

IMPLEMENTASI JARINGAN SARAF TIRUAN METODE BACKPROPAGATION UNTUK MEMPREDIKSI JUMLAH NILAI EKSPOR DI PROVINSI NTB

*(Implementation of Artificial Neural Network Algorithm Backpropagation Method
to Predict Amount of Export Value in the NTB Province)*

Biondi Bagasta Wiko Putra*, Moh. Ali Albar dan Budi Irmawati
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram.
Jl. Majapahit 62, Mataram, Lombok, NTB-INDONESIA.
Email: siapaayo39@gmail.com, mohalialbar@unram.ac.id, budi-i@unram.ac.id

Abstract

This paper presents the application of the Backpropagation method of the Artificial Neural Network algorithm in the case study to predict the amount of export value in NTB province. This forecasting process uses two scenarios namely forecasting the total value of total exports in NTB province and forecasting the amount of export value based on commodity. The activation functions used are Sigmoid Binary and ReLU - Linear. Based on results of the system tests that have been carried out, Sigmoid Binary activation function shows that the best network architecture is 12-4-1, best learning rate is 0.3 and best error limit is 0.0015, which in test phase produces an MSE value of 0.0161 and the MAPE value of 30.53%, while ReLU-Linear activation function shows that the best network architecture is 12-5-1, best learning rate is 0.1 and best error limit is 0.0012, which is at the test phase produce MSE value of 0.0309 and the MAPE value of 53.04%. Because the MAPE value generated in the system testing using each activation function does not have a value of less than 20%, the use of Backpropagation in the study is not suitable.

Keyword : Artificial Neural Network, Backpropagation, Value of Export, Forecasting, Activation Function

**Penulis Korespondensi*

1. PENDAHULUAN

Provinsi NTB terkenal akan komoditi-komoditi ekspornya seperti gerabah, kerajinan buah kering, produk pertanian terutama jagung, mutiara dan konsentrat tembaga. Berdasarkan data Badan Pusat Statistik (BPS) provinsi NTB pada bulan Juni 2018, nilai ekspor Provinsi NTB sebesar US\$ 78.225.629 dan mengalami kenaikan sebesar 75.44% jika dibandingkan dengan nilai ekspor pada bulan Mei 2018 yang bernilai US\$ 44.588.299. Kenaikan nilai ekspor provinsi NTB yang sangat pesat ini menunjukkan bahwa provinsi NTB berpotensi menjadi salah satu provinsi terbesar untuk menopang ekspor Indonesia [1].

Pada era digital saat ini perkembangan teknologi semakin pesat, dikarenakan orang-orang pada berlomba-lomba untuk menciptakan teknologi yang terbaru. Salah satu contoh teknologi yang perkembangannya semakin pesat saat ini adalah kecerdasan buatan. Saat ini kecerdasan buatan sudah bisa diterapkan berbagai macam studi kasus terutama

dalam studi kasus peramalan dengan berbagai macam algoritma yang digunakan. Salah satu algoritma yang biasa digunakan dalam studi kasus peramalan adalah algoritma jaringan saraf tiruan dengan metode *backpropagation*. Algoritma jaringan saraf tiruan dengan metode *backpropagation* diterapkan pada penelitian ini karena berdasarkan kemampuan belajar yang dimilikinya algoritma jaringan saraf tiruan dengan metode *backpropagation* dapat dilatih untuk mengenali, mempelajari dan menganalisa pola-pola data masa lalu, sehingga dengan pola data lalu tersebut dapat dicari suatu formula atau fungsi yang dapat menghubungkan pola data pada masa lalu dengan keluaran yang diinginkan pada saat ini, menghitung kesalahan pada keluaran yang didapatkan dan melakukan pembaharuan pola data pada setiap iterasi berdasarkan pelatihan data yang telah dilakukan untuk mengurangi tingkat kesalahan sehingga dapat meningkatkan akurasi pada keluaran [2].

Data yang digunakan dalam penelitian ini yaitu jumlah nilai ekspor di provinsi NTB selama lima tahun mulai dari tahun 2014 sampai dengan tahun 2018 yang dicatat setiap bulannya oleh Dinas Perdagangan provinsi NTB, yang mana pada data tersebut terdapat data jumlah nilai ekspor per komoditi dan data jumlah nilai ekspor total dari semua komoditi setiap bulannya di provinsi NTB. Data jumlah nilai ekspor per komoditi digunakan untuk memprediksi nilai ekspor komoditi dan data jumlah nilai ekspor total dari semua komoditi digunakan untuk memprediksi jumlah nilai ekspor di provinsi NTB, sehingga pada penelitian ini terdapat dua skenario yaitu memprediksi jumlah nilai ekspor di provinsi NTB dan memprediksi jumlah nilai ekspor suatu komoditi setiap bulannya. Prediksi jumlah nilai ekspor ini dilakukan dengan tujuan untuk mengetahui seberapa akurat algoritma Jaringan Saraf Tiruan metode *Backpropagation* dalam meramalkan jumlah nilai ekspor di provinsi NTB dan hasil prediksi jumlah nilai ekspor ini diharapkan dapat bermanfaat bagi Dinas Perdagangan provinsi NTB sebagai gambaran jumlah nilai ekspor di provinsi NTB untuk masa yang akan datang. Data Untuk skenario memprediksi jumlah nilai ekspor suatu komoditi hanya akan digunakan komoditi yang data jumlah nilai ekspornya lengkap dari tahun 2014 sampai tahun 2018 seperti Gerabah, Kerajinan Buah Kering, Mutiara dan Konsentrat Tembaga.

