



Tugas Akhir - TF181801

RANCANG BANGUN PREDIKTOR SEBARAN IKAN MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN DENGAN OPTIMASI *PARTICLE SWARM OPTIMIZATION*

Adelia Wahyu Puspita Sari
NRP. 0231154000016

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

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember



Tugas Akhir - TF181801

**RANCANG BANGUN PREDIKTOR SEBARAN
IKAN MENGGUNAKAN METODE JARINGAN
SYARAF TIRUAN DENGAN OPTIMASI *PARTICLE
SWARM OPTIMIZATION***

Adelia Wahyu Puspita Sari
NRP. 0231154000016

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
Surabaya 2019



Final Project - TF181801

DESIGN OF ABUNDANCE FISH PREDICTOR USING ARTIFICIAL NEURAL NETWORK AND *PARTICLE SWARM OPTIMIZATION*

Adelia Wahyu Puspita Sari
NRP. 0231154000016

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 : Adelia Wahyu Puspita Sari
NRP : 0231154000016
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 Sebaran Ikan Menggunakan Jaringan Syaraf Tiruan Dengan Optimasi *Particle Swarm Optimization*” 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,



Adelia Wahyu Puspita Sari
NRP. 0231154000016

LEMBAR PENGESAHAN

RANCANG BANGUN PREDIKTOR SEBARAN IKAN MENGUNAKAN METODE JARINGAN SYARAF TIRUAN DENGAN OPTIMASI *PARTICLE SWARM* *OPTIMIZATION*

TUGAS AKHIR

Oleh:



Adelia Wahyu Puspita Sari
NRP. 0231154000016

Surabaya, 19 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
Fisika FTI-ITS
Agus Muhammad Hatta, ST, MSi, PhD
NIP. 19780902 200312 1 002

**RANCANG BANGUN PREDIKTOR SEBARAN IKAN
MENGUNAKAN METODE JARINGAN SYARAF
TIRUAN DENGAN OPTIMASI *PARTICLE SWARM
OPTIMIZATION***





TUGAS AKHIR

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

Oleh :

ADELIA WAHYU PUSPITA SARI
NRP. 0231154000016

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Syamsul Arifin, M.T. (Pembimbing I)
2. Prof. Dr. Ir. Aulia Siti Aisjah, M.T. (Pembimbing II)
3. Dr. Suyanto, S.T, M.T. (Penguji I)
4. Moch. Kamalul Wafi, S.T,MSc.Dic. (Penguji III)

SURABAYA
2019

**RANCANG BANGUN PREDIKTOR SEBARAN IKAN
MENGUNAKAN METODE JARINGAN SYARAF
TIRUAN DENGAN OPTIMASI *PARTICLE SWARM
OPTIMIZATION***

Nama : Adelia Wahyu Puspita Sari
NRP : 0231154000016
Departemen : Teknik Fisika
Dosen Pembimbing : 1. Dr. Ir. Syamsul Arifin, MT
2. Prof. Dr. Ir. Aulia Siti Aisjah, MT

ABSTRAK

Badan Informasi Geospasial (BIG) menyatakan bahwa garis pantai Indonesia memiliki total panjang 99,093 kilometer. Hal ini menunjukkan bahwa Indonesia juga memiliki kekayaan laut yang sangat banyak terutama ikan. Sampai saat ini masih terdapat permasalahan dimana kurangnya informasi mengenai keberadaan ikan di perairan Indonesia, sehingga perlu adanya suatu perancangan orediksi untuk menginformasikan adanya keberadaan ikan di suatu wilayah. Tujuan penelitian ini adalah untuk memperoleh algoritma prediktor terbaik untuk menentukan keberadaan ikan dengan menghitung prediksi jumlah tangkapan ikan untuk ikan biji nangka, kerapu dan kuniran. Metode yang diusulkan pana penelitian ini yaitu JST, dan JST-PSO. Data sekunder didapatkan dari NOAA, BMKG, dan PPN Brondong. Data sekunder dependen dan independen diuji *missing data* , *outlier*, dan korelasi. Kemudian JST dan JST-PSO dirancang arsitektur nya. Arsitektur terbaik yang digunakan untuk memprediksi *real time* di pantai kenjeran . Arsitektur terbaik yang didapatkan pada JST-PSO dengan *input* berupa kecepatan angin, suhu permukaan laut, dan curah hujan. Gabungan JST-PSO dengan *hidden node* yang dipasang sebanyak 5, 9 dan 10 unit mampu menurunkan RMSE masing-masing sebesar 26 kg, 228,7 kg, dan 121,9 kg.

Kata Kunci: *Particle Swarm Optimization*, *hidden node*, jaringan saraf tiruan, *real time*

DESIGN OF ABUNDANCE FISH PREDICTOR USING ARTIFICIAL NEURAL NETWORK AND PARTICLE SWARM OPTIMIZATION

Name : Adelia Wahyu Puspita Sari
NRP : 0231154000016
Department : Engineering Physics
Supervisors : 1. Dr. Ir. Syamsul Arifin, MT
2. Prof. Dr. Ir. Aulia Siti Aisjah, MT

ABSTRACT

The Geospatial Information Agency (BIG) states that Indonesia's coastline has a total length of 99,093 kilometers. This shows that Indonesia also has a wealth of marine resources, especially fish. Until now there are still problems where there is a lack of information about the presence of fish in Indonesian waters, so there needs to be an edictional design to inform the existence of fish in an area. The purpose of this study was to obtain the best predictor algorithm for determining the presence of fish by calculating the prediction of the number of fish catches for fish of jackfruit seeds, groupers and turmeric. The method proposed for this research is ANN, and JST-PSO. Secondary data was obtained from NOAA, BMKG, and Brondong PPN. Dependent and independent secondary data were tested for missing data, outliers, and correlations. Then the JST and JST-PSO are designed by the architecture. The best architecture that is used to predict real time on kenjeran beach. The best architecture obtained on ANN-PSO with input in the form of wind speed, sea surface temperature and rainfall. Combined JST-PSO with hidden nodes installed 5, 9 and 10 units were able to reduce RMSE by 26 kg, 228.7 kg and 121.9 kg respectively.

Keywords: *Particle Swarm Optimization, hidden node, neural network, real, RMSE*

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 SEBARAN IKAN MENGUNAKAN METODE JARINGAN SYARAF TIRUAN DENGAN OPTIMASI *PARTICLE SWARM OPTIMIZATION*

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

1. Agus M. Hatta, S.T, M.Si, Ph.D 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 dan memberikan saran dan dukungan selama pengerjaan tugas akhir penulis.
3. 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.
4. Keluarga penulis yang selalu memberikan dukungan dan do'a.
5. Tim tugas akhir *Buoyweather*, Dina, Tika, Bela, Bima, Wildan, Abiyyu, dan Isa yang banyak membantu dan memberi support satu sama lain dalam proses pengerjaan melalui semua suka duka dari awal hingga akhir.

6. Sahabat-sahabat saya, Yesika, Cesa, Aul dan Ghina, yang bersedia membantu saya sekaligus memberikan dukungan agar saya selalu semangat mengerjakan tugas akhir saya ini.

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	vii
ABSTRAK	xiii
<i>ABSTRACT</i>	xv
DAFTAR GAMBAR	xxi
DAFTAR ISTILAH	xxvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Gelombang Laut.....	5
2.2 Jaringan Syaraf Tiruan	8
2.3 Algoritma Pelatihan Levenberg Marquardt.....	12
2.4 Optimasi <i>Particle Swarm Optimization</i> (PSO)	15
2.5 <i>Buoyweather Station Type II</i>	17
2.6 Validasi Prediktor.....	18
2.7 Sebaran Ikan di Area WPP-712.....	19
BAB III METODE PENELITIAN	21
3.1 Perumusan Masalah.....	22
3.2 Studi Literatur	22
3.3 Penentuan Variabel Penelitian.....	22
3.4 Pengumpulan Data	23
3.5 Pengolahan Data.....	24
3.6 Perancangan Algoritma Prediktor	26
3.7 Pemilihan Arsitektur Terbaik Pada Prediktor Sebaran Ikan	33
3.8 Analisis Hasil dari Sistem Prediksi Ikan	36
3.9 Pengujian Realtime	37
3.11 Kesimpulan dan Saran.....	39

BAB IV HASIL DAN PEMBAHASAN.....	41
4.1 Hasil Pengolahan Data	41
4.2 Prediksi Sebaran Ikan Menggunakan Jaringan Syaraf Tiruan	44
4.3 Prediksi Sebaran Ikan Menggunakan Jaringan Syaraf Tiruan dengan Optimasi <i>Particle Swarm Optimization</i> (PSO).....	53
4.4 Perbandingan Hasil Antara Metode Prediktor JST-LM dan JST-PSO	63
4.5 Prediktor Sebaran Ikan Menggunakan Data <i>Realtime</i> <i>Buoyweather Type II</i>	69
BAB V PENUTUP.....	71
1.1 Kesimpulan.....	71
1.2 Saran.....	71
DAFTAR RUJUKAN	73
LAMPIRAN A Nilai RMSE	77
BIODATA PENULIS.....	101

DAFTAR GAMBAR

Gambar 2. 1 Upheneus Mollucensis Mullidae (Semuaikan, 2018)	5
Gambar 2. 2 Epinephelus Coiodes Serranidae (Semuaikan,2018)	6
Gambar 2. 3 Upheneus Tragula Mullidae (Semuaikan, 2018)	7
Gambar 2. 4 Hubungan Antar Neuron dalam Arsitektur JST (Fausset, 1994).....	9
Gambar 2. 5 Jaringan Tipe Feed Forward (Jain, Deo, & Rajendran, 2011).....	10
Gambar 2. 6 Fungsi Aktivasi Linier (Fausset, 1994).....	11
Gambar 2. 7 Fungsi Aktivasi Sigmoid Bipolar (Fausset, 1994)	12
Gambar 2. 8 Fungsi Aktivasi Sigmoid Biner (Fausset, 1994) ...	12
Gambar 2. 9 Diagram Blok Algoritma Levenberg Marquardt (Yu & Wilamowski, 2011).....	14
Gambar 2. 10 Buoyweather Station Type II.....	17
Gambar 4. 1 Hasil RMSE Ikan Biji Nangka Jaringan Syaraf Tiruan.....	45
Gambar 4. 2 Hasil RMSE Ikan Kerapu Jaringan Syaraf Tiruan	46
Gambar 4. 3 Hasil RMSE Ikan Kuniran Jaringan Syaraf Tiruan	47
Gambar 4. 4 Hasil Uji Ikan Biji Nangka Jaringan Syaraf Tiruan <i>Hidden node 1</i>	48
Gambar 4. 5 Hasil Uji Ikan Kerapu Jaringan Syaraf Tiruan <i>Hidden node 3</i>	49
Gambar 4. 6 Hasil Uji Ikan Kuniran Jaringan Syaraf Tiruan <i>Hidden node 2</i>	49
Gambar 4. 7 Ilustrasi Penamaan Bobot Pada Arsitektur Jaringan Syaraf Tiruan.....	51
Gambar 4. 8 Hasil RMSE Ikan Biji Nangka Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	54

Gambar 4. 9	Hasil RMSE Ikan Kerapu Jaringan Syaraf Truan – <i>Particle Swarm Optimization</i>	55
Gambar 4. 10	Hasil RMSE Ikan Kuniran Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	56
Gambar 4. 11	Hasil Pengujian Ikan Biji Nangka Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization Hidden</i> <i>node 5</i>	57
Gambar 4. 12	Hasil Pengujian Ikan Kerapu Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization Hidden</i> <i>node 9</i>	58
Gambar 4. 13	Hasil Pengujian Ikan Kuniran Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	59
Gambar 4. 14	Hasil Validasi Ikan Biji Nangka Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	66
Gambar 4. 15	Hasil Validasi Ikan Kerapu Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	67
Gambar 4. 16	Hasil Validasi Ikan Kuniran Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	68

DAFTAR TABEL

Tabel 2. 1 Skala Penentuan Nilai Akurasi Suatu Prediktor	19
Tabel 3. 1 Nilai Interval Korelasi Terhadap Tingkat hubungannya (Sugiyono, 2007).....	26
Tabel 3. 2 Pembagian Data Pelatihan dan Validasi	27
Tabel 3. 3 Parameter Algoritma Jaringan Syaraf Tiruan	28
Tabel 3. 4 Parameter Algoritma Jaringan Syaraf Tiruan - <i>Particle Swarm Optimization</i>	31
Tabel 3. 5 Spesifikasi Sensor Suhu Permukaan Laut (Maxim Integrated, 2015)	38
Tabel 3. 6 Spesifikasi Sensor Curah Hujan (Depo Inovasi, 2011)	38
Tabel 4. 1 Jumlah Data <i>Outlier</i> Pada Variabel yang Digunakan	42
Tabel 4. 2 Hasil Uji Korelasi Nilai r.....	43
Tabel 4. 3 Hasil Uji Korelasi Nilai Signifikansi.....	44
Tabel 4. 4 Bobot Terbaik Prediksi Ikan Biji Nangka Jaringan Syaraf Tiruan	50
Tabel 4. 5 Bobot Terbaik Prediksi Ikan Kerapu Jaringan Syaraf Tiruan.....	52
Tabel 4. 6 Bobot Terbaik Prediksi Ikan Kuniran Jaringan Syaraf Tiruan.....	53
Tabel 4. 7 Bobot Terbaik Prediksi Ikan Biji Nangka Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	60
Tabel 4. 8 Bobot Terbaik Ikan Kerapu Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	61
Tabel 4. 9 Bobot Terbaik Prediksi Ikan Kuniran Jaringan Syaraf Tiruan – <i>Particle Swarm Optimization</i>	62
Tabel 4. 10 Perbandingan Hasil Keluaran Prediktor JST-LM dan JST-PSO	64
Tabel 4. 11 Hasil Pengukuran Sensor dan Prediksi Keluaran dari Prediktor JST-PSO di Pantai Kenjeran.....	69

Tabel 4. 12 Hasil Pengukuran Sensor dan Prediksi Keluaran dari Prediktor JST-PSO di Laut Jawa	70
---	----

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	-
μ	Konstanta pembelajaran	-
I	Matriks identitas	-
$d_{p,m}$	<i>Output</i> yang diharapkan	meter atau cm/s
$e_{p,m}$	Target	meter atau cm/s
σ	Standar deviasi	-
n	Jumlah data	-
net	<i>Output</i> 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	<i>Input</i> 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	-
RMSE	<i>Root Mean Square Error</i> atau parameter untuk mengevaluasi hasil prediktor	-

<i>MSE</i>	<i>Mean Square Error</i> atau parameter untuk mengevaluasi hasil prediktor	-
<i>MAPE</i>	<i>Mean Absolute Percentage Error</i> atau parameter untuk mengevaluasi hasil prediktor	%
<i>r</i>	<i>Pearson Correlation</i> atau korelasi hubungan antar variabel	-
<i>x'</i>	Data hasil normalisasi	-
<i>x</i>	Data asli	meter atau cm/s
<i>x_{max}</i>	Nilai maksimum data asli	meter atau cm/s
<i>x_{min}</i>	Nilai minimum data asli	meter atau cm/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 untuk mengobservasi keadaan laut secara *real time*

Buoyweather tipe II

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

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

Generalisasi model matematika dari jaringan saraf biologi

Particle Swarm Optimization (PSO)

Metode optimasi yang menirukan perilaku sosial organisme seperti kawanan burung atau ikan.

National Oceanic and Atmospheric Administration (NOAA)

Sebuah badan ilmiah di Departemen Perdagangan Amerika Serikat yang berfokus pada kondisi samudera dan atmosfer.

Tropical Atmosphere Ocean (TAO)

Salah satu *buoy* yang dikembangkan oleh NOAA untuk mengobservasi keadaan laut di Samudera Pasifik

Pelabuhan Perikanan Nusantara (PPN) Brondong

Pusat kegiatan perikanan yang dilengkapi dengan fasilitas di darat dan di perairan bagi masyarakat, nelayan, dan usaha perikanan

Research Moored Array for African –Asian-Australian Monsoon Analysis and Prediction (RAMA)

Salah satu *buoy* yang dikembangkan oleh NOAA untuk mengobservasi keadaan laut di Samudera Hindia

Root Mean Squared Error (RMSE)

Metode alternatif untuk mengevaluasi teknik peramalan yang digunakan untuk mengukur tingkat akurasi hasil prakiraan suatu model

Sebaran jenis ikan

Berat (dengan satuan kilogram) beberapa jenis ikan di satu lokasi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia yang memiliki banyak pulau tentu juga memiliki garis pantai yang panjang. Menurut Badan Informasi Geospasial (BIG), garis pantai Indonesia memiliki total sepanjang 99.093 kilometer (Samantha, 2013). Dengan garis pantai yang panjang menunjukkan bahwa Indonesia juga memiliki kekayaan laut yang sangat banyak terutama ikan. Kekayaan tersebut dapat dijadikan sebagai salah satu sumber pendapatan bagi pertumbuhan ekonomi nasional. Sampai saat ini masih terdapat permasalahan dimana kurangnya informasi mengenai keberadaan ikan di perairan Indonesia. Hal ini menyebabkan distribusi nelayan dan kapal ikan juga tidak merata. Lebih dari 90% armada kapal ikan Indonesia fokus pencariannya di perairan pesisir dan laut dangkal seperti selat Malaka, Pantura, Selat Bali, dan pesisir selatan Sulawesi (Jagad, 2016). Hal tersebut mengakibatkan adanya kelebihan tangkapan ikan pada wilayah tertentu yang menyebabkan nelayan semakin miskin dan sumber daya ikan lama kelamaan akan punah.

Oleh karena itu, dibutuhkan suatu informasi agar nelayan tahu tentang penyebaran ikan di suatu tempat agar penangkapan ikan lebih merata. Sehingga perlu dilakukan perancangan prediksi untuk mendeteksi keberadaan jenis ikan tersebut. Membuat perancangan suatu prediksi diperlukan variabel-variabel yang mempengaruhi lingkungan laut terhadap ikan. Suhu Permukaan Laut (SPL) merupakan variabel yang cukup berpengaruh terhadap persebaran ikan. Suhu Permukaan Laut adalah salah satu sifat fisika air laut yang dapat mempengaruhi metabolisme dan pertumbuhan organisme perairan. Di samping itu suhu sangat berpengaruh terhadap jumlah oksigen yang terlarut dalam air dan ini berguna untuk mengetahui stok ikan di masa depan (Soesono, 1974). Parameter kimia lain yang juga berpengaruh terhadap keberadaan ikan yaitu kecepatan angin. Begitu pula faktor lingkungan lain, seperti curah hujan (Hobday, Spillman, Paige Eveson, & Hartog, 2016). Curah hujan akan mempengaruhi

kecepatan angin (Daniel, Santiago, Juan, dan Patricia, 2013) dan suhu permukaan laut (Katsaros & Buettner, 1969).

Penggunaan JST untuk prediksi sebaran ikan sebelumnya pernah dilakukan oleh Srestha pada tahun 2018 mengenai prediktor sebaran ikan di Perairan Jawa menggunakan JST dengan algoritma pelatihan *Backpropagation* (BP) didapatkan nilai RMSE sebesar 1000 kg -7000 kg (Srestha, 2018). Pada penelitian tersebut penulis membandingkan keluaran prediksi yang didapatkan antara metode JST-BP dan JST-GA. Penelitian diatas dapat dikatakan bahwa algoritma pelatihan *Backpropagation* (BP) kurang maksimal dalam kecepatan mencapai konvergensi dalam proses pelatihan atau *training*. Sehingga dalam penelitian ini dilakukan 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 & K. Venugopalan, 2014).

Sehingga pada tugas akhir ini dilakukan penelitian untuk prediksi sebaran ikan menggunakan metode Jaringan Syaraf Tiruan dengan dua metode. Metode pertama yaitu Pelatihan Jaringan Syaraf Tiruan menggunakan algoritma pelatihan *Levenberg-Marquadt*. Sedangkan metode kedua yaitu Jaringan Syaraf Tiruan diptimasi menggunakan *Particle Swarm Optimization* (PSO). Dimana *Particle Swarm Optimiaztion* merupakan algortima evolusioner yang menjadi alat penting untuk optimasi dan penyelesaian masalah kompleks. PSO adalah teknik komputasi yang menarik dengan kemampuan tinggi dan aplikasi praktis karena memiliki prinsip kerja sederhana dan mudah digunakan dalam berbagai bidang (Khare & Rangnekar, 2013).

1.2 Perumusan Masalah

Permasalahan yang diangkat dalam penelitian tugas akhir ini antara lain:

- a. Apakah algoritma Jaringan Syaraf Tiruan dengan optimasi *Particle Swarm Optimization* dapat memprediksi sebaran ikan (kg) secara akurat dan tepat?
- b. Bagaimana performansi dari prediktor sebaran ikan (kg) secara realtime untuk masing-masing ikan?

1.3 Tujuan Penelitian

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

- a. Memeproleh hasil dari algoritma prediktor yang digunakan untuk penentuan jumlah berat dari 3 jenis ikan yang digunakan dalam penelitian.
- b. Menentukan performansi dari prediktor sebaran ikan (kg) secara realtime untuk masing-masing ikan.

1.4 Batasan Masalah

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

- a. Variabel ikan yang menjadi objek penelitian adalah suhu permukaan laut, kecepatan angin, salinitas, curah hujan, kecepatan angin dan arus laut.
- b. Variasi *hidden layer* dibatasi sebanyak 1 lapisan dan *hidden note* sebanyak 10 unit.
- c. Algoritma pelatihan Jaringan Syaraf Tiruan-*Particle Swarm Optimization* adalah Levenberg Marquadt.
- d. Lokasi uji pada kondisi realtime berada di Pantai Kenjeran, Laut Jawa.
- e. Jenis ikan yang diprediksi antara lain ikan biji nangka, kerapu, dan kuniran.

Halaman ini sengaja dikosongkan

BAB II TINJAUAN PUSTAKA

2.1 Gelombang Laut

Menurut data dari Pelabuhan Perikanan Nusantara (PPN) Brondong, terdapat beberapa jenis ikan yang hidup di Laut Jawa. Salah satunya adalah jenis ikan demersal. Ikan demersal adalah ikan yang hidup dan makan di dasar laut. Berikut ini adalah beberapa jenis ikan demersal antara lain ikan biji nangka (*Upheneus Mollucensis Mullidae*), kerapu (*Epinephelus Coiodes Serranidae*), dan kuniran (*Upheneus Tragula Mullidae*).

