WH(t)) \quad (3.6)$$

$CS(t + 12)$  = ketinggian gelombang 12 jam ke depan

$WS(t - 12)$  = kecepatan angin 12 jam sebelumnya

$WS(t)$  = kecepatan angin saat ini

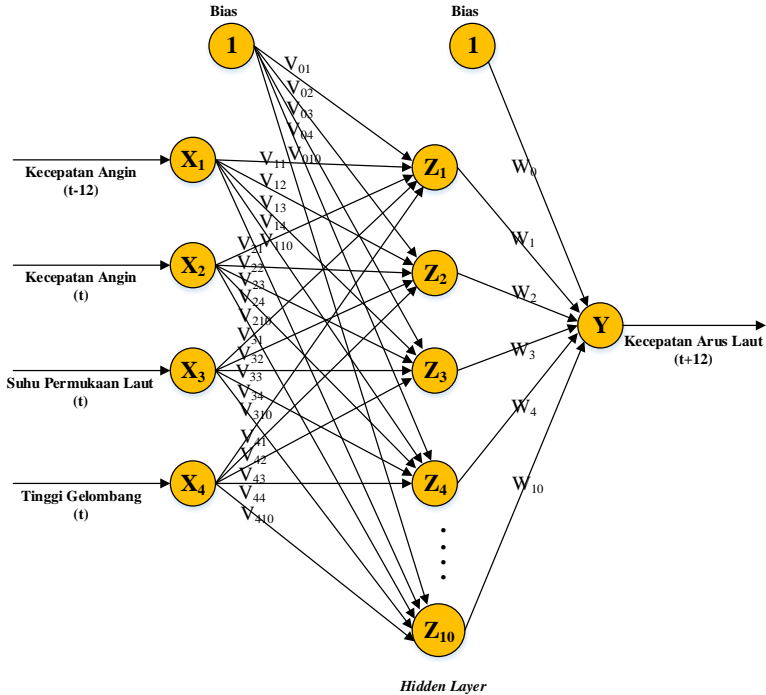
$SST(t)$  = suhu permukaan laut saat ini

$WH(t)$  = ketinggian gelombang saat ini

Prediktor kecepatan arus laut menggunakan *input layer* sebanyak 4 variabel input, *hidden layer* sebanyak 1 dengan *hidden node* sebanyak 4 dan *output layer* sebanyak 1 atau arsitektur 4-4-1. Variabel  $V_{ij}$  merupakan bobot dari input ke *hidden* sedangkan  $W_x$  merupakan bobot dari *hidden* ke output. Nilai  $j$  yaitu variasi *hidden node* bernilai 1, 2, 3, dan 4. Nilai  $x$  yaitu variasi *hidden node* bernilai 1, 2, 3, 4 dan 0 untuk nilai bias. Metode JST-LM dengan *hidden node* 4 menghasilkan nilai bobot sebanyak 25. Nilai bobot keseluruhan JST-LM pada prediktor kecepatan arus laut ini ditunjukkan pada Tabel 3.11 di bawah ini.

**Tabel 3. 11** Nilai Bobot pada Arsitektur 4-4-1 Metode JST-LM Prediktor Arus Laut

Bobot	Nilai	Bobot	Nilai	Bobot	Nilai
$V_{11}$	8,3105	$V_{32}$	2,1019	$V_{03}$	-15,8460
$V_{12}$	-8,3955	$V_{33}$	-2,7440	$V_{04}$	71,3875
$V_{13}$	26,9660	$V_{34}$	-2,0410	$W_1$	-12,5686
$V_{14}$	-84,9620	$V_{41}$	-5,3660	$W_2$	-12,4458
$V_{21}$	10,3354	$V_{42}$	15,5190	$W_3$	0,2801
$V_{22}$	-10,5247	$V_{43}$	127,559	$W_4$	-0,2020
$V_{23}$	16,5080	$V_{01}$	-0,9722	$W_0$	123,4253
$V_{24}$	-2,0653	$V_{02}$	0,9921		
$V_{31}$	-13,8270	$V_{44}$	-5,1555		



**Gambar 3. 10** Arsitektur JST-GA 4-10-1 Prediktor Arus Laut

Arsitektur terbaik pada prediktor kecepatan arus laut dengan metode JST-GA dengan RMSE terkecil yaitu arsitektur 4-10-1 (4 *input layer*, 1 *hidden layer* dengan 10 *hidden node* dan 1 *output layer*). Arsitektur terbaik prediktor kecepatan arus laut metode JST-GA ditunjukkan pada Gambar 3.10. Metode JST-GA dengan *hidden node* 10 menghasilkan nilai bobot sebanyak 61. Nilai bobot keseluruhan JST-GA ini ditunjukkan pada Tabel 3.12. Variabel  $V_{ij}$  merupakan bobot dari input ke *hidden* sedangkan  $W_x$  merupakan bobot dari *hidden* ke output. Bobot ini didapatkan dari proses pelatihan optimasi GA yang menghasilkan nilai *fitness* terbaik dengan *epoch* yang digunakan sebanyak 30 *epoch* dan lama waktu proses optimasi ini selama kurang lebih 45 menit.



**Tabel 3. 12** Nilai Bobot pada Arsitektur 4-10-1 Metode JST-GA Prediktor Arus Laut

<b>Bobot</b>	<b>Nilai</b>	<b>Bobot</b>	<b>Nilai</b>	<b>Bobot</b>	<b>Nilai</b>
V <sub>11</sub>	-0,1320	V <sub>32</sub>	0,0870	V <sub>03</sub>	-0,9042
V <sub>12</sub>	-0,4819	V <sub>33</sub>	0,0440	V <sub>04</sub>	-0,7713
V <sub>13</sub>	0,4702	V <sub>34</sub>	0,4135	V <sub>05</sub>	0,2297
V <sub>14</sub>	0,2473	V <sub>35</sub>	0,2043	V <sub>06</sub>	0,9824
V <sub>15</sub>	0,7752	V <sub>36</sub>	-0,2532	V <sub>07</sub>	0,3040
V <sub>16</sub>	-0,1202	V <sub>37</sub>	0,5816	V <sub>08</sub>	-0,5347
V <sub>17</sub>	-0,6442	V <sub>38</sub>	0,2512	V <sub>09</sub>	0,1730
V <sub>18</sub>	-0,5738	V <sub>39</sub>	0,6950	V <sub>010</sub>	0,3353
V <sub>19</sub>	0,6325	V <sub>310</sub>	0,0635	W <sub>1</sub>	0,5699
V <sub>110</sub>	0,6794	V <sub>41</sub>	-0,0264	W <sub>2</sub>	-0,6833
V <sub>21</sub>	-0,5855	V <sub>42</sub>	-0,3959	W <sub>3</sub>	0,8495
V <sub>22</sub>	-0,2160	V <sub>43</sub>	0,3763	W <sub>4</sub>	-0,9042
V <sub>23</sub>	0,1065	V <sub>44</sub>	0,2805	W <sub>5</sub>	0,8299
V <sub>24</sub>	-0,4037	V <sub>45</sub>	0,5112	W <sub>6</sub>	0,2239
V <sub>25</sub>	0,0186	V <sub>46</sub>	-0,8338	W <sub>7</sub>	-0,4330
V <sub>26</sub>	0,1652	V <sub>47</sub>	-0,8847	W <sub>8</sub>	0,3314
V <sub>27</sub>	0,2043	V <sub>48</sub>	-0,3255	W <sub>9</sub>	-0,7967
V <sub>28</sub>	0,0010	V <sub>49</sub>	0,3529	W <sub>10</sub>	0,7087
V <sub>29</sub>	-0,1613	V <sub>410</sub>	-0,4115	W <sub>0</sub>	-0,6931
V <sub>210</sub>	0,4976	V <sub>01</sub>	-0,0674		
V <sub>31</sub>	0,7126	V <sub>02</sub>	-0,9081		

Perbedaan nilai parameter (*epoch*, jenis fungsi aktivasi, dan *hidden node*) yang digunakan pada JST-LM dan JST-GA pada prediktor kecepatan arus laut ditunjukkan pada Tabel 3.13. Nilai parameter tersebut digunakan berdasarkan *trial and error* pada pemilihan laju pelatihan dan jenis fungsi aktivasi. Jumlah *epoch* yang digunakan berdasarkan *trial and error* pada arsitektur yang dibuat hingga mendapatkan nilai yang konvergen. Parameter tersebut kemudian diterapkan pada arsitektur jaringan yang dibuat dengan variasi *hidden node*. Keluaran prediksi kemudian diukur nilai performansinya melalui nilai RMSE yang terkecil. Nilai RMSE yang terukur merupakan selisih dari nilai prediksi dan target dari variabel ketinggian gelombang dan kecepatan arus laut.

**Tabel 3. 13** Perbedaan Parameter JST-LM dan JST-GA pada Prediktor Kecepatan Arus Laut

<b>Prediktor kecepatan Arus Laut</b>		
<b>Parameter</b>	<b>JST-LM</b>	<b>JST-GA</b>
<i>Epoch</i>	250	30
Fs Aktivasi Input ke <i>Hidden</i>	Logsig	Logsig
Fs Aktivasi <i>Hidden</i> ke Output	Logsig	Logsig
Waktu <i>running</i>	5 menit	35 menit
Jumlah <i>Hidden Node</i> Terbaik	4 <i>hidden node</i>	10 <i>hidden node</i>

### 3.7 Pengujian *Realtime*

Validasi dilakukan dengan menguji prediktor ketinggian gelombang dan kecepatan arus laut menggunakan JST-GA dengan data hasil pengukuran menggunakan *Buoyweather Station Type II* yang dilakukan di Pantai Kenjeran selama tiga hari, pada tanggal 10 Juni, 15 Juni, dan 18 Juni 2019.

### 3.8 Analisis Data dan Pembahasan

Analisis data dilakukan setelah semua simulasi dilakukan kemudian pembahasan mengenai performansi prediktor. Performansi prediktor ini diukur berdasarkan nilai RMSE yang dihasilkan. Nilai RMSE yang dihasilkan tersebut kemudian dibandingkan dengan penelitian sebelumnya mengenai prediksi ketinggian gelombang.

### 3.9 Kesimpulan dan Saran

Kesimpulan merupakan jawaban dari rumusan masalah yang diangkat, apabila kesimpulan masih belum menjawab rumusan masalah maka akan dimasukkan saran sebagai acuan untuk penelitian selanjutnya atau saran terkait penelitian ini agar lebih sempurna. Kesimpulan pada penelitian ini yaitu berisi tentang arsitektur terbaik yang digunakan pada prediksi ketinggian gelombang dan kecepatan arus laut serta membandingkan antara metode JST-LM dan JST-GA berdasarkan nilai RMSE yang didapatkan.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Pengujian Data

Data yang telah dikumpulkan dari BMKG II Perak Surabaya dilakukan evaluasi kelayakan data dan pengujian data secara statistik. Data yang digunakan yaitu data selama 2 tahun yaitu Januari 2017 hingga Desember 2018. Pengujian data dilakukan dalam 3 jenis pengujian yaitu uji *missing* data, uji data *outlier*, dan uji korelasi data. Pengujian data dilakukan dengan bantuan *software* IBM SPSS 16.0. Penjelasan ketiga pengujian data tersebut dijelaskan pada sub bab di bawah ini.

#### 4.1.1 Uji *Missing* Data

Hasil uji *missing data* independen dan dependen dapat menunjukkan bahwa masing-masing data memiliki nilai presentase *missing data* yang rendah yaitu di bawah 1% dari total data yang ada. Nilai tersebut menunjukkan bahwa data ketinggian gelombang air laut, kecepatan angin, arah angin, arus laut, dan suhu permukaan laut layak digunakan untuk proses selanjutnya. Hasil uji data *outlier* ditunjukkan pada Tabel 4.1.

**Tabel 4. 1** Uji *Missing* Data

Variabel	Jumlah Data Total	Jumlah Data <i>Missing</i>	%Jumlah Data <i>Missing</i>
Tinggi Gelombang	15173	-	-
Arus Laut	15173	-	-
Kecepatan Angin	15173	-	-
Arah Angin	15173	-	-
Suhu Permukaan Laut	15173	73	0.48%

#### 4.1.2 Uji Data *Outlier*

Hasil uji *outlier* semua variabel ditunjukkan pada Tabel 4.2 dimana variabel dikatakan *outlier* apabila nilai  $z$  melebihi dari 2,5. Variabel-variabel yang digunakan dalam penelitian ini memiliki

jumlah data *outlier* yang rendah yaitu dibawah 5% dari total data keseluruhan sehingga variabel tinggi gelombang, arus laut, kecepatan angin, arah angin, dan suhu permukaan laut layak untuk digunakan.

**Tabel 4. 2** Uji Data *Outlier*

Variabel	Jumlah Data Total	Jumlah Data <i>Outlier</i>	%Jumlah Data <i>Outlier</i>
Tinggi Gelombang	15173	181	1,19%
Arus Laut	15173	358	2,35%
Kecepatan Angin	15173	212	1,39%
Arah Angin	15173	-	-
Suhu Permukaan Laut	15173	73	0,48%

#### 4.1.3 Uji Korelasi Data

Uji korelasi data digunakan mencari hubungan 2 variabel antara variabel independen dan dependen. Uji korelasi digunakan untuk menyeleksi variabel independen yang tidak berpengaruh terhadap variabel dependen. Hasil uji korelasi ditunjukkan pada Tabel 4.3. Nilai uji korelasi mengacu pada Tabel 3.1. Hasil uji korelasi menunjukkan beberapa variabel independen memiliki nilai korelasi sangat rendah atau tidak berpengaruh terhadap variabel dependen.

**Tabel 4. 3** Uji Korelasi Variabel Prediksi Tinggi Gelombang

Variabel	Korelasi (Nilai r)	Signifikan
Kecepatan Angin	0,649	0,000
Arah Angin	-0,360	0,000
Suhu Permukaan Laut	-0,142	0,000
Tekanan	0,052	0,000
Kelembaban	0,105	0,303

Hasil uji korelasi pada variabel input prediksi ketinggian gelombang didapatkan bahwa kecepatan angin dan arah angin yang berkorelasi tinggi, sedangkan untuk suhu permukaan laut, tekanan, dan kelembaban memiliki korelasi yang rendah dengan tinggi gelombang. Hasil korelasi bernilai positif menunjukkan bahwa

hubungan antar variabel berbanding lurus sedangkan bernilai negatif menunjukkan bahwa hubungan antar variabel berbanding terbalik. Tingkat korelasi (koefisien  $r$ ) pada kecepatan angin termasuk dalam kategori kuat sedangkan arah angin termasuk dalam kategori rendah, meskipun arah angin berkategori rendah tetapi masih bisa digunakan karena nilai rata-rata signifikansi bernilai 0,00. Nilai signifikansi apabila dibawah 0,05 maka data layak digunakan atau dikatakan berkorelasi. Variabel suhu permukaan laut, tekanan, dan kelembaban memiliki nilai koefisien  $r$  yang sangat rendah sehingga tidak digunakan sebagai input. Variabel kelembaban tidak digunakan karena nilai koefisien  $r$  yang sangat rendah serta nilai signifikansi yang sangat tinggi yaitu 0,303. Hasil pengujian korelasi variabel input yang akan digunakan pada metode JST untuk prediksi ketinggian gelombang ditunjukkan pada Tabel 4.4 di bawah ini.

**Tabel 4. 4** Uji Korelasi Variabel Input JST Prediksi Ketinggian Gelombang

<b>Variabel Input</b>	<b>Korelasi</b>
Kecepatan Angin (t-12)	0,6482
Arah Angin (t-12)	0,2058
Tinggi Gelombang (t-12)	0,9063
Tinggi Gelombang (t)	0,9642

Hasil pengujian korelasi menunjukkan bahwa variabel input yang digunakan pada prediktor ketinggian gelombang air laut 12 jam kedepan adalah kecepatan angin 12 jam sebelumnya, arah angin 12 jam sebelumnya, tinggi gelombang 12 jam sebelumnya, dan tinggi gelombang saat ini. Hasil uji korelasi tersebut menunjukkan hasil bahwa variabel-variabel tersebut berkorelasi tinggi dengan variabel output berupa tinggi gelombang air laut 12 jam kedepan. Nilai korelasi relatif tinggi melebihi 0,5 dan termasuk kategori kuat sehingga dapat digunakan sebagai input.

Hasil pengujian korelasi variabel yang mempengaruhi kecepatan arus laut ditunjukkan pada Tabel 4.5. Variabel yang mempengaruhi yaitu gelombang, suhu permukaan laut, dan angin.

**Tabel 4. 5** Uji Korelasi Variabel Prediksi Arus Laut

Variabel	Korelasi (Nilai r)	Signifikansi
Kecepatan Angin	0,538	0,000
Arah Angin	-0,034	0,000
Tinggi Gelombang	0,571	0,000
Suhu Permukaan Laut	-0,238	0,000
Tekanan	0,080	0,000
Kelembaban	0,018	0,191

Hasil uji korelasi pada variabel input prediksi arus laut didapatkan bahwa kecepatan angin, tinggi gelombang, dan suhu permukaan laut berkorelasi tinggi, sedangkan untuk tekanan, arah angin, dan kelembaban memiliki korelasi yang rendah dengan arus laut. Hasil korelasi bernilai positif menunjukkan bahwa hubungan antar variabel berbanding lurus sedangkan bernilai negatif menunjukkan bahwa hubungan antar variabel berbanding terbalik. Tingkat korelasi (koefisien r) pada kecepatan angin dan tinggi gelombang termasuk dalam kategori kuat sedangkan suhu permukaan laut termasuk dalam kategori rendah, meskipun suhu permukaan laut berkategori rendah tetapi masih bisa digunakan karena nilai rata-rata signifikansi bernilai 0,00. Nilai signifikansi apabila dibawah 0,05 maka data layak digunakan atau dikatakan berkorelasi. Variabel arah angin, tekanan, dan kelembaban memiliki nilai koefisien r yang sangat rendah sehingga tidak digunakan sebagai input. Variabel kelembaban tidak digunakan karena nilai koefisien r yang sangat rendah serta nilai signifikansi yang sangat tinggi yaitu 0,191. Hasil pengujian korelasi variabel input yang akan digunakan pada metode JST untuk prediksi kecepatan arus laut ditunjukkan pada Tabel 4.6 di bawah ini.

**Tabel 4. 6** Uji Korelasi Variabel Input JST Prediksi Kecepatan Arus Laut

Variabel Input	Korelasi
Kecepatan Angin (t-12)	0,4414
Kecepatan Angin (t)	0,4930
Suhu Permukaan Laut (t)	0,4542
Tinggi Gelombang (t)	0,5104

Hasil pengujian korelasi menunjukkan bahwa variabel input yang digunakan pada prediktor kecepatan arus laut 12 jam kedepan adalah kecepatan angin 12 jam sebelumnya, kecepatan angin saat ini, suhu permukaan laut saat ini, dan tinggi gelombang saat ini. Hasil uji korelasi tersebut menunjukkan hasil bahwa variabel-variabel tersebut berkorelasi tinggi dengan variabel output berupa arus laut 12 jam kedepan. Nilai korelasi di atas termasuk kategori sedang untuk kecepatan angin dan suhu permukaan laut dan kategori tinggi untuk tinggi gelombang berdasarkan Tabel. 3.2 untuk tingkat hubungan korelasi antar variabel yang digunakan.

#### 4.2 Prediksi Gelombang Menggunakan JST-LM

Prediksi ketinggian gelombang laut menggunakan metode JST-LM melalui 2 tahap yaitu pelatihan dan pengujian. Pelatihan JST-LM dilakukan dengan menentukan nilai parameter laju pelatihan dan fungsi aktivasi yang digunakan. Laju pelatihan divariasikan dengan memberikan nilai 0,1; 0,01; 0,001, 0,5; 0,05; dan 0,005 melalui proses *trial and error*. Fungsi aktivasi yang divariasikan yaitu fungsi aktivasi logsig (sigmoid biner), fungsi aktivasi linier (*pureline*), dan tansig (sigmoid bipolar). Variasi laju pelatihan dan jenis fungsi aktivasi, dilakukan uji coba untuk memilih jaringan yang menghasilkan RMSE terkecil pada prediktor ketinggian gelombang air laut. Nilai RMSE hasil performansi berdasarkan variasi besarnya laju pelatihan dan jenis fungsi aktivasi ditunjukkan pada Tabel 4.7.

Nilai RMSE terkecil berdasarkan *trial and error* pemilihan laju pelatihan dan fungsi aktivasi pada Tabel 4.8 adalah 0,0573. Laju pelatihan yang digunakan yaitu sebesar 0,01. Fungsi aktivasi yang digunakan yaitu logsig (sigmoid biner) untuk input ke *hidden* dan tansig (sigmoid bipolar) untuk *hidden* ke output. Hasil pemilihan jenis fungsi aktivasi menunjukkan apabila menggunakan *pureline* (linier) menghasilkan nilai RMSE yang cukup besar, sehingga penggunaan *pureline* kurang tepat untuk prediktor ketinggian gelombang. Fungsi aktivasi logsig atau tansig menghasilkan nilai RMSE yang kecil.