2. TINJAUAN PUSTAKA

Tulisan dari Saragih [3] yang berisi penelitian tentang akurasi dari algoritma jaringan saraf tiruan metode *backpropagation* dalam memprediksi jumlah nilai ekspor di provinsi Sumatera Utara dengan data dari tahun 2012 sampai tahun 2017. Penelitiannya menggunakan lima model arsitektur Jaringan Saraf Tiruan yakni model 4-5-1, 4-7-1, 4-9-1, 4-10-1 dan 4-11-1. Model terbaik dari ke lima model yang diuji adalah model 4-7-1 dengan menghasilkan tingkat akurasi 100%, dengan waktu 27 detik. Batas *error* yang digunakan 0,001 – 0,05, dengan demikian maka model 4-7-1 cukup bagus untuk memprediksi Nilai Ekspor di Sumatera Utara, karena keakurasiannya mencapai 100%.

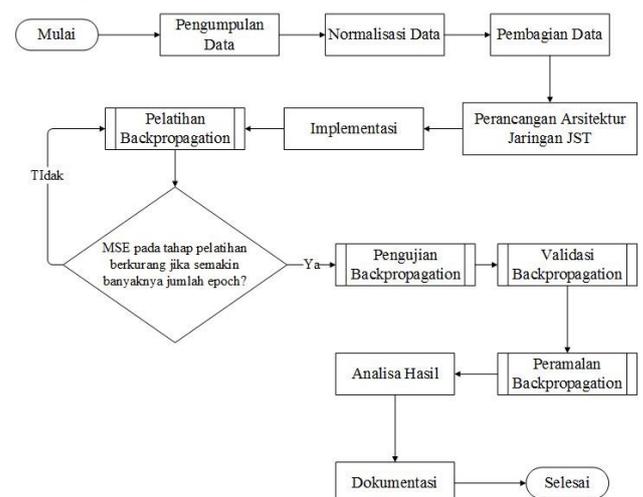
Selain itu terdapat tulisan dari Anggriningrum [4] yang berisi penelitian tentang perbandingan antara algoritma ARIMA (*Autoregressive Integrated Moving Average*) dengan algoritma Jaringan Saraf Tiruan metode *Backpropagation* dalam memprediksi harga saham PT. Asuransi Bina Dana Arta (ABDA) Tbk. Arsitektur jaringan yang digunakan dengan algoritma jaringan saraf tiruan metode *backpropagation* sendiri

adalah 1-1-1 sedangkan untuk algoritma ARIMA sendiri menggunakan model 1-1-0 (*difference first-order autoregressive model*). Berdasarkan hasil simulasi untuk algoritma Jaringan Saraf Tiruan metode *Backpropagation* sendiri menghasilkan nilai MSE (*Mean Squared Error*) dari tahap pelatihan sebesar 0,000206 dan nilai MSE untuk tahap pengujian sebesar 0,00140, sedangkan untuk algoritma ARIMA sendiri menghasilkan nilai MSE sebesar 0,01145. Karena selisih nilai MSE dari kedua metode tidak terlalu besar, maka kedua metode dapat digunakan untuk penelitian prediksi harga saham.

Selanjutnya terdapat tulisan Sutawinaya [5] yang berisi penelitian tentang perbandingan antara algoritma jaringan saraf tiruan metode *Backpropagation* dengan algoritma jaringan saraf tiruan metode ADALINE (*Adaptive Linear Neuron*) dalam memprediksi curah hujan di kota Denpasar. Arsitektur jaringan yang digunakan untuk masing-masing metode adalah 5-5-1. Berdasarkan hasil pengujian yang dilakukan sebanyak 1000 iterasi dihasilkan MSE sebesar 0,0019 untuk metode *Backpropagation* sedangkan untuk metode ADALINE dihasilkan MSE sebesar 0,0045, sehingga dapat disimpulkan bahwa metode *Backpropagation* lebih baik dari metode ADALINE dalam memprediksi curah hujan di kota Denpasar.

3. METODE PENELITIAN

Secara umum alur proses penelitian ini ditunjukkan pada gambar berikut :



Gambar 1. Alur proses penelitian

Penjelasan setiap langkah alur proses penelitian ini adalah pada sub bab berikut.

3.1. Pengumpulan Data

Untuk mendapatkan data-data yang dibutuhkan dalam membantu berjalannya proses penelitian ini, dilakukan dengan cara studi literatur.