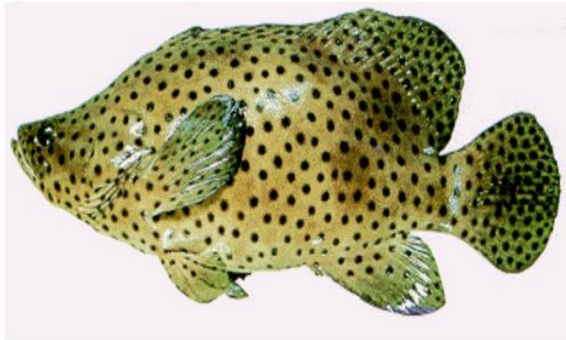


Gambar 2. 1 *Upheneus Mollucensis Mullidae* (Semuaikan, 2018)

Ikan biji nangka yang memiliki nama latin *Upheneus Mollucensis Mullidae* mempunyai ciri-ciri yang mendasar yaitu terdapat “janggut” yang terletak di bawah rahang bawahnya. Ikan biji angka juga memiliki dua sirip yang terletak di punggung nya. Bentuk tubuhnya pipih agak panjang dan terdapat jari-jari sirip keras, lemah, mengeras dan lemah. Bentuk tubuh dari ikan biji nangka dapat dilihat pada Gambar 2.1. Ikan ini tergolong ikan demersal dan hidup di perairan pantai pada kedalaman antara 10-

80 m. Ikan ini biasanya hidup secara bergerombol besar (Oseana, 1999)

Ikan kerapu yang memiliki nama latin *Epinephelus Coiodes Serranidae* biasanya hidup di perairan pantai dan termasuk ikan demersal. Ikan kerapu tersebar luas dari wilayah Asia Pasifik termasuk Laut Merah. Di Indonesia sendiri banyak terdapat di daerah perairan Pulau Jawa, Sulawesi, Sumatera, Ambon, dan Pulau Buru. Umumnya ikan ini mencapai panjang 150 cm, dan umunya antara 50-70 cm (Oseana, 1999). Seluruh permukaan tubuh kerapu berwarna putih keabuan. Memiliki sirip punggung yang semakin ke belakang akan semakin melebar. Bentuk tubuh ikan kerapu dapat dilihat pada Gambar 2.2.



Gambar 2. 2 *Epinephelus Coiodes Serranidae*
(Semuaikan,2018)

Ikan kuniran yang memiliki nama latin *Upheneus Tragula Mullidae* biasanya hidup di perairan pantai sampai kedalaman 40 m. Ikan ini tergolong dalam ikan demersal. Ikan kuniran merupakan jenis ikan yang memiliki bentuk badan sedikit memanjang, pipih samping dengan penampang melintang bagian depan punggung. Ikan kuniran termasuk ke dalam jenis ikan demersal, sebagai ikan yang dapat dikonsumsi. Kebiasaan makanan ikan kuniran adalah 59,49 % jenis udang, 14,51% ikan-ikan kecil, dan juga 13,51% moluska (Oseana, 1999). Lebih jelasnya gambar ikan kuniran dapat dilihat pada Gambar 2.3.



Gambar 2. 3 *Upheneus Tragula Mullidae*
(Semuaikan, 2018)

Untuk mengetahui keberadaan ikan di lingkungan laut, terdapat variabel-variabel yang mempengaruhi kondisi lingkungan laut tersebut. Variabel yang mempengaruhi keberadaan ikan adalah sebagai berikut:

a. Suhu Permukaan Laut

Keberadaan ikan sangat dipengaruhi oleh iklim maritim. Iklim maritim yang berpengaruh terhadap penyebaran ikan adalah permukaan laut. Beberapa variabel permukaan laut, di antaranya adalah suhu permukaan laut, warna laut, *phytoplankton*, *sea state*, *sea ice*, *salinity*, ketinggian permukaan, dan arus laut (National Oceanic and Atmospheric Administration, 2014). Variabel permukaan laut yang sangat berpengaruh terhadap penyebaran ikan adalah suhu permukaan laut (Torres-Faurrieta et al., 2016). Suhu permukaan laut merupakan salah satu parameter fisik fenomena *upwelling* (Balai Penelitian dan Observasi Laut, 2018).

Fenomena *upwelling* dapat menyebabkan mempertemukan larva dengan pemangsa melalui turbulensi (Life, Jolla, & Rotunda, 1990). *Upwelling* merupakan suatu fenomena naiknya aliran air dingin ke permukaan akibat adanya angin yang berhembus di permukaan laut (Life et al., 1990). Larva dan pemangsa ini yang akan mempengaruhi rantai makanan sehingga mengetahui keberadaan beberapa jenis ikan yang berada dalam rantai makan tersebut. Suhu permukaan laut biasanya berkisar antara 26oC – 29oC di daerah tropis dan 15oC – 20oC di daerah subtropis. Faktor-

faktor fisik yang mempengaruhi suhu permukaan laut adalah arus permukaan, keadaan awan, penguapan, gelombang, gerakan konveksi, upwelling, divergensi, pembekuan dan pencairan es di daerah kutub.

b. Kecepatan Angin

Angin adalah udara yang bergerak akibat adanya perbedaan tekanan udara dengan arah aliran angin dari tempat yang memiliki tekanan tinggi ke tempat yang bertekanan rendah atau dari daerah yang memiliki suhu rendah ke wilayah bersuhu tinggi. Selain itu, kecepatan angin juga memberi pengaruh terhadap tinggi gelombang serta kecepatan arus. Besarnya gelombang dapat mempengaruhi operasi penangkapan ikan (Daniel, Santiago, Juan, dan Patricia, 2013). Kecepatan angin juga mempengaruhi adanya arus pada laut. Ikan akan bereaksi secara langsung terhadap perubahan lingkungan yang dipengaruhi oleh arus dengan mengarahkan dirinya secara langsung pada arus.

c. Curah Hujan

Variabel lain yang berpengaruh pada keberadaan ikan adalah curah hujan. Curah hujan tidak memiliki pengaruh langsung terhadap keberadaan ikan, namun curah hujan memiliki pengaruh terhadap kondisi air laut. Hujan yang deras memiliki efek yang berbeda dengan hujan yang ringan. Hal tersebut memiliki pola yang sama terhadap suhu permukaan air laut, hanya saja perbedaannya kecil. Perubahan suhu terjadi pada kedalaman yang besar yang mengakibatkan turbulensi karena pengaruh tetesan air hujan deras, namun perubahan suhu hanya terjadi di sekitar permukaan saja karena lebih banyak hujan dingin bercampur dengan permukaan air laut (Hobday et al., 2016).

2.2 Jaringan Syaraf Tiruan

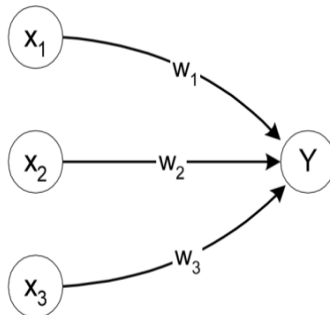
Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* merupakan salah satu metode *Artificial Intelligence* yang banyak digunakan dalam proses prediksi. Konsep dasar dari JST ini seperti jaringan syaraf biologis pada manusia yang banyak terdapat

neuron yang saling terhubung satu sama lain. JST dalam proses pengenalan pola sinyal melalui proses pembelajaran atau *training*. Sistem pembelajaran atau *training* ini melibatkan penyesuaian koneksi antar *neuron* (Shaban dkk, 2016).

JST ini memiliki karakteristik seperti kemampuan beradaptasi, mengorganikan diri sendiri, dan pembelajar serta *non-linearity*, *non-locality*, *non-steady*, dan *non-convex* sehingga menjadikan metode ini dapat mengatasi masalah yang kompleks (Yang, Chen, Wang, Li, & Li, 2016).

JST dapat digunakan untuk keperluan pemrosesan sinyal, pengendalian, pengenalan pola, pengobatan, penghasil suara, pengenalan suara, dan bisnis. JST ditentukan oleh tiga hal, yaitu (Fausett, 1994):

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



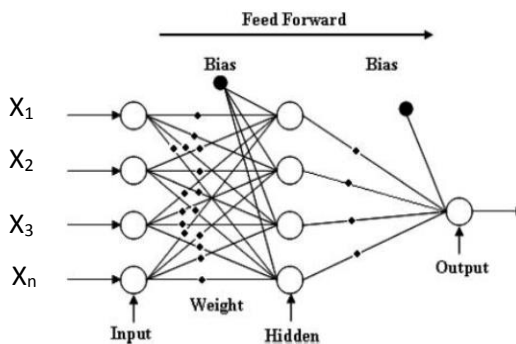
Gambar 2. 4 Hubungan Antar Neuron dalam Arsitektur JST (Fausset, 1994)

Ketiga aspek di atas dapat dilihat pada Gambar 2.2 dimana contoh persamaan *neuron* Y adalah Persamaan (2.1). 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 $y = f(\text{net})$. Sinyal akan meneruskan apabila fungsi aktivasi kuat. Dari fungsi aktivasi juga dapat merubah bobot yang terbaik.

$$net = x_1w_1 + x_2w_2 + x_3w_3 \quad (2.1)$$

Metode JST yang mengacu pada model dari jaringan syaraf pada makhluk biologis dalam mencerna informasi. Sistem ini kemudian digunakan oleh organisme untuk mengolah data yang didapatkan dari lingkungan untuk mengambil keputusan lebih lanjut. Sehingga 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).

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



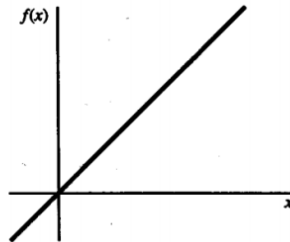
Gambar 2. 5 Jaringan Tipe *Feed Forward* (Jain, Deo, & Rajendran, 2011)

Terdapat beberapa macam fungsi aktivasi diantaranya *sigmoid biner*, *sigmoid bipolar*, dan linier. Fungsi aktivasi linier memiliki keluaran yang sama terhadap masukannya. Persamaan fungsi aktivasi linier dan turunannya dapat dilihat pada persamaan (2.2) dan (2.3) dengan grafik pada Gambar 2.4 (Fausett, 2005).

$$f(x) = x \quad (2.2)$$

$$f'(x) = 1 \quad (2.3)$$

Fungsi aktivasi *sigmoid bipolar* memiliki kisaran -1 hingga 1. Persamaan fungsi *sigmoid bipolar* dapat dilihat pada persamaan (2.4) dengan turunan persamaan (2.5). Grafik *sigmoid bipolar* ada pada Gambar 2.5 (Fausett, 1994).

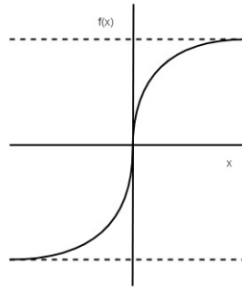


Gambar 2. 6 Fungsi Aktivasi Linier (Fausset, 1994)

$$f(x) = \frac{2}{1+e^{-\sigma t}} - 1 \quad (2.4)$$

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)] \quad (2.5)$$

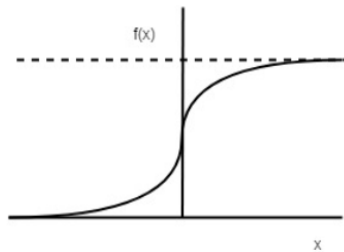
Fungsi aktivasi *sigmoid biner* bertindak sebagai fungsi yang mengompres fungsi *input* saat dibutuhkan nilai positif yang besar atau nilai negatif yang besar. Nilai positif yang besar secara asimtotik mendekati satu, sementara nilai negatif yang besar terkompres ke nol (Fausset, 1994). Fungsi aktivasi *sigmoid biner* diberikan oleh persamaan (2.6) dan (2.7) dan grafik fungsi dapat dilihat pada Gambar 2. 6.



Gambar 2. 7 Fungsi Aktivasi *Sigmoid Bipolar* (Fausset, 1994)

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

$$f'(x) = f(x)(1 - f(x)) \quad (2.7)$$



Gambar 2. 8 Fungsi Aktivasi *Sigmoid Biner* (Fausset, 1994)

2.3 Algoritma Pelatihan *Levenberg Marquardt*

Algoritma pelatihan atau *training* yang dapat digunakan untuk melakukan pelatihan Jaringan Syaraf Tiruan 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)

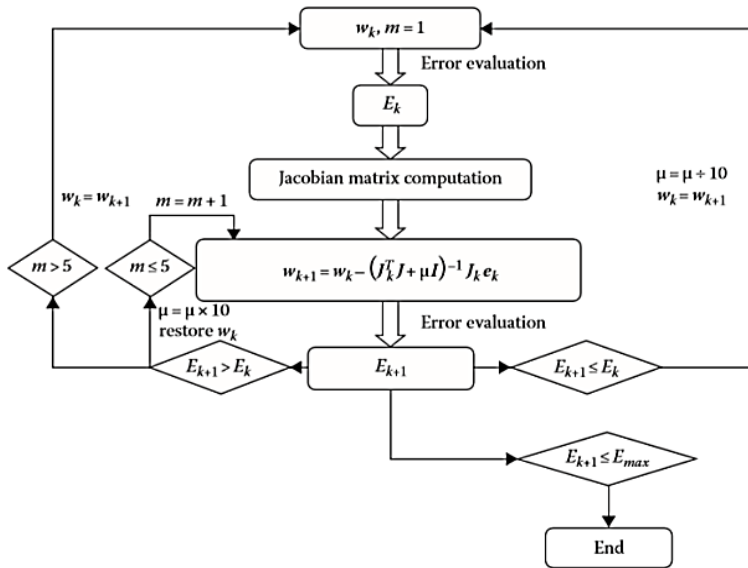
dianggap sebagai gabungan dari *Steepest Descent* dan *Gauss-Newton*.

Algoritma *Steepest Descent* yang secara luas dipakai dianggap kurang efisien karena lambatnya untuk mencapai konvergensi. Hal ini dikarenakan perlunya ukuran langkah yang tepat untuk gradien menuju konvergen, 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. Sehingga algoritma 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* (Eren, Yaqub, & Eyüpoğlu, 2016).

Algoritma pelatihan LM merupakan pengembangan dari algoritma *backpropagation* standar. Pada penggunaan algoritma *backpropagation*, 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.

Sehingga ketika μ sangat kecil (mendekati nol) maka pembaruan bobot mendekati algoritma *Gauss Newton*. Sedangkan ketika μ sangat besar, maka pembaruan bobot mendekati algoritma *Steepest Descent*. Serta apabila μ sangat besar dapat dianggap sebagai koefisien pembelajaran pada metode *Steepest Descent*.

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 Levenberg Marquardt proses *backpropagation* harus diulang. Perhitungan matriks *Jacobian*, *input node* y dapat dihitung dalam *forward computation*. Sementara parameter *error* δ diperoleh dalam perhitungan mundur atau *backward computation* (Yu & Wilamowski, 2011).



Gambar 2. 9 Diagram Blok Algoritma Levenberg Marquardt (Yu & Wilamowski, 2011)

$$H \approx J^T J + \mu I \quad (2.8)$$

Di mana,

μ adalah koefisien kombinasi (selalu bernilai positif),

I adalah matriks identitas,

J adalah matriks Jacobian.

$$\alpha = \frac{1}{\mu} \quad (2.9)$$

Sehingga persamaan pembaruan bobot untuk algoritma LM dapat dilihat pada Persamaan 2.10

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \quad (2.10)$$

2.4 Optimasi *Particle Swarm Optimization* (PSO)

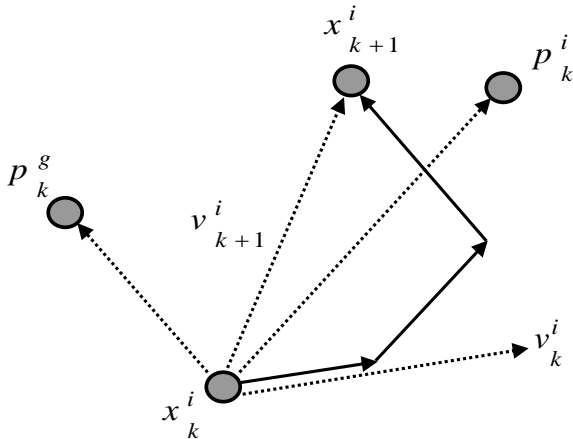
Algoritma *Particle Swarm Optimization* (PSO) meniru perilaku sosial organisme seperti kawanan burung atau ikan. Algoritma ini pertama kali diusulkan oleh Kennedy dan Eberhart pada tahun 1995 (Rao, 2009). Pada (Hassan, Cohanim, & Weck, 2004) metode optimasi PSO menghasilkan solusi yang berkualitas tinggi sama dengan metode optimasi Genetik Algoritma (GA). Secara komputasional bahwa PSO untuk mencapai solusi yang berkualitas tinggi memerlukan upaya yang lebih kecil daripada GA. *Particle Swarm Optimization* memiliki hubungan dengan kehidupan buatan secara umum, sekawanan burung, dan khususnya teori berkerumun. Juga berkaitan dengan perhitungan evolusioner dan memiliki kaitan baik pada algoritma genetika dan pemrograman evolusioner (Kennedy & Eberhart, 1995). PSO merupakan algoritma yang sederhana yang dapat digunakan untuk menyelesaikan permasalahan optimasi pada berbagai fungsi pada range yang luas. Ilustrasi perpindahan partikel yang dipengaruhi oleh kecepatan berdasarkan inersia, kognitif, dan sosial pada *Particle Swarm Optimization* diberikan pada Gambar 2. 10.

Particle Swarm Optimization (PSO) menirukan perilaku burung dalam kawanan mengikuti aturan sebagai berikut:

- a. Mencoba untuk tidak terlalu dekat dengan burung lain.
- b. Mengarah ke arah rata-rata burung lain.
- c. Mencoba untuk menyesuaikan "posisi rata-rata" antara burung-burung lain tanpa celah lebakawanan.

Perilaku kawanan atau kumpulan didasarkan pada kombinasi dari 3 faktor yaitu kohesi, pemisahan, dan alignment. Kemudian PSO dikembangkan berdasarkan pada model berikut:

- a. Ketika seekor burung menempatkan target maka otomatis mengirimkan informasi ke semua burung lainnya.
- b. Semua burung lainnya tertarik pada target tetapi tidak secara langsung.
- c. Ada komponen dari pemikiran independen masing-masing burung memiliki memori masa lalu.



Gambar 2. 10 Ilustrasi pembaruan posisi dan kecepatan pada partikel (Pitono, 2015).

Dengan demikian, secara bertahap semakin banyak iterasi, burung-burung terbang semakin mendekati target (yaitu fungsi maksimum atau fungsi obyektif) (Rao, 2009).

Skema algoritma PSO sebagai berikut (Marini & Walczak, 2015):

Langkah 1

Inisialisasi setiap jumlah n partikel

1. Inisialisasi posisi awal sekaligus kecepatan awalnya secara random.
2. Dihitung fitness untuk setiap partikel dan inisialisasi global best.
3. Untuk tiap partikel, Pbest awal akan sama dengan posisi awal.

Langkah 2

Diulangi langkah-langkah berikut hingga kriteria stop

1. Update kecepatan partikel
2. Update posisi partikel
3. Evaluasi fitness untuk tiap partikel

4. Update personal best
5. Update global best

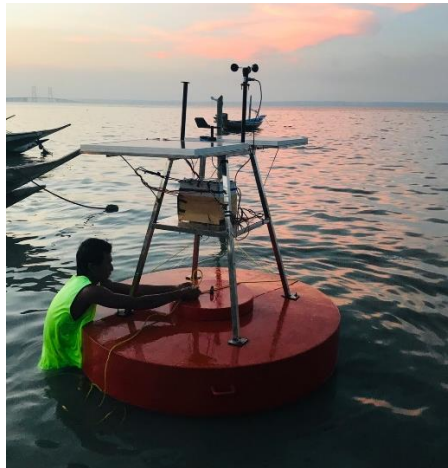
Langkah 3 Pada akhir proses iterasi, solusi terbaik direpresentasikan oleh global best

Persamaan update posisi dan kecepatan diberikan pada persamaan 2.13 dan 2.14

$$xi(t+1)=xi(t)+vi(t+1) \quad (2.13)$$

$$vi(t+1)=w*vi(t)+c1(x^L-xi(t))R1+c2(x^G-xi(t))R2 \quad (2.14).$$

2.5 *Buoyweather Station Type II*



Gambar 2. 10 *Buoyweather Station Type II*

Stasiun *buoyweather* yaitu pelampung tetap atau yang terapung di laut yang 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 kelembapan udara. *Buoyweather Station Type II* dapat dilihat pada Gambar 2.11.

2.6 Validasi Prediktor

Suatu proses/metode dapat memberikan hasil yang konsisten sesuai dengan spesifikasi yang telah ditetapkan dan terdokumentasi dengan baik. Hingga pada akhirnya setelah dilakukan simulasi maka akan dilakukan pula proses analisa yang didalamnya terdapat evaluasi kerja dari sistem prediktor ini. Untuk mengukur performansi jaringan syaraf tiruan dapat dilakukan menggunakan dua parameter yaitu *Root Mean Square Error* (RMSE), *Mean Absolute Percentage Error* (MAPE). Untuk menentukan nilai RMSE dapat dilihat pada persamaan 2.13.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x})^2} \quad (2.13)$$

Keakuratan metode estimasi kesalahan pengukuran diindikasikan dengan adanya RMSE yang kecil. Metode estimasi yang mempunyai RMSE lebih kecil dikatakan lebih akurat daripada metode estimasi yang mempunyai RMSE lebih besar (Kolen & Brennan, 1995).

Untuk menentukan nilai parameter MAPE (*Mean Absolute Percentage Error*) digunakan rumus pada persamaan 2.14

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}}{x_i} \right| \quad (2.14)$$

Dimana x_i adalah hasil keluaran dari prediktor, \hat{x} adalah target yang dihasilkan, n adalah jumlah data yang digunakan dan i adalah jumlah keluaran yang diinginkan (Santosh, Gopika, Ghosh, & Fernandes, 2018).