**Tabel 4.7** Pengujian Fungsi Aktivasi dan Laju Pelatihan JST Prediktor Ketinggian Gelombang

		RMSE (m)		
Input ke <i>Hidden</i>	<i>Hidden</i> ke Output	Fs Aktivasi		
		0,5	0,05	0,005
logsig	logsig	0,0590	0,0586	0,0581
tansig	logsig	0,0581	0,0581	0,0583
purelin	logsig	0,0640	0,0640	0,0640
logsig	tansig	0,0597	0,0580	0,0575
tansig	tansig	0,0578	0,0583	0,0581
purelin	tansig	0,0656	0,0656	0,0656
logsig	purelin	0,0576	0,0577	0,0589
tansig	purelin	0,0587	0,0581	0,0586
purelin	purelin	0,0624	0,0624	0,0624
Input ke <i>Hidden</i>	<i>Hidden</i> ke Output	Fs Aktivasi		
		0,1	<b>0,01</b>	0,001
logsig	logsig	0,0587	0,0582	0,0579
tansig	logsig	0,0581	0,0574	0,0590
purelin	logsig	0,0640	0,0640	0,0640
<b>logsig</b>	<b>tansig</b>	0,0590	<b>0,0573</b>	0,0589
tansig	tansig	0,0587	0,0579	0,0581
purelin	tansig	0,0656	0,0656	0,0656
logsig	purelin	0,0583	0,0581	0,0583
tansig	purelin	0,0578	0,0583	0,0576
purelin	purelin	0,0624	0,0624	0,0624

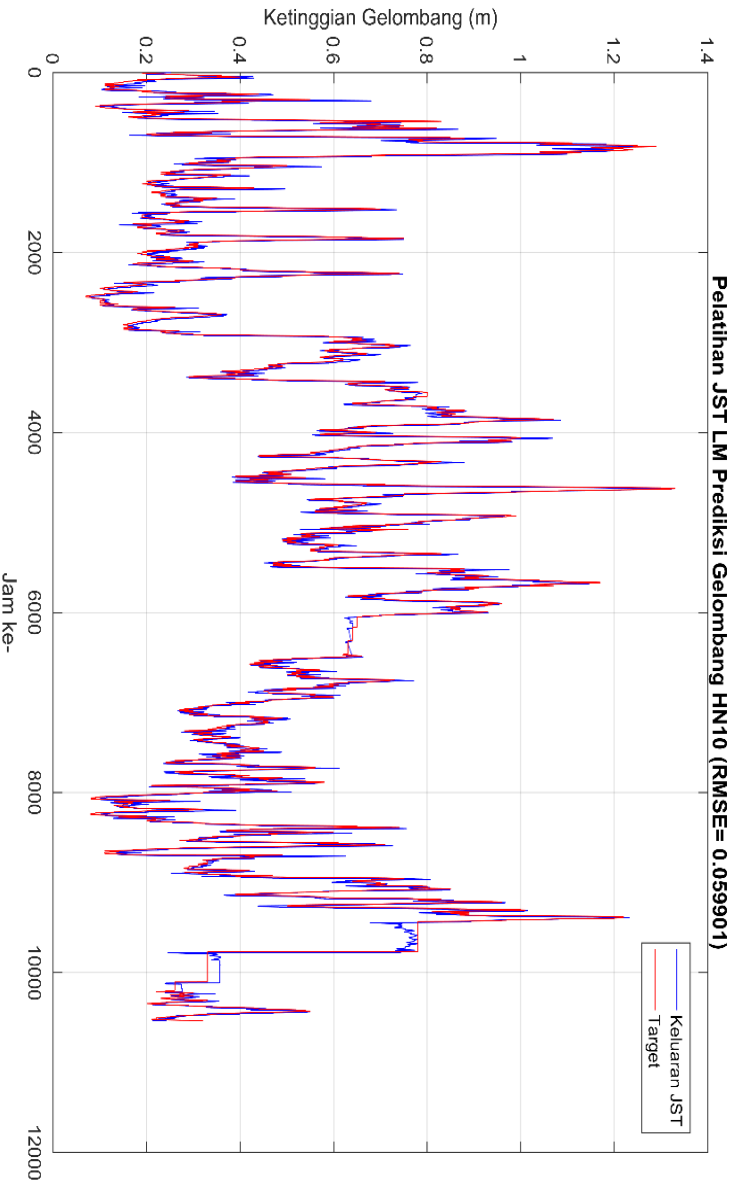
Tahap pertama penggunaan JST yaitu proses pelatihan dengan memvariasikan jumlah *hidden node* sebanyak 22 arsitektur JST-LM. Nilai RMSE yang didapatkan dari proses pelatihan dan pengujian prediksi ketinggian gelombang air laut menggunakan JST-LM berdasarkan variasi *hidden node* ditunjukkan pada Tabel 4.8. Terdapat dua macam nilai RMSE yaitu pada proses pelatihan (RMSE latih) dan pada proses pengujian (RMSE uji). Nilai RMSE terkecil didapatkan pada *hidden node* 10 atau arsitektur 4-10-1 yaitu sebesar 0,059061 m untuk pelatihan dan 0,059060 m untuk pengujian. Pemilihan arsitektur terbaik ini berdasarkan nilai RMSE terkecil pada proses pengujian.



**Tabel 4.8** Nilai RMSE Metode JST-LM Prediktor Ketinggian Gelombang

<b>Jumlah <i>Hidden Node</i></b>	<b>RMSE (m) Pelatihan JST LM</b>	<b>RMSE (m) Pengujian JST LM</b>
1	0,062363	0,059345
2	0,061890	0,068021
3	0,061761	0,059114
4	0,061246	0,060650
5	0,061137	0,063632
6	0,060371	0,069036
7	0,060652	0,061476
8	0,059595	0,059980
9	0,058560	0,065326
<b>10</b>	<b>0,059061</b>	<b>0,059060</b>
11	0,058258	0,072754
12	0,057905	0,076065
13	0,057154	0,078625
14	0,057176	0,065177
15	0,056673	0,067352
20	0,055332	0,102600
25	0,053981	0,064236
30	0,052588	0,073003
35	0,051759	0,091567
40	0,051449	0,121390
45	0,049952	0,136520
50	0,049516	0,072024

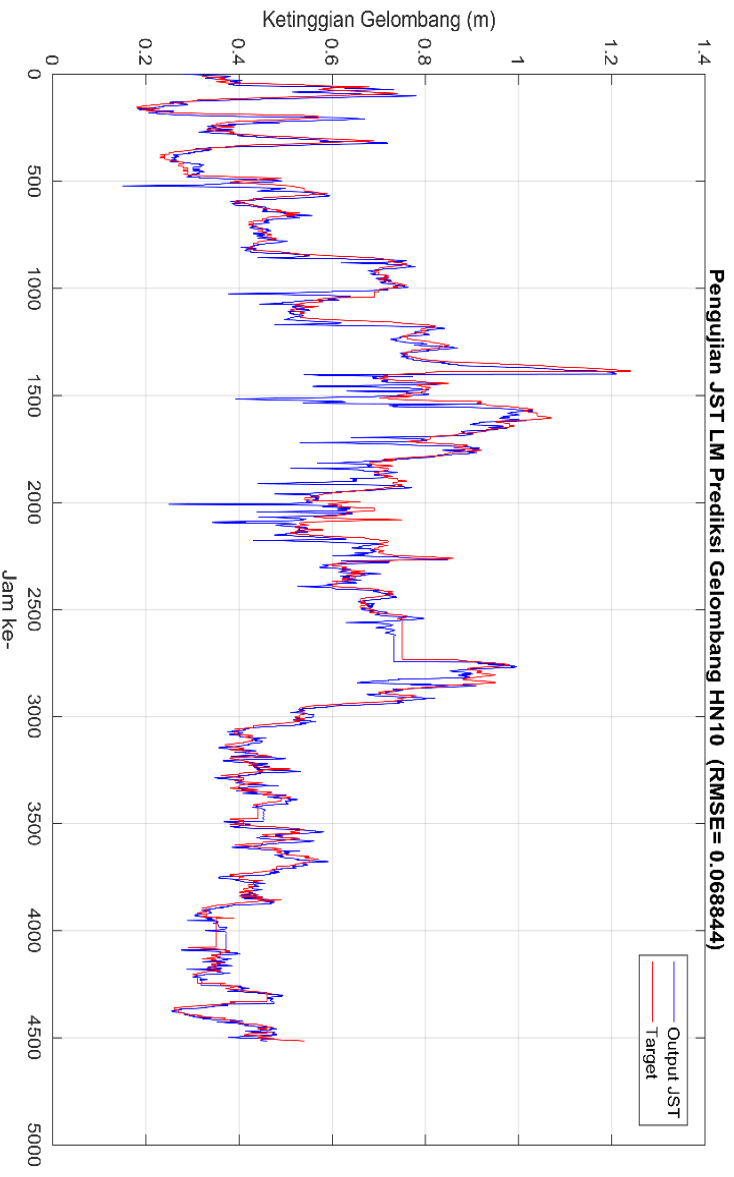
Grafik hasil pelatihan JST-LM pada arsitektur 4-10-1 ditunjukkan pada Gambar 4.1. Grafik hasil pelatihan tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada arsitektur 4-10-1 yaitu sebesar 0,059061 m.. Grafik hasil pelatihan JST-LM menghasilkan nilai RMSE lebih besar dari proses pengujian. Nilai RMSE didapatkan dari proses *running* program MATLAB JST-LM berdasarkan *source code* yang ditunjukkan pada Lampiran E.



**Gambar 4. 1** Hasil Pelatihan JST-LM Prediktor Ketinggian Gelombang Arsitektur 4-10-1

Grafik hasil pengujian JST-LM pada arsitektur 4-10-1 ditunjukkan pada Gambar 4.2. Grafik hasil pengujian tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada arsitektur 4-10-1 yaitu sebesar 0,059060 m. Proses pengujian JST-LM pada program MATLAB ini menggunakan bobot yang telah didapatkan pada proses pelatihan, kemudian dilakukan pengujian terhadap data di luar data yang digunakan pada proses pelatihan. Proses ini bertujuan untuk mengetahui tingkat keakuratan dari arsitektur jaringan prediktor yang telah dibuat. Grafik hasil pengujian JST-LM menghasilkan nilai RMSE lebih kecil dari pelatihan. Pemilihan arsitektur jaringan terbaik ini berdasarkan nilai RMSE pengujian yang terkecil, karena menunjukkan tingkat keakuratan prediktor. Nilai RMSE pelatihan terkecil yang didapatkan pada *hidden node* 50 tetapi pada proses pengujian menghasilkan nilai RMSE yang lebih besar. *Hidden node* 10 menghasilkan selisih yang sedikit antara nilai RMSE pelatihan dan pengujian. Penambahan *hidden node* pada jaringan belum tentu menghasilkan nilai RMSE lebih kecil. Jumlah *epoch* yang digunakan pada JST-LM yaitu sebanyak 250 *epoch* ini berdasarkan *trial and error* dalam mencapai nilai konvergensi yang dihasilkan setiap *running program*.

Nilai RMSE ini lebih kecil apabila dibandingkan dengan penelitian sebelumnya oleh Ukhti pada tahun 2018 pada penggunaan JST-LM untuk prediksi ketinggian gelombang. Hasil ini juga lebih baik apabila dibandingkan dengan penelitian Raharja yang menggunakan metode JST dengan algoritma pelatihan *backpropagation* pada tahun 2018 dari segi banyaknya jumlah iterasi yang digunakan. Penelitian ini hanya memerlukan 250 iterasi untuk mencapai konvergen sedangkan penelitian Raharja memerlukan 10.000 iterasi. Hasil penelitian ini juga sesuai dengan kajian dari Sharma pada tahun 2014 mengenai kecepatan dalam mencapai konvergensi pada metode JST-LM. Waktu yang diperlukan pada penelitian ini yaitu 300 detik untuk mencapai konvergensi pada 250 iterasi.



**Gambar 4. 2** Hasil Pengujian JST-LM Prediktor Ketinggian Gelombang Arsitektur 4-10-1

### 4.3 Prediksi Kecepatan Arus Laut Menggunakan JST-LM

Prediksi kecepatan arus laut menggunakan metode JST-LM melalui 2 tahap yaitu pelatihan dan pengujian. Pelatihan JST dilakukan dengan menentukan nilai parameter laju pelatihan dan fungsi aktivasi yang digunakan. Laju pelatihan divariasikan dengan memberikan nilai 0,1; 0,01; 0,001, 0,5; 0,05; dan 0,005 melalui proses *trial and error*. Fungsi aktivasi yang divariasikan yaitu fungsi aktivasi logsig (sigmoid biner), fungsi aktivasi linier (*pureline*) dan tansig (sigmoid bipolar). Nilai RMSE hasil performansi berdasarkan variasi besarnya laju pelatihan dan jenis fungsi aktivasi ditunjukkan pada Tabel 4.9.

**Tabel 4. 9** Pengujian Fungsi Aktivasi dan Laju Pelatihan JST Prediktor Arus Laut

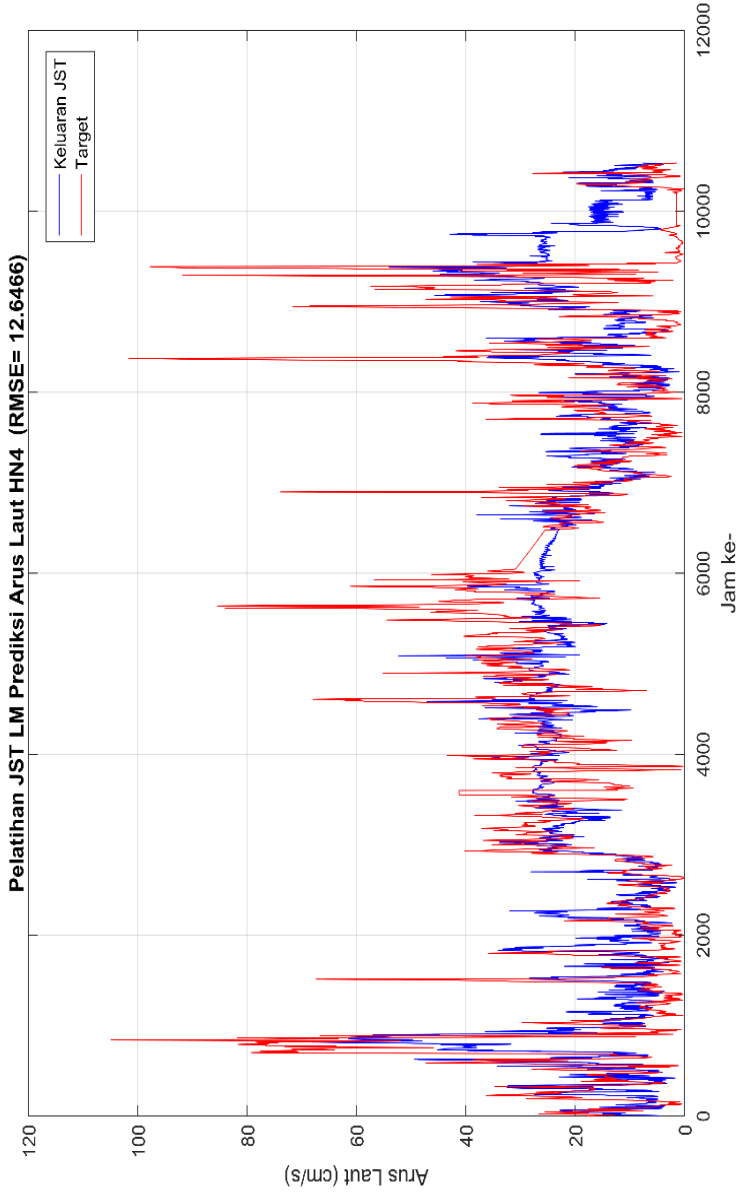
		RMSE (cm/s)		
Input ke <i>Hidden</i>	<i>Hidden</i> ke Output	Fs Aktivasi		
		0,5	<b>0,05</b>	0,005
logsig	logsig	11,0363	<b>10,6178</b>	10,9996
tansig	logsig	11,1089	10,9655	10,8048
purelin	logsig	13,0398	13,0398	13,0398
logsig	tansig	11,3686	10,7516	11,0973
tansig	tansig	11,2225	11,3324	11,435
purelin	tansig	13,0308	13,0308	13,0308
logsig	purelin	11,1289	10,9857	10,8194
tansig	purelin	10,8697	10,8698	10,8025
purelin	purelin	13,0164	13,0164	13,0164
Input ke <i>Hidden</i>	<i>Hidden</i> ke Output	Fs Aktivasi		
		0,1	0,01	0,001
logsig	logsig	11,0453	11,1121	11,178
tansig	logsig	10,7515	11,2053	11,1969
purelin	logsig	13,0398	13,0398	13,0398
logsig	tansig	11,2462	11,1846	11,0767
tansig	tansig	10,9192	10,6544	11,0841
purelin	tansig	13,0308	13,0308	13,0308
logsig	purelin	11,0256	11,2717	10,903
tansig	purelin	11,099	11,2254	11,0296
purelin	purelin	13,0164	13,0164	13,0164

Nilai RMSE terkecil berdasarkan *trial and error* pemilihan besarnya laju pelatihan dan jenis fungsi aktivasi pada Tabel 4.9 adalah 10,6178 cm/s. Laju pelatihan yang digunakan yaitu sebesar 0,05. Fungsi aktivasi yang digunakan yaitu logsig (sigmoid biner) untuk input ke *hidden* dan logsig (sigmoid biner) untuk *hidden* ke output.

Tahap pertama penggunaan JST yaitu proses pelatihan dengan memvariasikan jumlah *hidden node* sebanyak 22 arsitektur JST-LM. Nilai RMSE yang didapatkan dari proses pelatihan dan pengujian prediksi kecepatan arus laut menggunakan JST-LM berdasarkan variasi *hidden node* ditunjukkan pada Tabel 4.10.

**Tabel 4. 10** Nilai RMSE Metode JST-LM Prediktor Arus Laut

<b>Jumlah <i>Hidden Node</i></b>	<b>RMSE (cm/s) Pelatihan JST LM</b>	<b>RMSE (cm/s) Pengujian JST LM</b>
1	13,03	10,96
2	12,42	11,04
3	12,35	11,26
<b>4</b>	<b>12,24</b>	<b>10,93</b>
5	11,81	11,49
6	11,85	11,37
7	11,45	11,59
8	11,27	11,35
9	11,14	11,42
10	10,68	11,55
11	11,15	11,52
12	10,85	11,60
13	10,80	12,09
14	10,79	12,38
15	10,55	11,90
20	10,03	12,89
25	9,72	12,05
30	9,42	12,04
35	9,13	12,86
40	8,79	12,45
45	8,52	13,67
50	8,45	13,31

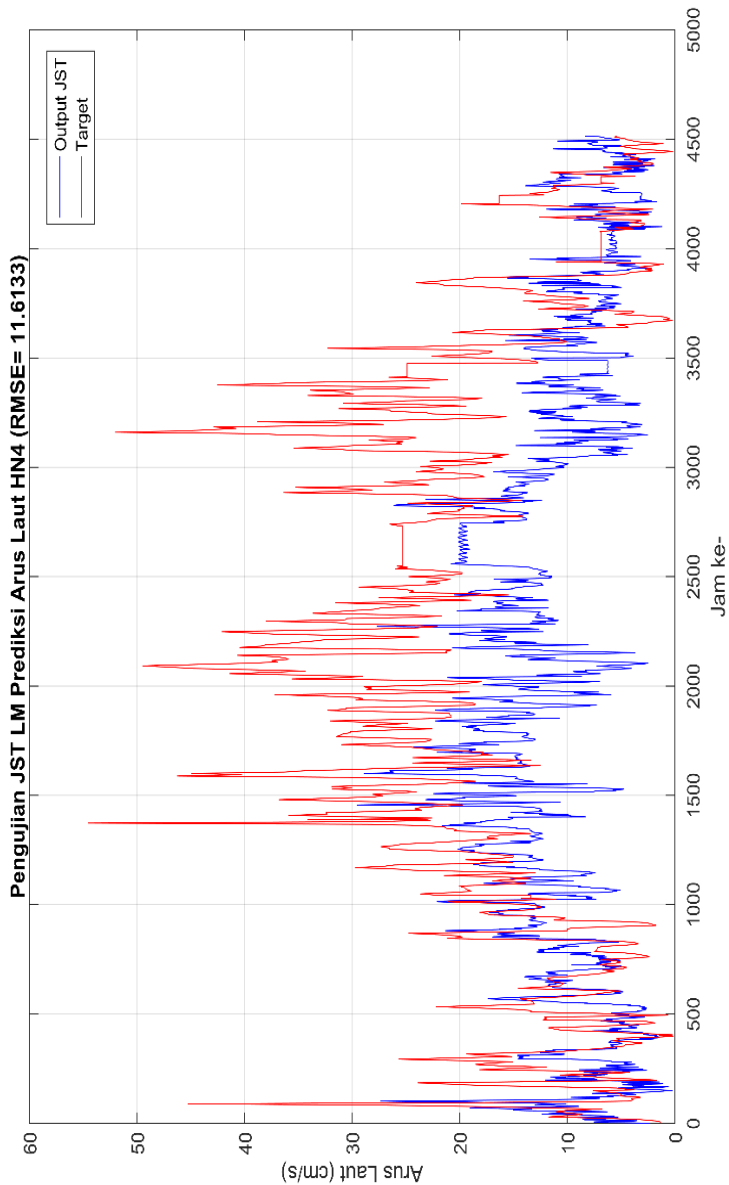


**Gambar 4. 3** Hasil Pelatihan JST-LM Prediktor Arus Laut Arsitektur 4-4-1

Grafik hasil pelatihan JST-LM pada prediktor kecepatan arus laut arsitektur 4-4-1 ditunjukkan pada Gambar 4.3. Grafik hasil pelatihan tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada arsitektur 4-4-1 yaitu sebesar 12,6466 cm/s. Nilai bobot hasil JST-LM pada *hidden node* 4 dapat dilihat pada Tabel 3.11 dengan bobot total yang dihasilkan sebanyak 25. Grafik hasil pelatihan JST-LM menghasilkan nilai RMSE lebih besar dari pengujian. Penambahan *hidden node* pada jaringan belum tentu menghasilkan nilai RMSE lebih kecil.

Grafik hasil pengujian JST-LM pada arsitektur 4-4-1 ditunjukkan pada Gambar 4.4. Grafik hasil pengujian tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada pengujian pada arsitektur 4-4-1 yaitu sebesar 11,6133 cm/s. Proses pengujian JST-LM pada program MATLAB ini menggunakan bobot yang telah didapatkan pada proses pelatihan, kemudian dilakukan pengujian terhadap data di luar data yang digunakan pada proses pelatihan. Proses ini bertujuan untuk mengetahui tingkat keakuratan dari arsitektur jaringan prediktor yang telah dibuat. Grafik hasil pengujian JST-LM menghasilkan nilai RMSE lebih kecil dari pengujian. Pemilihan arsitektur jaringan terbaik ini berdasarkan nilai RMSE pengujian yang terkecil. Nilai RMSE pelatihan terkecil yang didapatkan pada *hidden node* 50 tetapi pada proses pengujian menghasilkan nilai RMSE yang lebih besar. *Hidden node* 4 menghasilkan selisih yang sedikit antara nilai RMSE pelatihan dan pengujian. Penambahan *hidden node* pada jaringan belum tentu menghasilkan nilai RMSE lebih kecil. Jumlah *epoch* yang digunakan pada JST-LM yaitu sebanyak 250 *epoch* ini berdasarkan *trial and error* dalam mencapai nilai konvergensi yang dihasilkan setiap *running program*. Setiap *running program* nilai RMSE yang dihasilkan berubah-ubah, maka belum mencapai nilai konvergen. Penelitian ini mencapai nilai konvergen untuk JST-LM pada 250 *epoch*

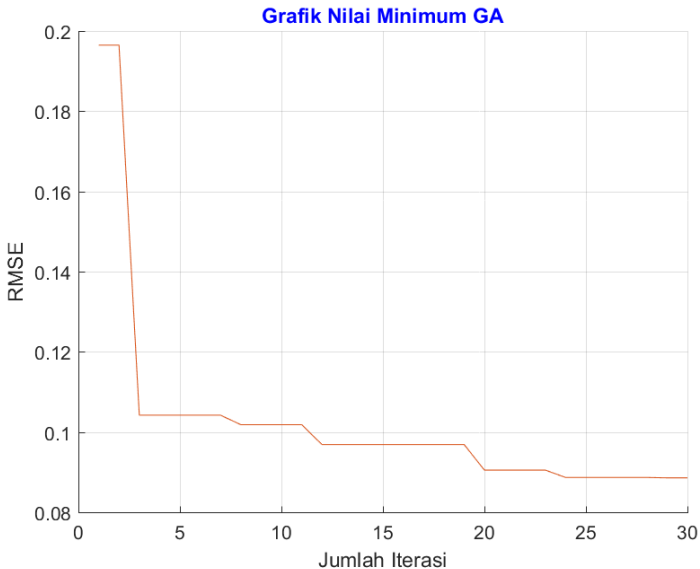




**Gambar 4. 4** Hasil Penguujian JST-LM Predikor Arus Laut Arsitektur 4-4-1

#### 4.4 Prediksi Gelombang Menggunakan JST-GA

Metode kedua yang digunakan pada prediktor ketinggian gelombang air laut ini yaitu JST-GA. Parameter yang digunakan pada proses optimasi ini telah dijelaskan pada sub bab 3.5.1. Grafik performansi pelatihan JST-GA ditunjukkan pada Gambar 4.5. Nilai minimum GA pada grafik yaitu nilai *fitness* dari persamaan fungsi objektif. Fungsi objektif yang digunakan yaitu nilai RMSE. Nilai RMSE paling optimal yang dihasilkan dari proses optimasi GA sebesar 0,0854 m.