3.2. Normalisasi Data

Data yang bersifat masih mentah harus dinormalisasi terlebih dahulu untuk meningkatkan performa pada jaringan. Ketika dilakukan normalisasi, data bisa ditransformasikan ke interval antara 0,1 sampai 0,9. Proses normalisasi selalu menghasilkan data terkecil yang bernilai 0,1 dan untuk data yang terbesar selalu bernilai 0,9. Berikut ini merupakan rumus untuk melakukan normalisasi data :

$$x' = \frac{0,8(x - a)}{b - a} + 0,1 \quad (1)$$

dimana :

- x' : Hasil normalisasi data.
- x : Data yang akan dinormalisasi.
- a : Data terkecil.
- b : Data terbesar.

3.3. Pembagian Data

Data yang diperoleh berupa data historis jumlah nilai ekspor di provinsi NTB yang didapatkan dari Dinas Perdagangan Provinsi NTB yang dicatat setiap bulannya dari tahun 2014 sampai dengan tahun 2018, akan dibagi menjadi empat bagian data yaitu data yang akan digunakan untuk tahap pelatihan, pengujian, validasi dan peramalan *backpropagation*.

3.4. Perancangan Arsitektur Jaringan JST

Pada tahap ini dirancang dan diteliti arsitektur jaringan yang akan digunakan pada tahap pelatihan, pengujian dan validasi untuk mencari yang mana merupakan arsitektur jaringan yang terbaik. Selanjutnya arsitektur jaringan yang terbaik tersebut akan digunakan untuk melakukan peramalan *backpropagation*. Adapun arsitektur jaringan yang digunakan dan diteliti yaitu arsitektur jaringan 12-1-1 (12 buah *neuron input layer*, satu buah *neuron hidden layer* dan satu buah *neuron output layer*), 12-2-1, 12-3-1, 12-4-1 dan 12-5-1 dengan tujuan untuk mengetahui apakah jumlah *neuron* pada *hidden layer* sangat mempengaruhi keakuratan hasil peramalan. Jumlah *hidden layer* yang digunakan hanya satu saja karena penggunaan jaringan dengan lebih dari satu *hidden layer* dalam kasus peramalan tidak memberikan hasil peramalan yang berbeda jauh dengan menggunakan satu lapisan dan dengan semakin banyaknya *hidden layer* atau semakin banyaknya *neuron* yang digunakan juga akan memperlambat proses pelatihan [6].

3.5. Implementasi

Pada penelitian ini dilakukan pengimplementasian algoritma jaringan saraf tiruan metode *backpropagation* kedalam sebuah bahasa pemrograman. Bahasa pemrograman yang digunakan adalah *Python* versi 3.6 dengan *library* seperti *Numpy*, dan *Pandas*. Proses *coding* ini dilakukan dengan empat tahap yaitu *coding* untuk tahap pelatihan, pengujian, validasi dan peramalan. Pada tahap pelatihan dilakukan perulangan mulai dari fase propagasi maju hingga fase propagasi balik berdasarkan jumlah perulangan yang digunakan. Pada tahap pengujian dilakukan hanya pada fase propagasi maju saja karena hanya mengujikan bobot yang dihasilkan pada tahap pelatihan. Pada tahap validasi dilakukan percobaan untuk meramalkan jumlah nilai ekspor di provinsi NTB pada tahun 2018, meskipun data jumlah nilai ekspor di provinsi NTB sudah ada. Pada tahap peramalan dilakukan peramalan jumlah nilai ekspor di provinsi NTB untuk masa yang akan datang yaitu pada tahun 2019 dan seterusnya. Fungsi aktivasi yang digunakan adalah *Sigmoid Biner*, ReLU dan *Linear*. Berikut ini merupakan rumus untuk masing – masing fungsi aktivasi :

1. Sigmoid Biner

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

dimana:

- y = Keluaran pada jaringan.
- x = Hasil perkalian bobot dan *input* ditambah bias.
- e = Bilangan eksponensial.

2. ReLU (*Rectified Linear Unit*)

$$f(x) = \max(0, x) \quad (3)$$

dimana :

- x = Hasil perkalian bobot dan *input* ditambah bias.
- \max = Nilai maksimum

3. Linear

$$y = x \quad (4)$$

dimana :

- x = Hasil perkalian bobot dan *input* ditambah bias.
- Berikut ini merupakan algoritma pelatihan *Backpropagation* [7] :

Langkah 1 : Definisikan pola data yang digunakan sebagai masukan dan target pelatihan.

Langkah 2 : Inisialisasi nilai bobot awal dan *bias* dengan bilangan acak dengan rentang [0, 1].

Langkah 3 : Tentukan jumlah *epoch*, target *error*, dan *learning rate* yang akan digunakan. *Epoch* adalah proses perhitungan pelatihan *Backpropagation* mulai dari tahap propagasi maju (*forward pass*) hingga propagasi balik (*backward pass*). Satu *epoch* berarti proses perhitungan hanya pada tahap propagasi maju sampai propagasi balik saja, sedangkan dua epoch berarti proses perhitungan mulai dari tahap propagasi maju hingga propagasi balik, setelah itu dilanjutkan lagi tahap propagasi maju hingga propagasi balik dengan menggunakan bobot baru yang didapatkan pada *epoch* sebelumnya sehingga terhitung terdapat dua *epoch*. Begitu juga seterusnya untuk *epoch* yang lebih tinggi.