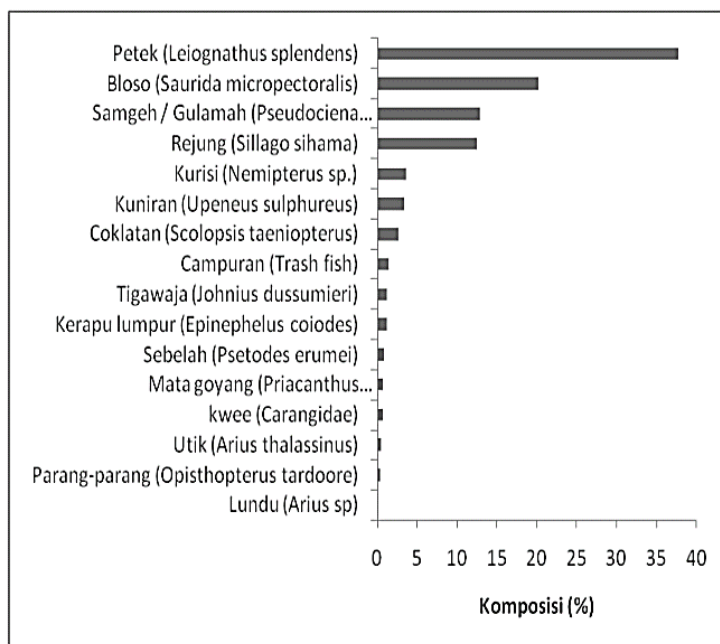
Klasifikasi akurasi asuatu prediktor yang telah dikembangkan oleh Lewis dengan perhitungan MAPE dapat ditunjukkan seperti pada Tabel 2.1.

Tabel 2. 1 Skala Penentuan Nilai Akurasi Suatu Prediktor

MAPE	Nilai Akurasi
$\leq 10\%$	<i>Highly accurate</i>
11% - 20%	<i>Good Forecast</i>
21% - 50%	<i>Reasonable forecast</i>
$\geq 51\%$	<i>Inacurate forecast</i>

2.7 Sebaran Ikan di Area WPP-712

Secara geografis penyebaran ikan demersal merata di pantai utara Jawa mulai perairan Tangerang, Karawng, Indramayu, Pamanukan hingga Cirebon, Tegal, Demak, Pati, Rembang hingga sebelah utara Madura.



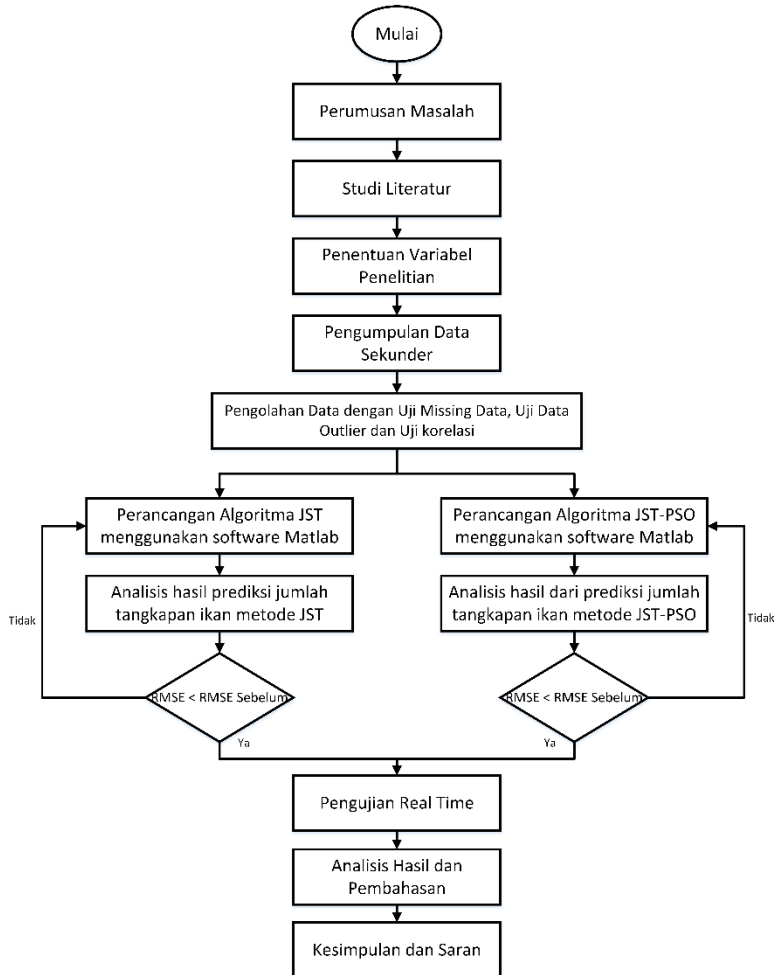
Gambar 2. 12 Komposisi Jenis Ikan Demersal di WPP 712 Laut Jawa Tahun 2012

Komposisi jenis ikan demersal terakhir diperoleh melalui survei Balai Penelitian Perikanan Laut tahun 2012 seperti tampak pada Gambar 2.12. Dapat dilihat bahwa hasil tangkapan ikan demersal didominasi oleh famili *Leognathidae* (peperék), *Nemipteridae* (kurisi) dan *Mullidae* (kuniran). Kelompok *Ariidae* (manyung), *Prichanthidae* (swangi) dan *Lutjanidae* (kakap) telah mengalami penurunan dengan komposisi kurang dari 2%.

Sementara itu perikanan demersal di perairan selatan Kalimantan yang berbasis di PPI Muara Kintap banyak ditangkap dengan lampara dasar. Komposisi hasil tangkapannya didominasi oleh ikan petek/peperék (*Leiognathus sp*), manyung (*Arius sp*) dan kurisi (*Nemipterus sp*). Komposisi hasil tangkapan ketiga jenis ikan ini juga berfluktuasi berdasarkan waktu dimana ikan manyung menjadi tangkapan dominan pada bulan Januari dan November sedangkan petek dominan pada bulan Oktober. Dominasi ikan petek menunjukkan bahwa kapal lampara dasar yang berbasis di PPI Kintap daerah operasinya tidak jauh dari pantai.

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 yang ada. Rumusan masalah ini didapatkan dari kajian beberapa penelitian yang didapatkan sebelumnya. Penelitian mengenai variabel yang menentukan kondisi lingkungan laut telah dijelaskan oleh Daniel Osvaldo Bruno. Pada penelitian tersebut dijelaskan bahwa variabel yang mempengaruhi kondisi lingkungan laut antara lain curah hujan, suhu permukaan laut, salinitas, kedalaman, kecepatan angin, dan arah angin.

Berdasarkan hasil penelitian tersebut di atas dilakukan strategi untuk mengetahui performansi atau tingkat keakuratan dari prediktor jumlah tangkapan ikan dengan menggunakan metode Jaringan Syaraf Tiruan. Penelitian ini membandingkan nilai RMSE dari JST algoritma pelatihan *Levenberg Marquardt* dengan JST yang dioptimasi *Particle Swarm Optimization* (PSO). Penelitian ini diharapkan dapat menghasilkan nilai RMSE yang lebih kecil daripada penelitian sebelumnya.

3.2 Studi Literatur

Studi literatur dilakukan untuk mengembangkan ide agar lebih sistematis melalui referensi yang ada untuk dapat memahami prinsip dasar dari metode yang dibuat agar tidak salah konsep. Studi literatur pada penelitian ini adalah mencari variabel-variabel independen yang berpengaruh pada variabel dependen. Studi literatur yang relevan dengan tema penelitian yaitu hasil penelitian terdahulu dengan tema yang sama, teorema mengenai JST, *simulink* Matlab yang diperoleh dari jurnal dan *handbook*.

3.3 Penentuan Variabel Penelitian

Tahap ini merupakan tahap untuk menentukan variabel yang berkorelasi dengan keluaran dan masukan prediktor. Variabel keluaran prediktor adalah jumlah tangkapan ikan. Jumlah tangkapan ikan berpengaruh terhadap kondisi lingkungan laut tempat ikan tersebut hidup.

Variabel masukan prediktor ditentukan terlebih dahulu, yaitu berupa variabel cuaca maritim. Dicari beberapa variabel yang memengaruhi variabel cuaca maritim tersebut. Variabel ini ditentukan berdasarkan studi literatur yang dilakukan, yaitu suhu permukaan laut, salinitas, curah hujan, kecepatan angin, dan arus laut. Data *output* yang tersedia diambil dari PPN Brondong adalah berupa berat data ikan demersal. Ikan demersal yang dipilih di antaranya ikan biji nangka, kerapu dan kuniran. Selain itu, penentuan variabel penelitian juga didasarkan pada nilai korelasi antar variabel *input* dan *ouput* yang digunakan dalam penelitian ini.

3.4 Pengumpulan Data

Pengumpulan data dilakukan melalui dua cara yaitu pengambilan data sekunder dari referensi terpercaya dan pengambilan data *realtime*. Data sekunder dicari sebagai data referensi untuk memodelkan prediktor. Data yang diambil dalam jangka waktu 12 bulan pada wilayah laut Indonesia. Variabel independen (suhu permukaan laut, kecepatan angin, arus laut, salinitas dan curah hujan). Data *buoy* yang diambil ada dua, yaitu *Tropical Atmosphere Oceanic* (TAO) dengan koordinat 8N; 137E dan *Research Moored Array for African –Asian-Australian Monsoon Analysis and Prediction* (RAMA) dengan koordinat 8S; 80,5E. Variabel independen yang berasal dari *buoy* NOAA diinterpolasi dengan persamaan (3.1) untuk mendapatkan perkiraan nilai variabel di suatu lokasi (Arifin, Aisjah, & Hakim, 2011).

$$T(A) = \frac{A-A_2}{A_0-A_1} T_0 + \frac{A-A_0}{A-A_0} T_1 \quad (3.1)$$

Hasil interpolasi memiliki nilai yang cukup besar apabila dimasukkan ke dalam *software* Matlab. Nilai variabel perlu dinormalisasi dengan rentang 0 sampai 1 agar memori tidak terlalu besar dan proses iterasinya semakin cepat. Persamaan normalisasi dapat dilihat pada persamaan 3.2 (Siang, 2005).

$$X' = \frac{0.8(X-b)}{a-b} \quad (3.2)$$

Variabel dependen beberapa jenis ikan diambil dari Pelabuhan Perikanan Nusantara (PPN) Brondong dengan koordinat 6S; 112E. Data yang diambil sebanyak 12 bulan atau 365 hari pada tahun 2018.

3.5 Pengolahan Data

Data sekunder yang telah dikumpulkan (data independen dan dependen), dilakukan uji dan analisis data cuaca. Hal ini dilakukan untuk menemukan variabel cuaca apa saja yang tepat sebagai variabel masukan untuk melakukan prediksi yang akan dilakukan. Dilakukan uji *missing data*, *outlier*, dan korelasi. Uji *missing* dan uji *outlier* digunakan untuk memperbaiki data yang cacat. Sedangkan uji korelasi digunakan untuk menentukan variabel masukan yang akan digunakan dalam penelitian. Besarnya korelasi ditentukan oleh persamaan 3.4. Pengujian data dibantu dengan software *IBM SPSS (Statistical Package for Social Science) Statistics 22*. Penjelasan mengenai uji data tersebut dijelaskan pada sub bab di bawah ini.

a. Uji *Missing Data*

Missing data merupakan suatu informasi yang tidak tersedia karena alasan tertentu. Hal ini disebabkan karena informasi untuk suatu objek tidak diberikan, sulit dicari, dan memang informasi tersebut tidak ada. *Missing data* tidak akan bermasalah jika jumlahnya hanya sedikit. Misal jika hanya sekitar 1% dari total data. Namun, jika persentasenya cukup besar, maka data tidak diproses lebih lanjut. Batasan persentase yang dipakai pada penelitian ini adalah 1% (Santoso, 2010).

Pengujian dilakukan dengan mencari *cell* mana yang tidak adanya atau kosong. Pencarian data yang kosong dibantu dengan software *IBM SPSS 22* secara otomatis. Langkah yang harus dilakukan adalah klik *Analyze – Descriptive Statistics – Frequencies* – semua variabel dipilih – OK.

b. Uji *Outlier*

Data *outlier* merupakan data yang benar-benar berbeda dari data-data pada umumnya. Data *outlier* bisa terjadi karena ada beberapa sebab, 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 dapat dilihat pada persamaan (3.3). Data dapat dikatakan *outlier* apabila nilai z lebih besar dari angka +2,5 atau lebih kecil dari angka -2,5 (Santoso, 2010).

$$z = \frac{x - \bar{x}}{\sigma} \quad (3.3)$$

c. 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. Arti tanda negatif merupakan korelasi antar variabel berlawanan. Sedangkan tanda positif merupakan korelasi antar variabel searah. Metodologi yang digunakan adalah *Pearson Correlation*. Persamaan *Pearson Correlation* dapat dilihat pada persamaan (3.4) (Zhou, Deng, Xia, & Fu, 2016). Data yang diuji korelasi adalah antar variabel independen dan dependen. Pengujian hanya bisa dilakukan untuk mencari hubungan dua variabel. Data independen suhu permukaan laut di uji korelasi dengan ikan biji nangka, ikan kerapu, dan ikan kuniran. Data independen yang lain juga dilakukan pada masing-masing ikan.

$$r = \frac{\sum(x_i - \bar{x}) \sum(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} \quad (3.4)$$

Tabel 3. 1 Nilai Interval Korelasi Terhadap Tingkat hubungannya (Sugiyono, 2007)

Interval Koefisien	Tingkat Hubungan
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. Namun 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%. Apabila nilai signifikansi di bawah 0,05 maka variabel tersebut dikatakan tidak berkorelasi atau memiliki hubungan antar variabel yang rendah sehingga tidak layak untuk digunakan sebagai variabel *input*.

3.6 Perancangan Algoritma Prediktor



Gambar 3. 2 Diagram Blok Sistem Prediktor

Rancangan prediktor ditunjukkan pada blok diagram Gambar 3.2. Perancangan algoritma prediktor sebaran ikan menggunakan metode Jaringan Syaraf Tiruan (JST). Masukan dari prediktor berupa variabel suhu permukaan laut, kecepatan angin, dan curah hujan. Keluaran dari sistem prediktor berupa jumlah tangkapan ikan dalam kilogram. Sistem prediktor menggunakan metode JST terdiri dari dua tahapan yaitu tahap pelatihan atau *training* dan tahap validasi. Pada Tabel merupakan pembagian data pelatihan dan validasi yaitu sebesar 80% data untuk pelatihan dan 20% data untuk validasi dari total data yang diperoleh. Sehingga didapatkan bahwa data yang digunakan untuk proses pelatihan JST yaitu sebanyak 292 data dan data yang digunakan untuk proses validasi

JST yaitu sebanyak 73 data. Data mengenai jumlah tangkapan ikan didapatkan dari PPN Brondong sesuai dengan letak koordinat data yang akan diambil. Data yang diambil memiliki rentang waktu dari bulan Januari hingga Desember 2018.

Tabel 3. 2 Pembagian Data Pelatihan dan Validasi

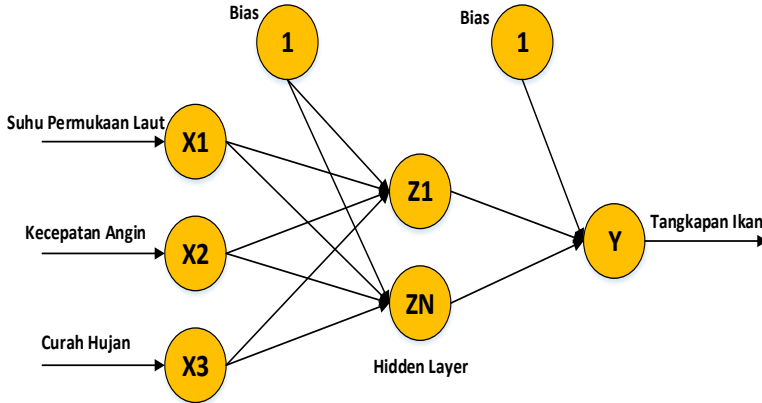
Jenis Data	Rentang Data	Jumlah Data
Data Pelatihan	Januari 2018-September 2018	292
Data Validasi	Oktober 2018-Desember 2018	73

a. Perancangan Algoritma JST-LM

Pelatihan Jaringan Syaraf Tiruan (JST) menggunakan arsitektur satu *input layer*, satu *hidden layer*, dan satu *output layer*. Algoritma JST yang digunakan yaitu algoritma *Lavenberg Marquardt* (LM). *Hidden layer* yang digunakan pada penelitian ini adalah 1 dengan variasi *hidden node* 1-10. Selain *hidden layer* dan *learning rate*, berikut ini adalah beberapa parameter yang digunakan pembuatan algoritma JST seperti pada Tabel 3.3.

Variabel kecepatan angin, curah hujan, dan suhu permukaan laut dijadikan sebagai unit *input* pada *input layer*. Jumlah tangkapan ikan merupakan unit output pada *output layer*. Pelatihan Jaringan Syaraf Tiruan dengan algoritma *pelatihan Levenberg-Marquardt* (LM) dilakukan dengan arsitektur 3-n-1. Arsitektur jaringan syaraf tiruan dapat dilihat pada Gambar 3.3. Notasi X1 adalah variabel kecepatan angin, X2 adalah variabel curah hujan, dan X3 adalah variabel suhu permukaan laut. Pada gambar tersebut *hidden layer* terdiri dari 10 *hidden node* yaitu Z1 – Z10. Keluaran hasil prediksi jaringan ditunjukkan oleh unit *output* Y. Perancangan prediktor telah dilakukan melalui 10 variasi arsitektur untuk tiap-tiap *hidden node* untuk masing-masing ikan. Dari pelatihan yang telah dilakukan dipilih arsitektur Jaringan Syaraf Tiruan yang memiliki nilai *Root Mean Square Error* (RMSE)

terkecil dan paling cocok untuk digunakan sebagai prediksi jumlah tangkapan ikan. Di mana n adalah jumlah *hidden* yang divariasikan.

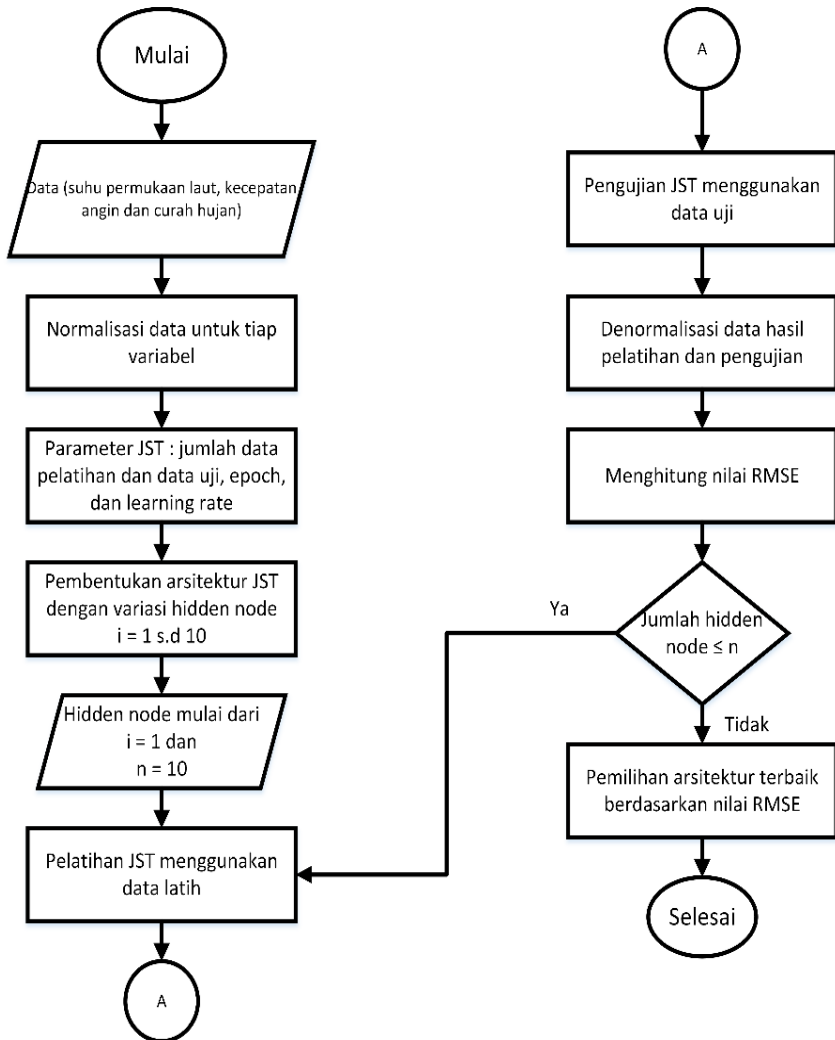


Gambar 3. 3 Arsitektur Jaringan Syaraf Tiruan 3-n-1

Parameter yang digunakan dalam pembuatan algoritma JST-LM dapat dilihat pada Tabel 3.3. Proses *training* dilakukan selama 1000 *epoch* dengan *learning rate* sebesar 0,01 dengan fungsi aktivasi tansig dari *input* ke *hidden* dan *purelin* dari *hidden* ke *output*. Proses algoritma JST secara lengkapnya dapat dilihat diagram alir pada Gambar 3.2.

Tabel 3. 3 Parameter Algoritma Jaringan Syaraf Tiruan

No	Parameter	Nilai
1	Jumlah data pelatihan (%)	80
2	Jumlah data validasi (%)	20
3	<i>Epoch</i>	1000
4	<i>Learning rate</i>	0,01
5	Momentum constant	0,9
6	Fungsi Aktivasi <i>Input</i> ke <i>Hidden</i>	Tansig
7	Fungsi Aktifasi <i>Hidden</i> ke <i>Output</i>	Purelin



Gambar 3. 4 Diagram Alir Algoritma Jaringan Syaraf Tiruan

b. Perancangan Algoritma JST-PSO

Pelatihan Jaringan Syaraf Tiruan (JST) menggunakan arsitektur satu *input* layer, satu *hidden layer*, dan satu *output* layer. Algoritma JST dioptimasi menggunakan *Particle Swarm Optimization* (PSO) untuk menemukan nilai bobot optimum pada pelatihan. Variabel kecepatan angin, curah hujan dan suhu permukaan laut sebagai unit *input* pada *input* layer. Jumlah tangkapan ikan merupakan unit output pada *output* layer.