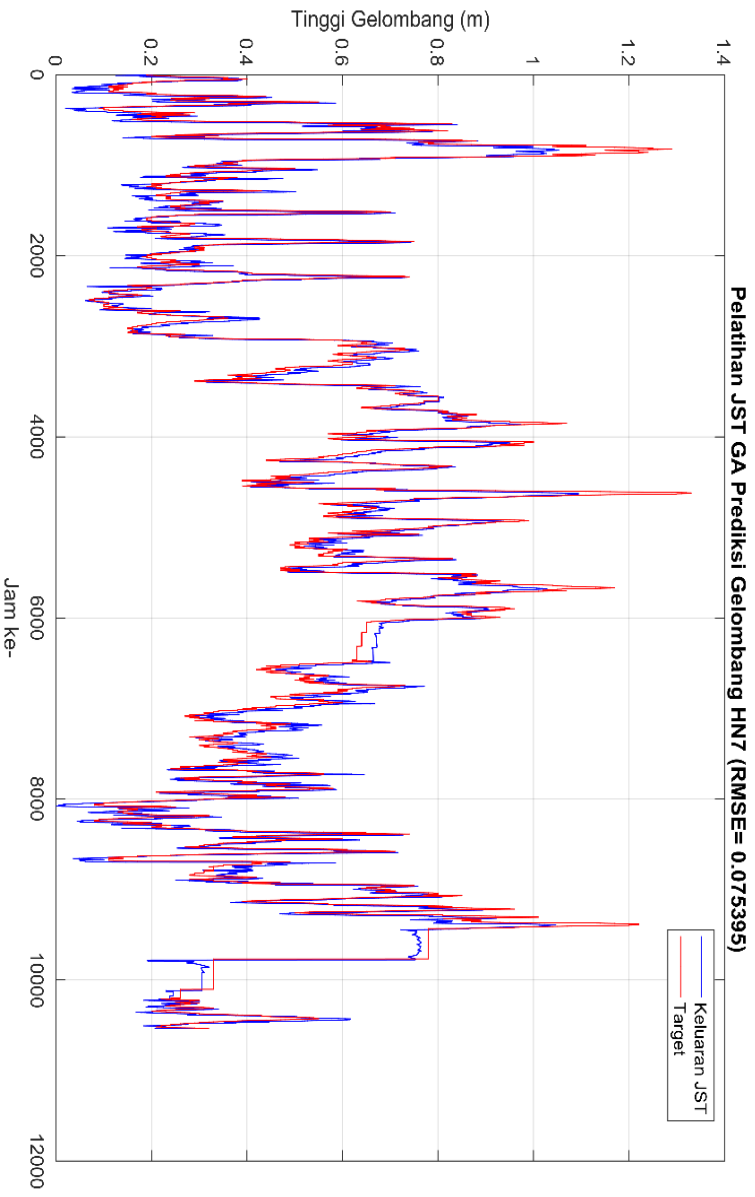
**Gambar 4. 5** Grafik Performansi Optimasi *Genetic Algorithm* (GA) Prediktor Gelombang

Hasil performansi nilai RMSE pelatihan dan pengujian pada prediktor ketinggian gelombang air laut dengan metode JST-GA ditunjukkan pada Tabel 4.11. Terdapat dua macam nilai RMSE yaitu pada proses pelatihan (RMSE latih) dan pada proses pengujian (RMSE uji). Variasi *hidden node* yang digunakan pada metode JST-GA ini yaitu sebanyak 15 variasi dengan 30 *epoch*.

Waktu yang dibutuhkan untuk memproses optimasi GA pada bobot-bobot JST ini yaitu 35 menit untuk 30 *epoch*. Nilai RMSE terbaik pada *hidden node* 7 dilakukan variasi jumlah *epoch* yang digunakan yaitu 30, 60, dan 90 *epoch* untuk mengetahui pengaruh *epoch* terhadap nilai RMSE. Nilai RMSE terkecil didapatkan pada *hidden node* 7 atau arsitektur 4-7-1 pada 30 *epoch* yaitu sebesar 0,075395 m untuk pelatihan dan 0,057556 m untuk pengujian. Jumlah bobot yang dioptimasi pada arsitektur ini yaitu sebanyak 43 bobot yang telah dituliskan pada Tabel 3.9. Pemilihan arsitektur terbaik ini berdasarkan nilai RMSE terkecil pada proses pengujian. Pemilihan arsitektur terbaik ini berdasarkan nilai RMSE terkecil pada proses pengujian serta selisih antara nilai RMSE pelatihan dan pengujian relatif kecil. Nilai RMSE didapatkan dari proses *running* program MATLAB JST-GA berdasarkan *source code* yang ditunjukkan pada Lampiran F.

**Tabel 4. 11** Nilai RMSE Hasil Pelatihan JST-GA Prediktor Ketinggian Gelombang

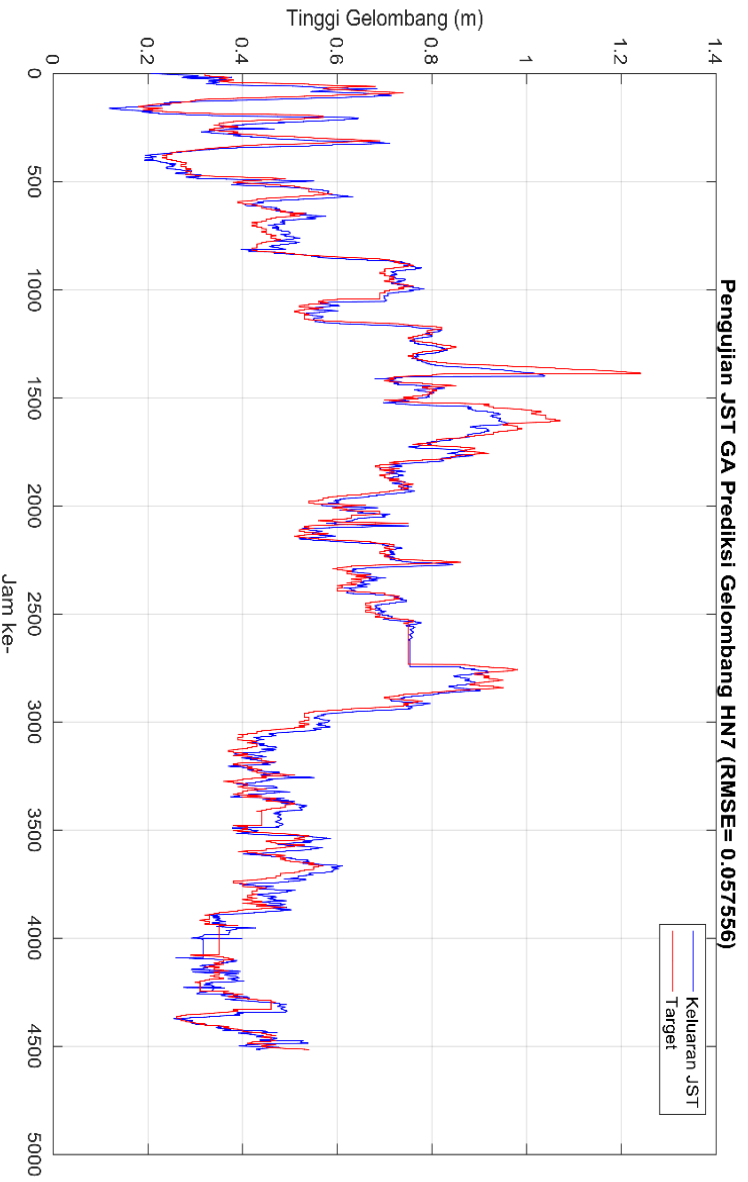
Jumlah <i>Hidden Node</i>	RMSE (m) Pelatihan JST GA	RMSE (m) Pengujian JST GA
1	0,124570	0,097725
2	0,103208	0,078521
3	0,124170	0,089322
4	0,074237	0,062711
5	0,088807	0,071033
6	0,099886	0,075870
<b>7</b>	<b>0,075395</b>	<b>0,057556</b>
7 iterasi 60	0,073603	0,059937
7 iterasi 90	0,073603	0,059937
8	0,099395	0,091250
9	0,080087	0,071890
10	0,105230	0,086897
11	0,081165	0,063449
12	0,154970	0,111730
13	0,125210	0,095166
14	0,094358	0,085813
15	0,124000	0,092405



**Gambar 4. 6** Hasil Pelatihan JST-GA Prediktor Gelombang Arsitektur 4-7-1

Grafik hasil pelatihan JST-GA pada prediktor ketinggian gelombang arsitektur 4-7-1 ditunjukkan pada Gambar 4.6. Grafik hasil pelatihan tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada arsitektur 4-7-1 yaitu sebesar 0,075395. Nilai bobot hasil JST-GA pada *hidden node 7* ditunjukkan pada Tabel 3.9 dengan bobot total hasil optimasi yang dihasilkan sebanyak 43. Bobot hasil optimasi GA merupakan hasil dari bobot JST yang telah dioptimasi. Grafik hasil pelatihan JST-GA menghasilkan nilai RMSE lebih besar dari pengujian. Penambahan *hidden node* dan jumlah *epoch* belum tentu menghasilkan nilai RMSE lebih kecil. Semakin banyak jumlah *epoch* yang digunakan maka waktu yang dibutuhkan juga akan semakin lama. Penambahan 30 *epoch* menyebabkan waktu yang dibutuhkan 2 kali lebih lama yaitu 60 - 90 menit. Grafik hasil pelatihan JST-GA untuk *hidden node 1, 2, 3, 4, .....*, 15 ditunjukkan pada Lampiran A.

Nilai RMSE yang dihasilkan dari proses pelatihan dan pengujian dengan metode JST-GA ini menghasilkan nilai yang hampir sama. Nilai RMSE pada pelatihan kecil maka nilai RMSE pada pengujian juga kecil begitu juga sebaliknya. *Hidden node 7* dengan 30 *epoch* menghasilkan nilai RMSE yang paling optimal apabila dibandingkan dengan 60 *epoch* dan 90 *epoch*. Pemilihan jumlah 30 *epoch* pada penelitian ini berdasarkan *trial and error*, yaitu dengan memvariasikan jumlah *epoch* dari 5, 10, 15, 20, 25, dan 30 *epoch*. Jumlah *epoch* yang dipilih berdasarkan nilai RMSE yang dihasilkan paling kecil serta kemampuan dalam mencapai nilai konvergen pada proses *running data* di MATLAB. Penambahan *epoch* pada *hidden node 7* menghasilkan nilai RMSE yang semakin kecil pada proses pelatihan, tetapi pada proses pengujian nilai RMSE semakin besar. Grafik keluaran prediksi ketinggian gelombang dengan metode JST-GA pada proses pelatihan dapat mengikuti target keluaran tetapi pada puncak-puncak tinggi gelombang tertentu yang ekstrem seperti pada tinggi gelombang 1,25 m, 1,3 m, dan 1,4 m keluaran JST tidak dapat mengikuti target.



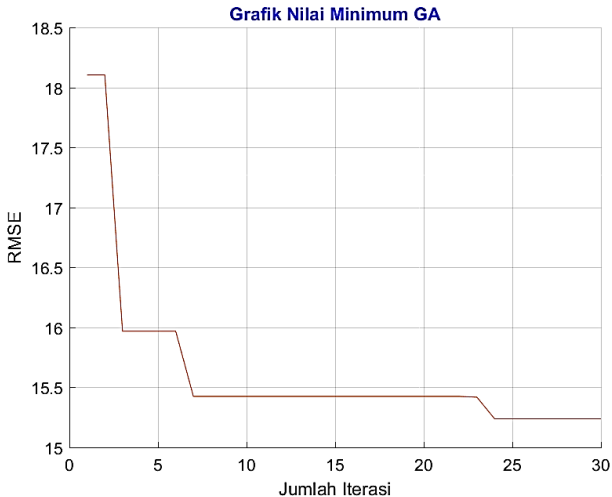
**Gambar 4. 7** Hasil Pengujian JST-GA Prediktor Gelombang Arsitektur 4-7-1

Grafik hasil pengujian JST-GA pada arsitektur 4-7-1 ditunjukkan pada Gambar 4.7. Grafik hasil pengujian tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada arsitektur 4-7-1 yaitu sebesar 0,057556. Proses pengujian JST-GA pada program MATLAB ini menggunakan bobot yang telah didapatkan pada proses pelatihan melalui optimasi GA, kemudian dilakukan pengujian terhadap data di luar data yang digunakan pada proses pelatihan untuk mengetahui keakuratan dari prediktor. Grafik hasil pengujian JST-GA menghasilkan nilai RMSE lebih kecil dari pelatihan. Pemilihan arsitektur jaringan terbaik ini berdasarkan nilai RMSE pengujian yang terkecil. Penambahan *hidden node* belum tentu menghasilkan nilai RMSE lebih kecil. Nilai RMSE ini lebih kecil apabila dibandingkan dengan penelitian sebelumnya oleh Ukhti pada tahun 2018 pada penggunaan JST-PSO untuk prediksi ketinggian gelombang. Grafik hasil pengujian JST-GA *hidden node* 1, 2, 3, ..., 15 ditunjukkan pada Lampiran B.

Metode JST-GA pada prediktor ketinggian gelombang menghasilkan nilai RMSE lebih kecil yaitu 0,0575 m apabila dibandingkan dengan metode JST-PSO yang dilakukan oleh Ukhti pada tahun 2018 menghasilkan RMSE sebesar 0,0590 m. Waktu yang diperlukan untuk memproses JST-GA ini juga lebih cepat apabila dibandingkan dengan JST-PSO. Penelitian ini juga sesuai dengan penelitian yang telah dilakukan oleh Zhang pada tahun 2009 bahwa penggunaan optimasi GA pada JST dapat menghasilkan akurasi yang baik dengan waktu yang cepat. Metode JST-GA memerlukan waktu 35 menit untuk 30 *epoch* sedangkan metode JST-PSO memerlukan waktu 85 menit untuk 30 *epoch*. Nilai RMSE yang dihasilkan pada prediktor ketinggian gelombang dengan JST-GA lebih kecil daripada JST-LM pada penelitian ini serta JST-LM yang pernah dilakukan Ukhti pada tahun 2018. Hasil ini juga lebih baik apabila dibandingkan dengan metode JST *backpropagation* yang pernah dilakukan oleh Raharja pada tahun 2018 dan Rizianiza pada tahun 2015 dari segi nilai RMSe dan banyaknya jumlah epoch yang digunakan.

#### 4.5 Prediksi Kecepatan Arus Laut Menggunakan JST-GA

Metode kedua yang digunakan pada prediktor kecepatan arus laut yaitu JST-GA. Parameter yang digunakan pada proses optimasi ini telah dijelaskan pada sub bab 3.5.2. Grafik performansi pelatihan JST-GA ditunjukkan pada Gambar 4.8. Nilai minimum GA pada grafik yaitu nilai *fitness* dari persamaan fungsi objektif. Fungsi objektif yang digunakan yaitu nilai RMSE. Nilai RMSE paling optimal yang dihasilkan dari proses optimasi GA sebesar 15,24 cm/s.



**Gambar 4. 8** Grafik Performansi Optimasi Genetic Algorithm (GA) Prediktor Arus Laut

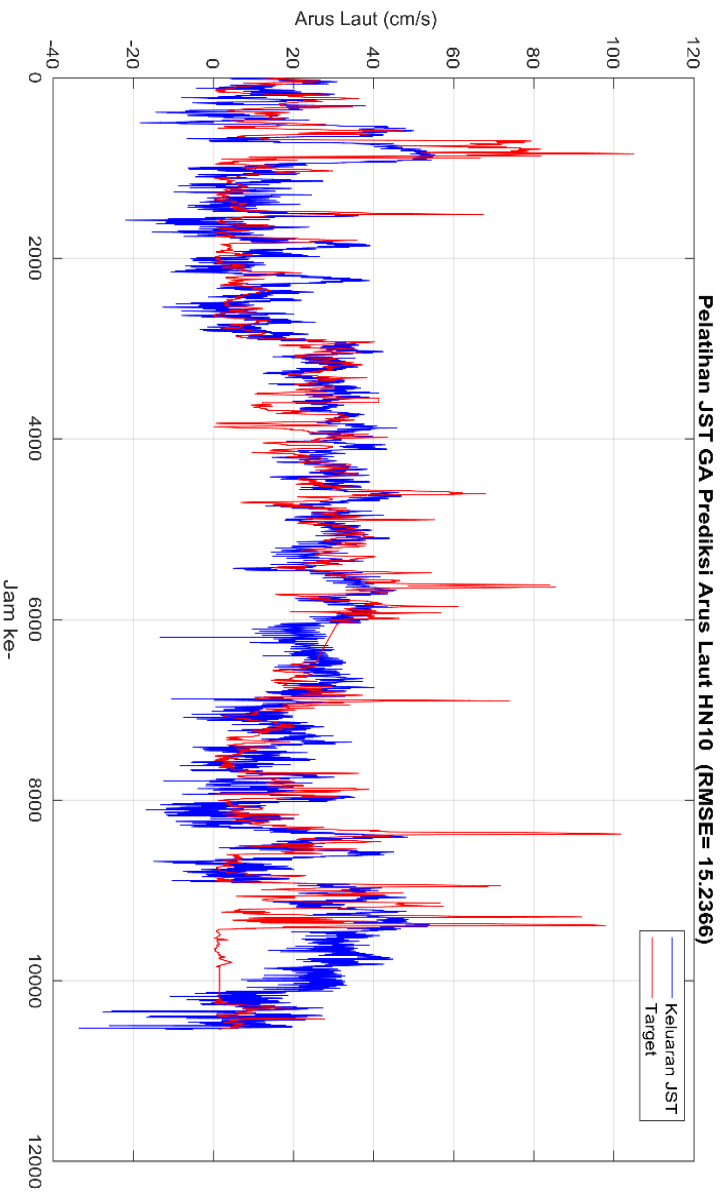
Hasil performansi nilai RMSE pelatihan dan pengujian pada prediktor kecepatan arus laut dengan metode JST-GA ditunjukkan pada Tabel 4.12. Terdapat dua macam nilai RMSE yaitu pada proses pelatihan (RMSE latih) dan pada proses pengujian (RMSE uji). Variasi *hidden node* yang digunakan pada metode JST-GA ini yaitu sebanyak 15 variasi dengan 30 *epoch*. Waktu yang dibutuhkan untuk memproses optimasi GA pada bobot-bobot JST ini yaitu 35 menit untuk 30 *epoch*. Nilai RMSE terbaik pada *hidden node* 10



dilakukan variasi jumlah *epoch* yang digunakan yaitu 30, 60, dan 90 *epoch* untuk mengetahui pengaruh *epoch* terhadap nilai RMSE. Nilai RMSE terkecil didapatkan pada *hidden node* 10 atau arsitektur 4-10-1 pada 30 *epoch* yaitu sebesar 15,24 cm/s untuk pelatihan dan 8,66 cm/s untuk pengujian. Jumlah bobot yang dioptimasi pada arsitektur ini yaitu sebanyak 61. *Hidden node* 10 dengan 30 *epoch* menghasilkan nilai RMSE terbaik dibandingkan dengan 60 dan 90 *epoch*. Penambahan jumlah *epoch* belum tentu menghasilkan nilai RMSE yang lebih kecil. Proses pelatihan pada *hidden node* 10 dengan penambahan *epoch* menghasilkan nilai RMSE yang lebih kecil tetapi pada proses pengujian nilai RMSE lebih besar. Pemilihan arsitektur terbaik ini berdasarkan RMSE terkecil pada proses pengujian serta selisih antara RMSE pelatihan dan pengujian relatif kecil. Nilai RMSE yang kecil pada pelatihan belum tentu menghasilkan RMSE yang kecil pada pengujian

**Tabel 4. 12** Nilai RMSE Hasil Pelatihan JST-GA Prediktor Arus laut

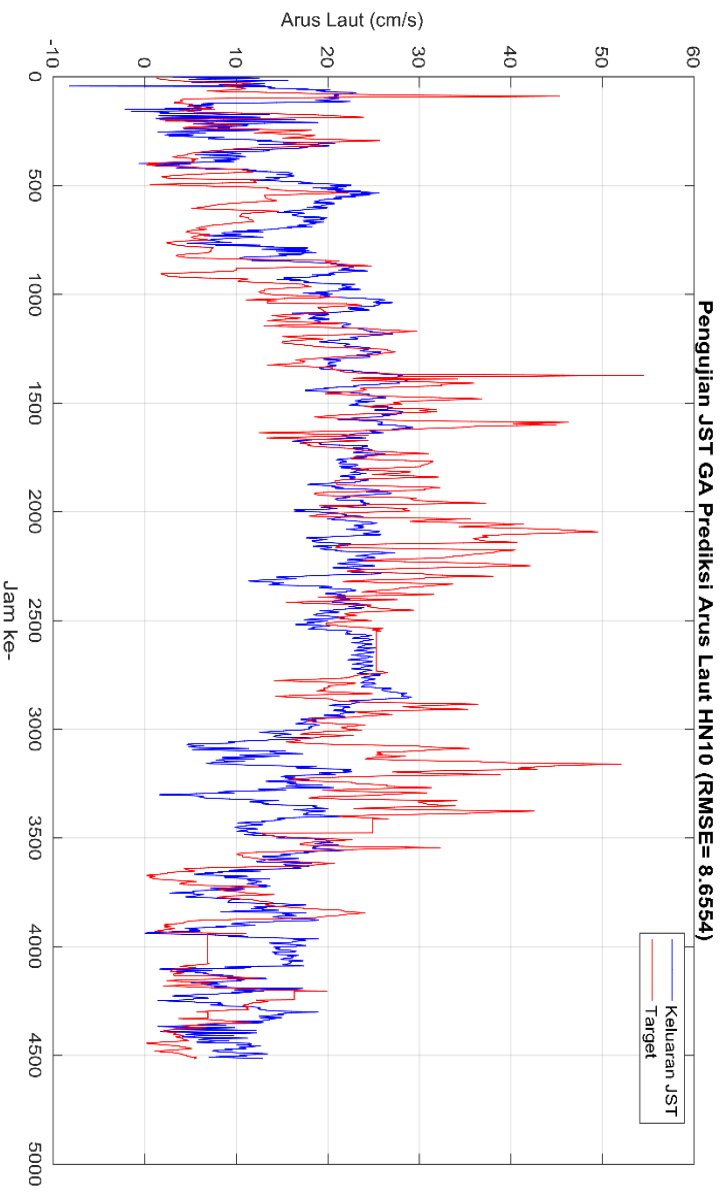
<b>Jumlah <i>Hidden Node</i></b>	<b>RMSE (cm/s) Pelatihan JST GA</b>	<b>RMSE (cm/s) Pengujian JST GA</b>
1	13,59	12,07
2	13,76	12,71
3	13,65	10,94
4	14,78	12,88
5	13,40	10,49
6	14,55	9,54
7	14,50	13,08
8	14,95	10,05
9	13,68	10,76
<b>10</b>	<b>15,24</b>	<b>8,66</b>
10 iterasi 60	13,67	10,00
10 iterasi 90	13,38	10,84
11	14,58	12,12
12	15,38	15,73
13	15,53	10,99
14	14,27	11,97
15	14,36	11,70



**Gambar 4. 9** Hasil Pelatihan JST-GA Prediktor Arus Laut Arsitektur 4-10-1

Grafik hasil pelatihan prediktor kecepatan arus laut menggunakan JST-GA pada prediktor kecepatan arus laut arsitektur 4-10-1 ditunjukkan pada Gambar 4.9. Grafik hasil pelatihan tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Nilai RMSE terbaik pada arsitektur 4-10-1 yaitu sebesar 15,24 cm/s. Nilai bobot hasil JST-GA pada *hidden node* 10 dapat dilihat pada Tabel 3.12 dengan bobot total yang dihasilkan sebanyak 61. Grafik hasil pelatihan JST-GA menghasilkan nilai RMSE lebih besar dari pengujian. Penambahan *hidden node* dan jumlah *epoch* belum tentu menghasilkan nilai RMSE lebih kecil. Semakin banyak jumlah *epoch* yang digunakan maka waktu yang dibutuhkan untuk memproses bobot yang akan dioptimalkan juga akan semakin lama. Penambahan 30 *epoch* menyebabkan waktu yang dibutuhkan 2 kali lebih lama yaitu 60 menit hingga 90 menit. Grafik hasil pelatihan JST-GA *hidden node* 1, 2, 3, ..., 15 ditunjukkan pada Lampiran C.