Langkah 4 : Kerjakan langkah-langkah selanjutnya selama *epoch* sudah mencapai target jumlah *epoch* dan nilai *error* sudah mencapai nilai batas *error* yang digunakan. Target *error* adalah nilai *error* yang diinginkan untuk menghentikan proses pelatihan. Jika target *error* sudah memenuhi walaupun *epoch* belum selesai, maka proses pelatihan tetap bisa dihentikan.

Fase I : Propagasi Maju.

Langkah 5 : Jumlahkan semua sinyal dari *neuron input layer* yang masuk ke *neuron hidden layer* (z_{net_j}) ($x_i, i = 1,2,3,\dots,n$)($v_j, j = 1,2,3,\dots,n$).

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (5)$$

dimana :

z_{net_j} : Jumlah sinyal neuron input layer yang masuk menuju neuron hidden layer.

x_i : Pola masukan.

v_j : Bobot sinyal dari neuron input layer yang menuju ke neuron hidden layer.

v_{j0} : Bobot bias hidden layer.

Selanjutnya hitung output masing-masing *neuron* pada *hidden layer* dengan menggunakan fungsi aktivasi, misalnya dengan fungsi aktivasi *sigmoid biner*.

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (6)$$

dimana:

z_j : Keluaran sinyal neuron hidden layer.

e : Bilangan eksponensial.

Langkah 6 : Jumlahkan semua *output* sinyal dari *neuron hidden layer* yang menuju ke *neuron output layer* (y_{net_k}) ($k = 1, 2, \dots, m$).

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (7)$$

dimana:

y_{net_k} : Jumlah sinyal yang masuk menuju neuron output layer.

w_{kj} : Bobot sinyal dari neuron hidden layer yang masuk ke neuron output layer.

w_{k0} : Bobot bias output layer.

Lalu gunakan fungsi aktivasi untuk menghitung *output* pada jaringan.

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (8)$$

dimana:

y_k : Keluaran pada jaringan.

Fase II : Propagasi Mundur.

Langkah 7 : Hitung faktor kesalahan (δ) pada *output* jaringan y_k ($k = 1, 2, \dots, m$).

$$\delta_k = (t_k - y_k)y_k(1 - y_k) \quad (9)$$

dimana:

δ_k : Faktor kesalahan output jaringan.

t_k : Target pada pola.

Hitung suku perubahan bobot w_{kj} (yang akan digunakan untuk memperbaharui bobot w_{kj}) dengan *learning rate* (α).

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (10)$$

dimana:

Δw_{kj} = Suku perubahan bobot w_{kj} .

Hitung suku perubahan bobot bias pada *output layer* dengan *learning rate* (α).

$$\Delta w_b = \alpha \delta_k x_b \quad (11)$$

dimana:

Δw_b = Suku perubahan bobot bias pada output layer.

x_b = Nilai input dummy (biasanya = 1)

Langkah 8 : Hitung faktor kesalahan (δ) sinyal yang masuk menuju *neuron* pada *hidden layer* berdasarkan faktor kesalahan disetiap *neuron* pada *hidden layer* (z_j) dengan *output* jaringan (δ_k).

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (12)$$

dimana:

δ_{net_j} : Faktor kesalahan sinyal yang masuk ke *neuron hidden layer*.

Kalikan nilai δ_{net_j} dengan turunan dari fungsi aktivasinya untuk mencari faktor kesalahan setiap *neuron hidden layer* :

$$\delta_j = \delta_{net_j} z_j (1 - z_j) \quad (13)$$

dimana:

δ_j : Faktor kesalahan *neuron hidden layer*.

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk memperbaharui bobot v_{ji}).

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (14)$$

dimana:

Δv_{ji} = Suku perubahan bobot sinyal *neuron input layer* yang menuju ke *neuron hidden layer*.

Hitung suku perubahan bobot bias pada *hidden layer*

$$\Delta v_b = \alpha \delta_j x_b \quad (15)$$

dimana:

Δv_b = Suku perubahan bobot bias pada *hidden layer*.

x_b = Nilai *input dummy* (biasanya bernilai = 1)

Fase III : Perubahan bobot.

Langkah 9 : Hitung semua perubahan bobot garis.

- Hitung perubahan bobot sinyal *neuron hidden layer* yang menuju ke *neuron output layer* :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (16)$$

- Hitung perubahan bobot bias pada *output layer* :

$$w_{k0}(\text{baru}) = w_{k0}(\text{lama}) + \Delta w_b \quad (17)$$

- Hitung perubahan bobot garis dari *neuron input layer* ke *neuron hidden layer* :

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (18)$$

- Hitung perubahan bobot bias pada *output layer* :

$$v_{j0}(\text{baru}) = v_{j0}(\text{lama}) + \Delta v_b \quad (19)$$

- Setelah dilakukan pembaruan bobot, selanjutnya dilakukan perhitungan nilai MSE (Mean Square Error) dengan menggunakan rumus berikut :

$$MSE = \frac{1}{k} \sum_{k=1}^n (y_k - t_k)^2 \quad (20)$$

dimana:

n : jumlah data; t_k : nilai target data

y_k : nilai *output* data

- Selanjutnya hitung nilai MAPE (Mean Absolute Percent Error) untuk menghitung rata-rata

persentase *error* hasil peramalan dengan rumus :

$$MAPE = \frac{1}{n} \sum_{k=1}^n \frac{|t_k - y_k|}{t_k} \times 100\% \quad (21)$$

dimana:

n : jumlah data t_k : nilai target data

y_k : nilai *output* data

Langkah 10: Uji kondisi berhenti (akhir iterasi) dengan melihat apakah target *epoch* dan target *error* telah tercapai..