Pelatihan Jaringan Syaraf Tiruan yang dioptimasi menggunakan *Particle Swarm Optimization* untuk menentukan nilai bobot optimalnya. Arsitektur yang digunakan yaitu 3-m-1 di mana m adalah jumlah *hidden node*. Jaringan terdiri dari satu *input* layer yang terdiri dari 3 unit *input*, satu *hidden layer* yang terdiri dari *hidden node* dengan variasi jumlah *hidden node* 1-10, dan satu *output* layer yang terdiri dari 1 unit *output* jumlah tangkapan ikan.

Dalam pelatihan Jaringan Syaraf Tiruan – *Particle Swarm Optimization* dilakukan dalam dua tahap, yaitu penentuan fungsi objektif dan optimasi berdasarkan fungsi objektif tersebut. Fungsi objektif berupa nilai *Root Mean Square Error* (RMSE) didapatkan dari perhitungan algoritma jaringan syaraf tiruan LM dengan variabel yang dioptimasi adalah nilai bobot dari masing-masing koneksi jaringan . Optimasi yang dilakukan yaitu untuk mendapatkan bobot optimal sebagai variabel perhitungan algoritma untuk mendapatkan nilai RMSE minimum. Secara ringkas, diagram alir algoritma pelatihan JST-PSO disajikan pada Gambar 3.3

Menurut Chen & Shih posisi dari tiap partikel dianggap sebagai calon solusi (*candidate solution*) bagi suatu masalah optimasi. Tiap-tiap partikel nantinya akan diberi suatu fungsi *fitness* merancang sesuai dengan menunjuk masalah yang bersesuaian. Dengan keuntungan dari aplikasi yang mudah dan sederhana, lebih sedikit parameter yang dibutuhkan dan mendapatkan hasil yang lebih baik.

Dalam merancang sebuah prediktor JST-PSO dibutuhkan yang namanya parameter yang tentunya berpengaruh terhadap performansi prediktor yang dihasilkan. Parameter yang dibutuhkan

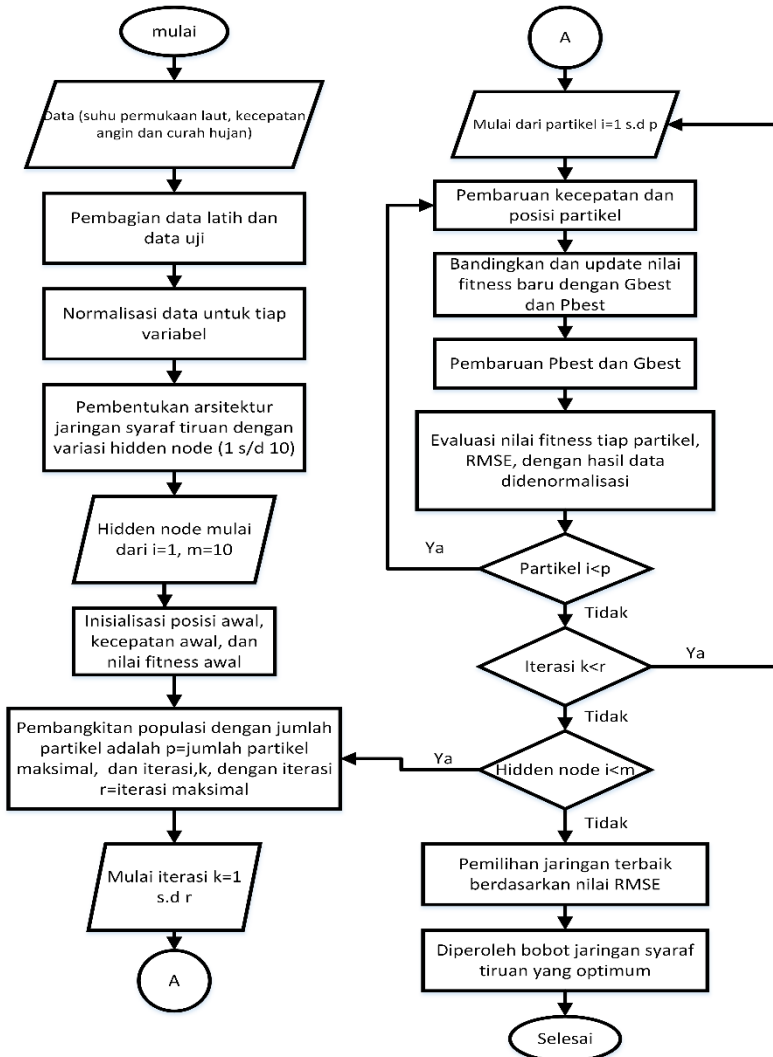
dalam optimasi PSO antara lain jumlah partikel, *epoch* maksimum, bobot inersia, koefisien percepatan kognitif (c_1) dan koefisien percepatan sosial (c_2). Nilai dari parameter yang digunakan dapat dilihat pada Tabel 3.4.

Penentuan angka untuk masing-masing parameter tersebut didasarkan dari literatur mengenai *Particle Swarm Optimization* (PSO). Jumlah partikel dalam kawanan PSO memiliki jumlah bebas tidak ditentukan. Biasanya terdapat antara rentang 20-40 partikel. Semakin besar jumlah partikel proses jalannya algoritma semakin lambat dan jika semakin sedikit jumlah partikel akan merusak kawanan dalam PSO tersebut. Jalannya partikel pada PSO bersifat acak tidak menentu. Oleh karena itu, perlu adanya bobot inersia yang digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan oleh suatu partikel. Biasanya nilai bobot inersia berkisar antar 0,4 – 0,9.

Koefisien percepatan pada algoritma PSO terdiri dari dua koefisien, yaitu koefisien percepatan kognitif (c_1) dan koefisien percepatan sosial (c_2). Koefisien percepatan kognitif (c_1) menunjukkan seberapa nyaman partikel tersebut terhadap dirinya sendiri. Sedangkan, koefisien percepatan sosial (c_2) menunjukkan seberapa nyaman partikel tersebut terhadap tetangganya dalam suatu kawanan. Kedua koefisien ini saling berhubungan satu sama lain dalam menentukan performa yang akan dihasilkan dari algoritma tersebut. Biasanya nilai dari koefisien percepatan ini memiliki rentang antara 0-4.

Tabel 3. 4 Parameter Algoritma Jaringan Syaraf Tiruan - *Particle Swarm Optimization*

No	Parameter	Nilai
1	Jumlah partikel	25
2	<i>Epoch</i> maksimum	30
3	Bobot inersia	0,9
5	Koefisien percepatan kognitif	1,5
6	Koefisien percepatan sosial	2

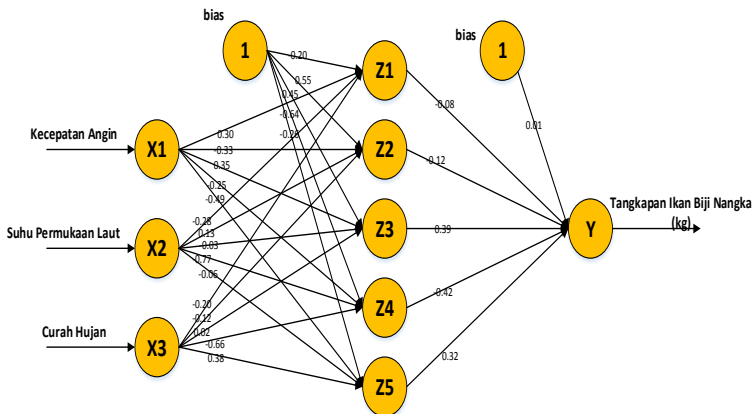


Gambar 3. 5 Diagram Alir Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

3.7 Pemilihan Arsitektur Terbaik Pada Prediktor Sebaran Ikan

Berdasarkan proses pelatihan dengan menggunakan metode JST-LM yang telah dilakukan, didapatkan hasil bahwa prediktor sebaran ikan biji nangka menghasilkan RMSE terkecil pada arsitektur 3-1-1 sebesar 9664,6 kg. Pada ikan kerapu juga didapatkan nilai RMSE terbaik pada arsitektur 3-3-1 dengan nilai sebesar 1071,8 kg. Sedangkan prediktor sebaran ikan kuniran didapatkan nilai RMSE terkecil pada arsitektur 3-2-1 dengan nilai sebesar 8623,6 kg.

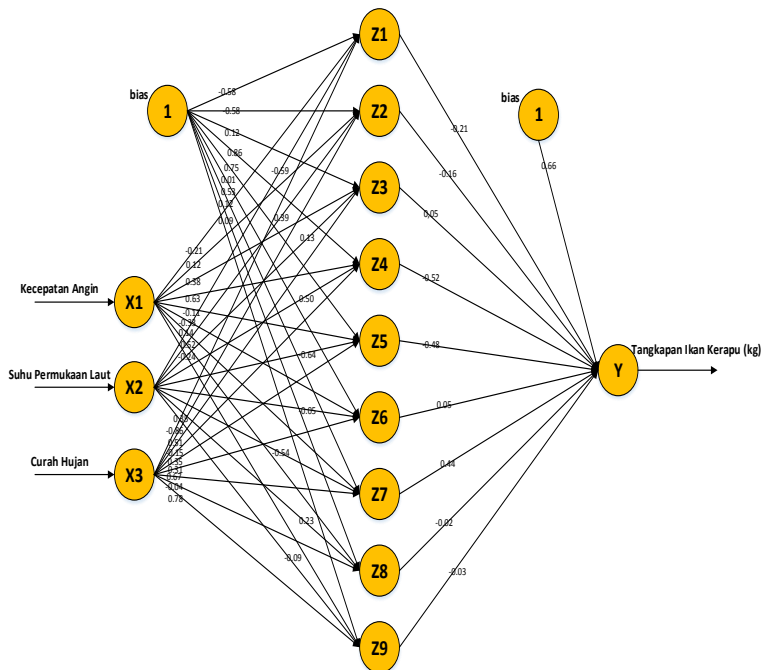
Setelah didapatkan arsitektur terbaik untuk masing-masing ikan dengan metode JST-LM, didapatkan pula arsitektur terbaik untuk metode JST-PSO. Pada metode JST-PSO arsitektur terbaik untuk ikan biji nangka adalah arsitektur 3-5-1 dengan nilai RMSE sebesar 9638,6 kg. Pada ikan kerapu arsitektur terbaik terdapat pada arsitektur 3-9-1 dengan nilai RMSE sebesar 843,1 kg. Pada ikan kuniran arsitektur terbaik terdapat pada arsitektur 3-10-1 dengan nilai RMSE sebesar 8501,7 kg. Jika dilihat perubahan RMSE antara metode JST-LM dan JST-PSO didapatkan nilai RMSE yang menurun. Hal ini menunjukkan bahwa performa JST-PSO lebih baik dibandingkan dengan metode JST-LM.



Gambar 3. 6 Arsitektur Jaringan Syaraf Tiruan-*Particle Swarm Optimization* Ikan Biji Nangka 3-5-1

Arsitektur JST-PSO masing-masing ikan biji nangka, ikan kerapu dan ikan kuniran dapat dilihat pada Gambar 3.6, Gambar 3.7 dan Gambar 3.8. Arsitektur prediksi ikan biji nangka memiliki *hidden layer* 1 lapisan dengan *hidden node* 5 unit. *Input* diberikan ada tiga variabel independen, yaitu kecepatan angin (X1), curah hujan (X2), dan suhu permukaan laut (X3).

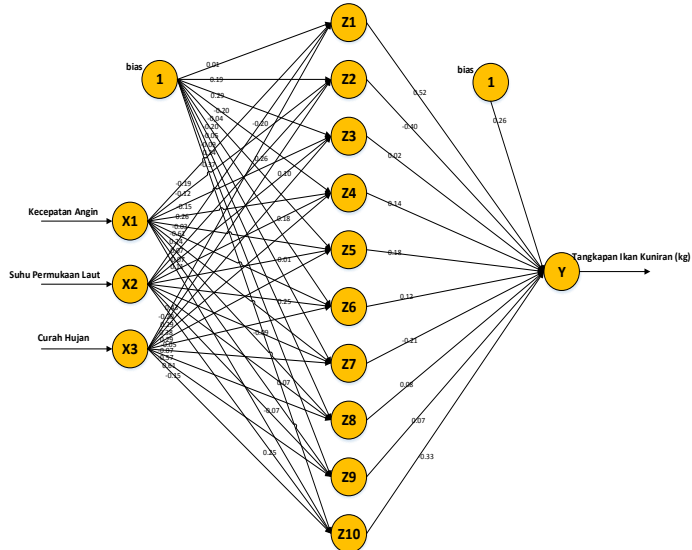
Arsitektur terbaik ikan biji nangka metode JST-PSO dapat dilihat seperti pada Gambar 3.4. Pada arsitektur tersebut menunjukkan jumlah *hidden node* yang digunakan untuk ikan biji nangka adalah sebanyak 5 *hidden node*. Pada arsitektur 3-5-1 ikan biji nangka didapatkan nilai RMSE terendah dibandingkan *hidden node* yang lainnya, yaitu sebesar 9638,6 kg.



Gambar 3. 7 Arsitektur Jaringan Syaraf Tiruan - *Particle Swarm Optimization* Ikan Kerapu 3-9-1

Arsitektur terbaik pada ikan kerapu memiliki *input* yang sama seperti ikan biji nangka. Arsitektur ikan kerapu terbaik metode JST-PSO ditunjukkan seperti pada Gambar 3.5. Arsitektur tersebut menunjukkan jumlah *input* yang sama dengan ikan biji nangka. Pada arsitektur ini memiliki satu jumlah *output* yang menunjukkan jumlah tangkapan ikan kerapu dalam satuan kilogram. Jumlah *hidden node* yang digunakan pada arsitektur ini adalah sebesar 9 buah *hidden node*. Pada arsitektur 3-9-1 ikan kerapu didapatkan nilai RMSE terbaik dibandingkan *hidden node* yang lainnya, yaitu sebesar 843,1 kg.

Arsitektur terbaik ikan kuniran metode JST-PSO terdapat pada arsitektur 3-10-1 yang dapat ditunjukkan pada Gambar 3.6. Pada arsitektur memiliki *input* yang sama dengan ikan biji nangka dan ikan kerapu. *Output* pada arsitektur ini berupa jumlah tangkapan ikan kuniran dalam satuan kilogram. Pada arsitektur JST-PSO ikan kuniran didapatkan nilai RMSE terbaik dibandingkan *hidden node* yang lainnya adalah sebesar 8501,7 kg.



Gambar 3.8 Arsitektur Jaringan Syaraf Tiruan – *Particle Swarm Optimization* Ikan Kuniran 3-10-1

Parameter eror yang menentukan arsitektur terbaik selain nilai RMSE adalah MAPE. Seperti yang sudah dijelaskan pada sub bab sebelumnya, bahwa nilai MAPE ini berupa presentase *error* yang dapat mengukur skala akurasi apakah suatu prediktor bernilai akurat atau tidak. Algoritma Jaringan Syaraf Tiruan-*Particle Swarm Optimization* memiliki presentase MAPE antara 20%-50% yang berarti bahwa prediktor tersebut termasuk dalam kategori *Reasonable Forecast*. Penjelasan lebih rinci mengenai perbandingan besar presentase MAPE untuk masing-masing algoritma akan dijelaskan pada Bab 4.

3.8 Analisis Hasil dari Sistem Prediksi Ikan

Sistem prediksi ikan dapat dikatakan telah berhasil dilakukan atau dapat bekerja dengan baik apabila nilai RMSE dari prediksi yang dihasilkan konvergen dari RMSE sebelumnya. Prediksi sebaran ikan yang dapat dikatakan memiliki akurasi yang lebih tinggi dan hasil yang lebih baik apabila nilai RMSE yang dihasilkan semakin mendekati nilai nol. Hasil pelatihan menggunakan Jaringan Syaraf Tiruan dengan fungsi pembaruan pelatihan algoritma pelatihan *Levenberg-Marquardt* dibandingkan dengan jaringan syaraf tiruan dengan optimasi bobot menggunakan *Particle Swarm Optimization* (PSO). Kemudian dipilih arsitektur jaringan terbaik yaitu dengan nilai RMSE terkecil untuk digunakan sebagai prediktor sebaran ikan.

Fluktuasi keadaan lingkungan laut mempunyai pengaruh yang besar terhadap periode migrasi musiman serta terdapatnya ikan di suatu tempat (Gunarso, 1985). Faktor oseanografi yang secara tidak langsung mempengaruhi keberadaan ikan dalam penelitian ini antara lain suhu permukaan laut, curah hujan, dan kecepatan angin. Persebaran ikan-ikan tersebut juga sering mengikuti persebaran dari arus laut yang disebabkan oleh pergerakan angin. Daerah pertemuan antara arus panas dan arus dingin merupakan daerah yang banyak organisme dan diduga daerah tersebut merupakan *fishing ground* yang baik bagi sektor perikanan (Hela and Laevastu, 1981). Oleh karena itu, jumlah tangkapan ikan yang

menjadi keluaran dari prediktor dapat digunakan untuk mengetahui banyaknya sebaran ikan di daerah tersebut.

3.9 Pengujian Realtime

Pengujian secara *realtime* dilakukan selama dua hari yaitu tanggal 16 Juni dan 18 Juni 2018. Lokasi tempat pengujian berupa di Pantai Kenjeran dengan menggunakan *Buoyweather Type II*. Sistem instrumen yang dipakai terdiri dari 3 sensor: sensor suhu DS18B20, sensor kecepatan angin, dan sensor *rain gauge*. Ketiga sensor dihubungkan dengan *arduino due*. Sensor tersebut ditempatkan di *buoyweather* untuk mengukur kondisi laut. Sensor suhu permukaan laut dipasang di *buoyweather* dengan kedalaman 10 cm terhadap permukaan *buoyweather*, sedangkan sensor curah hujan dipasang di atas kerangka *buoyweather* dekat dengan *photovoltaic*. Spesifikasi sensor prediksi ikan yang perlu ditambahkan adalah sebagai berikut:

a. Sensor Suhu Permukaan Laut

Sensor suhu yang digunakan adalah tipe DS18B20. Kelebihan sensor ini yang tahan terhadap air. Ujung sensor terdapat lapisan *stainless steel* sehingga mampu tahan karat lebih lama. Sensor suhu permukaan laut dapat dilihat pada Gambar 3.7 dan spesifikasinya pada Tabel 3.5.



Gambar 3. 9 Sensor Suhu Permukaan Laut (Depoinovasi, 2018)

Tabel 3. 5 Spesifikasi Sensor Suhu Permukaan Laut (Maxim Integrated, 2015)

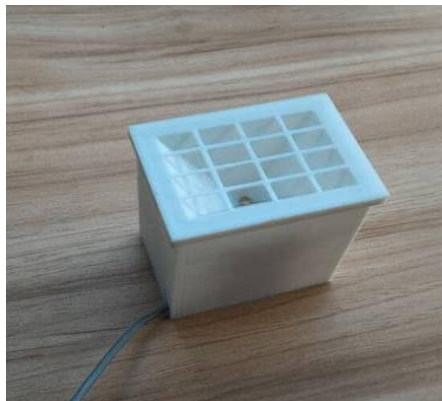
Type Sensor	DS18B20
Voltage Range (V)	0.5 sampai 6.0
Range (°C)	-55 sampai 125
Accuracy (°C)	±0.5

b. Sensor Curah Hujan

Sensor curah hujan seperti pada Gambar 3.8 yang digunakan yaitu *hall effect*. Ketika ada magnet yang mengenai sensor *hall* maka akan didapatkan spesifikasi seperti pada Tabel 3.6

Tabel 3. 6 Spesifikasi Sensor Curah Hujan (Depo Inovasi, 2011)

Type Sensor	<i>Magnetic</i>
Tegangan Masukan (volt)	0.5 sampai 6.0
Nilai per tip (mm)	1.346
Output	Pulse digital TTL



Gambar 3. 10 Sensor Curah Hujan
(Depo Inovasi, 2018)

3.10 Analisis Data dan Pembahasan

Arsitektur terbaik JST-PSO digunakan untuk memprediksi berat ikan di suatu lokasi. Hasil prediksi algoritma terbaik dianalisis jenis ikan mana yang berpotensi. Hal yang perlu dianalisis adalah hasil prediksi apakah memiliki nilai positif atau nol. Jika positif, maka hasil prediksi dapat dikatakan berhasil. Sebaliknya jika nol, maka perlu dianalisis hal apa yang menyebabkan prediksi menjadi nol. Ketika prediksi bernilai nol maka perlu ada perbandingan dengan lokasi lain. Hasil analisis digunakan untuk rekomendasi ke nelayan sebagai pertimbangan penangkapan ikan di lokasi yang diukur.

Hasil yang didapatkan dari masing-masing sensor dapat mengidentifikasi secara langsung kondisi lingkungan laut dari tempat uji saat itu. Sensor yang digunakan mampu mendukung keseluruhan dari sistem prediktor, sehingga hasil keluaran prediktor tidak bernilai nol.

3.11 Kesimpulan dan Saran

Kesimpulan merupakan jawaban dari rumusan masalah yang diangkat. Jika kesimpulan masih belum menjawab rumusan masalah, maka akan dimasukkan saran sebagai acuan untuk penelitian selanjutnya atau saran terkait penelitian ini agar lebih sempurna. Penulis juga memasukkan kelemahan-kelemahan apabila ada hasil yang tidak baik atau tidak sesuai dengan referensi-referensi sebelumnya. Adanya kesimpulan dan saran diharapkan mampu mengembangkan penelitian tentang maritim agar lebih baik dan bisa diaplikasikan di masyarakat.