Grafik hasil pengujian prediktor kecepatan arus laut menggunakan JST-GA pada arsitektur 4-10-1 ditunjukkan pada Gambar 4.10. Grafik hasil pengujian tersebut menunjukkan bahwa hasil keluaran prediksi dapat mengikuti target dimana keluaran JST ditunjukkan oleh garis biru sedangkan target keluaran ditunjukkan oleh garis merah. Prediksi JST-GA ini masih belum bisa mengikuti target pada puncak-puncak tertentu seperti pada kecepatan arus laut 40 cm/s hingga 50 cm/s. Nilai RMSE terbaik pada arsitektur 4-10-1 dengan 30 *epoch* yaitu sebesar 8,66 cm/s. Proses pengujian JST-GA pada program MATLAB ini menggunakan bobot yang telah didapatkan pada proses pelatihan melalui optimasi GA, kemudian dilakukan pengujian terhadap data di luar data yang digunakan pada proses pelatihan untuk keakuratan dari prediktor yang telah dibuat. Grafik hasil pengujian JST-GA menghasilkan nilai RMSE lebih kecil dari pelatihan. Pemilihan arsitektur jaringan terbaik ini berdasarkan nilai RMSE pengujian yang terkecil. Penambahan *hidden node* dan jumlah *epoch* belum tentu menghasilkan nilai RMSE lebih kecil. Grafik hasil pengujian JST-GA *hidden node* 1, 2, 3, ....., 15 ditunjukkan pada Lampiran D.



**Gambar 4. 10** Hasil Pengujian JST-GA Prediktor Arus Laut Arsitektur 4-10-1

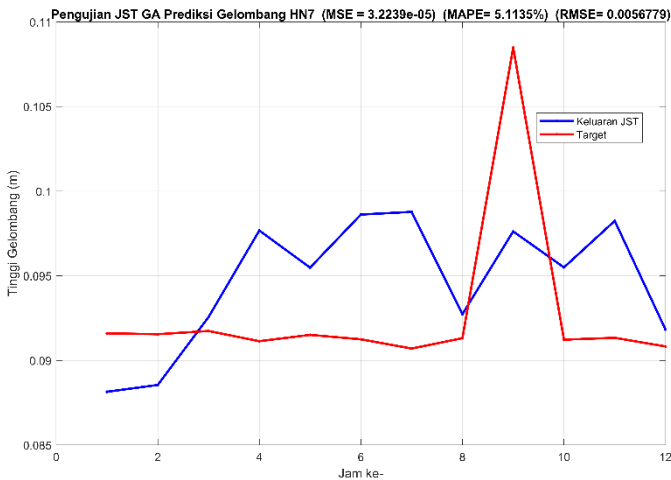
#### 4.6 Prediktor Ketinggian Gelombang dengan Data *Realtime*

Validasi prediktor dilakukan dengan pengambilan data *realtime* langsung dari sensor wahana apung Buoy. Data asli dari sensor yang ditunjukkan pada Tabel 3.1 kemudian dilakukan prediksi ketinggian gelombang dengan menggunakan JST-GA arsitektur terbaik yaitu 4-7-1. Hasil prediksi ketinggian gelombang memiliki nilai selisih yang tinggi dengan tinggi gelombang terukur dikarenakan pengaruh dari akurasi sensor arah angin yang rendah sehingga data yang terukur tidak sesuai dengan sebenarnya. Ketinggian gelombang hasil prediksi yang cenderung lebih rendah dari nilai terukur. Penyebab selisihnya nilai ketinggian terukur dan hasil prediksi disebabkan karena pengambilan data *realtime* dilakukan di pinggir pantai sehingga gelombang yang dihasilkan kecil berbeda dengan data gelombang yang digunakan proses pelatihan jaringan. Hasil prediksi ketinggian gelombang air laut menggunakan JST-GA dibandingkan dengan data *realtime* dari sensor ditunjukkan pada Tabel 4.13 di bawah ini.

**Tabel 4. 13** Data Hasil Pengujian *Realtime* Prediktor Gelombang

<b>Kecepatan Angin (m/s)</b>	<b>Arah Angin (derajat)</b>	<b>Tinggi Gelombang Hasil Terukur (m)</b>	<b>Tinggi Gelombang Hasil Prediksi (m)</b>
4,96	45,00	0,091597	0,088142
3,90	44,25	0,091544	0,088545
4,01	66,78	0,091734	0,092556
2,02	99,21	0,091131	0,097672
2,17	78,71	0,091506	0,095480
2,12	113,75	0,091244	0,098616
1,16	114,03	0,090697	0,098774
1,87	59,56	0,091309	0,092727
1,45	96,59	0,108488	0,097619
1,66	76,81	0,091222	0,095492
1,49	105,40	0,091331	0,098241
1,38	52,59	0,090822	0,091819

Grafik keluaran prediktor ketinggian gelombang menggunakan JST-GA berdasarkan data *realtime* ditunjukkan pada Gambar 4.11 dimana hasil keluaran prediksi gelombang menghasilkan keluaran dibawah nilai ukur dari sensor. Keluaran JST ditunjukkan dengan garis warna biru dan target keluaran sensor ditunjukkan dengan warna merah. Hasil prediksi ketinggian gelombang dikatakan baik dan akurat karena MAPE hasil pengujian yaitu 5,11% atau di bawah 10% serta pada proses pengujian data *realtime* di Pantai Kenjeran. Hasil keluaran prediksi ketinggian gelombang JST-GA dengan arsitektur 4-7-1 menghasilkan nilai *error* yang kecil yaitu 1% - 8% data ketinggian gelombang yang terukur. Prediktor yang telah dibuat dengan arsitektur 4-7-1 menggunakan data input berupa kecepatan angin dan arah angin yang berasal dari database *Thingspeak* yang mengambil data secara *realtime* dari sensor yang dipasang di wahana apung *Buoyweather*. Hasil prediksi yang telah diolah di PC user kemudian dikirimkan ke *database online* berupa *Thingspeak*. Data prediksi di *Thingspeak* kemudian ditampilkan pada aplikasi *android* yang dapat diakses secara umum.



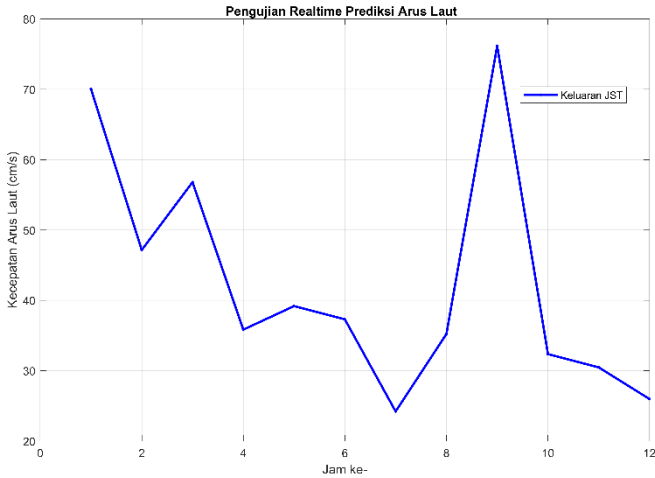
**Gambar 4. 11** Hasil Pengujian *Realtime* Prediktor Gelombang JST-GA 4-7-1

#### 4.7 Prediktor Kecepatan Arus Laut dengan Data *Realtime*

Pengujian *realtime* prediktor kecepatan arus laut dilakukan pengambilan data secara langsung dari *Buoyweather station type II* di Pantai Kenjeran. Data sensor yang digunakan untuk prediksi kecepatan arus laut yaitu nilai suhu permukaan air laut, kecepatan angin, dan ketinggian gelombang air laut. Hasil pengujian secara *realtime* prediktor kecepatan arus laut menggunakan metode JST-GA arsitektur 4-10-1 ditunjukkan pada Tabel 4.14. Pengambilan data *realtime* dari *Buoyweather* dilakukan selama 3 hari yaitu tanggal 11 Juni 2019, 15 Juni 2019, dan 18 Juni 2019. Data yang masuk dari pembacaan sensor seperti kecepatan angin, arah angin, suhu permukaan laut, dan tinggi gelombang yaitu setiap 20 detik sekali, sehingga untuk mendapatkan nilai data variabel setiap 1 jam dilakukan rata-rata dari data yang diperoleh. Program MATLAB untuk prediktor mengambil data variabel masukan dari *Thingspeak* yang membaca data secara *realtime* dari sensor yang terpasang pada wahana apung *Buoyweather*, kemudian data hasil prediksi ditampilkan pada GUI cuaca maritim yang telah dibuat di PC *user* dan dikirimkan kembali ke *Thingspeak*.

**Tabel 4. 14** Data Hasil Pengujian *Realtime* Prediktor Arus Laut

Kecepatan Angin (m/s)	Suhu Permukaan Laut (°C)	Ketinggian Gelombang (cm)	Arus Laut Hasil Prediksi (cm/s)
4,96	45,04	9,15	70,047
3,90	19,73	9,15	47,148
4,01	28,84	9,17	56,788
2,02	28,94	9,11	35,834
2,17	28,89	9,15	39,177
2,12	28,68	9,12	37,298
1,16	29,02	9,06	24,209
1,87	29,15	9,13	35,209
1,45	28,97	10,84	76,151
1,66	29,00	9,12	32,355
1,49	28,34	9,13	30,467
1,38	27,15	9,08	25,972



**Gambar 4. 12** Hasil Pengujian *Realtime* Prediktor Arus Laut JST-GA 4-10-1

Hasil prediksi kecepatan arus laut ditunjukkan pada Gambar 4.12 dengan nilai prediksi berkisar antara 30 cm/s hingga 70 cm/s. Hasil tersebut sesuai dengan rentang data arus laut yang digunakan pada saat pelatihan JST. Buoyweather ini tidak terdapat sensor kecepatan arus laut sehingga tidak ada data validasi untuk mengetahui nilai error antara pengukuran dan hasil prediksi yang telah dilakukan. Hasil prediktor hanya menampilkan hasil prediksi saja berdasarkan input yang ada. Data hasil prediksi ini kemudian ditampilkan pada *Graphic User Interface* (GUI) untuk prediktor cuaca maritime yang ada di PC user. Tampilan GUI prediktor ditunjukkan pada Gambar 4.13 di bawah ini. GUI ini terdapat beberapa prediktor cuaca maritim diantaranya prediktor kecepatan angin, arah angin, ketinggian gelombang, kecepatan arus laut, prediksi jumlah tangkapan ikan biji nangka, ikan kerapu, dan ikan kuniran. Prediktor yang ditampilkan yaitu 12 jam ke depan, 24 jam ke depan, 36 jam ke depan, dan 48 jam ke depan untuk prediksi cuaca. Prediktor yang ditampilkan 1 hari ke depan, 2 hari ke depan, dan 3 hari ke depan untuk prediksi tangkapan ikan.



BUOY WEATHER

**Saturday**  
29-06-2019

Day

Time

## Predictor of Maritime Weather

Sea Surface Temperature **33** Celcius

Pressure **100** Pa

Wave Height **0.09** m

Wind Speed **3** m/s

Degree **30**

Rainfall **0.01** mm

Relative Humidity **50** %

Air Temperature **27** Celcius

**1.3437 mm**  
Rainfall

0.09375 mm  
0.09375 mm  
0.09375 mm

**1.1286 m**  
Wave Height

1.1488 m  
1.1485 m  
0.6206 m  
0.043789 m

**6005.9565 Kg**  
Kerapu

6005.9565 Kg  
1302.5346 Kg  
1302.5346 Kg

**0.625 m/s**  
Wind Speed

0.625 m/s  
0.625 m/s  
0.625 m/s

**28.3798 cm/s**  
Current Speed

29.327 cm/s  
52.2918 cm/s  
23.8413 cm/s  
86.2402 cm/s

**34273.1518 Kg**  
Kuniran

34273.1518 Kg  
61852.9218 Kg  
61852.9218 Kg

**22065.0231 Kg**  
Biji Nangka

22065.0231 Kg  
72808.2068 Kg  
72808.2068 Kg

**1.3437 mm**  
Rainfall

0.09375 mm  
0.09375 mm  
0.09375 mm

**0.625 m/s**  
Wind Speed

0.625 m/s  
0.625 m/s  
0.625 m/s

**33.75 °**  
Wind Direction

33.75 °  
33.75 °  
33.75 °

Gambar 4. 13 Tampilan GUI Prediktor Cuaca

Tampilan GUI prediktor cuaca maritim di atas berisi hasil prediksi cuaca, gelombang, arus laut, dan jumlah tangkapan ikan. Data yang diperoleh dari hasil pengambilan data secara *realtime* dikirimkan secara *realtime* ke *Thingspeak* yang dapat diakses dengan koneksi internet. Data yang telah dikirim tersebut kemudian dipanggil oleh program integrasi prediktor di MATLAB. Hasil prediktor ditampilkan pada *Graphic User Interface* (GUI) yang ada di MATLAB serta diupload kembali ke *Thingspeak*, hasil prediksi yang ada di *Thingspeak* kemudian ditampilkan pada *android* yang bisa diakses oleh masyarakat umum.

## **BAB V PENUTUP**

### **4.1 Kesimpulan**

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Hasil prediksi ketinggian gelombang menggunakan JST-GA menghasilkan nilai RMSE yang lebih kecil daripada JST-LM. Arsitektur terbaik pada JST-LM adalah 4-10-1. Nilai RMSE pada pelatihan JST-LM sebesar 0,059061 m sedangkan pada pengujian sebesar 0,059060 m. Arsitektur terbaik pada JST-GA adalah 4-7-1. Nilai RMSE pada pelatihan JST-GA sebesar 0,075395 m sedangkan pada pengujian sebesar 0,057556 m.
2. Hasil prediksi kecepatan arus laut menggunakan JST-GA menghasilkan nilai RMSE yang lebih kecil daripada JST-LM. Arsitektur terbaik pada JST-LM adalah 4-4-1. Nilai RMSE pada pelatihan JST-LM sebesar 12,24 cm/s sedangkan pada pengujian sebesar 10,39 cm/s. Arsitektur terbaik pada JST-GA adalah 4-10-1. Nilai RMSE pada pelatihan JST-GA sebesar 15,24 cm/s sedangkan pada pengujian sebesar 8,66 cm/s.
3. Prediksi ketinggian gelombang secara *real time* dengan metode JST-GA menghasilkan data prediksi yaitu 0,088142 m; 0,088545 m; 0,092556 m; 0,097672 m; 0,095480 m; 0,098616 m, 0,098774 m, 0,092727 m; 0,097619 m; 0,095492 m; 0,098241 m; dan 0,091819 m.
4. Prediksi kecepatan arus laut secara *real time* dengan metode JST-GA menghasilkan data prediksi yaitu 70,047 cm/s; 47,148 cm/s; 56,788 cm/s; 35,834 cm/s; 39,177 cm/s; 37,298 cm/s; 24,209 cm/s; 35,209 cm/s; 76,151 cm/s; 32,355 cm/s; 30,467 cm/s; dan 25,972 cm/s.

### **4.2 Saran**

Beberapa saran dalam penelitian ini adalah sebagai berikut:

1. Penambahan variasi untuk parameter-parameter di dalam penggunaan metode JST-LM dan JST-GA agar mendapatkan hasil yang lebih optimal.

2. Pengumpulan data sekunder dilakukan jauh hari serta pengumpulan data dari beberapa tahun terakhir.
3. Pengukuran letak koordinat di lokasi yang akan dilakukan pengambilan data *realtime Buoyweather* sebelum meminta data dari BMKG II Perak.

## DAFTAR RUJUKAN

- Ainsworth T. (2017). *Significant wave height a closer look at wave forecasts retrieved from marine exchange of alaska*, <https://www.mxak.org/weather/pdfs/waves.pdf>.
- Aisjah, A. S. (2011). Maritime weather prediction using fuzzy logic in java sea. 205-208.
- Aliev, A., & Guirimov, B. G. (2014). *Type-2 fuzzy neural networks and their applications*. Springer International Publishing Switzerland.
- Fausset, L. (1996). *Fundamental of neural network: theory and application*,. Prentice Hall, SIngapore.
- Hakim, J. A. (2011). *Perancangan sistem prediktor maritim dengan metode logika fuzzy untuk meningkatkan jangkauan ramalan studi kasus pelayaran surabaya-banjarmasin*.
- Jain, P., Deo, M. C., & Rajendran, V. (2011). Real time wave forecasting using wind time history and numerical model. *Ocean Modelling* 36, 26-39.
- Kusumadewi, S., & Purnomo, H. (2005). *Penyelesaian masalah optimasi dengan teknik-teknik heuristik*. Yogyakarta: Graha Ilmu.
- Man, K. F., Tang, K. S., & Kwong, S. (1999). *Genetic algorithms: concepts and design*. Berlin: Springer.
- Raharja, M. A., & Astra, I. M. (2018). Prediksi ketinggian gelombang laut menggunakan metode backpropagation pada pantai lebih ginyar. *Jurnal Ilmiah ILMU KOMPUTER Universitas Udayana Vol. XI, No. 1, April 2018*, ISSN 1979 – 5661.
- Rizianiza, I., & Aisjah, A. S. (2015). Perancangan prediktor ketinggian gelombang di perairan sumatera-jawa berbasis artificial neural network. *Seminar Nasional Teknologi Informasi dan Komunikasi 2015 (SENTIKA 2015)*, ISSN: 2089-9815 Yogyakarta, 28 Maret 2015.
- Sampurna. (2011). *Perancangan sistem prediktor dengan menggunakan metode anfis untuk meningkatkan ramalan studi kasus pelayaran surabaya-banjarmasin*.

- Saputra. (2016, Desember 29). *Liputan 6*. Retrieved from Tenggelamnya Kapal Marina [www.liputan6.com](http://www.liputan6.com)
- Sharma, B., & Venugopalan, K. (2014). Comparison of neural network training functions for hematoma classification in brain ct images. *IOSR Journal of Computer Engineering*, 31-35.
- Sudarsono, R. (2017). Perancangan prediktor cuaca maritim menggunakan fuzzy tipe 2 sebagai pendukung keselamatan nelayan dengan user interface android.
- Triatmodjo, B. (1999). *Teknik Pantai*. Yogyakarta: Beta Offset.
- Ukhti, M. F. (2018). *Prediksi ketinggian gelombang laut menggunakan jaringan syaraf tiruan particle swarm optimization*. Surabaya: Teknik Fisika FTI ITS.
- Wellyantama, P. (2015). Prediksi ketinggian gelombang laut perairan laut jawa bagian barat sebelah utara jakarta dengan metode jaringan syaraf tiruan propagasi balik bmkg stasiun meteorologi maritim pontianak. *POSITRON, Vol. V, No. 1*, 30 - 35 ISSN: 2301-4970.
- Word Meteorological, O. (1998). *Guide to wave analysis and forecasting* (2 ed). Switzerland: Geneva.
- 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*, 1-13 <https://doi.org/10.1016/j.asoc.2016.07.053>.
- Yaqub, M., Eren, B., & Eyupoglu, V. (2016). Assessment of neural network training algorithms for the Prediction of Polymeric Inclusion Membranes Efficiency. *SAÜ Fen Bil Der*, 533-542.
- Yu, H., & Wilamowski, B. M. (2011). *Levenberg marquardt training industrial electronics handbook chapter 12 2nd Edition (pp. 12-1 to 12-15)*. CRS Press.
- Zhang, S., Wang, H., Liu, L., Du, C., & Lu, J. (2009). Optimization of neural network based on improved genetic algorithm. *International Conference on Computational Intelligence*

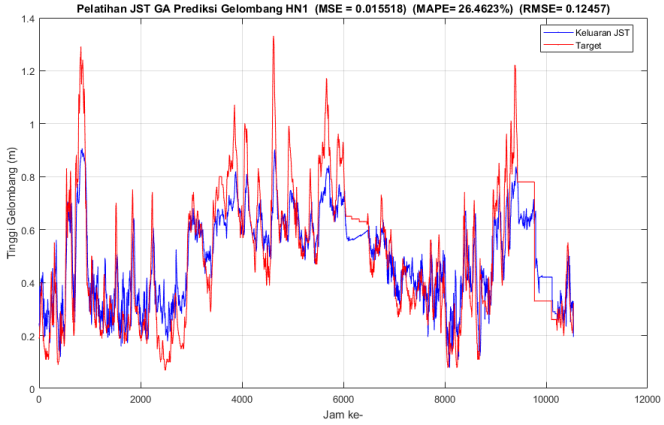
*and Software Engineering, (Cctot),* 203–207.  
<https://doi.org/10.1109/CISE.2009.5365287>.