3.6. Analisa Hasil

Pada Tahap ini dilakukan analisa hasil implentasi *coding* metode *Backpropagation* yang telah dibuat seperti menganalisa variabel yang terbaik dan analisa akurasi hasil peramalan.

3.7. Dokumentasi

Pada tahap ini dilakukan pendokumentasian hasil analisa *coding* kedalam bentuk laporan maupun jurnal.

4. HASIL DAN PEMBAHASAN

4.1. Perancangan dan Pembangunan Sistem

Sistem ini dibangun dengan menggunakan bahasa pemrograman *python* versi 3.7.3. Adapun library yang digunakan seperti *Numpy*, *Pandas* dan *Matplotlib*. Pada sistem ini terdapat empat buah *file* ber-format *.py* untuk tahap pelatihan, pengujian, validasi dan peramalan *Backpropagation*. Selanjutnya terdapat dua buah *file* ber-format *.csv* yaitu *dataset.csv* membuat *dataset* dan *bobot_pengujian.csv* untuk menyimpan bobot hasil pelatihan data dan digunakan pada tahap pengujian, validasi dan peramalan.

4.2. Pengujian Sistem

Pengujian sistem ini dilakukan dengan tujuan untuk menguji sistem yang telah dibangun dengan cara melakukan analisis terhadap hasil, nilai MSE dan nilai MAPE dari proses pelatihan, pengujian dan validasi *Backpropagation* dengan menggunakan berbagai macam jumlah data pelatihan – pengujian, jumlah *neuron* pada *hidden layer* dan beragam nilai *learning rate*. Pengujian sistem ini dilakukan dengan menggunakan dua skenario yaitu dengan menggunakan fungsi aktivasi *Sigmoid Biner* pada *hidden layer* dan *output layer* dan dengan menggunakan fungsi aktivasi ReLU pada *hidden layer* dan *Linear* pada *output layer*.

4.2.1. Pengujian Sistem dengan Fungsi Aktivasi

Sigmoid Biner

Berikut ini merupakan pengujian sistem dengan menggunakan fungsi aktivasi *Sigmoid Biner*.

4.2.1.1. Pengujian Pengaruh Pembagian Jumlah Data Pelatihan – Pengujian

Pengujian ini bertujuan untuk mengetahui pembagian jumlah data pelatihan dan pengujian yang mana sebaiknya digunakan pada peramalan ini. Arsitektur jaringan yang digunakan adalah 12-3-1, inisialisasi bobot awal = $[-0,1, 0,1]$, batas *error* = 0,0015 dan *learning rate* = 0.3. Pada Tabel I merupakan hasil pengujian pengaruh pembagian jumlah data pelatihan dan pengujian :

TABEL I. HASIL PENGUJIAN PENGARUH JUMLAH PEMBAGIAN DATA PELATIHAN - PENGUJIAN

No	Data Pelatihan - Pengujian	Epoch	MSE Pengujian	MAPE Pengujian
1	(2014-2016), (2016-2017)	5190	0,1249	133,34%
2	(2014-2015), (2016-2017)	4890	0,1294	99,27%
3	(2014-2016), (2015-2017)	5190	0,0627	69,02%

Berdasarkan TABEL I bisa dilihat bahwa pembagian data (2014-2016) untuk pelatihan dan (2015-2017) untuk pengujian menghasilkan nilai MSE dan MAPE terendah pada tahap pengujian sehingga dapat disimpulkan bahwa pembagian data (2014-2016) untuk pelatihan dan (2015-2017) merupakan yang terbaik.

4.2.1.2. Pengujian Pengaruh Jumlah Neuron Hidden layer

Pengujian ini bertujuan untuk mencari arsitektur jaringan yang terbaik berdasarkan nilai MSE dan MAPE yang dihasilkan pada tahap pengujian. Pengujian pengaruh jumlah *neuron hidden layer* ini dilakukan dengan menggunakan pembagian data pelatihan = (2014-2016), pengujian = (2015-2017), jumlah *neuron hidden layer* dari satu sampai dengan lima buah, inisialisasi bobot awal dengan rentang $[-0,1, 0,1]$, *learning rate* = 0,3, batas *error* = 0,0015 dan *epoch* = 20000. Pada TABEL II merupakan hasil pengujian pengaruh jumlah *neuron* pada *hidden layer* terhadap hasil pelatihan dan pengujian.