Halaman ini sengaja dikosongkan

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Pengolahan Data

Data sekunder yang telah dikumpulkan, selanjutnya akan dilakukan evaluasi kelayakan data dan pengujian data secara statistik. Data yang digunakan yaitu data selama 12 bulan antara bulan Januari 2018 – Desember 2018. Pengolahan data dilakukan dalam 3 jenis pengujian yaitu uji *missing* data, uji data *outlier*, dan uji korelasi variabel. Semua pengolahan data dilakukan dengan bantuan *software* IBM SPSS 16.0.

a. Missing Data

Hasil uji missing data independen dan dependen dapat menunjukkan bahwa masing-masing data memiliki nilai presentase missing data yang sama, yaitu 0%. Missing data tidak akan bermasalah jika nilai presentase kurang dari 1% (Santoso, 2010). Jika presentase missing data lebih dari 1% dikhawatirkan data tersebut tidak merepresentasikan keadaan selama 3 bulan (Januari, Februari, dan Maret) sebagai data latih. Nilai tersebut menunjukkan bahwa data suhu permukaan laut, kecepatan angin, curah hujan, arus laut, kecepatan angin, ikan biji nangka, ikan kerapu, dan ikan kuniran dapat digunakan untuk proses selanjutnya. Proses selanjutnya, yaitu uji *outlier*.

b. Outlier

Hasil uji *outlier* semua variabel dapat dilihat pada Tabel 4.1. *Outlier* merupakan nilai ekstrem di antara nilai yang lain. Jumlah nilai ekstrem pada variabel yang digunakan dapat dilihat dari nilai z pada Tabel 4.1. Nilai z yang memiliki nilai lebih dari 2.5 atau -2.5 dianggap sebagai data *outlier*. Ada beberapa perlakuan terhadap data *outlier*, yaitu mempertahankan data apa adanya atau menghilangkan data *outlier* (Santoso, 2010). Penelitian ini mengambil perlakuan mempertahankan data apa adanya. Data *outlier* dianggap dimana kondisi variabel independen memang terjadi fenomena ekstrem. Begitu juga data *outlier* variabel

dependen. Hari tersebut merupakan hari dimana banyak nelayan yang sedang panen ikan. Alasan lain tetap mempertahankan data *outlier* adalah pengukuran dianggap valid karena instrumen yang digunakan berasal dari badan yang terpercaya, yaitu NOAA dan PPN Brondong.

Tabel 4. 1 Jumlah Data *Outlier* Pada Variabel yang Digunakan

Variabel	Jumlah Data <i>Outlier</i>
Kecepatan Angin	5
Arus Laut	2
Curah Hujan	10
Salinitas	5
Suhu permukaan laut	3
Ikan Biji Nangka	19
Ikan Kerapu	5
Ikan Kuniran	3

c. Korelasi

Uji korelasi digunakan untuk mencari hubungan antara variabel independen dan dependen. Uji korelasi ini digunakan untuk mencari variabel independen yang tidak berpengaruh dengan variabel dependen. Terdapat beberapa variabel independen yang memiliki nilai sangat rendah atau tidak berpengaruh terhadap variabel dependen. Salinitas dan arus laut memiliki nilai korelasi yang paling rendah. Sehingga variabel tersebut dapat dihilangkan dalam penelitian ini. Nilai korelasi koefisien r dapat terlihat pada Tabel 4.2. Walaupun kelima variabel independen memiliki nilai yang sama-sama relatif rendah, terlihat bahwa variabel suhu permukaan laut, curah hujan, dan kecepatan memiliki nilai korelasi yang lebih besar dibandingkan variabel salinitas dan arus laut yang bernilai korelasi negatif. Jika nilai signifikansi lebih dari 0,05 maka variabel tersebut tidak ada hubungannya. Oleh karena nilai korelasi signifikansi dicari nilai yang berada dibawah 0,05. Rata-rata variabel terkecil dari ketiga variabel tersebut adalah arus laut dan signifikan terbesar adalah salinitas. Semakin besar nilai signifikansi semakin sedikit hubungan yang dimiliki antara

variabel. Hubungan signifikansi berbanding terbalik dengan nilai korelasi. Rata-rata signifikansi terkecil dari kedua variabel yang tidak berhubungan adalah arus laut. Sedangkan nilai signifikansi terbesar dari kedua variabel yang tidak berhubungan tersebut adalah salinitas.

Tabel 4. 2 Hasil Uji Korelasi Nilai r

Variabel	Koefisien r			
	Ikan biji nangka	Ikan kerapu	Ikan kuniran	Rata- rata
Suhu permukaan laut	0,247	0,235	0,260	0,247
Curah hujan	-0,110	-0,110	-0,109	-0,109
Kecepatan angin	0,236	0,300	0,249	0,261
Salinitas	-0,006	-0,047	0,008	-0,020
Arus laut	-0,103	-0,083	-0,098	-0,094

Tiga variabel independen lain yang memiliki hubungan dengan variabel dependen adalah suhu permukaan laut, curah hujan dan kecepatan angin. Suhu permukaan laut memiliki hubungan sangat rendah dengan ikan biji nangka, dan rendah dengan ikan kerapu dan ikan kuniran. Curah hujan menunjukkan nilai negatif yang berarti berkebalikan dari variabel dependen. Curah hujan memiliki hubungan sangat rendah terhadap ikan biji nangka, ikan kerapu, Kecepatan angin memiliki hubungan rendah dengan ikan biji nangka, ikan kerapu, dan ikan kuniran. Selain nilai r, nilai signifikansi juga menunjukkan bahwa ketiga variabel independen tersebut menunjukkan nilai kurang dari 0,05. Nilai signifikansi terkecil didapatkan pada kecepatan angin yang bernilai 0 sehingga dapat dikatakan kecepatan angin memiliki hubungan terhadap variabel dependen. Nilai signifikansi terbesar dari tiga variabel yang dipilih adalah curah hujan dengan nilai signifikansi

menunjukkan angka 0,108. Nilai tersebut masih dapat dikatakan memiliki hubungan dengan variabel dependen karena masih memiliki nilai korelasi lebih besar dibanding variabel independen lainnya.

Tabel 4. 3 Hasil Uji Korelasi Nilai Signifikansi

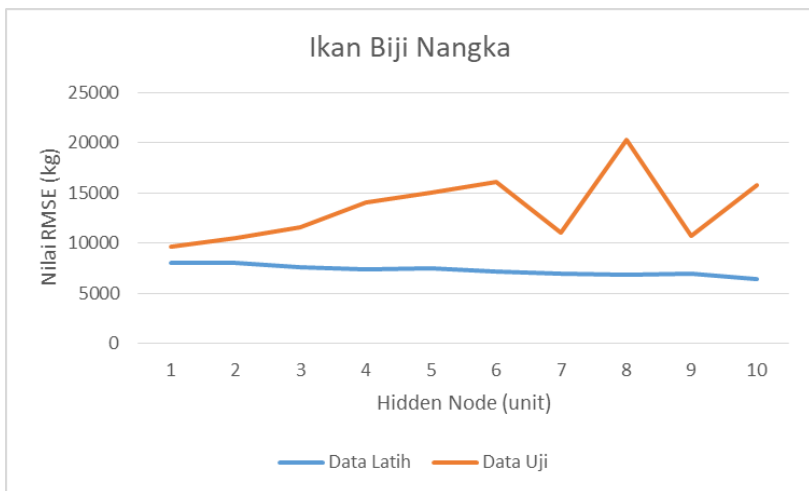
Variabel	Koefisien Signifikansi			
	Ikan biji nangka	Ikan kerapu	Ikan kuniran	Rata- rata
Suhu permukaan laut	0,023	0,000	0,017	0,013
Curah hujan	0,251	0,036	0,037	0,108
Kecepatan angin	0,000	0,000	0,000	0,000
Salinitas	0,909	0,370	0,884	0,721
Arus laut	0,251	0,111	0,060	0,140

4.2 Prediksi Sebaran Ikan Menggunakan Jaringan Syaraf Tiruan

Input yang digunakan sebanyak tiga variabel, yaitu suhu permukaan laut, kecepatan angin, dan curah hujan. Ketiga variabel tersebut yang memiliki korelasi paling tinggi dibanding yang lainnya diharapkan mampu memprediksi kelimpahan ikan dengan baik. Hasil RMSE data latih dan uji setelah dirun dengan software matlab dapat dilihat pada Gambar 4.1, Gambar 4.2, dan Gambar 4.3.

Gambar 4.1 menunjukkan hasil uji RMSE data latih dan data uji dari ikan biji nangka. Garis biru menunjukkan hasil RMSE data latih dan garis oranye menunjukkan hasil RMSE data uji dengan variasi *hidden node* 1-10. Dari Gambar 4.1 dapat dilihat bahwa *hidden node* terbaik sebagai validasi untuk memprediksi ikan biji nangka adalah 1 *hidden node*. *Hidden node* 1 memiliki nilai RMSE sebesar 9664,2323 kg.

Data pelatihan ikan biji nangka yang digunakan adalah dengan distribusi data 80% sebanyak 292 data dimulai dari tanggal 1 Januari 2018 hingga 19 Oktober 2018. Dari hasil pelatihan ikan biji nangka mampu menghasilkan RMSE data latih terkecil sebesar 6446,4368 kg yang terdapat pada *hidden node* 10 (Gambar 4.1). Hal ini menunjukkan bahwa RMSE data latih terkecil belum tentu menghasilkan nilai RMSE data uji yang terkecil pula.

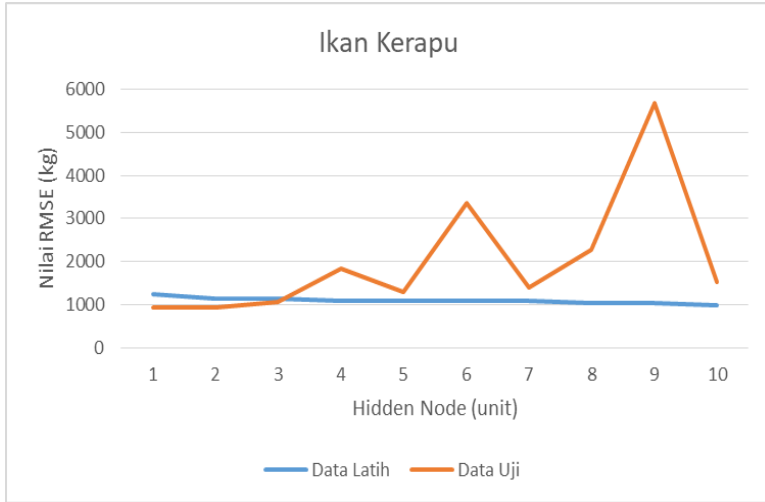


Gambar 4. 1 Hasil RMSE Ikan Biji Nangka Jaringan Syaraf Tiruan

Pada Gambar 4.2 menunjukkan hasil pelatihan RMSE data latih dan data uji pada ikan kerapu. Garis biru menunjukkan hasil RMSE data latih dan garis oranye menunjukkan hasil RMSE data uji dengan variasi *hidden node* 1-10. Dari grafik tersebut didapatkan nilai RMSE data uji terbaik atau terkecil didapatkan pada *hidden node* 3 sebesar 1071,8671 kg.

Dari Gambar 4.2, data pelatihan ikan kerapu dengan distribusi 80% sebanyak 292 data dari tanggal 1 Januari 2018 hingga 19 Oktober 2018 mampu menghasilkan nilai RMSE data latih terkecil sebesar 987,3092 kg terdapat pada *hidden node* 10. Sedangkan nilai RMSE data uji pada *hidden node* 10 adalah sebesar 1534,8985

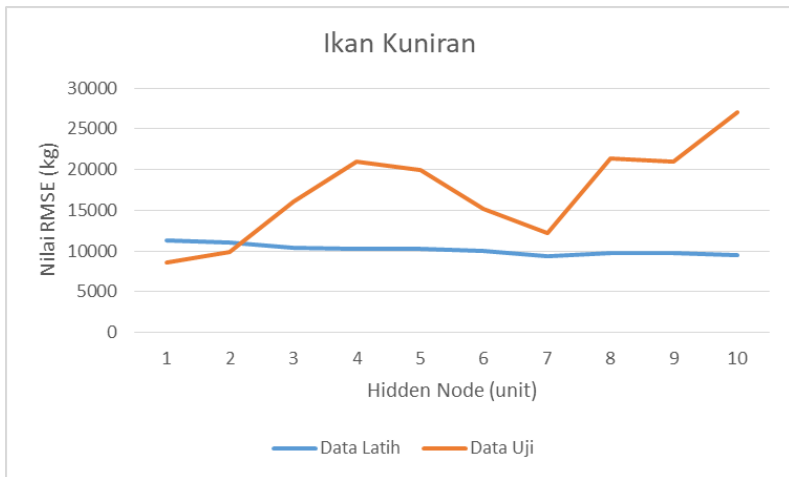
kg. Sehingga RMSE data uji terkecil belum tentu menghasilkan RMSE data latih yang terkecil pula.



Gambar 4. 2 Hasil RMSE Ikan Kerapu Jaringan Syaraf Tiruan

RMSE ikan kuniran pada Gambar 4.3 juga menunjukkan hal yang sama. Tidak ada hubungan linier atau berbanding terbalik antara jumlah *hidden node* dengan nilai RMSE. Grafik biru menunjukkan RMSE data latih dan grafik oranye menunjukkan RMSE data uji. Jika dilihat pada Gambar 4.3, data pelatihan ikan kuniran dengan distribusi data 80% sebesar 292 data dari 1 Januari 2018 hingga 19 Oktober 2018 mampu menghasilkan uji validasi RMSE terkecil dengan nilai RMSE sebesar 9861,9855 kg. RMSE data uji terbaik ikan kuniran didapatkan pada *hidden node* 2.

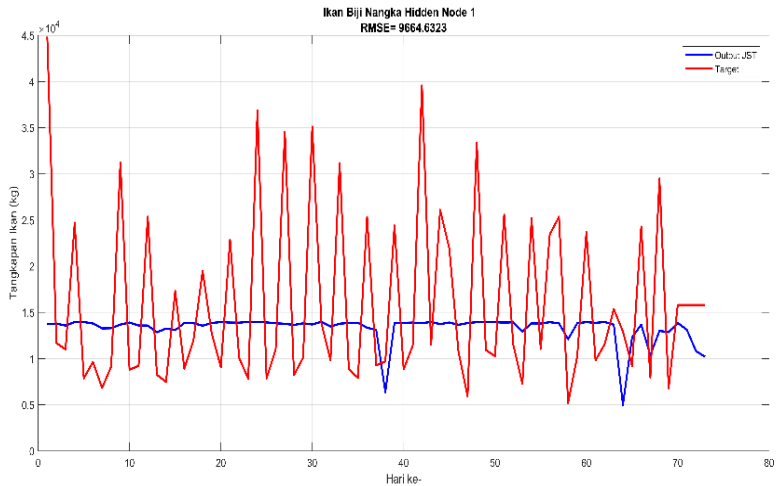
Grafik ikan kuniran menunjukkan bahwa RMSE data latih pada *hidden node* 2 bukan merupakan yang terkecil. RMSE data latih *hidden node* 2 memiliki nilai sebesar 11019,0268 kg. Hal itulah mengapa RMSE data uji kecil belum tentu memiliki nilai RMSE data latih yang kecil pula. Grafik tersebut juga memperlihatkan bahwa nilai RMSE data latih yang semakin lama semakin menurun namun tidak untuk nilai RMSE data uji nya.



Gambar 4. 3 Hasil RMSE Ikan Kuniran Jaringan Syaraf Tiruan

Model Jaringan Saraf Tiruan telah dianggap sebagai kotak hitam. Hubungan antara *input* dan *output* tidak bisa dijelaskan secara jelas sehingga sulit untuk menafsirkan arti dari perubahan *hidden node* (Torres-Faurrieta et al., 2016). Hal ini memperkuat penelitian sebelumnya. Variasi *hidden node* menunjukkan data yang fluktuatif (Gao, Chen, & Qin, 2010; Thomas, Manoj Kumar, Annappa, 2016). Tidak ada metode yang tepat untuk menentukan *hidden layer* dan *hidden node* untuk menentukan nilai *error* terkecil (Pitambare, 2016).

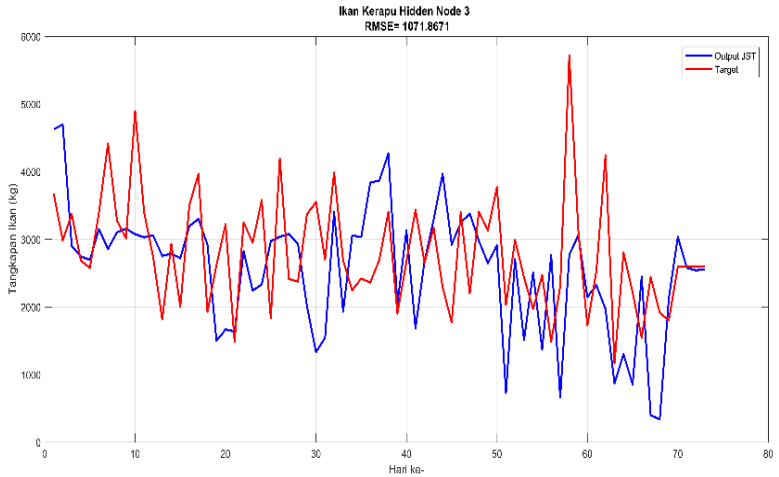
Grafik hasil uji ikan biji angka dengan *hidden node* 1 dapat dilihat pada Gambar 4.4. Garis biru menunjukkan hasil keluaran algoritma JST dan garis merah menunjukkan target yang didapatkan dari data sekunder ke-293 hingga ke-365 atau sebanyak 20% dari tanggal 20 Oktober 2018 hingga 31 Desember 2018. Gambar grafik tersebut menunjukkan bahwa keluaran JST masih dapat mengikuti pola target secara baik. RMSE data uji didapatkan sebesar 9664,6323 kg masih menunjukkan *error* yang cukup besar sebagai prediktor sehingga perlu adanya optimasi agar diperoleh hasil yang lebih baik.



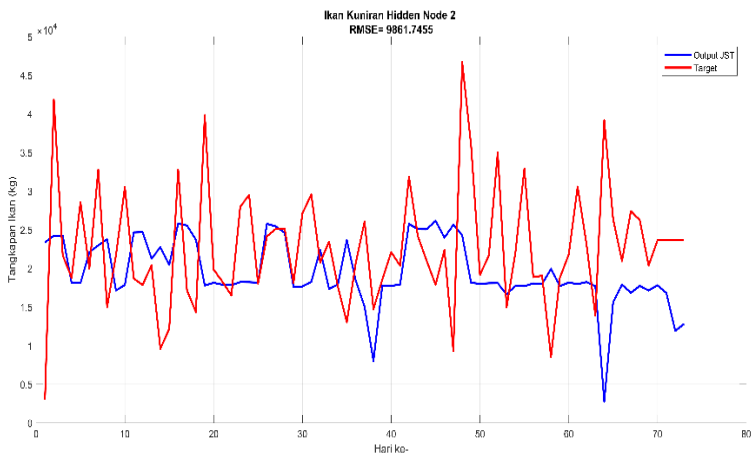
Gambar 4. 4 Hasil Uji Ikan Biji Nangka Jaringan Syaraf Tiruan *Hidden node 1*

Hasil uji ikan kerapu JST *hidden node 3* dapat dilihat pada Gambar 4.5. Grafik keluaran JST yang berwarna biru juga masih belum bisa mengikuti pola target dengan tepat terhadap garis yang berwarna merah. RMSE data uji 1071,8671 kg juga masih terlalu besar untuk dijadikan sebagai prediktor. Prediktor dengan *hidden node 3* masih belum bisa dipercaya karena masih memiliki *error* yang cukup besar sehingga perlu dioptimasi agar RMSE data uji bernilai lebih kecil daripada sebelumnya.

Grafik pengujian ikan kuniran JST pada *hidden node 2* dapat dilihat pada Gambar 4.6. Terlihat dari grafik pengujian keluaran JST ikan kuniran belum mengikuti garis dari target. Keluaran JST ikan kuniran memiliki keluaran yang sama pada beberapa hari. Selain itu RMSE data uji sebesar 9861,7455 kg masih dianggap memiliki eror yang lumayan besar. Hal ini menunjukkan bahwa algoritma JST ikan kuniran *hidden node 2* masih belum bisa dijadikan prediktor sehingga tetap perlu adanya optimasi.



Gambar 4. 5 Hasil Uji Ikan Kerapu Jaringan Syaraf Tiruan
Hidden Node 3



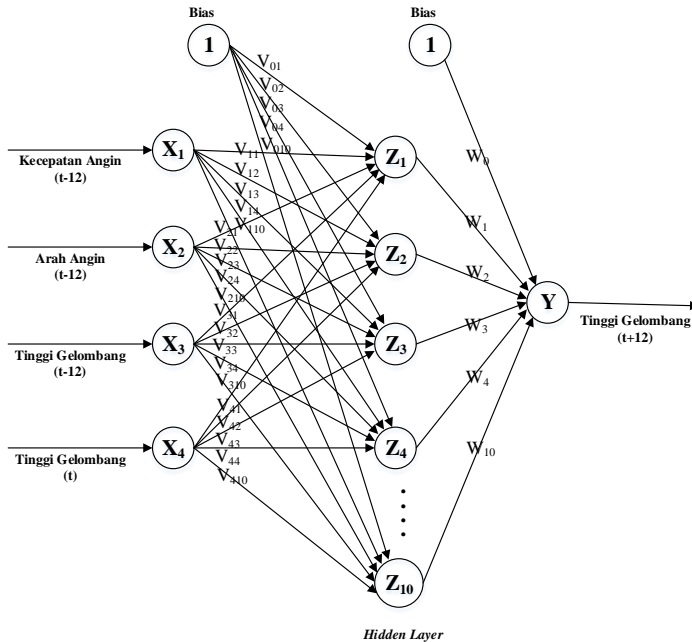
Gambar 4. 6 Hasil Uji Ikan Kuniran Jaringan Syaraf Tiruan
Hidden Node 2

Iterasi pada pelatihan ikan biji nangka memberikan bobot JST seperti pada Tabel 4.4. Penamaan bobot JST mengacu pada buku *Fundamental of Neural Network Architecture, Algorithms, and Application* (Fausett, 1994). Ilustrasi terkait penamaan bobot pada arsitektur dapat dilihat pada Gambar 4.7. Penanaman dasar adalah V_{ij} . i merupakan *node* awal, sedangkan j merupakan *node* yang dituju. Bila bobot maksimal dan minimal masing-masing secara berurutan adalah 5,8106 dan -3,7930. Tabel 4.4 menunjukkan bahwa variabel kecepatan angin (V_{11}) dan curah hujan (V_{21}) memiliki nilai bobot positif sedangkan variabel suhu permukaan laut (V_{31}) memiliki nilai bobot negatif. Tanda positif yang sama menunjukkan bahwa antar variabel saling menguatkan ketersediaan ikan biji nangka, sedangkan tanda negatif sebaliknya. Tanda negatif menunjukkan bahwa antar variabel saling menguatkan untuk mengurangi ketersediaan ikan biji nangka.

Perhatikan kolom 2 bobot W_0 hingga W_1 pada Tabel 4.4. Terlihat bahwa hanya bobot dari bias ke *output* yang bernilai positif sedangkan dari *hidden layer* 1 memiliki nilai negatif yang berarti tidak mendukung adanya ketersediaan ikan biji nangka. Nilai bobot dengan dengan rentang maksimal dan minimal tersebut masih belum mampu memberikan prediksi dengan RMSE terbaik.