*Halaman ini sengaja dikosongkan*

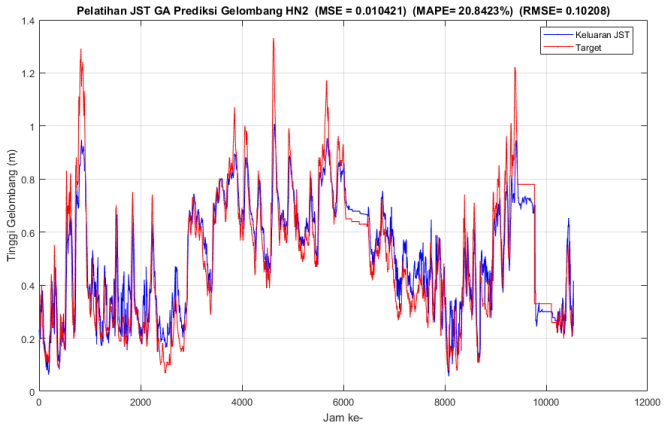


# LAMPIRAN

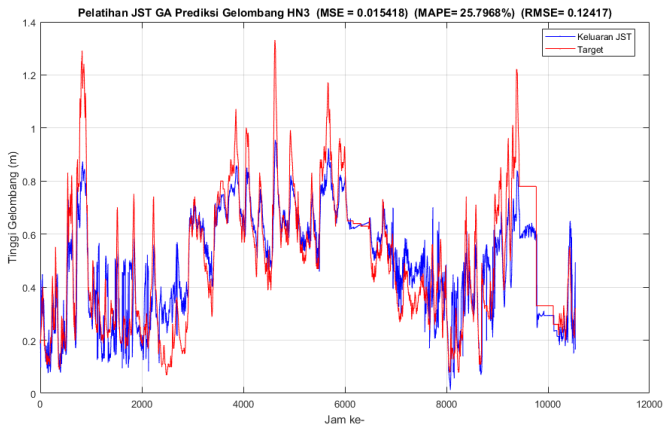
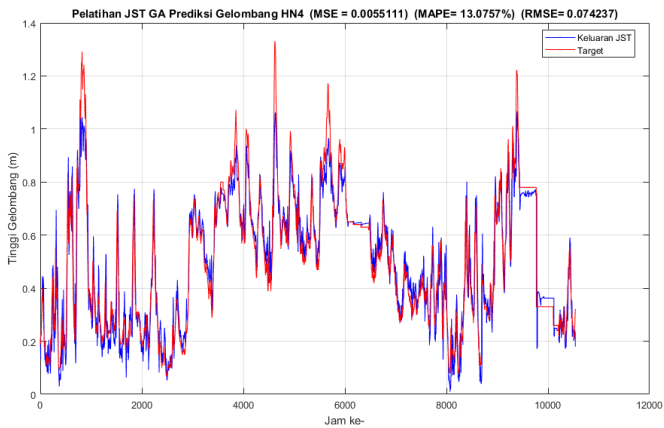
## Lampiran A. Pelatihan JST-GA Prediktor Gelombang

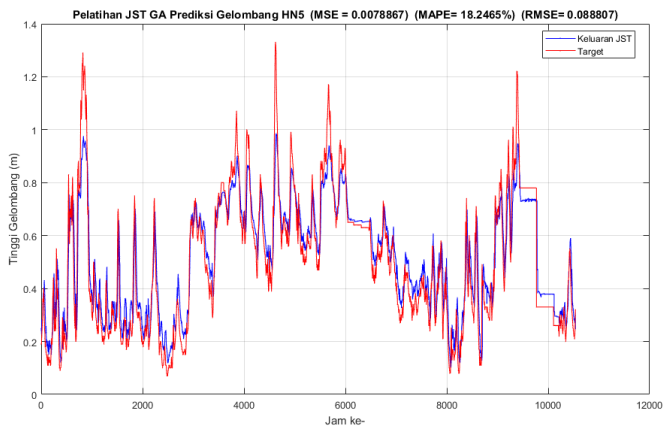
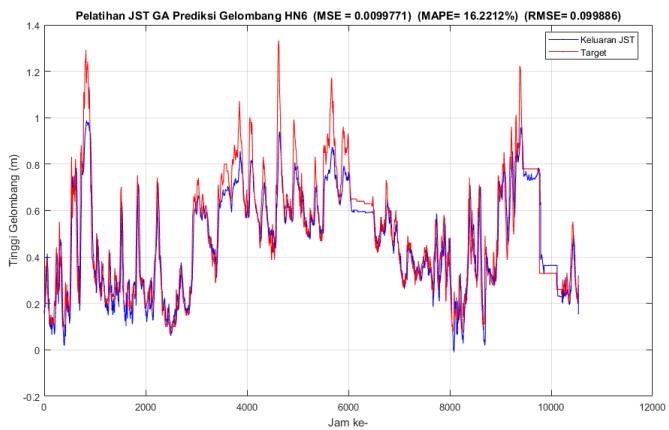


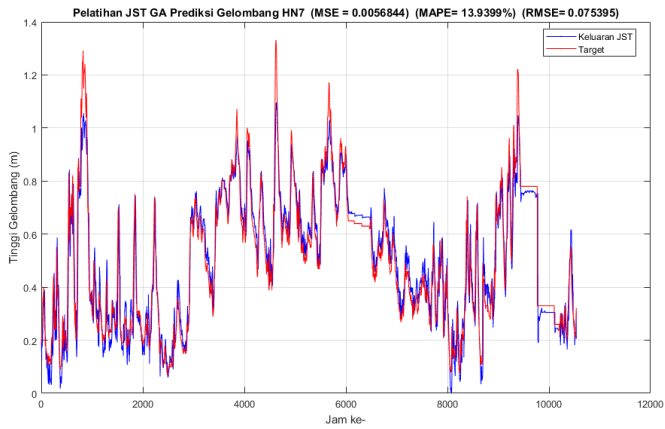
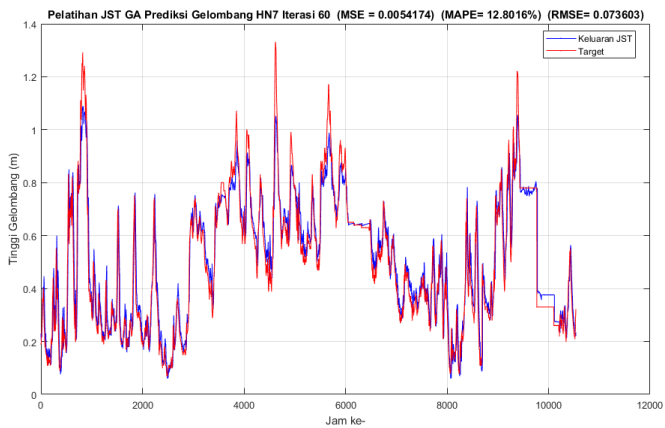
Gambar A- 1 Grafik Pelatihan JST-GA *Hidden Node 1*

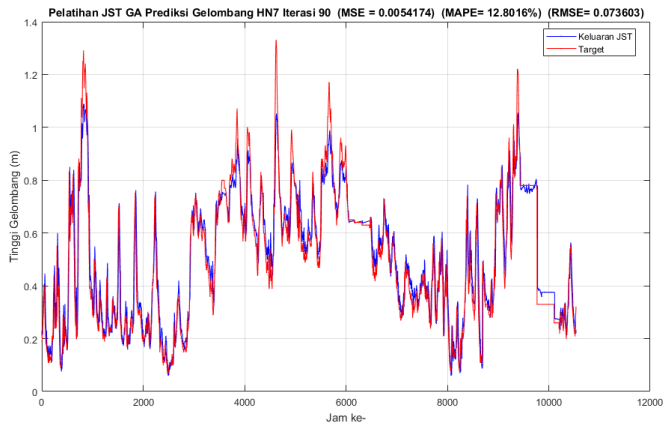
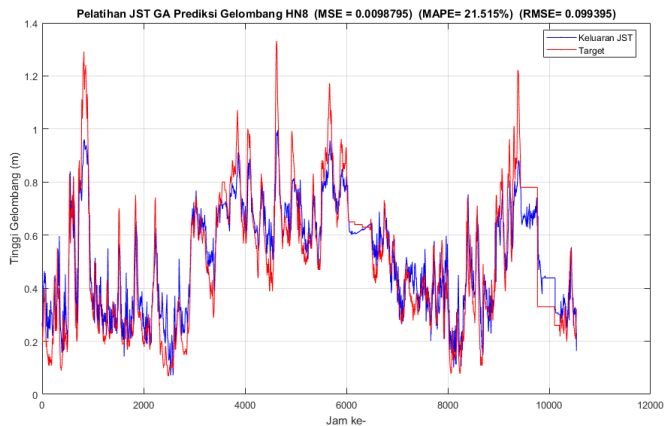


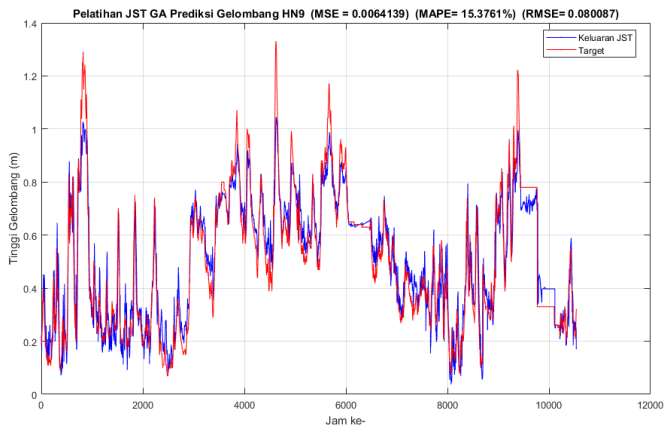
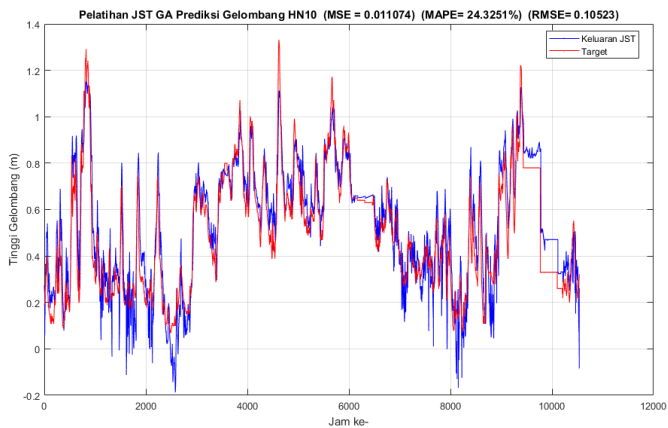
Gambar A- 2 Grafik Pelatihan JST-GA *Hidden Node 2*

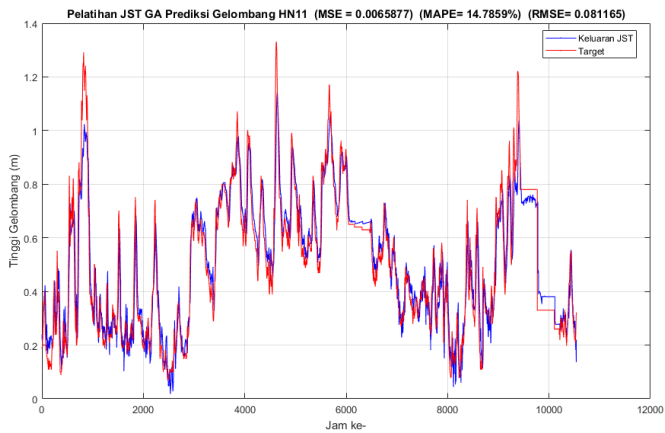
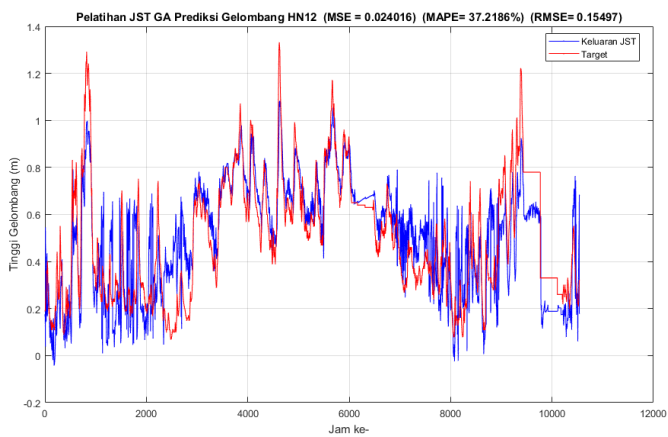
Gambar A- 3 Grafik Pelatihan JST-GA *Hidden Node 3*Gambar A- 4 Grafik Pelatihan JST-GA *Hidden Node 4*

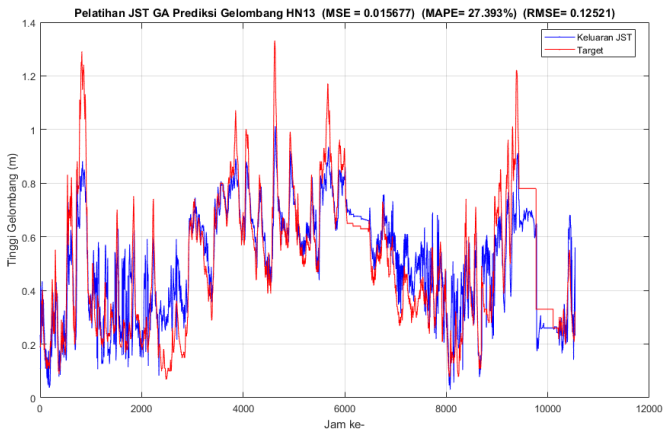
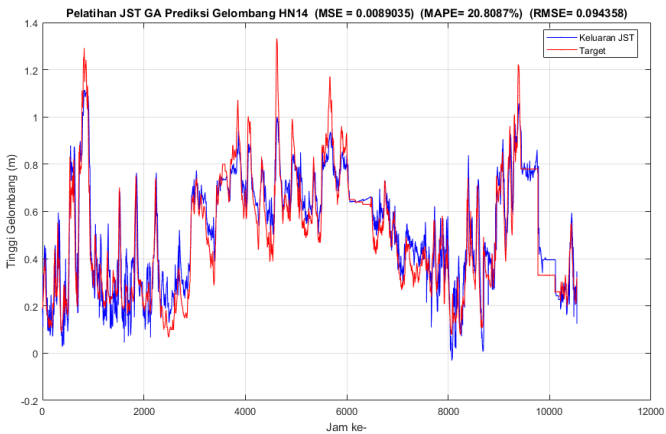
Gambar A- 5 Grafik Pelatihan JST-GA *Hidden Node 5*Gambar A- 6 Grafik Pelatihan JST-GA *Hidden Node 6*

Gambar A- 7 Grafik Pelatihan JST-GA *Hidden Node 7* Iterasi 30Gambar A- 8 Grafik Pelatihan JST-GA *Hidden Node 7* Iterasi 60

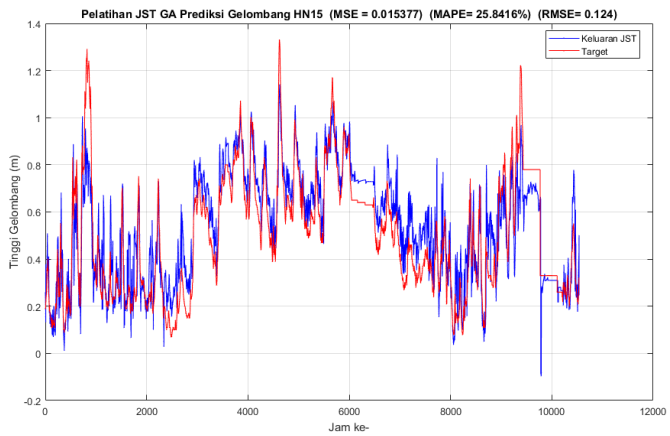
Gambar A- 9 Grafik Pelatihan JST-GA *Hidden Node 7* Iterasi 90Gambar A- 10 Grafik Pelatihan JST-GA *Hidden Node 8*

Gambar A- 11 Grafik Pelatihan JST-GA *Hidden Node 9*Gambar A- 12 Grafik Pelatihan JST-GA *Hidden Node 10*

Gambar A- 13 Grafik Pelatihan JST-GA *Hidden Node 11*Gambar A- 14 Grafik Pelatihan JST-GA *Hidden Node 12*

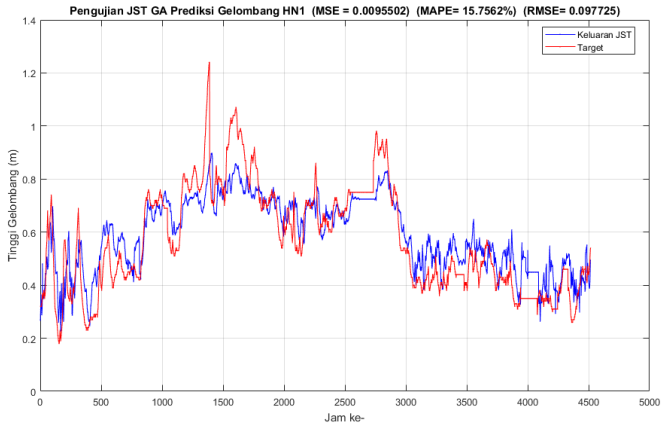
Gambar A- 15 Grafik Pelatihan JST-GA *Hidden Node 13*Gambar A- 16 Grafik Pelatihan JST-GA *Hidden Node 14*



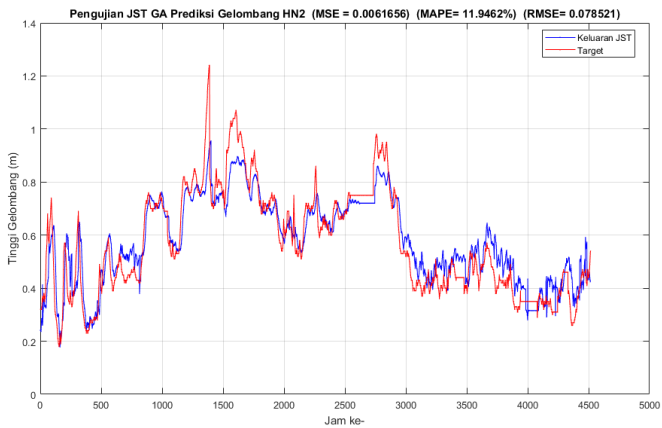


Gambar A- 17 Grafik Pelatihan JST-GA *Hidden Node 15*

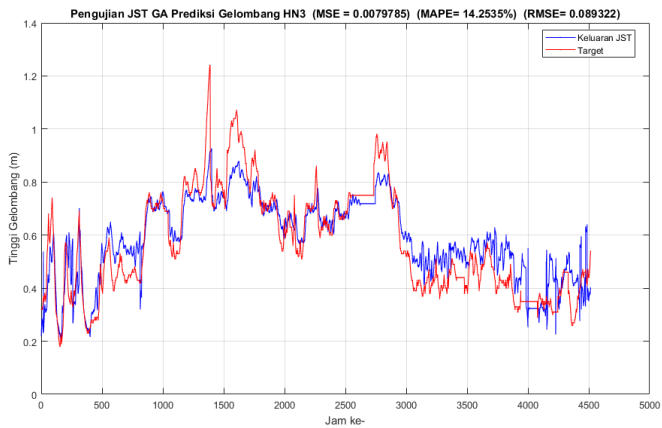
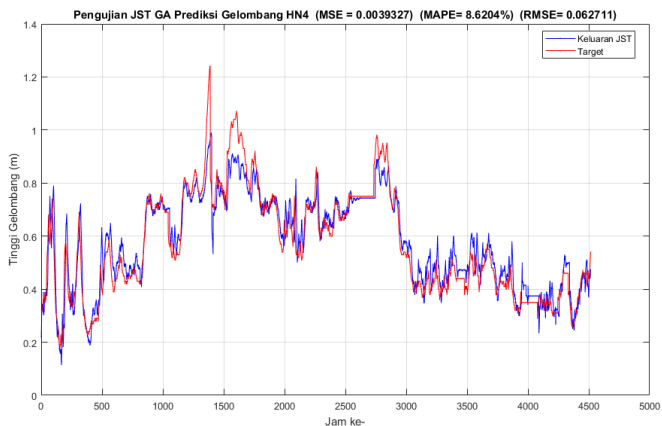
## Lampiran B. Pengujian JST-GA Prediktor Gelombang

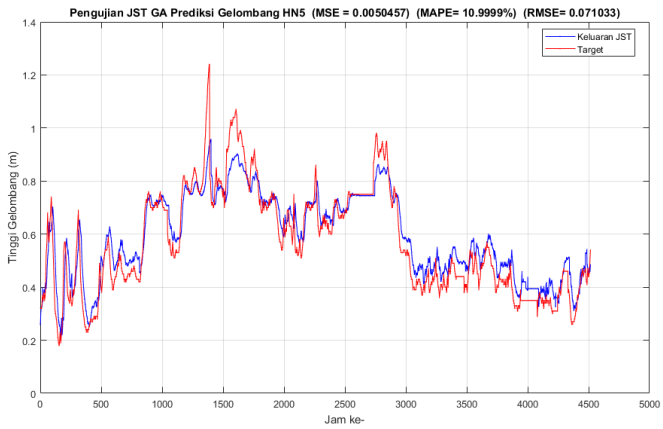
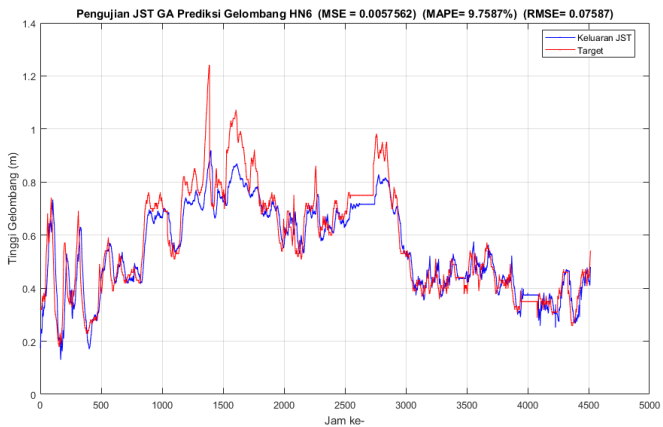


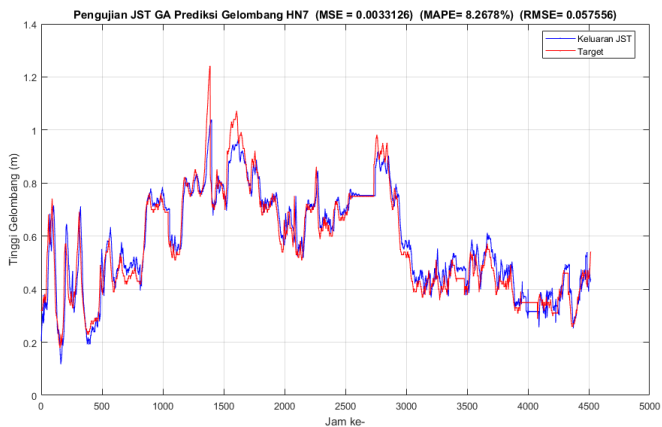
Gambar B- 1 Grafik Pengujian JST-GA *Hidden Node 1*



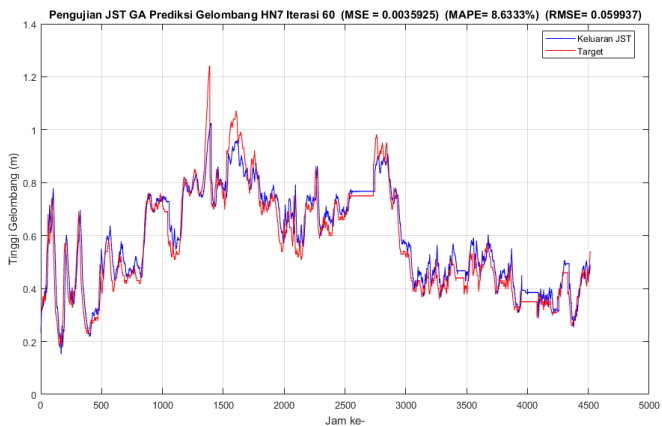
Gambar B- 2 Grafik Pengujian JST-GA *Hidden Node 2*

Gambar B- 3 Grafik Pengujian JST-GA *Hidden Node 3*Gambar B- 4 Grafik Pengujian JST-GA *Hidden Node 4*

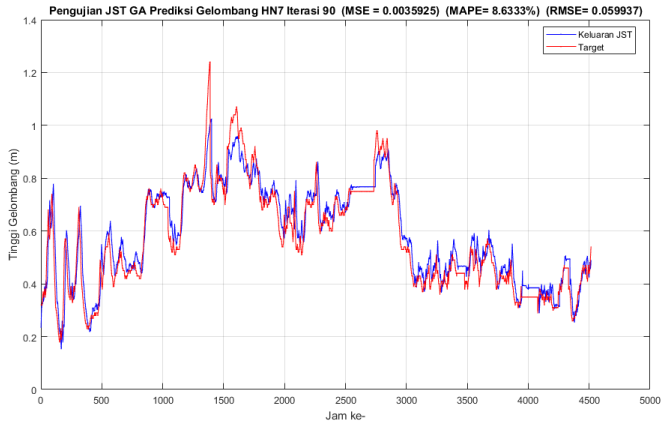
Gambar B- 5 Grafik Pengujian JST-GA *Hidden Node 5*Gambar B- 6 Grafik Pengujian JST-GA *Hidden Node 6*