TABEL II. HASIL PENGUJIAN PENGARUH JUMLAH NEURON PADA HIDDEN LAYER

No	Arsitektur Jaringan	Epoch	MSE Pengujian	MAPE Pengujian
1	12-1-1	20000	0,0526	86,13%
2	12-2-1	20000	0,0523	67,33%
3	12-3-1	5190	0,0627	69,02%
4	12-4-1	4440	0,0161	30,53%
5	12-5-1	4940	0,0269	50,22%

Berdasarkan Tabel II dapat disimpulkan bahwa terdapat dua arsitektur jaringan terbaik yaitu arsitektur 12-4-1 karena menghasilkan nilai MSE dan MAPE terendah pada tahap pengujian dengan *epoch* = 4390. Berdasarkan Tabel II juga dapat dilihat bahwa arsitektur jaringan 12-1-1 dan arsitektur 12-2-1 belum mampu mencapai target *error* meskipun pada *epoch* = 20000. Hal ini menunjukkan bahwa jumlah *neuron hidden layer* terlalu sedikit tidak baik, karena dapat memperlambat proses pembelajaran.

4.2.1.3. Pengujian Pengaruh Nilai Learning Rate

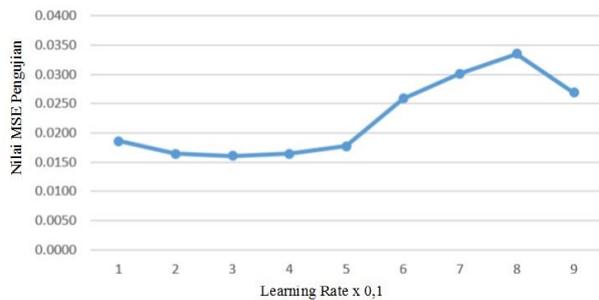
Pengujian ini bertujuan untuk mengetahui nilai *learning rate* berapa merupakan yang terbaik dengan menghasilkan nilai MSE dan MAPE terendah pada tahap pengujian. Pada pengujian *learning rate* ini dilakukan dengan nilai *learning rate* sebesar (0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8 dan 0,9) dengan arsitektur jaringan 12-4-1, inisialisasi bobot awal dengan rentang $[-0,1, 0,1]$, batas *error* = 0,0015 dan jumlah *epoch* = 20000. Pada Tabel III merupakan hasil pengujian pengaruh *learning rate*.

TABEL III. HASIL PENGUJIAN PENGARUH NILAI LEARNING RATE

No	Learning Rate	Epoch	MSE Pengujian	MAPE Pengujian
1	0,1	14540	0,0186	35,14%
2	0,2	6950	0,0164	31,28%
3	0,3	4440	0,0161	30,53%
4	0,4	3320	0,0164	29,54%
5	0,5	2660	0,0177	25,90%
6	0,6	2320	0,0259	34,49%
7	0,7	2070	0,0301	39,45%
8	0,8	1900	0,0335	43,35%
9	0,9	2020	0,0269	42,22%

Berdasarkan Tabel III bisa dilihat bahwa *learning rate* = 0,3 menghasilkan nilai MSE yang terkecil, sedangkan *learning rate* = 0,5 menghasilkan nilai MAPE terkecil. Berdasarkan hasil penelitian Hyndman[8], nilai MAPE lebih bagus digunakan jika semua data positif dan jauh lebih besar dari nol, sedangkan pada penelitian ini dilakukan normalisasi data dengan interval antara $[0,1, 0,9]$ yang mana nilainya mendekati

nol dan *range* keluarannya berada pada rentang [0, 1], sehingga berpatokan berdasarkan nilai MAPE kurang cocok pada pengujian ini dan lebih bagus untuk menggunakan nilai MSE sebagai patokan hasil akurasi pengujian data. Berdasarkan hal tersebut, maka nilai *learning rate* = 0,3 akan digunakan pada tahap validasi dan peramalan. Pada Gambar 2 merupakan grafik perbandingan hasil pengujian *learning rate* sesuai dengan hasil pengujian pada Tabel III.



Gambar 2. Grafik hasil pengujian pengaruh nilai *learning rate*

4.2.1.4. Pengujian Pengaruh Nilai Batas Error

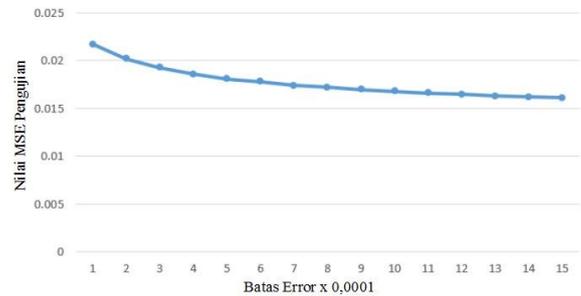
Pengujian ini bertujuan untuk mencari nilai batas *error* yang terbaik dengan menghasilkan nilai MSE dan MAPE terendah pada tahap pengujian. Pengujian pengaruh nilai batas *error* ini dilakukan dengan arsitektur jaringan 12-4-1, *learning rate* = 0,3 dan nilai batas *error* dari 0,0001 sampai dengan 0,0015. Pada Tabel IV merupakan hasil pengujian pengaruh nilai batas *error*.