Tabel 4. 4 Bobot Terbaik Prediksi Ikan Biji Nangka Jaringan Syaraf Tiruan

Posisi	Bobot
V_{01}	-3,7930
V_{11}	4,6408
V_{21}	5,8106
V_{31}	-0,6059
W_0	0,2632
W_1	-0,0807



Gambar 4. 7 Ilustrasi Penamaan Bobot Pada Arsitektur Jaringan Syaraf Tiruan

Iterasi pada ikan kerapu memberikan hasil bobot JST pada Tabel 4.5. Bobot ikan kerapu memiliki rentang diantara 52,2437 dan -4,0537. Bobot curah hujan (V_{21}) dan suhu permukaan laut (V_{31}) memiliki tanda positif sedangkan bobot kecepatan (V_{11}) bernilai negatif. Tanda ini berarti bahwa variabel kecepatan angin dan suhu permukaan laut saling menguatkan adanya ketersediaan ikan kerapu. Bobot *input* ke *hidden layer* yang saling menguatkan adalah *hidden 3*. *Hidden layer 3* memiliki tanda positif sehingga saling menguatkan adanya ketersediaan ikan kerapu. Sedangkan pada *hidden layer 1* dan *2* bobot untuk variabel kecepatan angin memiliki bobot bertanda negatif. Hal ini berarti variabel ini akan menguatkan untuk mengurangi ketersediaan ikan kerapu. Bobot *hidden layer* ke *output* yang saling menguatkan antar *hidden layer*

adalah W_2 dan W_3 . Ketiga bobot ini saling menguatkan untuk mengurangi ketersediaan ikan kerapu. Sedangkan bobot *hidden layer* ke *output* yang lain seperti W_0 dan W_1 juga saling menguatkan terkait adanya ketersediaan ikan kerapu. Bobot dengan variasi tersebut masih belum mampu memprediksi ikan kerapu karena RMSE masih belum mencapai yang terkecil.

Tabel 4. 5 Bobot Terbaik Prediksi Ikan Kerapu Jaringan Syaraf Tiruan

Posisi	Bobot	Posisi	Bobot
V_{01}	-2,9996	V_{23}	1,7469
V_{02}	-4,3496	V_{31}	2,3999
V_{03}	-10,6228	V_{32}	3,3057
V_{11}	-0,5538	V_{33}	7,7860
V_{12}	-4,0537	W_0	1,1287
V_{13}	52,2437	W_1	1,3755
V_{21}	1,6279	W_2	-0,4816
V_{22}	4,6286	W_3	-0,1373

Iterasi pada pelatihan ikan kuniran memberikan hasil bobot JST seperti pada Tabel 4.6. Nilai bobot minimum dan maksimum secara berurutan adalah 57,2585 dan -36,1369. Jika dilihat pada Tabel 4.6, antar variabel yang saling menguatkan adalah *hidden layer 2*, sehingga antar variabel kecepatan angin, curah hujan, dan salinitas saling menguatkan ketersediaan ikan kuniran. Berbeda pada *hidden layer 2* dan 3 yang memiliki tanda berbeda antar variabel sehingga tidak saling menguatkan antar variabel. Bobot bias menuju *hidden layer* semuanya saling menguatkan antar variabel namun bernilai negatif. Jika dilihat pada bobot *hidden layer* ke *output* hanya bobot dari *hidden layer 2* yang bernilai negatif. Hanya bobot bias dan bobot dari *hidden layer 1* yang mendukung ketersediaan ikan kuniran. Bobot tersebut masih

belum mampu memprediksi ikan kuniran karena RMSE yang dirasa masih belum mencapai yang terkecil.

Tabel 4. 6 Bobot Terbaik Prediksi Ikan Kuniran Jaringan Syaraf Tiruan

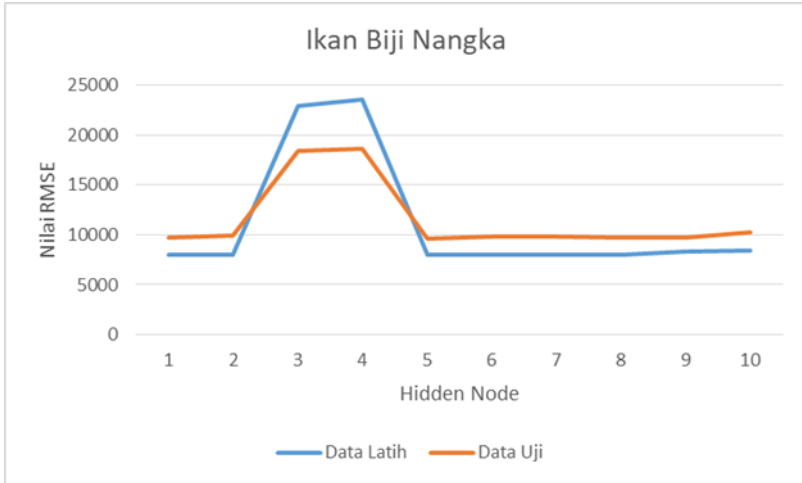
Posisi	Bobot	Posisi	Bobot
V ₀₁	-36,1369	V ₃₁	34,2331
V ₀₂	-4,4894	V ₃₂	1,0432
V ₁₁	-25,3524	W ₀	0,3016
V ₁₂	3,2247	W ₁	0,0608
V ₂₁	57,2585	W ₂	-0,1139
V ₂₂	7,1277		

Hasil analisis bobot prediksi terbaik dari masing-masing ikan memiliki rentang yang cukup besar hingga puluhan, namun RMSE yang dihasilkan masih belum optimal sehingga perlu optimasi untuk mencapai yang lebih baik. Dari hasil analisa bobot jaringan syarat tiruan untuk masing-masing ikan, didapatkan nilai maksimum dan minimum dari bobot tersebut secara berurutan adalah sebesar 57 dan -36. Setelah diptimasi dengan menggunakan *Particle Swarm Optimization* diharapkan akan didapatkan bobot yang lebih dibandingkan bobot yang dihasilkan oleh Jaringan Syaraf Tiruan saja.

4.3 Prediksi Sebaran Ikan Menggunakan Jaringan Syaraf Tiruan dengan Optimasi *Particle Swarm Optimization* (PSO)

Hasil running program matlab JST-PSO dengan *hidden node* 1-10 angka dapat dilihat pada Gambar 4.7. Dapat terlihat bahwa hubungan antara nilai RMSE dengan jumlah *hidden node* tidak linier. Hasil RMSE prediktor gabungan JST-PSO ikan biji angka dapat dilihat pada Gambar 4.7. Garis biru merupakan hasil RMSE data latih dan garis oranye merupakan hasil RMSE data uji. Sumbu x dan sumbu y masing-masing adalah jumlah *hidden node* dan nilai RMSE. Dari Gambar 4.7 menunjukkan bahwa data latih gabungan

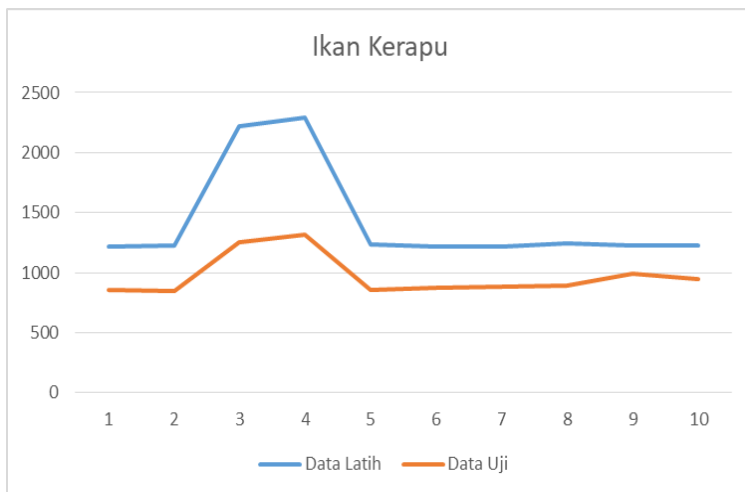
JST-PSO mampu menghasilkan RMSE antara 7000 kg hingga 25.000 kg.



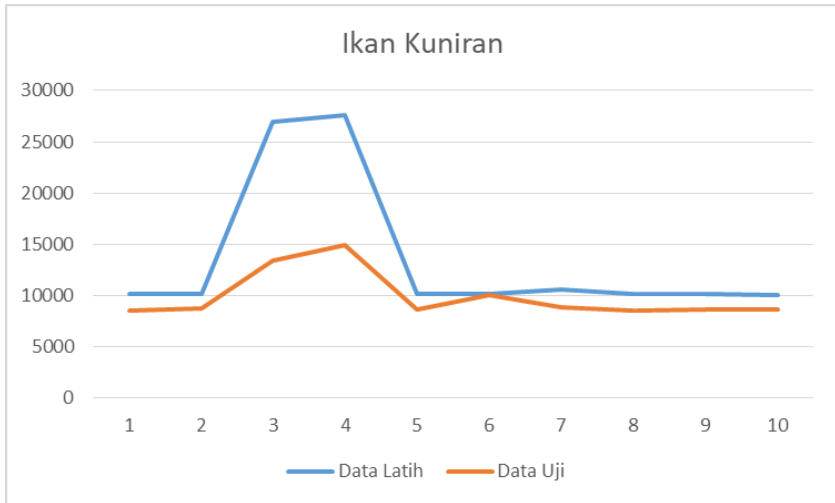
Gambar 4. 8 Hasil RMSE Ikan Biji Nangka Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

Grafik data latih dan data uji cenderung lebih teratur dibandingkan dengan grafik prediktor JST saja. Walaupun nilainya tidak berkurang terlalu jauh bahkan ada yang semakin besar, grafik data latih dan data uji terlihat memiliki hubungan yang linier. Terlihat dari grafik bahwa mulai *hidden layer* 5 sampai 10 tidak mengalami perubahan RMSE yang signifikan baik dari data latih maupun data uji. Hasil terbaik untuk data uji dari grafik ikan biji nangka JST-PSO terdapat pada *hidden node* 5 dengan nilai RMSE sebesar 9638,6741 kg. Jika dilihat prediktor JST uji terbaik ikan biji nangka sebesar 9664,6323 kg, prediktor ini mampu menurunkan RMSE data uji sebanyak 25,9582 kg. Walau tidak mengalami penurunan RMSE yang banyak, namun masih dapat dikatakan bahwa nilai RMSE prediktor JST-PSO menghasilkan nilai RMSE data uji yang lebih rendah dibandingkan dengan nilai RMSE prediktor JST.

Grafik RMSE data latih dan data uji dari prediktor JST-PSO ikan kerapu dapat dilihat pada Gambar 4.8. Garis biru menunjukkan RMSE data latih sedangkan garis oranye menunjukkan data uji dengan sumbu x dan y masing-masing menunjukkan jumlah *hidden node* dan nilai RMSE. Gambar 4.8 menunjukkan mampu menurunkan rentang RMSE data uji dari 1000-6000 kg menjadi 800-2000 kg dengan beda selisih masing-masing 5000kg dan 1200 kg. Jika dilihat RMSE data uji, prediktor JST-PSO mampu menurunkan RMSE data uji semua *hidden node*. prediktor JST-PSO terbaik didapatkan pada jumlah *hidden node* 9 dengan nilai RMSE data uji sebesar 993,9949 kg. Sehingga prediktor JST-PSO mampu menurunkan RMSE sebesar 4675,5357 kg.



Gambar 4. 9 Hasil RMSE Ikan Kerapu Jaringan Syaraf Truan – *Particle Swarm Optimization*

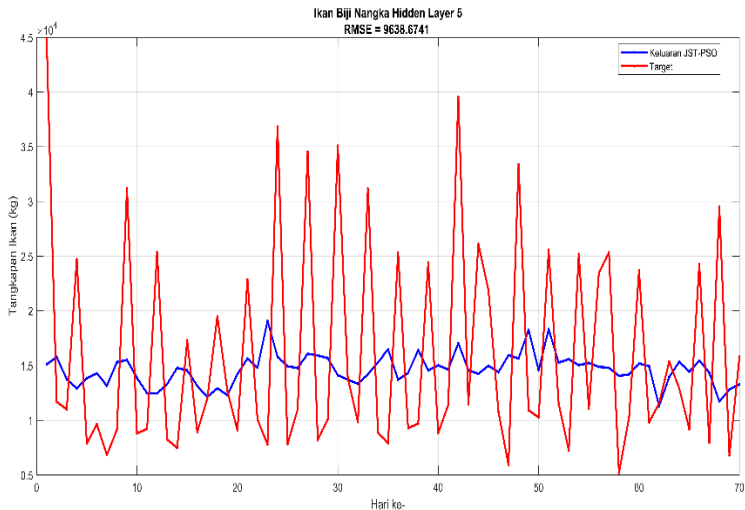


Gambar 4. 10 Hasil RMSE Ikan Kuniran Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

Hal yang sama ditunjukkan oleh RMSE ikan kuniran pada Gambar 4.9. Garis biru menunjukkan RMSE data latih sedangkan garis oranye menunjukkan RMSE data uji dengan sumbu x dan y masing-masing menunjukkan jumlah *hidden node* dan nilai RMSE. Prediktor JST-PSO mampu menurunkan rentang RMSE data uji dari rentang 8.000-30.000 kg menjadi 8.000-15.000 kg dengan selisih masing-masing sebesar 22.000 kg dan 7.000 kg. Penurunan ini mampu mencapai angka 15.000 kg. Dari hasil grafik diatas, didapatkan nilai uji terbaik dari ikan kuniran terdapat pada *hidden layer* 10 dengan nilai RMSE sebesar 8634,8335 kg.

Perbandingan prediktor JST dan JST-PSO dalam hal RMSE data latih dan data uji mampu merubah beberapa hal. Hasil analisis dari Gambar 4.10, Gambar 4.11 dan Gambar 4.12 memberikan informasi bahwa prediktor yang dioptimasi dengan PSO mampu menurunkan rentang RMSE data uji. Tidak hanya itu, RMSE data latih juga mengalami penurunan walau tidak signifikan. Ini membuktikan bahwa dalam hal RMSE data latih maupun data uji,

prediktor JST-PSO merupakan prediktor yang lebih baik daripada prediktor JST.

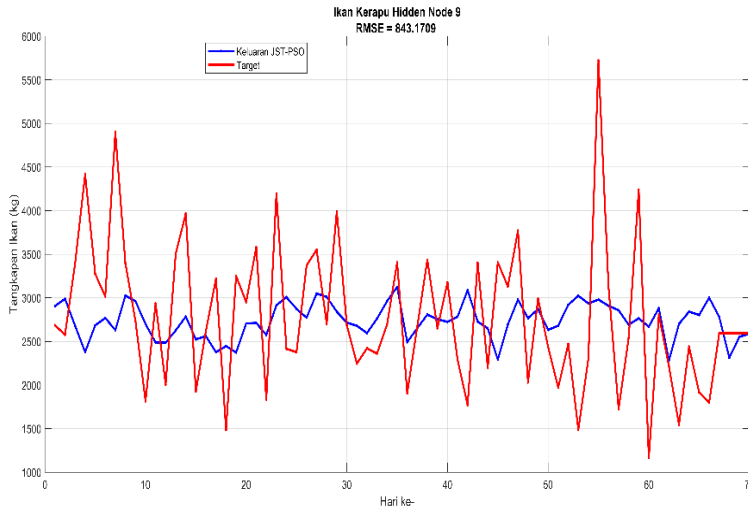


Gambar 4. 11 Hasil Pengujian Ikan Biji Nangka Jaringan Syaraf Tiruan – *Particle Swarm Optimization Hidden node 5*

Grafik hasil pengujian ikan biji nangka JST-PSO dapat dilihat pada Gambar 4.10. Pada grafik tersebut, garis biru menunjukkan keluaran prediktor JST-PSO dan garis merah menunjukkan target. Pada gambar pengujian ikan biji nangka terlihat bahwa pola garis dari keluaran JST-PSO mengikuti pola dari target data asli. Hasil nilai RMSE uji terbaik pada ikan biji nangka yang didapatkan pada *hidden layer 5* tidak menghasilkan nilai RMSE latih yang terbaik pula. Nilai RMSE data latih yang dihasilkan dari *hidden layer 5* adalah sebesar 9638,6741 kg.

Grafik hasil pengujian ikan kerapu JST-PSO dapat dilihat pada Gambar 4.11. Sama seperti penjelasan pada grafik ikan biji nangka, pada grafik ikan kerapu garis biru menunjukkan keluaran prediktor JST-PSO sedangkan garis merah menunjukkan target asli. Nilai RMSE data uji ikan kerapu terbaik terdapat pada *hidden*

layer 9 dengan nilai RMSE sebesar 843,1 kg. Namun, meskipun terdapat RMSE uji terkecil pada *hidden layer* 9 namun tidak menghasilkan nilai RMSE data latih terkecil pula pada *hidden layer* tersebut. Hasil garis keluaran JST-PSO sudah mulai mengikuti garis target walaupun belum sempurna.

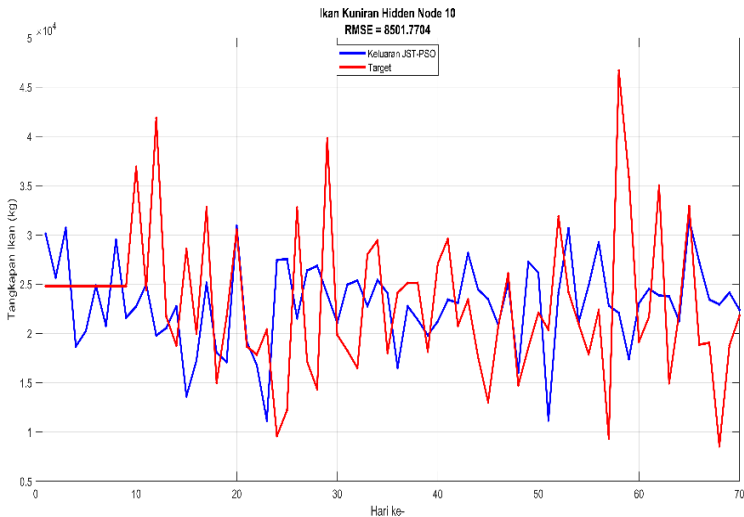


Gambar 4. 12 Hasil Pengujian Ikan Kerapu Jaringan Syaraf Tiruan – *Particle Swarm Optimization Hidden node 9*

Grafik hasil pengujian ikan kuniran JST-PSO dapat dilihat pada Gambar 4.12. Nilai RMSE data uji terkecil didapatkan pada *hidden layer* 9 dengan nilai RMSE sebesar 8501,7 kg. Data pengujian didapatkan dengan rentang waktu dari tanggal 20 Oktober 2018 sampai 31 Desember 2018. Garis keluaran JST-PSO lebih terlihat mengikuti garis target asli. Hal ini dibuktikan juga dengan nilai RMSE uji yang lebih baik dibandingkan RMSE uji prediktor JST.

Bobot algoritma JST-PSO ikan biji angka memiliki *constraint* antara -1 hingga 1. Namun algoritma JST-PSO mampu memberikan nilai RMSE yang lebih baik dibandingkan dengan algoritma JST. Bobot yang kecil akan lebih efektif perhitungannya

dibanding dengan bobot yang bernilai besar. Bobot JST-PSO ikan biji angka dapat dilihat pada Tabel 4.7.



Gambar 4. 13 Hasil Pengujian Ikan Kuniran Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

Jika dilihat bobot *input* ke *hidden layer*, antar variabel yang saling mendukung adalah *hidden layer* 4 namun bernilai negatif. Bobot *hidden layer* yang lain menunjukkan tanda yang berbeda antar variabel. Perhatikan kolom 2 pada Tabel 4.7 ,bobot *hidden layer* ke *output* yaitu w_1, w_2 dan w_4 yang memiliki tanda negatif yang berarti bahwa bobot saling menguatkan untuk mengurangi ketersediaan ikan biji angka. Sedangkan untuk bobot w_0, w_3 , dan w_5 yang memiliki tanda positif menunjukkan bahwa bobot tersebut aling menguatkan untuk mendukung ketersediaan ikan biji angka.

Tabel 4. 7 Bobot Terbaik Prediksi Ikan Biji Nangka Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

Posisi	Bobot	Posisi	Bobot
V ₀₁	-0,2041	V ₂₄	-0,7745
V ₀₂	0,5588	V ₂₅	-0,0685
V ₀₃	0,4539	V ₃₁	-0,2021
V ₀₄	-0,6499	V ₃₂	-0,1270
V ₀₅	-0,2610	V ₃₃	0,0254
V ₁₁	0,3043	V ₃₄	-0,6602
V ₁₂	-0,3329	V ₃₅	0,3854
V ₁₃	-0,3558	W ₀	0,0183
V ₁₄	-0,2550	W ₁	-0,0826
V ₁₅	-0,4986	W ₂	-0,1283
V ₂₁	-0,2892	W ₃	0,3903
V ₂₂	0,1375	W ₄	-0,4260
V ₂₃	-0,0305	W ₅	0,3224

Initial constraint pada ikan kerapu juga berada diantara -1 hingga 1. *constraint* tersebut mampu menghasilkan RMSE data uji yang lebih baik daripada prediktor JST. Bobot prediktor JST-PSO ikan kerapu dapat dilihat pada Tabel 4.8. Pada bobot *input* ke *hidden layer*, antar variabel yang saling mendukung adalah *hidden layer* 3 dan 5. Bobot masing-masing kedua *hidden layer* tersebut bernilai positif yang berarti variabel tersebut saling mendukung untuk menguatkan ketersediaan ikan kerapu. Pada bobot *hidden layer* ke *output*, W₁, W₂, W₄, W₅, dan W₉ memiliki tanda yang negatif, hal ini berarti bahwa bobot itu akan saling menguatkan untuk mengurangi ketersediaan ikan kerapu. Sedangkan untuk bobot W₀, W₃, W₆, W₇, dan W₈ memiliki nilai bobot bertanda positif. Hal ini berarti bahwa bobot tersebut akan saling menguatkan untuk mendukung ketersediaan ikan kerapu.