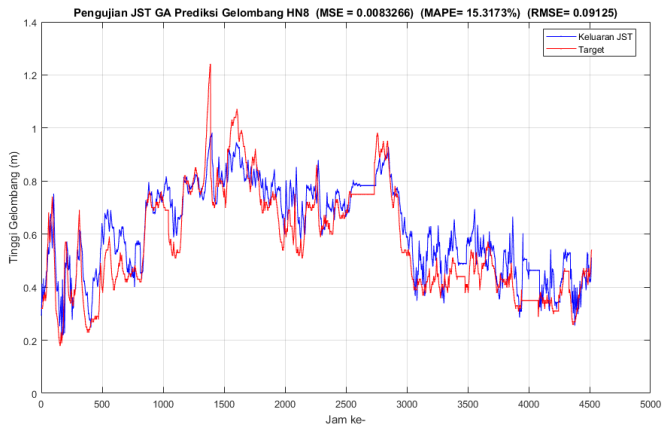
Gambar B- 7 Grafik Pengujian JST-GA *Hidden Node 7* Iterasi 30



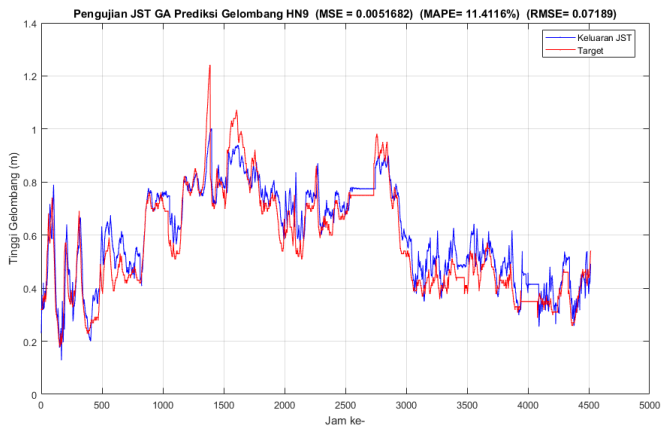
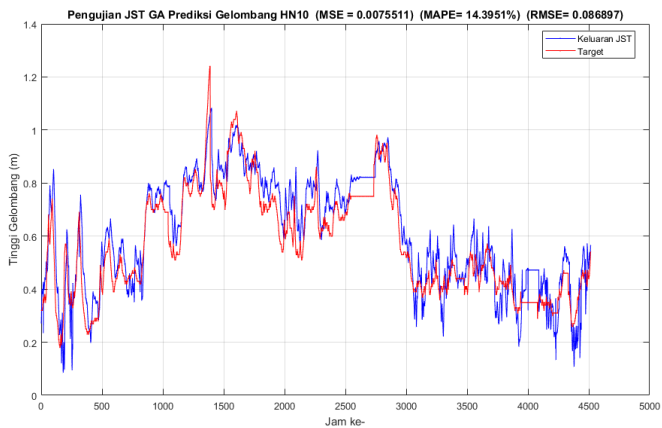
Gambar B- 8 Grafik Pengujian JST-GA *Hidden Node 7* Iterasi 60

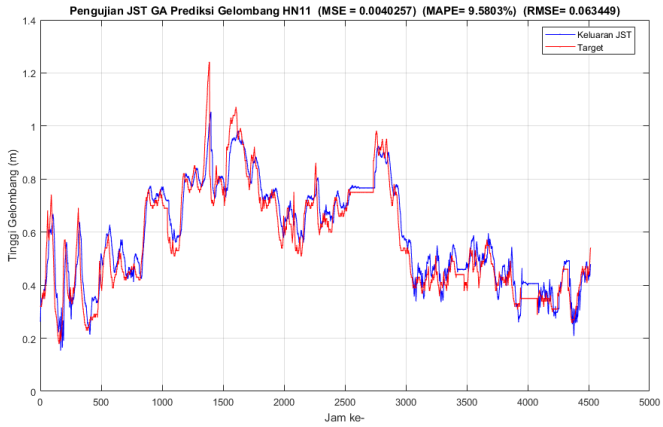
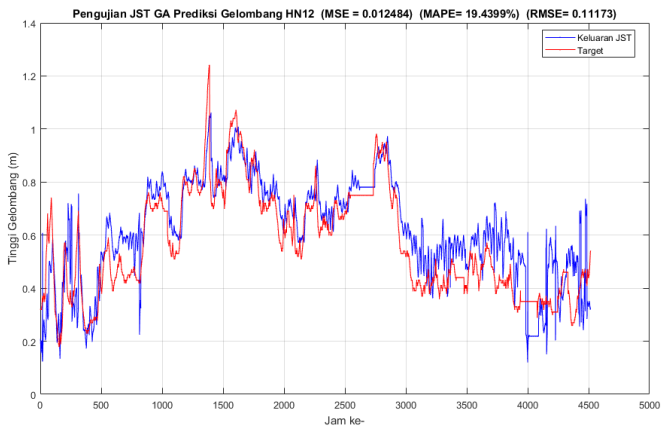


Gambar B- 9 Grafik Pengujian JST-GA Hidden Node 7 Iterasi 90

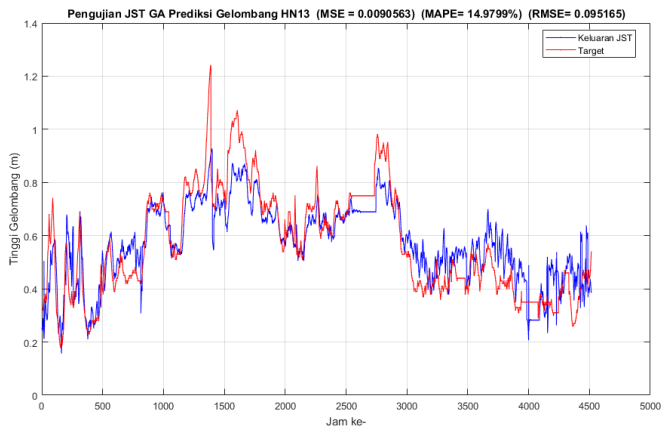


Gambar B- 10 Grafik Pengujian JST-GA Hidden Node 8

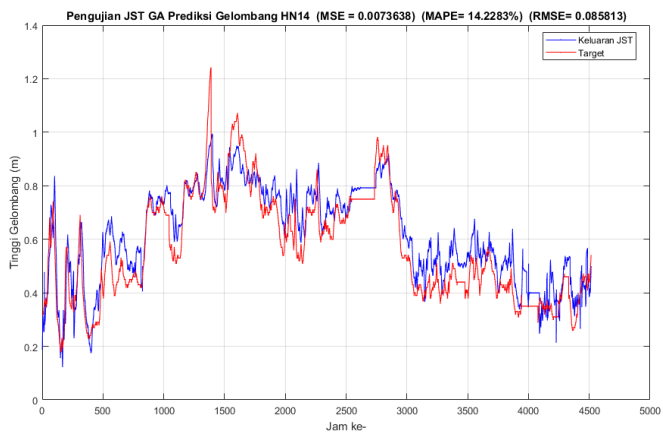
Gambar B- 11 Grafik Pengujian JST-GA *Hidden Node 9*Gambar B- 12 Grafik Pengujian JST-GA *Hidden Node 10*

Gambar B- 13 Grafik Pengujian JST-GA *Hidden Node 11*Gambar B- 14 Grafik Pengujian JST-GA *Hidden Node 12*

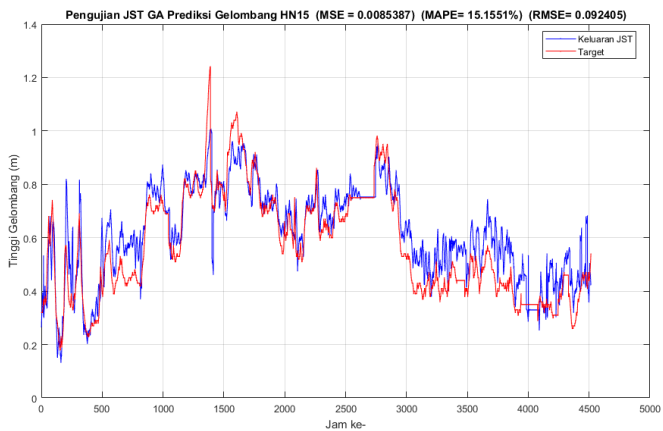




Gambar B- 15 Grafik Pengujian JST-GA *Hidden Node 13*

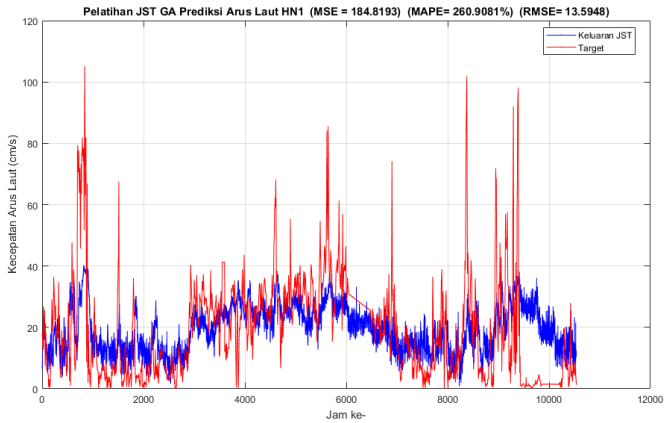


Gambar B- 16 Grafik Pengujian JST-GA *Hidden Node 14*

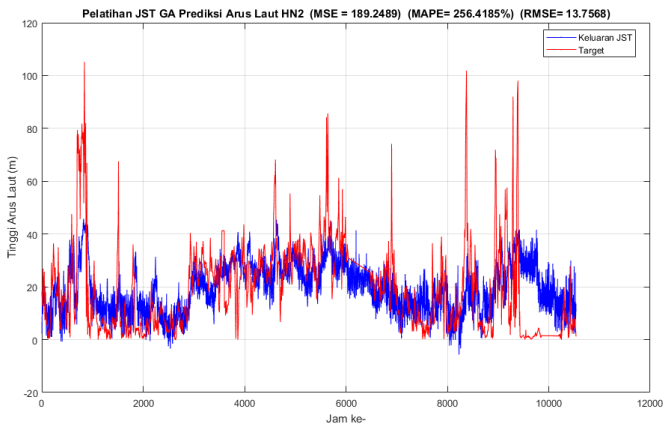


Gambar B- 17 Grafik Pengujian JST-GA *Hidden Node 15*

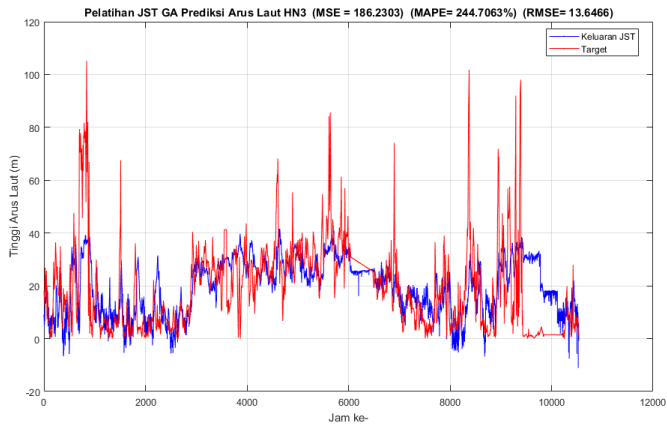
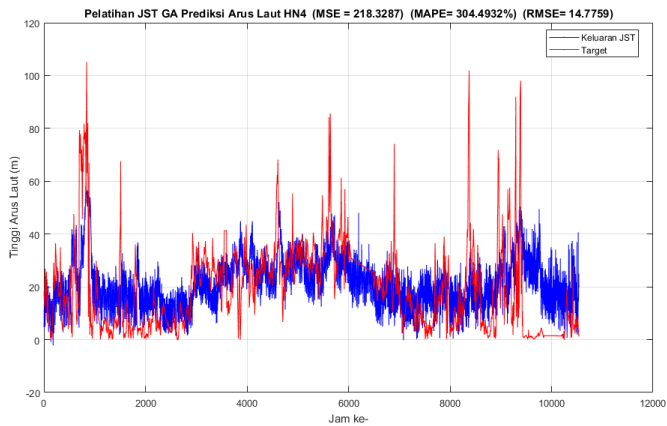
## Lampiran C. Pelatihan JST-GA Prediktor Arus Laut

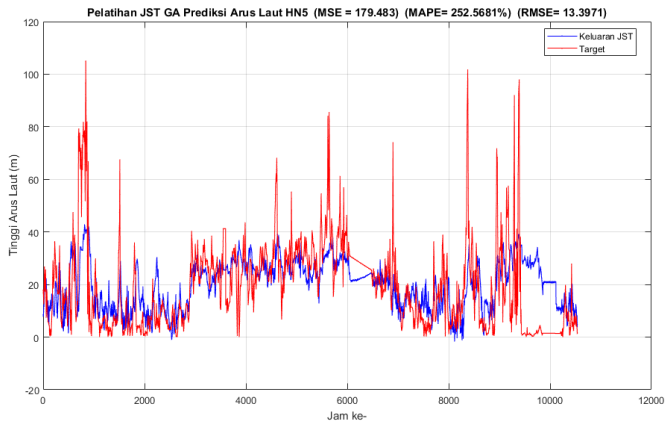
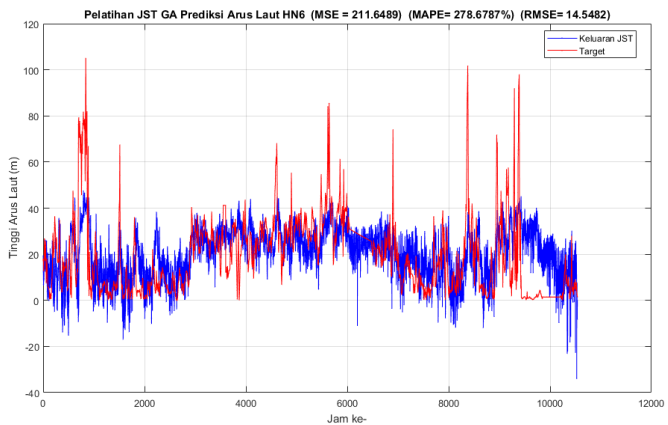


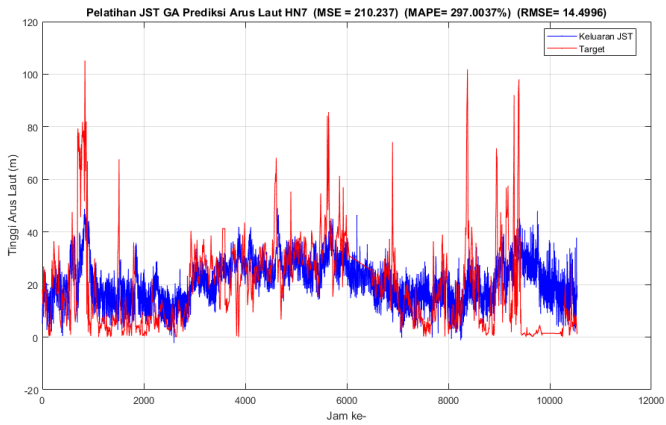
Gambar C- 1 Grafik Pelatihan JST-GA *Hidden Node 1*



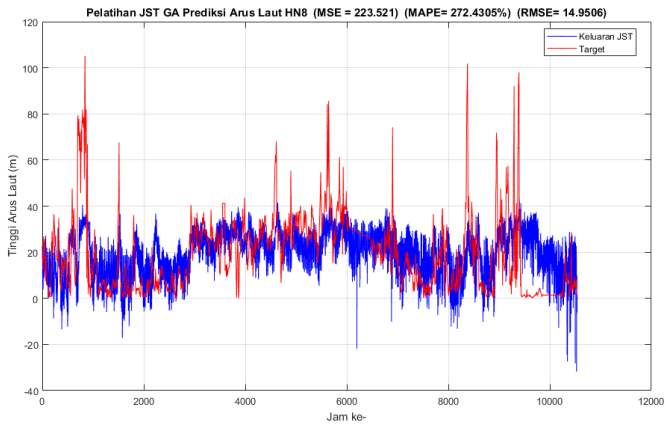
Gambar C- 2 Grafik Pelatihan JST-GA *Hidden Node 2*

Gambar C- 3 Grafik Pelatihan JST-GA *Hidden Node 3*Gambar C- 4 Grafik Pelatihan JST-GA *Hidden Node 4*

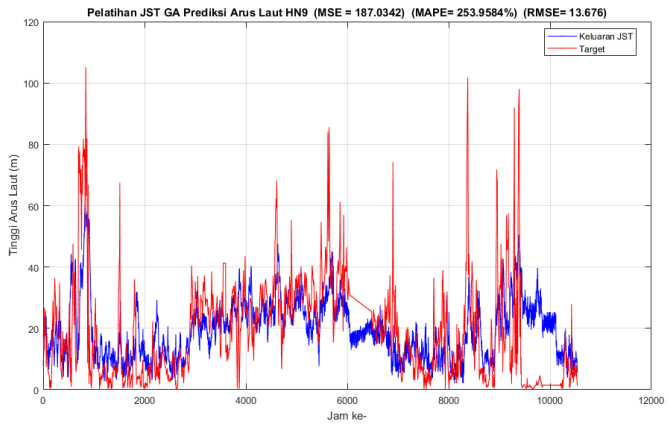
Gambar C- 5 Grafik Pelatihan JST-GA *Hidden Node 5*Gambar C- 6 Grafik Pelatihan JST-GA *Hidden Node 6*



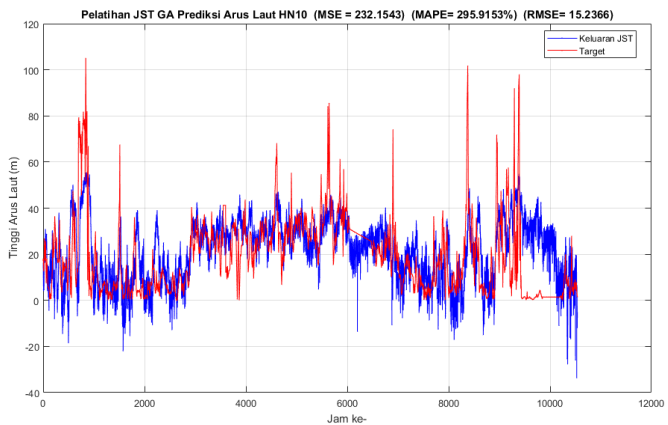
Gambar C- 7 Grafik Pelatihan JST-GA *Hidden Node 7*



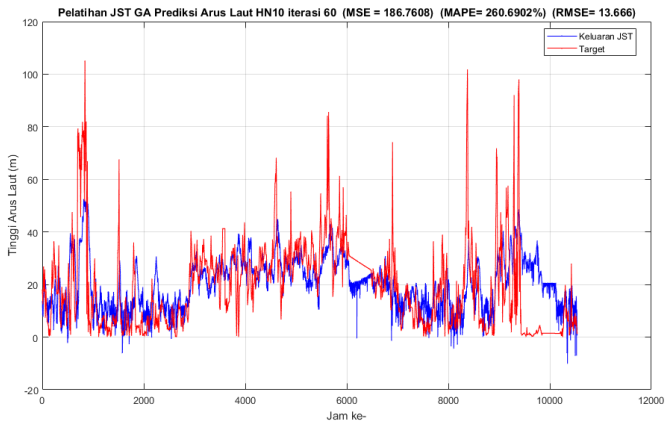
Gambar C- 8 Grafik Pelatihan JST-GA *Hidden Node 8*



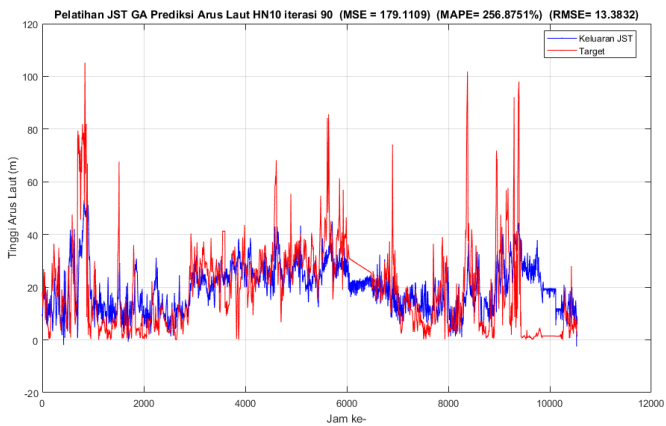
Gambar C- 9 Grafik Pelatihan JST-GA *Hidden Node 9*



Gambar C- 10 Grafik Pelatihan JST-GA *Hidden Node 10* Iterasi 30

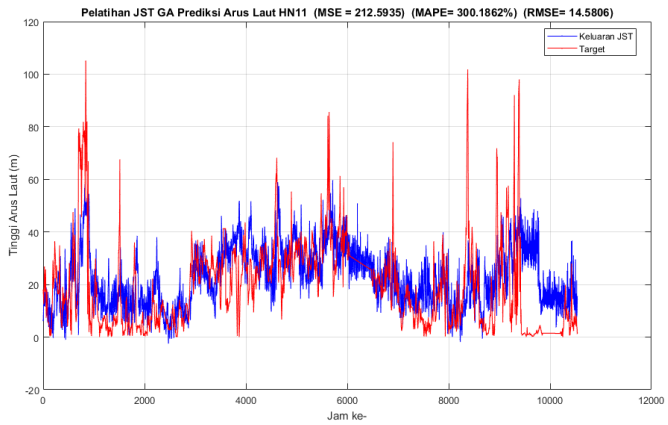


Gambar C- 11 Grafik Pelatihan JST-GA *Hidden Node* 10 iterasi 60

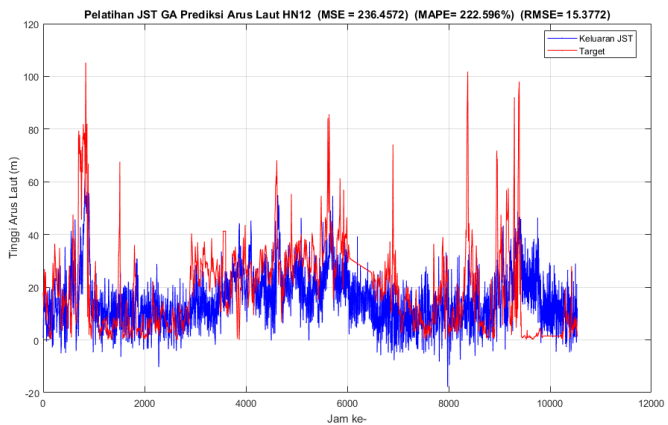


Gambar C- 12 Grafik Pelatihan JST-GA *Hidden Node* 10 iterasi 90

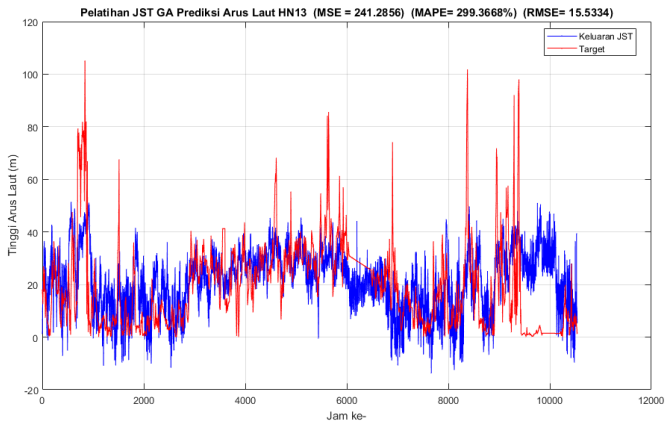
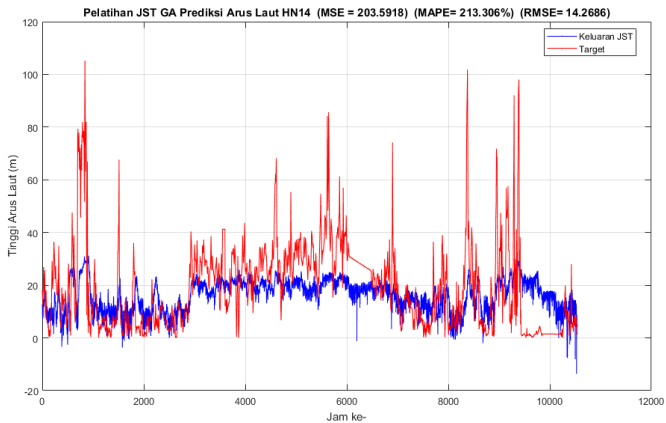