TABEL IV. HASIL PENGUJIAN PENGARUH NILAI BATAS ERROR

No	Batas Error	Epoch	MSE Pengujian	MAPE Pengujian
1	0,0001	7840	0,0217	34,55%
2	0,0002	7050	0,0202	33,43%
3	0,0003	6560	0,0193	32,67%
4	0,0004	6190	0,0186	32,18%
5	0,0005	5900	0,0181	31,77%
6	0,0006	5660	0,0178	31,53%
7	0,0007	5450	0,0174	31,26%
8	0,0008	5270	0,0172	31,12%
9	0,0009	5110	0,0170	31,00%
10	0,0010	4970	0,0168	30,87%
11	0,0011	4840	0,0166	30,75%
12	0,0012	4730	0,0165	30,73%
13	0,0013	4620	0,0163	30,64%
14	0,0014	4530	0,0162	30,55%
15	0,0015	4440	0,0161	30,53%

Berdasarkan Tabel IV dapat disimpulkan bahwa nilai batas *error* = 0,0015 adalah yang terbaik karena

menghasilkan nilai MSE dan MAPE terendah pada tahap pengujian. Pada Gambar 3 merupakan grafik hasil pengujian pengaruh nilai batas *error*.



Gambar 3. Grafik hasil pengujian pengaruh nilai batas *error*

4.2.1.5. Validasi Backpropagation

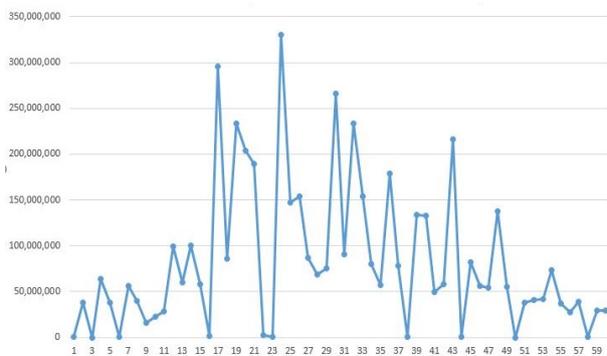
Pada tahap ini dilakukan peramalan terhadap data yang ada namun tidak digunakan pada tahap pelatihan dan pengujian yaitu data pada tahun 2018, dengan menggunakan nilai variabel berdasarkan hasil dari pengujian sistem yang telah dilakukan. Pada Tabel V merupakan hasil dari tahap validasi *Backpropagation* dengan meramalkan data pada tahun 2018 yang mana tersebut sudah ada.

TABEL V. HASIL DARI TAHAP VALIDASI BACKPROPAGATION DENGAN MERAMALKAN DATA PADA TAHUN 2018

No	Data Aktual 2018	Prediksi 2018
1	55.623.525	185.620.016
2	49.500	-18.639.762
3	37.925.880	77.668.640
4	40.510.456	169.522.848
5	41.392.078	35.958.744
6	73.611.411	137.864.880
7	36.691.259	219.132.336
8	27.526.174	-15.101.917
9	39.206.347	84.789.128
10	733.763	179.705.168
11	29.533.386	26.020.544
12	29.452.168	256.743.264

Berdasarkan Tabel V bisa dilihat bahwa terdapat hasil peramalan yang bernilai negatif atau hasil peramalan yang lebih kecil dari 0,1 ketika masih dinormalisasi. Hal ini disebabkan karena tidak stabilnya data jumlah nilai ekspor di provinsi NTB sehingga membutuhkan proses pelatihan yang sangat lama pada *epoch* yang besar. Dengan semakin besarnya jumlah *epoch* tersebut dapat menyebabkan terjadinya *vanishing gradient* dan nilai bobot yang dihasilkan terlalu besar pada fungsi aktivasi *Sigmoid Biner*. *Vanishing gradient* adalah terlalu kecilnya nilai

gradient yang dihasilkan ketika proses propagasi balik sehingga terjadi pembaruan bobot yang sangat kecil [9]. Dengan terjadinya *vanishing gradient* ini dapat menimbulkan ketidak seimbangan bobot yang dihasilkan ketika tahap propagasi balik, seperti terdapat sebagian bobot yang perubahannya sangat besar dan terdapat sebagian bobot yang perubahannya sangat kecil sehingga menyebabkan terjadinya ketidak seimbangan. Ketika bobot yang dihasilkan dari banyaknya *epoch* dan terdapat *vanishing gradient* ini diterapkan pada tahap peramalan memungkinkan terjadinya hasil peramalan yang bernilai negatif. Pada Gambar 4 merupakan pola data jumlah nilai ekspor di provinsi NTB yang memungkinkan terjadinya hasil validasi yang bernilai negatif.



Gambar 4. Pola data jumlah nilai ekspor di provinsi NTB

Berdasarkan grafik pada Gambar 4, dapat dilihat bahwa pola data jumlah nilai ekspor di provinsi NTB sangat tidak stabil sehingga memungkinkan hasil peramalan yang bernilai negatif.

4.2.2. Pengujian Sistem dengan Fungsi Aktivasi

ReLU - Linear

Pada skenario ini dilakukan pengujian sistem dengan menggunakan fungsi aktivasi ReLU pada *hidden layer* dan *Linear* pada *output layer*. Berikut ini merupakan pengujian sistem dengan menggunakan fungsi aktivasi ReLU - *Linear*.