Tabel 4. 8 Bobot Terbaik Ikan Kerapu Jaringan Syaraf Tiruan – Particle Swarm Optimization

Posisi	Bobot	Posisi	Bobot	Posisi	Bobot
V ₀₁	-0,4593	V ₁₈	0,3747	V ₃₆	0,3549
V ₀₂	0,2217	V ₁₉	-0,3540	V ₃₇	-0,1138
V ₀₃	-0,2877	V ₂₁	-0,3655	V ₃₈	-0,4472
V ₀₄	-0,1624	V ₂₂	0,2866	V ₃₉	0,1151
V ₀₅	0,0547	V ₂₃	0,0904	W ₀	0,6644
V ₀₆	0,7565	V ₂₄	0,0609	W ₁	-0,2138
V ₀₇	-0,2838	V ₂₅	0,4660	W ₂	-0,1624
V ₀₈	-0,1747	V ₂₆	0,7104	W ₃	0,0579
V ₀₉	0,2537	V ₂₇	-0,2494	W ₄	-0,5278
V ₁₁	0,8931	V ₂₈	-0,1114	W ₅	-0,4808
V ₁₂	-0,0049	V ₂₉	0,1134	W ₆	0,0505
V ₁₃	0,4828	V ₃₁	0,1362	W ₇	0,4446
V ₁₄	-0,0845	V ₃₂	-0,8334	W ₈	0,0294
V ₁₅	0,8280	V ₃₃	0,3854	W ₉	-0,0319
V ₁₆	-0,2296	V ₃₄	-0,0120		
V ₁₇	-0,1801	V ₃₅	0,0769		

Pada ikan kuniran menunjukkan bahwa *constraint* di antara -1 hingga 1 mampu memberikan RMSE data uji terbaik daripada prediktor JST. Bobot terbaik prediktor JST-PSO pada ikan kuniran dapat dilihat pada Tabel 4.9. Bobot *input* suhu permukaan laut, curah hujan, dan kecepatan angin ke *hidden layer* yang memiliki hubungan saling menguatkan adalah pada *hidden node* 4 dan 10 yang memiliki bobot bertanda positif. Bobot *input* menuju hidden node yang lain memiliki tanda negatif yang berarti variabel tidak saling menguatkan satu sama lain.

Tabel 4. 9 Bobot Terbaik Prediksi Ikan Kuniran Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

Posisi	Bobot	Posisi	Bobot	Posisi	Bobot
V ₀₁	0,0157	V ₁₉	0,2148	V ₃₇	-0,0706
V ₀₂	0,1930	V ₀₁₀	0,1191	V ₃₈	-0,5781
V ₀₃	0,2980	V ₂₁	-0,2095	V ₃₉	0,6174
V ₀₄	-0,2036	V ₂₂	0,2609	V ₃₁₀	-0,1568
V ₀₅	-0,0400	V ₂₃	0,1017	W ₀	0,2612
V ₀₆	-0,2076	V ₂₄	0,1818	W ₁	0,5271
V ₀₇	-0,0588	V ₂₅	0,0195	W ₂	-0,4012
V ₀₈	-0,0974	V ₂₆	0,2509	W ₃	0,0228
V ₀₉	0,2494	V ₂₇	-0,0918	W ₄	0,1421
V ₀₁₀	-0,3759	V ₂₈	0,0752	W ₅	0,1824
V ₁₁	-0,1980	V ₂₉	-0,0721	W ₆	0,1238
V ₁₂	-0,1267	V ₂₁₀	0,2526	W ₇	-0,2107
V ₁₃	-0,1590	V ₃₁	0,4347	W ₈	0,0885
V ₁₄	0,2655	V ₃₂	-0,2066	W ₉	0,0731
V ₁₅	-0,0317	V ₃₃	0,2951	W ₁₀	-0,3321
V ₁₆	-0,6143	V ₃₄	0,3840		
V ₁₇	0,2419	V ₃₅	0,2901		
V ₁₈	0,0708	V ₃₆	-0,0534		

Hasil analisis bobot prediktor terbaik JST-PSO semua jenis ikan menunjukkan bahwa dengan *constraint* -1 hingga 1 mampu memberikan RMSE data uji lebih kecil dibandingkan dengan prediktor JST. Hal ini juga menunjukkan bahwa prediktor JST-PSO cukup efektif jika digunakan sebagai prediktor dibandingkan dengan prediktor JST saja.

4.4 Perbandingan Hasil Antara Metode Prediktor JST-LM dan JST-PSO

Arsitektur jaringan terbaik yang telah didapatkan dari kedua metode JST dan JST-PSO kemudian dibandingkan berdasarkan nilai RMSE dan jumlah *epoch* dapat dilihat pada Tabel 4.12. Jaringan Saraf Tiruan – *Particle Swarm Optimization* menghasilkan prediktor terbaik pada RMSE data uji semua jenis ikan. Dari Tabel 4.12 dapat diketahui bahwa nilai RMSE terkecil didapatkan saat menggunakan metode JST-PSO.

Berdasarkan grafik pola pelatihannya yang telah dibahas pada sub bab sebelumnya, dibandingkan dengan JST bahwa JST-PSO memiliki grafik pola pelatihan yang lebih baik yaitu dari hasil prediksi persebaran jumlah tangkapan ikan dapat mengikuti pola target jumlah ikan yang asli.

Dari hasil perbandingan diatas dilihat bahwa hasil RMSE nilai uji JST-PSO memiliki nilai lebih rendah dibandingkan nilai RMSE prediktor JST. Optimasi PSO untuk menentukan nilai bobot optimal dapat memberikan hasil klasifikasi yang akurat sebanding dengan waktu konvergensi yang dapat diterima. Arsitektur jaringan, pemilihan parameter pada PSO, dan banyaknya data seperti jumlah *hidden node* yang memberikan banyaknya variabel bobot yang dioptimasi mempengaruhi performansi dan kecepatan waktu konvergensinya. (Hamed, Shamsuddin, & Salim, 2008).

Nilai MAPE pada Tabel 4.10 menunjukkan bahwa nilai presentasi MAPE algoritma prediktor JST-PSO memiliki presentase *error* MAPE lebih kecil daripada algoritma prediktor JST-LM. Hal ini membuktikan bahwa rancangan prediktor yang menggunakan algoritma JST-PSO memiliki performa yang lebih baik dibandingkan dengan JST-LM.

Pada nilai MAPE untuk algoritma prediktor JST-LM dapat dilihat bahwa ikan kerapu dan ikan kuniran termasuk dalam kategori *reasonable forecast*. Ikan biji nangka yang memiliki nilai MAPE diatas 50% berada dalam kategori *inaccurate forecast*. Presentase nilai MAPE pada prediktor algoritma JST-PSO

mengalami penurunan sebesar 5%-8%. Selain nilai MAPE, dua parameter lain yang menjadi pembandingan antara kedua algoritma ini adalah RMSE dan *epoch*. Nilai dari kedua parameter tersebut juga mengalami penurunan, sehingga dapat disimpulkan bahwa algoritma JST-PSO merupakan algoritma yang tepat untuk memprediksi sebaran ikan.

Tabel 4. 10 Perbandingan Hasil Keluaran Prediktor JST-LM dan JST-PSO

Parameter	Variabel	JST-LM	JST-PSO
RMSE (kg)	Ikan Biji Nangka	9664,6323	9638,6741
	Ikan Kerapu	1071,8671	843,1
	Ikan Kuniran	9861,7455	8501,7
Epoch	Ikan Biji Nangka	41	30
	Ikan Kerapu	500	30
	Ikan Kuniran	182	30
MAPE (%)	Ikan Biji Nangka	53,46	46,68
	Ikan Kerapu	30,42	25,58
	Ikan Kuniran	40,63	32,69

Berdasarkan Tabel 4.10 dapat dilihat bahwa nilai RMSE algoritma JST-PSO mengalami penurunan RMSE. Ikan biji nangka mengalami penurunan RMSE sebesar 25,9582 kg. Ikan kerapu mengalami penurunan RMSE sebesar 228,7671 kg. Ikan kuniran mengalami penurunan RMSE sebesar 1360,0455 kg.

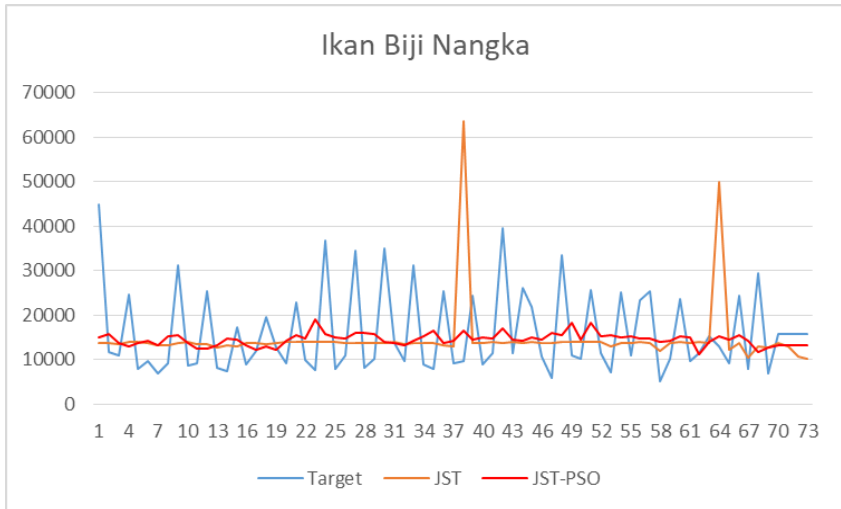
Meskipun JST-PSO membutuhkan iterasi/*epoch* yang lebih sedikit daripada JST, namun waktu yang dibutuhkan untuk satu *epoch* tersebut membutuhkan waktu konvergensinya. (Muhtar, Mustika, & Suharyanto, 2017). Program komputasi yang meminta

nilai best cost di setiap partikel dan iterasi juga dapat mempengaruhi kecepatannya.

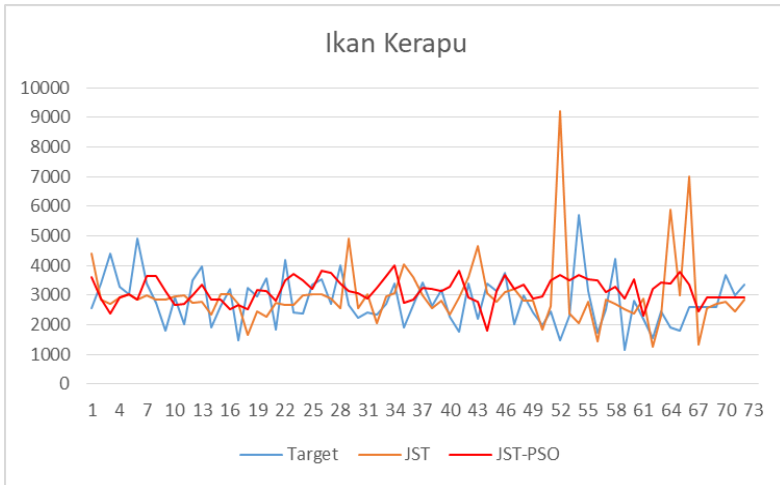
Grafik perbandingan keluaran JST-PSO dengan target ikan biji angka dapat dilihat pada Gambar 4.13. Garis biru menunjukkan target dari hari ke-293 hingga 365. Mulai tanggal 20 Oktober 2018 sampai 31 Desember 2018. Garis merah adalah keluaran dari JST-PSO. Garis warna oranye menunjukkan hasil keluaran dari algoritma JST-LM. Sumbu x dan y masing-masing menunjukkan data hari dan jumlah tangkapan ikan dalam kg. Gambar 4.13 menunjukkan bahwa selisih keluaran JST-PSO ikan biji angka terhadap target lebih sedikit dibandingkan dengan keluaran JST ikan biji angka terhadap target. Hal ini yang membuat RMSE data uji menjadi semakin lebih kecil ketika dipasang prediktor JST-PSO.

Hasil ini juga ditunjukkan oleh keluaran JST-PSO ikan kerapu dan ikan kuniran seperti pada Gambar 5.14 dan Gambar 4.15. Garis merah pada gambar menunjukkan keluaran dari algoritma JST-PSO. Garis warna oranye menunjukkan keluaran dari algoritma JST-LM. Garis warna menunjukkan target dari hari ke-293 hingga 365. Sumbu x dan y masing-masing menunjukkan data hari dan jumlah tangkapan ikan dalam kg. Selisih keluaran JST-PSO ikan tersebut terhadap target lebih sedikit dibandingkan dengan keluaran JST terhadap target. Sehingga mampu memperkecil data uji dari masing-masing ikan.

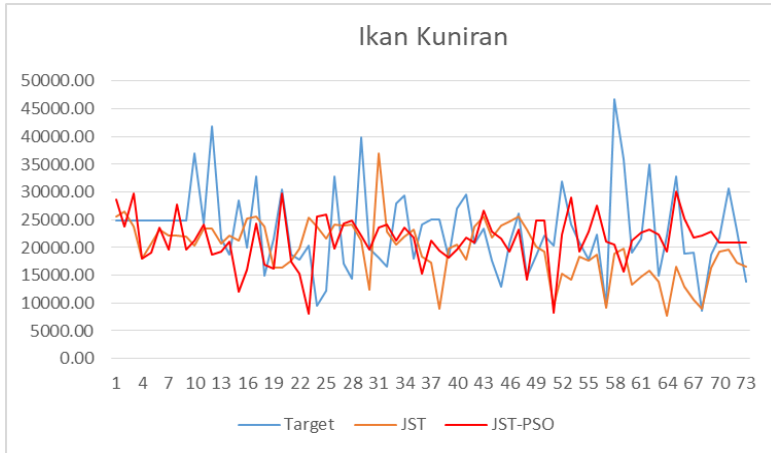
Pada grafik ikan kerapu dan ikan kuniran terlihat bahwa garis keluaran JST-PSO sudah lebih mengikuti garis target dibandingkan grafik pada keluaran prediktor JST. Hal ini menunjukkan bahwa keluaran JST-PSO memiliki nilai optimal dibandingkan pada keluaran JST-LM.



Gambar 4. 14 Hasil Validasi Ikan Biji Nangka Jaringan Syaraf Tiruan – *Particle Swarm Optimization*



Gambar 4. 15 Hasil Validasi Ikan Kerapu Jaringan Syaraf Tiruan
– *Particle Swarm Optimization*



Gambar 4. 16 Hasil Validasi Ikan Kuniran Jaringan Syaraf Tiruan – *Particle Swarm Optimization*

4.5 Prediktor Sebaran Ikan Menggunakan Data *Realtime Buoyweather Type II*

Pengukuran real time dilakukan pada tanggal 16, dan 18 Juni 2019 di Pantai Kenjeran Surabaya. Hasil rata-rata pengukuran dapat dilihat pada Tabel 4.12. Data di pantai kenjeran didapatkan melalui pengukuran langsung oleh penulis. Pengukuran dibantu dengan *buoyweather tipe ii*. Sensor suhu permukaan laut diletakkan di bawah *buoy* agar bisa mengenai air. Sensor curah hujan dan kecepatan angin terletak di atas *buoy* disamping *photovoltaic*. Hasil pengukuran suhu permukaan laut, kecepatan angin dan curah hujan masing-masing dapat dilihat pada kolom dua, tiga, dan empat. Perkiraan jumlah ikan dapat dilihat pada Tabel 4.13.

Prediksi ikan di tengah laut (Laut Jawa) menunjukkan hasil yang berbeda terhadap prediksi real time Pantai Kenjeran. Untuk hasil prediksi di Pantai Kenjeran didapatkan perkiraan jumlah ikan biji nangka, kerapu dan kuniran secara berurutan pada tanggal 18 Juni 2019 adalah sebesar 17076 kg, 410,58 kg, dan 6433,8 kg. Sedangkan hasil keluaran prediksi di Laut Jawa pada tanggal 18 Juni 2019 untuk ikan biji nangka, kerapu, dan kuniran secara berurutan adalah sebesar 9591,8 kg, 1391,7 kg, dan 12343 kg.

Tabel 4. 11 Hasil Pengukuran Sensor dan Prediksi Keluaran dari Prediktor JST-PSO di Pantai Kenjeran

Waktu Pengambilan	Suhu Permukaan Laut (°C)	Kecepatan Angin (m/s)	Curah hujan (mm/hari)	Prediksi Jumlah Tangkapan Ikan (kg)		
				Biji Nangka	Kerapu	Kuniran
16/06/19	28,76	2,20	0,09	13215	495,56	5438,3
18/06/19	29,09	1,38	0,11	17076	410,58	6433,8

Data yang dihasilkan oleh masing-masing sensor akan tersimpan dalam sebuah database yang bernama *thingspeak*. Database ini akan menyimpan setiap perubahan dari nilai yang dihasilkan sensor sehingga didapatkan lebih dari satu nilai untuk

tiap-tap sensor. Selanjutnya akan di rata-rata data-data tersebut sehingga didapatkan data dari sensor untuk per hari nya.

Tabel 4. 12 Hasil Pengukuran Sensor dan Prediksi Keluaran dari Prediktor JST-PSO di Laut Jawa

Waktu Pengambilan	Suhu Permukaan Laut (°C)	Kecepatan Angin (m/s)	Curah Hujan (mm/hari)	Prediksi Jumlah Tangkapan Ikan (kg)		
				Biji Nangka	Kerapu	Kumiran
16/06/19	29,78	2,23	0	16029	739,9	9419,1
18 /06/19	29,74	2,28	0	9591,8	1391,7	12343

Hasil pengukuran di Pantai Kenjeran untuk suhu permukaan laut, kecepatan angin, dan curah hujan masing-masing memiliki nilai berada pada range pemodelan, sehingga prediksi ikan tidak menunjukkan jumlah nol. Hasil analisis data pengukuran real time di pantai Kenjeran yang telah dilakukan menunjukkan bahwa model prediktor yang dibuat tidak bisa digunakan untuk memprediksi jumlah ikan di lautan yang lain, karena range pemodelan yang terbatas. Sehingga perlu dilakukan pemodelan ulang dengan data maksimum dan minimum sesuai standar laut.

BAB V

PENUTUP

1.1 Kesimpulan

Hasil analisis yang telah dilakukan menunjukkan bahwa:

1. Gabungan algoritma Jaringan Saraf Tiruan-*Particle Swarm Optimization* merupakan algoritma yang lebih baik daripada Jaringan Saraf Tiruan untuk memprediksi kelimpahan ikan. JST-PSO mampu menurunkan RMSE data uji JST masing-masing ikan biji nangka, kerapu kuniran adalah sebesar 9.638,6741 kg, 993,9949 kg, dan 8.634,8335 kg.
2. Prediksi ikan biji nangka, ikan kerapu, dan ikan kuniran pada tanggal 18 Juni 2019 di Pantai Kenjeran adalah sebesar 17.076,00 kg; 410,58 kg; dan 6.433,80 kg.

1.2 Saran

Adapun beberapa saran dalam penelitian ini adalah sebagai berikut:

1. Penambahan variasi untuk parameter-parameter di dalam penggunaan metode JST-LM dan JST-PSO agar mendapatkan hasil yang lebih optimal.
2. Pengumpulan data sekunder dilakukan jauh hari serta pengumpulan data dari beberapa tahun terakhir.
3. Mencari korelasi variabel-variabel lain yang berhubungan dengan kelimpahan ikan selain yang disebutkan dalam laporan.
4. Jumlah jenis ikan yang akan diprediksi sebaiknya ditambah variasi ikan yang lain.

Halaman ini memang sengaja dikosongkan

DAFTAR RUJUKAN

- Aliev, A., & Guirimov, B. G. (2014). *Type-2 Fuzzy Neural Networks and Their Applications*. Springer International Publishing Switzerland.
- Badan Penelitian dan Pengembangan Kelautan dan Perikanan. (2014). *Potensi dan Tingkat Pemanfaatan Sumberdaya Ikan di Wilayah Pengelolaan Perikanan Republik Indonesia (WPP RI)*. Ref Graphika, Jakarta.
- Depoinovasi. (2011a). Sensor Curah Hujan (Rain Gauge) Support Arduino. Retrieved July 21, 2018, from <http://depoinovasi.com/produk-960-sensor-curah-hujan-rain-gauge-support-arduino.html>
- Fausett, L. (1994). *Fundamentals of Neural Networks Architecture, Algorithms, and Application*. Prentice Hall.
- Bruno, D. O., Barbini, A. B., Martos P., & Dias J. M (2013). Fish Abundance and Distribution Patterns Related to Enviromental Factors in a Choked Temperate Coastal Lagoon (Argentina). *Brazillian Journal of Oceanography*
- Gao, P., Chen, C., & Qin, S. (2010). An optimization method of *hiddenK nodes* for neural network. *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, 2, 53–56.
- Hobday, A. J., Spillman, C. M., Paige Eveson, J., & Hartog, J. R. (2016). Seasonal forecasting for decision support in marine fisheries and aquaculture. *Fisheries Oceanography*, 25, 45–56. <https://doi.org/10.1111/fog.12083>
- Hamed, H. N., Shamsuddin, S. M., & Salim, N. (2008). *Particle Swarm Optimization For Neural Network Learning Enhancement*. Jurnal Teknologi, 49 Universiti Teknologi Malaysia, 13-26.
- Katsaros, K., & Buettner, K. J. K. (1969). Influence of Rainfall on Temperature and Salinity of the Ocean Surface. *Journal of Applied Meteorology*. [https://doi.org/10.1175/1520-0450\(1969\)008<0015:IOROTA>2.0.CO;2](https://doi.org/10.1175/1520-0450(1969)008<0015:IOROTA>2.0.CO;2)

- Life, M., Jolla, L., & Rotunda, T. (1990). Effects of Turbulent Shear Flow on Zooplankton Distribution. *Deep Sea Research*, 37(3), 447–461.
- Muhtar, A., Mustika, W., & Suharyanto. (2017). The Comparison of ANN-BP and ANN-PSO as Learning Algorithm to Track MPP in PV System. 7th International Annual Engineering Seminar (InAES. National Oceanic and Atmospheric Administration. (2014). Ocean-Oceanic Climate Variables. Retrieved January 5, 2018, from <https://www.climate.gov/maps-data/primer/ocean-oceanic-climate-variables>
- Oseana. (1999). Pengenalan Jenis-Jenis Ikan Laut Ekonomi Penting di Indonesia. Volume XXIV, from www.oseanografi.lipi.go.id
- Parra, Hugo E., Pham, Christopher K., Menezes, Gui, & Rosa, Alexandra. Predictive Modelling of deep-sea fish distribution in the Azores. <http://dx.doi.org/10.1016/j.dsr2.2016.01.004>
- Pitono, J. 2015. *Optimasi Aliran Daya Berbasis Particle Swarm Optimization (PSO) untuk Menekan Emisi dan Biaya Operasional*. Disertasi Teknik Elektro ITS Surabaya.
- Putra, B. W., Arifin, S., & Aisjah, S. A. (2014). Perancangan prediktor suhu permukaan laut menggunakan interpolasi lagrange dan logika fuzzy di perairan laut jawa. *Teknik POMITS*, 3(1), 1–6.
- Samantha, G. (2013). Terbaru: Panjang Garis Pantai Indonesia Capai 99.000 Kilometer. Retrieved January 15, 2018, from <http://nationalgeographic.co.id/berita/2013/10/terbaru-panjang-garis-pantai-indonesia-capai-99000-kilometer>
- Sharma, B., & K. Venugopalan, P. (2014). Comparison of Neural Network Training Functions for Hematoma Classification in Brain CT Images. *IOSR Journal of Computer Engineering*, 16(1), 31–35. <https://doi.org/10.9790/0661-16123135>
- Soesono, S. (1974). *Limnologi*. Dirjen Perikanan Departemen Pertanian Institut Pertanian Bogor.