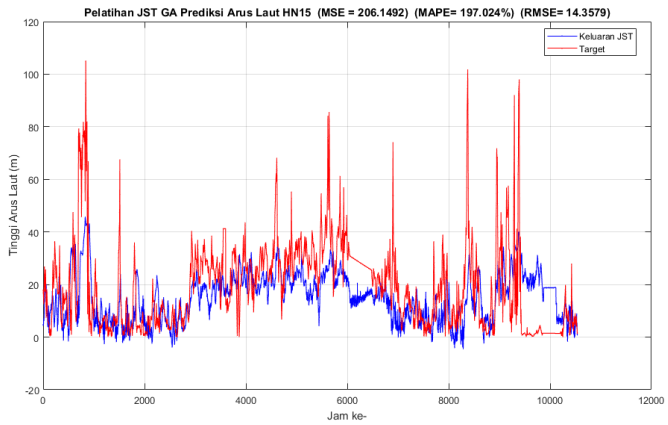


Gambar C- 13 Grafik Pelatihan JST-GA *Hidden Node 11*



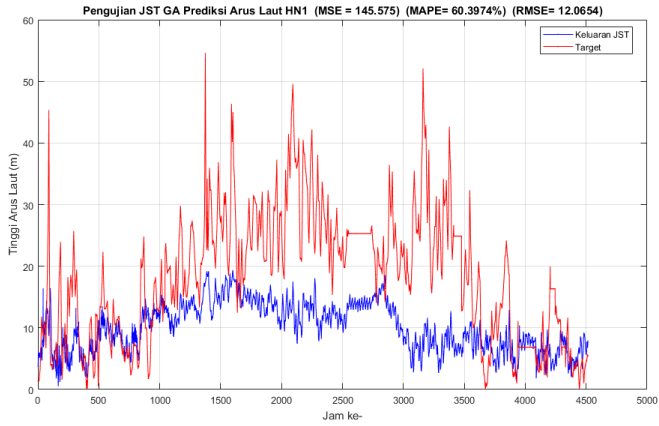
Gambar C- 14 Grafik Pelatihan JST-GA *Hidden Node 12*

Gambar C- 15 Grafik Pelatihan JST-GA *Hidden Node 14*Gambar C- 16 Grafik Pelatihan JST-GA *Hidden Node 14*

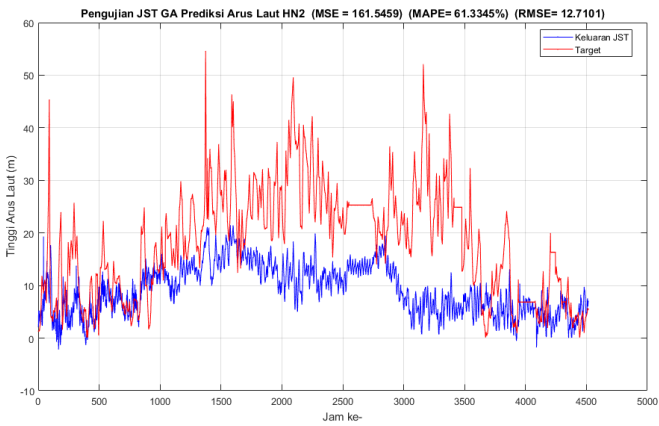


Gambar B- 18 Grafik Pelatihan JST-GA *Hidden Node 15*

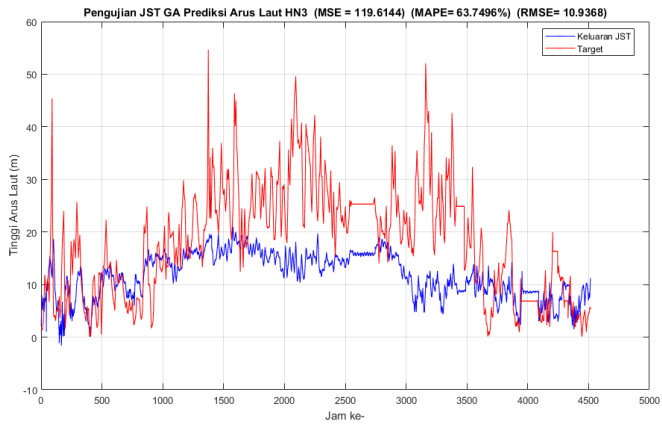
### Lampiran D. Pengujian JST-GA Prediktor Arus Laut



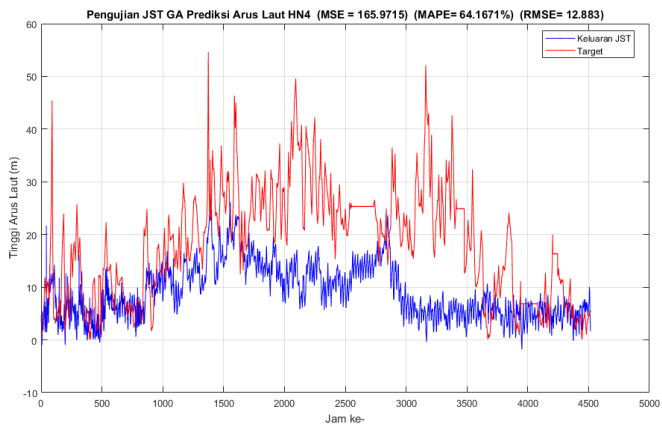
Gambar D- 1 Grafik Pengujian JST-GA *Hidden Node 1*



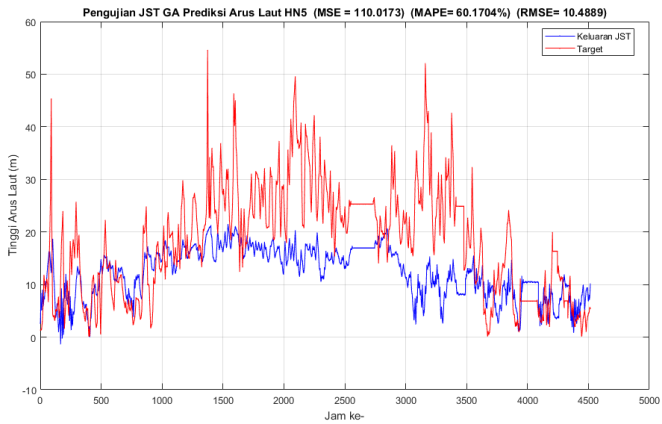
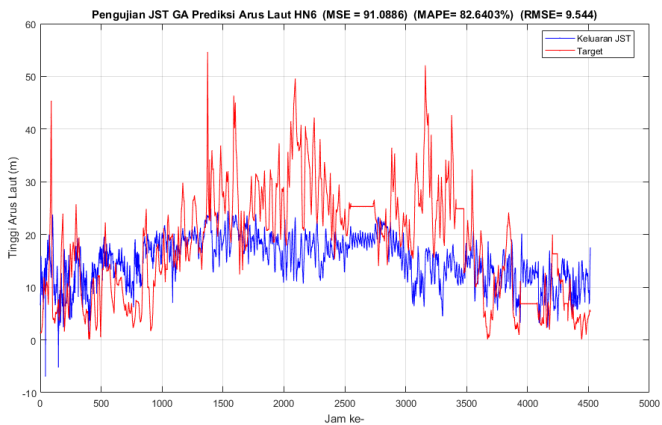
Gambar D- 2 Grafik Pengujian JST-GA *Hidden Node 2*

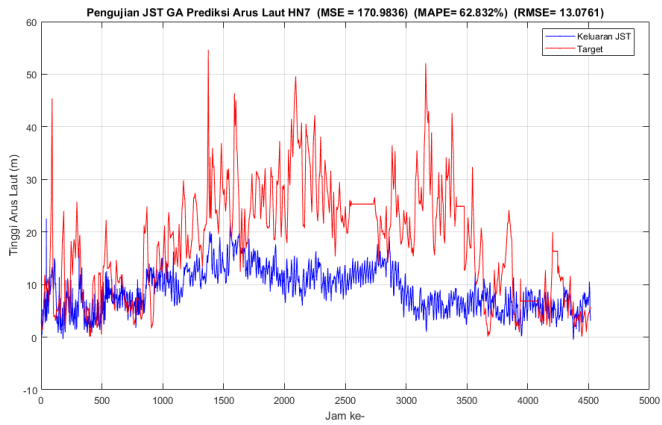


Gambar D- 3 Grafik Pengujian JST-GA *Hidden Node 3*

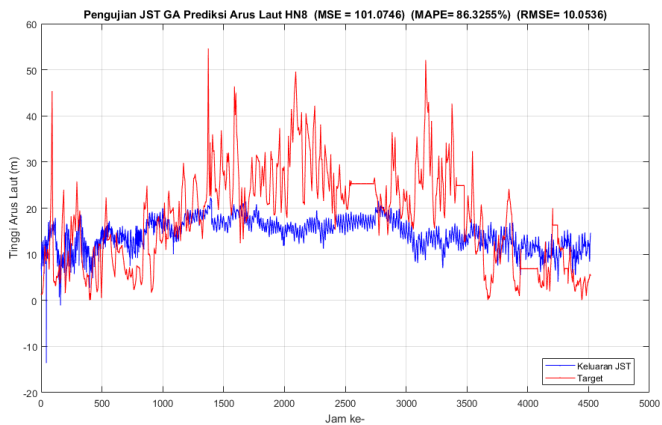


Gambar D- 4 Grafik Pengujian JST-GA *Hidden Node 4*

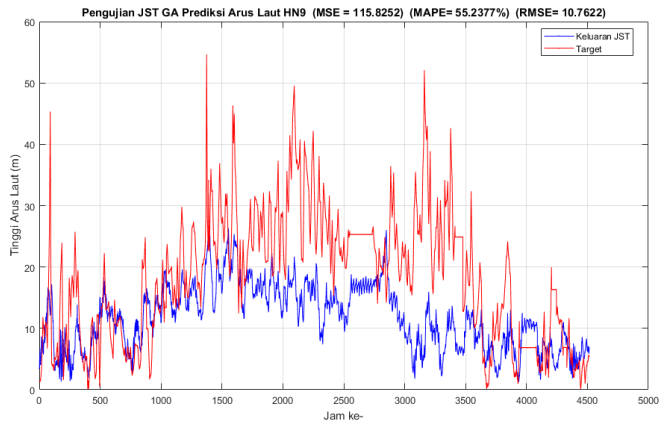
Gambar D- 5 Grafik Pengujian JST-GA *Hidden Node 5*Gambar D- 6 Grafik Pengujian JST-GA *Hidden Node 6*



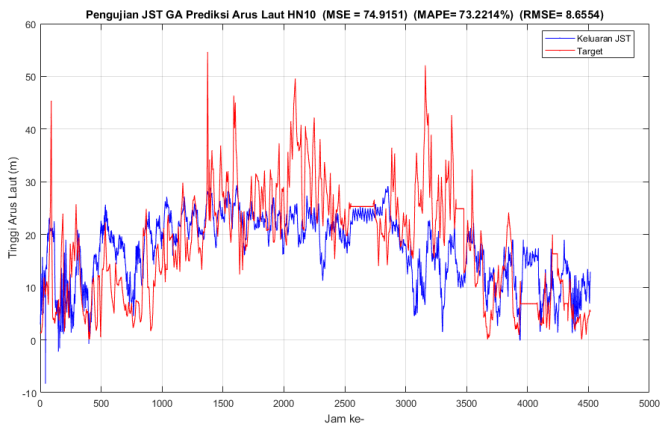
Gambar D- 7 Grafik Pengujian JST-GA *Hidden Node 7*



Gambar D- 8 Grafik Pengujian JST-GA *Hidden Node 8*

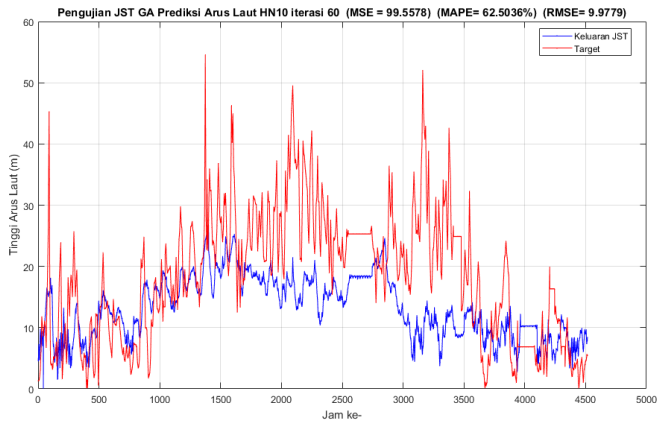


Gambar D- 9 Grafik Pengujian JST-GA *Hidden Node 9*

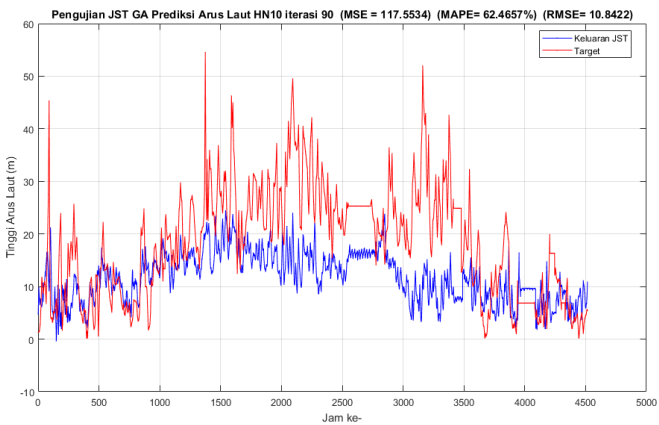


Gambar D- 10 Grafik Pengujian JST-GA *Hidden Node 10* Iterasi

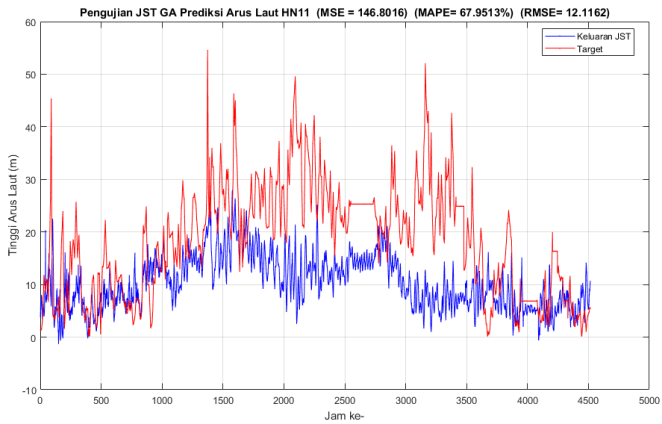
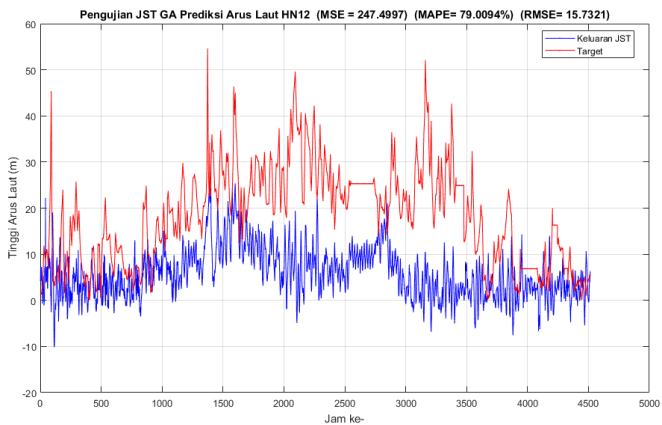


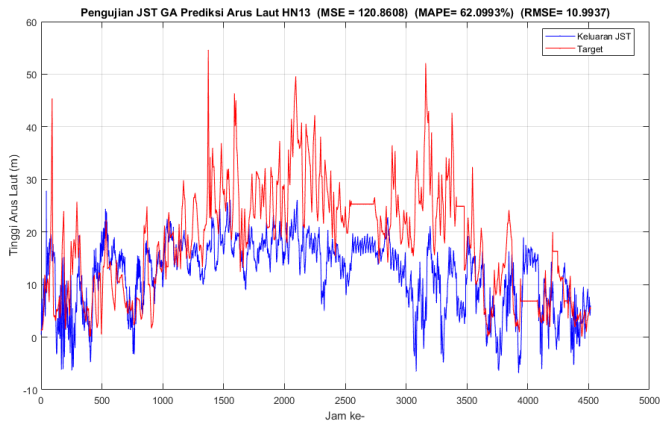


Gambar D- 11 Grafik Pengujian JST-GA *Hidden Node* 10 iterasi 60

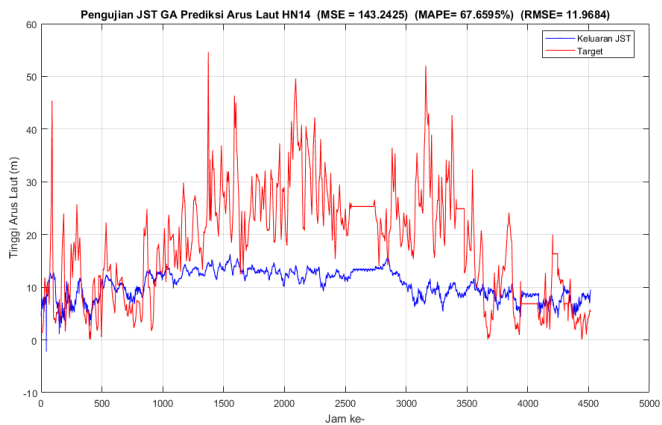


Gambar D- 12 Grafik Pengujian JST-GA *Hidden Node* 10 iterasi 90

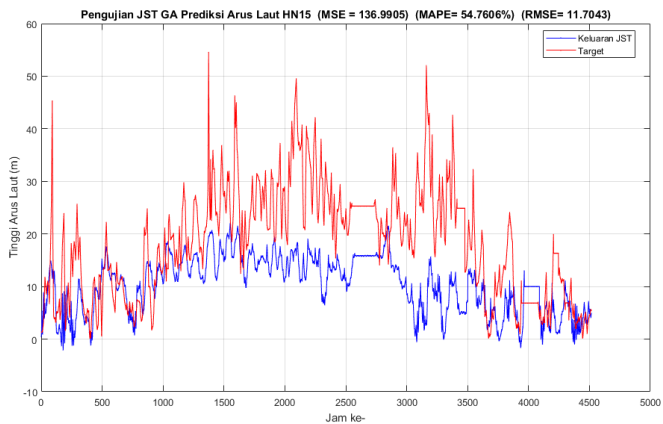
Gambar D- 13 Grafik Pengujian JST-GA *Hidden Node 11*Gambar D- 14 Grafik Pengujian JST-GA *Hidden Node 12*



Gambar D- 15 Grafik Pengujian JST-GA *Hidden Node 13*



Gambar D- 16 Grafik Pengujian JST-GA *Hidden Node 14*



Gambar D- 17 Grafik Pengujian JST-GA Hidden Node 15

## Lampiran E. Kode Program JST-LM

### E.1 Pelatihan JST LM

```

clc;
clear;
close all;
warning off;

% proses pembacaan data latih dari excel dengan
data normalisasi
filename = 'DATA GELOMBANG.xlsx';
sheet = 1;
xlRange = 'G2:K10539';
Data = xlsread(filename, sheet, xlRange);
input_latih = Data(:,1:4)'; %akses kolom 1-3
target_latih = Data(:,5)'; %akses kolom 4
[m,n] = size(input_latih);

%max dan min data ketinggian gelombang (target)
max_data = 1.33;
min_data = 0.07;

%pembuatan jaringan
net=newff(minmax(input_latih),[10,1],{'logsig','tansig'},'trainlm');

%parameter pelatihan
net.performFcn = 'mse';
net.trainParam.goal = 1e-5;
net.trainParam.show = 25;
net.trainParam.epochs = 100;
net.trainParam.lr = 0.01;
net.trainParam.min_grad = 10e-20;

%proses training
[net_output,tr,Y,E] =
train(net,input_latih,target_latih);

```

```

%hasil pelatihan102
bobot_hidden1 = net_output.IW{1,1};
bobot_keluaran = net_output.LW{2,1};
bias_hidden1 = net_output.b{1};
bias_keluaran = net_output.b{2};
jumlah_epoch = tr.num_epochs
nilai_output = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2) %error data
normalisasi

%menyimpan jaringan
save net.mat net_output

%hasil prediksi
hasil_latih = sim(net_output,input_latih);
hasil_latih_asli = (((hasil_latih -
0.1)*(max_data-min_data))/0.8)+min_data;

%nilai target asli
filename = 'DATA GELOMBANG.xlsx';
sheet = 1;
xlRange = 'E2:E10539';
target_latih_asli = xlsread(filename, sheet,
xlRange);
Y1 = hasil_latih_asli;
T1 = transpose(target_latih_asli);
E1 = abs(Y1-T1); %error dengan data asli
(denormalisasi)
error_MSE1 = (1/n)*sum(E1.^2) %MSE data data asli
(denormalisasi)
RMSE = sqrt(error_MSE1)
MAPE = ((sum(abs(E1./T1)))*100)/n;

%gambar plot regresi
figure,
plotregression(T1,hasil_latih_asli,'Regression');

```

```

%gambar performansi jaringan103
figure,
plotperform(tr); %plot hasil vs target
figure, plot(hasil_latih_asli,'b-');
hold on;
plot (target_latih_asli,'r-');
hold off;
grid on;
title(strcat(['Pelatihan JST LM Prediksi
Gelombang HN3 baru', ' (MSE = ',...
num2str(error_MSE1), ')', ' (MAPE= ',....
num2str(MAPE), '%)', ' (RMSE= ',...
num2str(RMSE), ')']));
xlabel('Jam ke-');
ylabel('Ketinggian Gelombang (m)');
legend('Keluaran JST', 'Target');

%menyimpan data pelatihan
save('training_351.mat')

```

## E.2 Pengujian JST LM

```

clc;
clear;
close all;
warning off;

%load jaringan yang sudah dibuat pada proses
pelatihan
load net.mat

%proses membaca data uji dari excel yang sudah
dinormalisasi
filename = 'DATA GELOMBANG.xlsx';
sheet = 2;
xlRange = 'G2:K4516';
Data = xlsread(filename, sheet, xlRange);
input_uji = Data(:,1:4)'; %akses kolom 1-3
target_uji = Data(:,5)'; %akses kolom 4
[m,n] = size(input_uji);

%Hasil prediksi
hasil_uji=sim(net_output,input_uji);
nilai_error = hasil_uji-target_uji;

%max dan min data target
max_data = 1.24;
min_data = 0.18;
hasil_uji_asli =(((hasil_uji - 0.1)*(max_data-
min_data))/0.8)+min_data;

%performansi hasil prediksi
error_MSE = (1/n)*sum(nilai_error.^2); %MSE data
normalisasi

%pembacaan data target asli
filename = 'DATA GELOMBANG.xlsx';
sheet = 2;