- Semuaikan. (2018). Morfologi dan Klasifikasi Ikan Cucut (Rhizoprionodon acutus). Retrieved July 22, 2018, from <http://www.semuaikan.com/klasifikasi-dan-morfologi-ikan-cucut/>
- Torres-Faurrieta, L. K., Dreyfus-León, M. J., & Rivas, D. (2016). Recruitment forecasting of yellowfin tuna in the eastern Pacific Ocean with artificial neuronal networks. *Ecological Informatics*, 36, 106–113. <https://doi.org/10.1016/j.ecoinf.2016.10.005>
- Wadi, F. (2016). Ikan Biji Nangka. Retrieved June 8, 2018, from <https://www.biodiversitywarriors.org/isi-katalog.php?idk=5595&judul=ikan-biji-nangka>
- Wellyantama, P., Meteorologi, S., Pontianak, M., & Utara, L. (2015). *Utara Jakarta dengan Metode Jaringan Syaraf Tiruan Propagasi Balik*. V(1), 30–35.
- Yáñez, E., Plaza, F., Gutiérrez-Estrada, J. C., Rodríguez, N., Barbieri, M. A., Pulido-Calvo, I., & Bórquez, C. (2010). Anchovy (*Engraulis ringens*) and sardine (*Sardinops sagax*) abundance forecast off northern Chile: A multivariate ecosystemic neural network approach. *Progress in Oceanography*, 87(1–4), 242–250. <https://doi.org/10.1016/j.pocean.2010.09.015>
- Yu, H., & Wilamowski, B. M. (2011). *LevenbergMarquardt Training Industrial Electronics Handbook CHapter 12*. In H. Yu, & B. M. Wilamowski, *LevenbergMarquardt Training Industrial Electronics Handbook, Vol. 5 Intelligent Systems, 2nd Edition (pp. 12-1 to 12-15)*. CRS Press.

Halaman ini memang sengaja dikosongkan

LAMPIRAN A Nilai RMSE

Jaringan Saraf Tiruan						
HN	Ikan Biji Nangka		Ikan Kerapu		Ikan Kuniran	
	RMSE Data Latih	RMSE Data Uji	RMSE Data Latih	RMSE Data Uji	RMSE Data Latih	RMSE Data Uji
1	8062.7	9664.6	1243.1	1931.4	11244.3	9861.7
2	7998.8	10535.9	1149.4	1942.7	11019.0	8623.6
3	7577.7	11634.7	1144.7	1071.8	10413.3	16085.7
4	7395.4	14031.8	1097.7	1842.8	10296.2	21012.9
5	7522.4	15064.9	1091.7	1289.1	10222.6	19914.0
6	7217.4	16125.3	1097.2	3359.9	10053.3	15124.7
7	6934.8	11060.9	1089.3	1401.4	9414.4	12233.2
8	6841.4	20282.9	1042.7	2270.3	9756.3	21310.3
9	6943.7	10759.1	1036.6	5669.5	9733.4	20931.4
10	6446.4	15747.4	987.3	1534.8	9468.0	26987.0
Jaringan Saraf Tiruan – Particle Swarm Optimization						
HN	Ikan Biji Nangka		Ikan Kerapu		Ikan Kuniran	
	RMSE Data Latih	RMSE Data Uji	RMSE Data Latih	RMSE Data Uji	RMSE Data Latih	RMSE Data Uji
1	7987.5	9726.7	1218.3	856.7	10093.6	8634.2
2	8046.4	9894.5	1228.7	993.9	10123.6	8790.1
3	22929.8	18396.5	2218.5	1252.1	26974.5	13425.0
4	23542.3	18698.1	2292.0	1318.6	27630.5	14974.2
5	8038.3	9638.6	1239.5	858.1	10160.2	8647.8
6	8027.3	9798.7	1216.6	873.3	10197.1	9995.8
7	8013.5	9840.8	1217.6	880.2	10552.8	8852.5
8	8022.8	9701.8	1244.1	887.7	10135.1	8563.6
9	8282.6	9731.1	1222.6	843.1	10174.1	8608.2
10	8415.1	10296.8	1230.3	947.4	10106.3	8501.7

LAMPIRAN B Source Code Matlab Jaringan Syaraf Tiruan Pelatihan

```

clc;
clear;
close all;
warning off;
% proses pembacaan data latih dari excel
dengan data normalisasi
filename = 'DATA BIJI NANGKA.xlsx';
sheet = 1;
xlRange = 'A2:D293';
Data = xlsread(filename, sheet, xlRange);
input_latih = Data(:,1:3)'; %akses kolom 1-3
target_latih = Data(:,4)'; %akses kolom 4
[m,n] = size(input_latih);

%max dan min data arus laut(target)
max_data = 45927;
min_data = 0;

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

%parameter pelatihan
net.performFcn = 'mse';
net.trainParam.goal = 0.001;
net.trainParam.show = 20;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.95;
net.trainParam.lr = 0.001;

```

```

%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 BIJI NANGKA.xlsx';
sheet = 1;
xlRange = 'M2:M293';
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
on');

%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([' (RMSE= ',...
num2str(RMSE),')']));
xlabel('Hari ke-');
ylabel('Tangkapan Ikan (kg)');
legend('Keluaran JST','Target');

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

```

LAMPIRAN C Source Code Matlab Jaringan Syaraf Tiruan Pengujian

```
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 BIJI NANGKA.xlsx';
sheet = 2;
xlRange = 'A2:D74';
Data = xlsread(filename, sheet, xlRange);
input_uji = Data(:,1:3)'; %akses kolom 1-3
target_uji = Data(:,4)'; %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 = 45927;
min_data = 0;
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 BIJI NANGKA.xlsx';
sheet = 2;
xlRange = 'M2:M74';
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([' (MAPE= ',...
num2str(MAPE),'%'),' (RMSE= ',...
num2str(RMSE),'')]);

xlabel('Hari ke-');
ylabel('Tangkapan Ikan (kg)');
legend('Output JST','Target');

```


**LAMPIRAN D Source Code Matlab Fungsi Objektif
Jaringan Saraf Tiruan – *Genetic Algorithm***

```
function RMSE=objf5_nangka(x)

%pembacaan data dari excel
inputs=xlsread('DATA PREDIKSI
REVISI.xlsx',1,'A2:D293');
targets=xlsread('DATA PREDIKSI
REVISI.xlsx',1,'E2:E293');

data=292 ;

%1 partikel mengandung seluruh bobot (5
hidden nodes = 26 bobot)
t1=x(1);
t2=x(2);
t3=x(3);
t4=x(4);

t5=x(5);
t6=x(6);
t7=x(7);
t8=x(8);

t9=x(9);
t10=x(10);
t11=x(11);
t12=x(12);

t13=x(13);
t14=x(14);
t15=x(15);
t16=x(16);
```

```
t17=x(17);
t18=x(18);
t19=x(19);
t20=x(20);
t21=x(21);
t22=x(22);
t23=x(23);
t24=x(24);
t25=x(25);
t26=x(26);


a1= inputs(1:data,1)*t1;
b1= inputs(1:data,2)*t2;
c1= inputs(1:data,3)*t3;
d1= ones(data,1)*t4;

hn1= [a1 b1 c1 d1];

a2= inputs(1:data,1)*t5;
b2= inputs(1:data,2)*t6;
c2= inputs(1:data,3)*t7;
d2= ones(data,1)*t8;

hn2= [a2 b2 c2 d2];

a3= inputs(1:data,1)*t9;
b3= inputs(1:data,2)*t10;
c3= inputs(1:data,3)*t11;
d3= ones(data,1)*t12;

hn3= [a3 b3 c3 d3];
```

```
a4= inputs(1:data,1)*t13;
b4= inputs(1:data,2)*t14;
c4= inputs(1:data,3)*t15;
d4= ones(data,1)*t16;

hn4= [a4 b4 c4 d4];

a5= inputs(1:data,1)*t17;
b5= inputs(1:data,2)*t18;
c5= inputs(1:data,3)*t19;
d5= ones(data,1)*t20;

hn5= [a5 b5 c5 d5];

%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

%hidden ke output
fhn1=tansig(nilaihn1)*t21;
fhn2=tansig(nilaihn2)*t22;
fhn3=tansig(nilaihn3)*t23;
fhn4=tansig(nilaihn4)*t24;
fhn5=tansig(nilaihn5)*t25;
fhn6=ones(data,1)*t26;

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

for i=1:data;
    nilaion(i,1)=sum(matrix(i,1:6));
end

%fungsi aktivasi output layer
prediksi=purelin(nilaion);

max_data= 45927;
min_data=0;
prediksi_asli=((prediksi - 0.1)*(max_data-
min_data))/0.8)+min_data;

% pembacaan data target asli
target_asli = xlsread('DATA PREDIKSI
REVISI.xlsx',1,'G2:G293');

%objective function
selisih = prediksi_asli - target_asli;

```

```
selisihkuadrat = selisih.^2;  
RMSE = sqrt((sum(selisihkuadrat))/data);
```

LAMPIRAN E Source Code Matlab Jaringan Saraf Tiruan – *Particle Swarm Optimization* Pelatihan

```

variabelinput = xlsread('DATA PREDIKSI
BIJI.xlsx', 'Pengujian', 'A2:D71');
validasi = xlsread('DATA PREDIKSI
BIJI.xlsx', 'Pengujian', 'G2:G71');

min_design_variabel = [0.304386116262054 -
0.289280440019286 -0.202156722589864 -
0.204121258559901 -0.332991931414741
0.137568840364363 -0.127058464315827
0.558857866232162 -0.355854911448230 -
0.0305164978507796 0.0254726967101908
0.453949702542299 -0.255062103460363 -
0.774584374851462 -0.0660299720072467 -
0.649926484497131 -0.498636707775369 -
0.0685632921052194 0.3854413444953945 -
0.261081371547499 -0.0826679679889874 -
0.128339512668493 0.3903064548893482 -
0.426074863825978 0.322488460077962
0.0183650702337309];

data = 70;

% 1 individu = seluruh bobot
t1=min_design_variabel(:,1)';      %input
variabel independen SPL ke hidden node 1
t2=min_design_variabel(:,2)';      %input
variabel independen Salinitas ke hidden
node 1
t3=min_design_variabel(:,3)';      %input
variabel independen Curah Hujan ke hidden
node 1

```

```

t4=min_design_variabel(:,4)';      %bias ke
hidden node 1

t5=min_design_variabel(:,5)';      %input
variabel independen SPL ke hidden node 2
t6=min_design_variabel(:,6)';      %input
variabel independen Salinitas ke hidden
node 2
t7=min_design_variabel(:,7)';      %input
variabel independen Curah Hujan ke hidden
node 2
t8=min_design_variabel(:,8)';      %bias ke
hidden node 2

t9=min_design_variabel(:,9)';      %input
variabel independen SPL ke hidden node 3
t10=min_design_variabel(:,10)';    %input
variabel independen Salinitas ke hidden
node 3
t11=min_design_variabel(:,11)';    %input
variabel independen Curah Hujan ke hidden
node 3
t12=min_design_variabel(:,12)';    %bias ke
hidden node 3

t13=min_design_variabel(:,13)';    %input
variabel independen SPL ke hidden node 4
t14=min_design_variabel(:,14)';    %input
variabel independen Salinitas ke hidden
node 4
t15=min_design_variabel(:,15)';    %input
variabel independen Curah Hujan ke hidden
node 4

```

```

t16=min_design_variabel(:,16)';    %bias ke
hidden node 4

t17=min_design_variabel(:,17)';    %input
variabel independen SPL ke hidden node 5
t18=min_design_variabel(:,18)';    %input
variabel independen Salinitas ke hidden
node 5
t19=min_design_variabel(:,19)';    %input
variabel independen Curah Hujan ke hidden
node 5
t20=min_design_variabel(:,20)';    %bias ke
hidden node 5

t21=min_design_variabel(:,21)';    %hidden
node 1 ke output
t22=min_design_variabel(:,22)';    %hidden
node 2 ke output
t23=min_design_variabel(:,23)';    %hidden
node 3 ke output
t24=min_design_variabel(:,24)';    %hidden
node 4 ke output
t25=min_design_variabel(:,25)';    %hidden
node 5 ke output
t26=min_design_variabel(:,26)';    %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 = ones(data,1)*t4;
%output bias
hn1 = [a1 b1 c1 d1];

```



```
a2 = variabelinput(1:data,1)*t5;
b2 = variabelinput(1:data,2)*t6;
c2 = variabelinput(1:data,3)*t7;
d2 = ones(data,1)*t8;
hn2 = [a2 b2 c2 d2];

a3 = variabelinput(1:data,1)*t9;
b3 = variabelinput(1:data,2)*t10;
c3 = variabelinput(1:data,3)*t11;
d3 = ones(data,1)*t12;
hn3 = [a3 b3 c3 d3];

a4 = variabelinput(1:data,1)*t13;
b4 = variabelinput(1:data,2)*t14;
c4 = variabelinput(1:data,3)*t15;
d4 = ones(data,1)*t16;
hn4 = [a4 b4 c4 d4];

a5 = variabelinput(1:data,1)*t17;
b5 = variabelinput(1:data,2)*t18;
c5 = variabelinput(1:data,3)*t19;
d5 = ones(data,1)*t20;
hn5 = [a5 b5 c5 d5];

% 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));
%Langkah 4 JST
end;

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

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

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

% Hidden ke output
fhn1 = tansig(nilaihn1)*t21;
fhn2 = tansig(nilaihn2)*t22;
fhn3 = tansig(nilaihn3)*t23;
fhn4 = tansig(nilaihn4)*t24;
fhn5 = tansig(nilaihn5)*t25;
fhn6 = ones(data,1)*t26; %bias pada output
layer

```

```

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

for i = 1:data;
    nilaion(i,1)= sum(matrix(i,1:6));
    %Langkah 5 JST
end;

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

max_data = 45927;           %Ganti nilai max
min_data = 0;              %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_Data_Uji =
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(['RMSE =  
,num2str(RMSE_Data_Uji), ' dan MAPE =  
,num2str(MAPE), '%'])))  
xlabel('Hari ke-')  
ylabel('Tangkapan Ikan (kg)')  
legend('Keluaran JST-  
PSO', 'Target', 'Location', 'Best')  
  
%saveas(gcf, 'Data Latih Ikan Biji Nangka HN  
5 PSO.png')
```

**LAMPIRAN E Source Code Matlab Jaringan Saraf Tiruan –
Particle Swarm Optimization Pengujian**

```

variabelinput = xlsread('DATA PREDIKSI
BIJI.xlsx','Pengujian','A2:D71');
validasi = xlsread('DATA PREDIKSI
BIJI.xlsx','Pengujian','G2:G71');

min_design_variabel = [0.304386116262054 -
0.289280440019286 -0.202156722589864 -
0.204121258559901 -0.332991931414741
0.137568840364363 -0.127058464315827
0.558857866232162 -0.355854911448230 -
0.0305164978507796 0.0254726967101908
0.453949702542299 -0.255062103460363 -
0.774584374851462 -0.0660299720072467 -
0.649926484497131 -0.498636707775369 -
0.0685632921052194 0.385441344953945 -
0.261081371547499 -0.0826679679889874 -
0.128339512668493 0.390306454893482 -
0.426074863825978 0.322488460077962
0.0183650702337309];

data = 70;

% 1 individu = seluruh bobot
t1=min_design_variabel(:,1)';      %input
variabel independen SPL ke hidden node 1
t2=min_design_variabel(:,2)';      %input
variabel independen Salinintas ke hidden
node 1
t3=min_design_variabel(:,3)';      %input
variabel independen Curah Hujan ke hidden
node 1

```

```
t4=min_design_variabel(:,4)';      %bias ke  
hidden node 1  
  
t5=min_design_variabel(:,5)';      %input  
variabel independen SPL ke hidden node 2  
t6=min_design_variabel(:,6)';      %input  
variabel independen Salinitas ke hidden  
node 2  
t7=min_design_variabel(:,7)';      %input  
variabel independen Curah Hujan ke hidden  
node 2  
t8=min_design_variabel(:,8)';      %bias ke  
hidden node 2  
  
t9=min_design_variabel(:,9)';      %input  
variabel independen SPL ke hidden node 3  
t10=min_design_variabel(:,10)';    %input  
variabel independen Salinitas ke hidden  
node 3  
t11=min_design_variabel(:,11)';    %input  
variabel independen Curah Hujan ke hidden  
node 3  
t12=min_design_variabel(:,12)';    %bias ke  
hidden node 3  
  
t13=min_design_variabel(:,13)';    %input  
variabel independen SPL ke hidden node 4  
t14=min_design_variabel(:,14)';    %input  
variabel independen Salinitas ke hidden  
node 4  
t15=min_design_variabel(:,15)';    %input  
variabel independen Curah Hujan ke hidden  
node 4
```

```

t16=min_design_variabel(:,16)';    %bias ke
hidden node 4

t17=min_design_variabel(:,17)';    %input
variabel independen SPL ke hidden node 5
t18=min_design_variabel(:,18)';    %input
variabel independen Salinitas ke hidden
node 5
t19=min_design_variabel(:,19)';    %input
variabel independen Curah Hujan ke hidden
node 5
t20=min_design_variabel(:,20)';    %bias ke
hidden node 5

t21=min_design_variabel(:,21)';    %hidden
node 1 ke output
t22=min_design_variabel(:,22)';    %hidden
node 2 ke output
t23=min_design_variabel(:,23)';    %hidden
node 3 ke output
t24=min_design_variabel(:,24)';    %hidden
node 4 ke output
t25=min_design_variabel(:,25)';    %hidden
node 5 ke output
t26=min_design_variabel(:,26)';    %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 = ones(data,1)*t4;
%output bias
hn1 = [a1 b1 c1 d1];

```

```

a2 = variabelinput(1:data,1)*t5;
b2 = variabelinput(1:data,2)*t6;
c2 = variabelinput(1:data,3)*t7;
d2 = ones(data,1)*t8;
hn2 = [a2 b2 c2 d2];

a3 = variabelinput(1:data,1)*t9;
b3 = variabelinput(1:data,2)*t10;
c3 = variabelinput(1:data,3)*t11;
d3 = ones(data,1)*t12;
hn3 = [a3 b3 c3 d3];

a4 = variabelinput(1:data,1)*t13;
b4 = variabelinput(1:data,2)*t14;
c4 = variabelinput(1:data,3)*t15;
d4 = ones(data,1)*t16;
hn4 = [a4 b4 c4 d4];

a5 = variabelinput(1:data,1)*t17;
b5 = variabelinput(1:data,2)*t18;
c5 = variabelinput(1:data,3)*t19;
d5 = ones(data,1)*t20;
hn5 = [a5 b5 c5 d5];

% 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));
    %Langkah 4 JST
end;

```



```

for i = 1:data;
    nilaihn3(i,1) = sum(hn3(i,1:4));
%Langkah 4 JST
end;
for i = 1:data;
    nilaihn4(i,1) = sum(hn4(i,1:4));
%Langkah 4 JST
end;
for i = 1:data;
    nilaihn5(i,1) = sum(hn5(i,1:4));
%Langkah 4 JST
end;

% Hidden ke output
fhn1 = tansig(nilaihn1)*t21;
fhn2 = tansig(nilaihn2)*t22;
fhn3 = tansig(nilaihn3)*t23;
fhn4 = tansig(nilaihn4)*t24;
fhn5 = tansig(nilaihn5)*t25;
fhn6 = ones(data,1)*t26; %bias pada output
layer
matrix =[fhn1,fhn2,fhn3,fhn4,fhn5,fhn6];
for i = 1:data;
    nilaion(i,1)= sum(matrix(i,1:6));
%Langkah 5 JST
end;

% Fungsi aktivasi output layer
prediksi = purelin(nilaion);
max_data = 45927; %Ganti nilai max
min_data = 0; %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_Data_Uji =
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(['RMSE =
', num2str(RMSE_Data_Uji), ' dan MAPE =
', num2str(MAPE), '%']))
xlabel('Hari ke-')
ylabel('Tangkapan Ikan (kg)')
legend('Keluaran JST-
PSO', 'Target', 'Location', 'Best')
saveas(gcf, 'Data Uji Ikan Biji Nangka HN 5
PSO.png')
```


BIODATA PENULIS



Penulis mempunyai nama lengkap Adelia Wahyu Puspita Sari, lahir di Tuban pada tanggal 29 Juli 1997, merupakan anak ke pertama dari dua bersaudara. Penulis menempuh pendidikan sekolah dasar di SD Negeri Sidokumpul 2 Gresik, pendidikan menengah di SMP Negeri 1 Gresik, dan melanjutkan pendidikan menengah atas di SMA Negeri 1 Gresik yang kemudian dilanjutkan ke ITS Jurusan Teknik Fisika pada tahun 2015. Hingga pada tahun 2019 penulis akhirnya melaksanakan Tugas Akhir sebagai syarat untuk menyelesaikan studi S1 di Departemen Teknik Fisika dengan judul “Rancang Bangun Prediktor Sebaran Ikan Menggunakan Metode Jaringan Syaraf Tiruan Dengan Optimasi *Particle Swarm Optimization*”. Bagi pembaca yang memiliki saran, kritik, atau ingin berdiskusi lebih lanjut tentang Tugas Akhir ini, maka bisa menghubungi penulis melalui adeliawahyupuspitasari@gmail.com.