```



```

xlRange = 'E2:E4516';
target_uji_asli =
xlsread(filename, sheet, xlRange);
Y1 = hasil_uji_asli;
T1 = transpose(target_uji_asli);
E1 = abs(Y1-T1); %error dengan data asli
error_MSE1 = (1/n)*sum(E1.^2) %MSE data asli
(denormalisasi)
RMSE = sqrt(error_MSE1)
MAPE = ((sum(abs(E1./T1)))*100)/n;

%gambar keluaran jaringan vs target
figure,
plot(hasil_uji_asli, 'b-');
hold on;
plot (target_uji_asli, 'r-');
hold off;
grid on;
title(strcat(['Pengujian JST LM Prediksi
Gelombang HN3', ' (MSE = ', ...
num2str(error_MSE1), ')', ' (MAPE= ', ...
num2str(MAPE), '%)', ' (RMSE= ', ...
num2str(RMSE), ')']));
xlabel('Jam ke-');
ylabel('Ketinggian Gelombang (m)');
legend('Output JST', 'Target');

```

## Lampiran F. Kode Program JST-GA

### F.1 Fungsi Objektif

```
function [NilaiFitness, output]=fob7GABaru(x,variab
elinput,validasi)
variabelinput = xlsread('DATA
GELOMBANG.xlsx', 'Pelatihan', 'G2:J10539');
%pembacaan validasi = xlsread('DATA
GELOMBANG.xlsx', 'Pelatihan', 'E2:E10539');

data = 10538;          %Jumlah Data Latih

% 1 individu = seluruh bobot
t1=x(1);              %input 1 ke hidden node 1
t2=x(2);              %input 2 ke hidden node 1
t3=x(3);              %input 3 ke hidden node 1
t4=x(4);              %input 4 ke hidden node 1
t5=x(5);              %bias ke hidden node 1

t6=x(6);              %input 1 ke hidden node 2
t7=x(7);              %input 2 ke hidden node 2
t8=x(8);              %input 3 ke hidden node 2
t9=x(9);              %input 4 ke hidden node 2
t10=x(10);           %bias ke hidden node 2

t11=x(11);            %input 1 ke hidden node 3
t12=x(12);            %input 2 ke hidden node 3
t13=x(13);            %input 3 ke hidden node 3
t14=x(14);            %input 4 ke hidden node 3
t15=x(15);            %bias ke hidden node 3

t16=x(16);            %input 1 ke hidden node 4
t17=x(17);            %input 2 ke hidden node 4
t18=x(18);            %input 3 ke hidden node 4
t19=x(19);            %input 4 ke hidden node 4
t20=x(20);            %bias ke hidden node 4
```

```

t21=x(21);      %input 1 ke hidden node 5
t22=x(22);      %input 2 ke hidden node 5
t23=x(23);      %input 3 ke hidden node 5
t24=x(24);      %input 4 ke hidden node 5
t25=x(25);      %bias ke hidden node 5

```

```

t26=x(26);      %input 1 ke hidden node 6
t27=x(27);      %input 2 ke hidden node 6
t28=x(28);      %input 3 ke hidden node 6
t29=x(29);      %input 4 ke hidden node 6
t30=x(30);      %bias ke hidden node 6

```

```

t31=x(31);      %input 1 ke hidden node 7
t32=x(32);      %input 2 ke hidden node 7
t33=x(33);      %input 3 ke hidden node 7
t34=x(34);      %input 4 ke hidden node 7
t35=x(35);      %bias ke hidden node 7

```

```

t36=x(36);      %node 1 ke output
t37=x(37);      %node 2 ke output
t38=x(38);      %node 3 ke output
t39=x(39);      %node 4 ke output
t40=x(40);      %node 5 ke output
t41=x(41);      %node 6 ke output
t42=x(42);      %node 7 ke output
t43=x(43);      %bias ke output

```

%Input ke hidden

```

a1 = variabelinput(1:data,1)*t1;
b1 = variabelinput(1:data,2)*t2;
c1 = variabelinput(1:data,3)*t3;
d1 = variabelinput(1:data,4)*t4;
e1 = ones(data,1)*t5;
hn1 = [a1 b1 c1 d1 e1];

```

```

a2 = variabelinput(1:data,1)*t6;
b2 = variabelinput(1:data,2)*t7;
c2 = variabelinput(1:data,3)*t8;
d2 = variabelinput(1:data,4)*t9;

```

```
e2 = ones(data,1)*t10;  
hn2 = [a2 b2 c2 d2 e2];
```

```
a3 = variabelinput(1:data,1)*t11;  
b3 = variabelinput(1:data,2)*t12;  
c3 = variabelinput(1:data,3)*t13;  
d3 = variabelinput(1:data,4)*t14;  
e3 = ones(data,1)*t15;  
hn3 = [a3 b3 c3 d3 e3];
```

```
a4 = variabelinput(1:data,1)*t16;  
b4 = variabelinput(1:data,2)*t17;  
c4 = variabelinput(1:data,3)*t18;  
d4 = variabelinput(1:data,4)*t19;  
e4 = ones(data,1)*t20;  
hn4 = [a4 b4 c4 d4 e4];
```

```
a5 = variabelinput(1:data,1)*t21;  
b5 = variabelinput(1:data,2)*t22;  
c5 = variabelinput(1:data,3)*t23;  
d5 = variabelinput(1:data,4)*t24;  
e5 = ones(data,1)*t25;  
hn5 = [a5 b5 c5 d5 e5];
```

```
a6 = variabelinput(1:data,1)*t26;  
b6 = variabelinput(1:data,2)*t27;  
c6 = variabelinput(1:data,3)*t28;  
d6 = variabelinput(1:data,4)*t29;  
e6 = ones(data,1)*t30;  
hn6 = [a6 b6 c6 d6 e6];
```

```
a7 = variabelinput(1:data,1)*t31;  
b7 = variabelinput(1:data,2)*t32;  
c7 = variabelinput(1:data,3)*t33;  
d7 = variabelinput(1:data,4)*t34;  
e7 = ones(data,1)*t35;  
hn7 = [a7 b7 c7 d7 e7];
```

```

% Fungsi aktivasi hidden node
for i = 1:data;
    nilaihn1(i,1) = sum(hn1(i,1:4));           %Langkah
4 JST
end;

for i = 1:data;
    nilaihn2(i,1) = sum(hn2(i,1:4));
end;

for i = 1:data;
    nilaihn3(i,1) = sum(hn3(i,1:4));
end;

for i = 1:data;
    nilaihn4(i,1) = sum(hn4(i,1:4));
end;

for i = 1:data;
    nilaihn5(i,1) = sum(hn5(i,1:4));
end;

for i = 1:data;
    nilaihn6(i,1) = sum(hn6(i,1:4));
end;

for i = 1:data;
    nilaihn7(i,1) = sum(hn7(i,1:4));
end;

% Hidden ke output
fhn1 = tansig(nilaihn1)*t36;
fhn2 = tansig(nilaihn2)*t37;
fhn3 = tansig(nilaihn3)*t38;
fhn4 = tansig(nilaihn4)*t39;
fhn5 = tansig(nilaihn5)*t40;
fhn6 = tansig(nilaihn6)*t41;
fhn7 = tansig(nilaihn6)*t42;
fhn8 = ones(data,1)*t43; %bias pada output layer

```

```

matrix = [fhn1, fhn2, fhn3, fhn4, fhn5, fhn6, fhn7,
fhn8];

for i = 1:data;
    nilaion(i,1) = sum(matrix(i,1:8));
%Langkah 5 JST
end;

% Fungsi aktivasi output layer
prediksi = tansig(nilaion);

max_data = 1.33;           %Ganti nilai max
min_data = 0.07;          %Ganti nilai min
output = ((prediksi-0.1)*(max_data-
min_data)/0.8)+min_data;

% Objective function
selisih = output - validasi;
selisihkuadrat = selisih.^2;
rmse = sqrt((sum(selisihkuadrat))/data);
NilaiFitness = 1/rmse;

```

## F.2 Optimasi *Genetic Algorithm*

```
%GENETIC ALGORITHM%

clear all;
close all;
clc

% Untuk tiap hidden node
% 1. Ganti dimension
% 2. Ganti 2 fungsi objektif
% 3. Ganti nama untuk save

%Pembangkitan Populasi dan Parameter
Npop      = 20;           %jumlah individu
Maxit     = 90;          %iterasi
el        = 0.8;         %elatism
Pc        = 0.8;         %probabilitas crossover
Pm        = 0.033;       %probabilitas mutasi
Nbit      = 10;          %jumlah bit kromoson
Dimension = 43;          %jumlah variabel yg
dioptimasi

%Constraint
Low = [-1]; % Lower Bound of Variables
Up = [1];  % Upper Bound of Variables
LB = repmat(Low,1,Dimension);
UB = repmat(Up,1,Dimension);

%sesuai dengan constraint fungsi objektif
eBangkit    = [];
Individu    = [];
eIndividu   = [];
david       = [];
Dadatfit    = [];
Datfit      = [];
summary     = [];
eDadatfit   = [];
efitnessmax = [];
```

```

eIndividuMax = [];
Bangkit = round(rand(Npop,Nbit*Dimension));

popsize = size(Bangkit,1);
for i = 1:Dimension
    batas(i) = UB(i)-LB(i);
end
for i =1:Npop
    for j = 1:Dimension
        Desimal(i,j) = bi2de(Bangkit(i,((j*Nbit)-(
(Nbit-1)):(j*Nbit)), 'left-msb'));
        Individu(i,j) = (Desimal(i,j)*batas(:,j)-
batas(:,j)+LB(:,j)*(2^Nbit-1))/(2^Nbit-1);
    end
end
Datfit = [];
variabel = [];

% Fungsi Objektif -----
for i = 1:size(Individu,1)
    [fitness,output] = fob7GABaru(Individu(i,:));
    Datfit = [Datfit;fitness];
    [fitemax,nmax]=max(Datfit);
end

Dadatfit = [];
for generasi=1:Maxit
    disp('GA processing')
    disp(['Iteration : ' num2str(generasi)])
    clear command windows
    clear command history
    clear memory
    if generasi > 1
        sort_fit =
sortrows(sort,Nbit*Dimension+1);
        Individu1 = sort_fit(round((1-
el)*Npop+1):Npop,:);
        remain =
sort_fit(round(el*Npop)+1:Npop,:);
        X = Individu1;

```



```

M = size(X,1);
sumfitness = sum(Datfit);
for i=1:M
    Prob(i) = Datfit(i)/sumfitness;
end
for i=2:M
    Prob(i) = Prob(i)+Prob(i-1);
end
for i=1:M
    n=rand;
    k=1;
    for j=1:M-1
        if (n>Prob(j))
            k=j+1;
        end
    end
    Xparents(i,:) = X(k,:);
end

% Crossover
[M,d] = size(Xparents);
Xcrossed = Xparents;
for i=1:2:M-1
    c=rand;
    if (c<=Pc)
        p=ceil((d-1)*rand);
        Xcrossed(i,:) = [Xparents(i,1:p)
Xparents(i+1,p+1:d)];
        Xcrossed(i+1,:) = [Xparents(i+1,1:p)
Xparents(i,p+1:d)];
    end
end
if (M/2~=floor(M/2))
    c=rand;
    if (c<=Pc)
        p=ceil((d-1)*rand);
        str=ceil((M-1)*rand);
        Xcrossed(M,:) = [Xparents(M,1:p)
Xparents(str,p+1:d)]; %the first child is chosen
    end
end

```

```

end

% Mutasi
[M,d] = size(Xcrossed);
Xnew=Xcrossed;
for i=1:M
    for j=1:d
        p=rand;
        if (p<=Pm)
            Xnew(i,j)=1-Xcrossed(i,j);
        end
    end
end
disp('New fitness calculation');
Bangkit =
[Xnew(:,1:Nbit*Dimension);remain(:,1:Nbit*Dimensi
on)];
end
eBangkit = [eBangkit; Bangkit];
for i =1:Npop
    for j = 1:Dimension;
        Desimal(i,j) =
bi2de(Bangkit(i, ((j*Nbit)-(Nbit-
1)): (j*Nbit)), 'left-msb');
        Individu(i,j) =
(Desimal(i,j)*batas(:,j)-
batas(:,j)+LB(:,j)*(2^Nbit-1))/(2^Nbit-1);
    end
end
Datfit = [];

% Fungsi Objektif
for i = 1:Npop
    [fitness,output] = fob7GAbaru(Individu(i,:))
    Datfit = [Datfit;fitness];
    [fitemax,nmax] = max(Datfit);
end
Dadatfit = Datfit;
eDadatfit = [eDadatfit;Dadatfit];
eIndividu = [eIndividu;Individu];

```

```

[fitnessmax,nmax] = max(eDadatfit);
efitnessmax = [efitnessmax;fitnessmax];

balikdata = 1./efitnessmax

BangkitMax = eBangkit(nmax,:);
IndividuMax = eIndividu(nmax,:);
eIndividuMax = [eIndividuMax;IndividuMax];
BangkitMaxlast = BangkitMax;
schedmax = BangkitMax;
sort = [Bangkit Dadatfit];
summary = [summary; sort];
david = [david; Dadatfit];
clc

min_design_variabel=IndividuMax(1,:)
min_objective_function=fitness(1,:)
RMSE_Data_Latih = 1/min_objective_function
figure(gcf)
title('Grafik Nilai Minimum GA','color','b')
xlabel('Jumlah Iterasi')
ylabel('RMSE')
grid on
hold on
plot(balikdata, 'DisplayName', 'efitnessmax',
'YDataSource', 'efitnessmax');
hold on
end

saveas(gcf, 'Baru RMSE Pelatihan Data Latih
Gelombang HN7.png')

%SIMPAN SEMUA DATA-----
save ('Baru Pelatihan GA Gelombang HN 7.mat')

```

### F.3 Pengujian Bobot Optimasi

```

variabelinput = xlsread('DATA
GELOMBANG.xlsx', 'Pengujian', 'G2:J4516');
validasi = xlsread('DATA
GELOMBANG.xlsx', 'Pengujian', 'E2:E4516');

min_design_variabel =      [0.6579      0.6794
0.1711      -0.5679      0.6266      0.3236      0.1535
-0.1105      -0.9374      0.4252      -0.8495      -0.0987
0.6755      0.5699      0.8475      -0.4154      0.1085
0.3236      -0.6129      0.9707      -0.4995      0.6774
-0.9003      0.4213      -0.7498      -0.1281      -0.9022
-0.7048      0.8045      0.9472      -0.6657      0.6168
-0.4174      -0.1222      -0.8201      0.2923      -0.9609
0.1437      -0.7439      0.1593      -0.3118      0.4878
0.0225];

data = 4515;          %Jumlah Data Latih

% 1 individu = seluruh bobot
t1=min_design_variabel(:,1)'; %input 1 ke node 1
t2=min_design_variabel(:,2)'; %input 2 ke node 1
t3=min_design_variabel(:,3)'; %input 3 ke node 1
t4=min_design_variabel(:,4)'; %input 4 ke node 1
t5=min_design_variabel(:,5)'; %bias ke node 1

t6=min_design_variabel(:,6)'; %input 1 ke node 2
t7=min_design_variabel(:,7)'; %input 2 ke node 2
t8=min_design_variabel(:,8)'; %input 3 ke node 2
t9=min_design_variabel(:,9)'; %input 4 ke node 2
t10=min_design_variabel(:,10)'; %bias ke node 2

t11=min_design_variabel(:,11)'; %input 1 ke node 3
t12=min_design_variabel(:,12)'; %input 2 ke node 3
t13=min_design_variabel(:,13)'; %input 3 ke node 3
t14=min_design_variabel(:,14)'; %input 4 ke node 3
t15=min_design_variabel(:,15)'; %bias ke node 3

```

```

t16=min_design_variabel(:,16)';%input 1 ke node 4
t17=min_design_variabel(:,17)';%input 2 ke node 4
t18=min_design_variabel(:,18)';%input 3 ke node 4
t19=min_design_variabel(:,19)';%input 4 ke node 4
t20=min_design_variabel(:,20)';%bias ke node 4

t21=min_design_variabel(:,21)';%input 1 ke node 5
t22=min_design_variabel(:,22)';%input 2 ke node 5
t23=min_design_variabel(:,23)';%input 3 ke node 5
t24=min_design_variabel(:,24)';%input 4 ke node 5
t25=min_design_variabel(:,25)';%bias ke node 5

t26=min_design_variabel(:,26)';%input 1 ke node 6
t27=min_design_variabel(:,27)';%input 2 ke node 6
t28=min_design_variabel(:,28)';%input 3 ke node 6
t29=min_design_variabel(:,29)';%input 4 ke node 6
t30=min_design_variabel(:,30)';%bias ke node 6

t31=min_design_variabel(:,31)';%input 1 ke node 7
t32=min_design_variabel(:,32)';%input 2 ke node 7
t33=min_design_variabel(:,33)';%input 3 ke node 7
t34=min_design_variabel(:,34)';%input 4 ke node 7
t35=min_design_variabel(:,35)';%bias ke node 7

t36=min_design_variabel(:,36)'; %node 1 ke output
t37=min_design_variabel(:,37)'; %node 2 ke output
t38=min_design_variabel(:,38)'; %node 3 ke output
t39=min_design_variabel(:,39)'; %node 4 ke output
t40=min_design_variabel(:,40)'; %node 5 ke output
t41=min_design_variabel(:,41)'; %node 6 ke output
t42=min_design_variabel(:,42)'; %node 7 ke output
t43=min_design_variabel(:,43)'; %bias ke output

%Input ke hidden
a1 = variabelinput(1:data,1)*t1;
b1 = variabelinput(1:data,2)*t2;
c1 = variabelinput(1:data,3)*t3;
d1 = variabelinput(1:data,4)*t4;
e1 = ones(data,1)*t5;
hn1 = [a1 b1 c1 d1 e1];

```

```
a2 = variabelinput(1:data,1)*t6;  
b2 = variabelinput(1:data,2)*t7;  
c2 = variabelinput(1:data,3)*t8;  
d2 = variabelinput(1:data,4)*t9;  
e2 = ones(data,1)*t10;  
hn2 = [a2 b2 c2 d2 e2];
```

```
a3 = variabelinput(1:data,1)*t11;  
b3 = variabelinput(1:data,2)*t12;  
c3 = variabelinput(1:data,3)*t13;  
d3 = variabelinput(1:data,4)*t14;  
e3 = ones(data,1)*t15;  
hn3 = [a3 b3 c3 d3 e3];
```

```
a4 = variabelinput(1:data,1)*t16;  
b4 = variabelinput(1:data,2)*t17;  
c4 = variabelinput(1:data,3)*t18;  
d4 = variabelinput(1:data,4)*t19;  
e4 = ones(data,1)*t20;  
hn4 = [a4 b4 c4 d4 e4];
```

```
a5 = variabelinput(1:data,1)*t21;  
b5 = variabelinput(1:data,2)*t22;  
c5 = variabelinput(1:data,3)*t23;  
d5 = variabelinput(1:data,4)*t24;  
e5 = ones(data,1)*t25;  
hn5 = [a5 b5 c5 d5 e5];
```

```
a6 = variabelinput(1:data,1)*t26;  
b6 = variabelinput(1:data,2)*t27;  
c6 = variabelinput(1:data,3)*t28;  
d6 = variabelinput(1:data,4)*t29;  
e6 = ones(data,1)*t30;  
hn6 = [a6 b6 c6 d6 e6];
```

```
a7 = variabelinput(1:data,1)*t31;  
b7 = variabelinput(1:data,2)*t32;  
c7 = variabelinput(1:data,3)*t33;  
d7 = variabelinput(1:data,4)*t34;
```

```
e7 = ones(data,1)*t35;
hn7 = [a7 b7 c7 d7 e7];

% Fungsi aktivasi hidden node
for i = 1:data;
    nilaihn1(i,1) = sum(hn1(i,1:4));
end;

for i = 1:data;
    nilaihn2(i,1) = sum(hn2(i,1:4));
end;

for i = 1:data;
    nilaihn3(i,1) = sum(hn3(i,1:4));
end;

for i = 1:data;
    nilaihn4(i,1) = sum(hn4(i,1:4));
end;

for i = 1:data;
    nilaihn5(i,1) = sum(hn5(i,1:4));
end;

for i = 1:data;
    nilaihn6(i,1) = sum(hn6(i,1:4));
end;

for i = 1:data;
    nilaihn7(i,1) = sum(hn7(i,1:4));
end;

% Hidden ke output
fhn1 = tansig(nilaihn1)*t36;
fhn2 = tansig(nilaihn2)*t37;
fhn3 = tansig(nilaihn3)*t38;
fhn4 = tansig(nilaihn4)*t39;
fhn5 = tansig(nilaihn5)*t40;
fhn6 = tansig(nilaihn6)*t41;
```

```

fhn7 = tansig(nilaihn6)*t42;
fhn8 = ones(data,1)*t43; %bias pada output layer
matrix =[fhn1,fhn2,fhn3,fhn4, fhn5, fhn6, fhn7,
fhn8];

for i = 1:data;
    nilaion(i,1)= sum(matrix(i,1:8));
%Langkah 5 JST
end;

% Fungsi aktivasi output layer
prediksi = tansig(nilaion);

max_data = 1.24; %Ganti nilai max
min_data = 0.18; %Ganti nilai min
output = ((prediksi-0.1)*(max_data-
min_data)/0.8)+min_data;
selisih = output - validasi;
selisihkuadrat = selisih.^2;
jumlahselisihkuadrat = sum(selisihkuadrat);
mse = jumlahselisihkuadrat/data;
RMSE = sqrt(jumlahselisihkuadrat/data)
MAPE = ((sum(abs(selisih./validasi)))*100)/data;

figure,
plot(output, 'bo-')
hold on
plot(validasi, 'ro-')
hold off
grid on
title(strcat(['Pengujian JST GA Prediksi
Gelombang HN7', ' (MSE = ',...
num2str(mse), ')', ' (MAPE= ',...
num2str(MAPE), '%)', ' (RMSE= ',...
num2str(RMSE), ')']));
xlabel('Jam ke-')
ylabel('Tinggi Gelombang (m)')
legend('Keluaran JST', 'Target', 'Location', 'Best')
saveas(gcf, 'Pengujian Data Uji Gelombang HN 7
GA.png')

```



## BIODATA PENULIS



Penulis mempunyai nama lengkap Vidiyan Nabila, lahir di Gresik pada tanggal 13 Januari 1997, merupakan anak bungsu dari tiga bersaudara. Penulis menempuh pendidikan sekolah dasar di MI. Ma'arif Sidomukti Kebomas Gresik, pendidikan menengah pertama di SMP Negeri 3 Gresik, pendidikan menengah atas di SMA Negeri 1 Gresik, dan melanjutkan kuliah di Departemen Teknik Fisika, Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2015. Pada tahun 2019 penulis menyelesaikan Tugas Akhir sebagai syarat untuk menyelesaikan studi S1 di Departemen Teknik Fisika dengan judul “**Rancang Bangun Prediktor Ketinggian Gelombang Air Laut Menggunakan Metode Jaringan Syaraf Tiruan dengan Optimasi *Genetic Algorithm* (GA)**”. Bagi pembaca yang mempunyai saran atau kritik tentang tugas akhir ini bisa menghubungi penulis melalui **vidiyan.nabila.vn@gmail.com**.