4.2.1. Pengujian Pengaruh Jumlah Neuron pada Hidden layer

Pengujian ini bertujuan untuk mencari arsitektur jaringan terbaik dengan menggunakan fungsi aktivasi ReLU - *Linear*. Pengujian ini dilakukan dengan cara membandingkan jumlah *neuron* pada *hidden layer* dari satu buah sampai dengan lima buah. Pada Tabel VI merupakan hasil pengujian pengaruh jumlah *neuron* pada *hidden layer* dengan menggunakan *epoch* = 20000, *learning rate* = 0,1 dan batas *error* = 0,0015.

TABEL VI. HASIL PENGUJIAN PENGARUH JUMLAH NEURON PADA HIDDEN LAYER

No	Arsitektur Jaringan	Epoch	MSE Pengujian	MAPE Pengujian
1	12-1-1	20000	0,0320	66,58%
2	12-2-1	20000	0,0016	38,94%
3	12-3-1	2000	0,0014	122,20%
4	12-4-1	400	0,0008	63,36%
5	12-5-1	450	0,0008	53,07%

Berdasarkan Tabel VI dapat disimpulkan bahwa arsitektur jaringan 12-5-1 merupakan yang terbaik karena menghasilkan nilai MSE dan MAPE terkecil pada tahap pengujian.

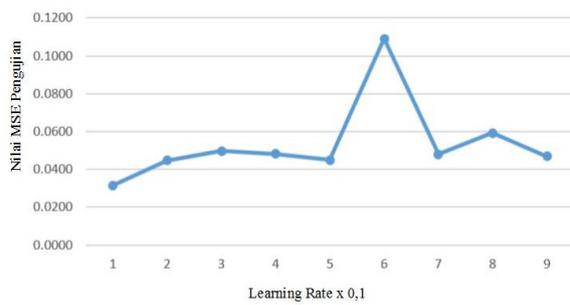
4.2.2. Pengujian Nilai Learning Rate

Pengujian ini bertujuan untuk mengetahui nilai *learning rate* yang terbaik dengan menggunakan fungsi aktivasi ReLU dan *Linear*. Pengujian ini dilakukan dengan menggunakan arsitektur jaringan = 12-5-1, batas *error* = 0,0008, *epoch* = 20000 dan nilai *learning rate* yang digunakan mulai dari 0,1 sampai dengan 0,9. Pada Tabel VII merupakan hasil pengujian pengaruh nilai *learning rate* pada fungsi aktivasi ReLU dan *Linear*.

TABEL VII. HASIL PENGUJIAN PENGARUH NILAI LEARNING RATE

No	Learning Rate	Epoch	MSE Pengujian	MAPE Pengujian
1	0,1	450	0,0316	53,07%
2	0,2	20000	0,0448	72,48%
3	0,3	20000	0,0498	68,06%
4	0,4	20000	0,0482	78,16%
5	0,5	20000	0,0449	81,63%
6	0,6	20000	0,1090	117,72%
7	0,7	20000	0,0479	84,22%
8	0,8	20000	0,0592	73,86%
9	0,9	20000	0,0469	85,37%

Berdasarkan Tabel V dapat disimpulkan bahwa nilai *learning rate* = 0,1 adalah yang terbaik dengan menggunakan fungsi aktivasi ReLU - *Linear*, karena menghasilkan nilai MSE dan MAPE terkecil pada tahap pengujian. Berdasarkan Tabel V juga bisa dilihat bahwa nilai *learning rate* diatas 0,1 belum mampu mencapai target *error* meskipun sudah berada pada *epoch* 20000. Hal ini menunjukkan bahwa jika ingin menggunakan fungsi aktivasi ReLU dan *Linear*, harus hati - hati dalam menetapkan nilai *learning rate* karena jika kebesaran akan menyebabkan "dying ReLU" target *error* menjadi tidak tercapai [10]. Pada Gambar 6 merupakan grafik hasil pengujian pengaruh nilai *learning rate* terhadap nilai MSE yang dihasilkan pada tahap pengujian dengan menggunakan fungsi aktivasi ReLU - *Linear*.



Gambar 5. Grafik hasil pengujian pengaruh nilai *learning rate* dengan fungsi aktivasi ReLU – *Linear*.

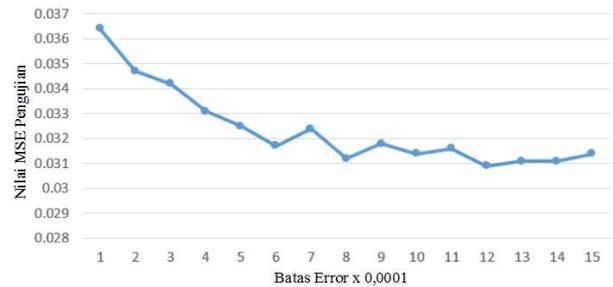
4.2.3. Pengujian Pengaruh Nilai Batas Error

Pengujian ini bertujuan untuk mengetahui nilai batas *error* berapa yang terbaik dengan menggunakan fungsi aktivasi ReLU – *Linear*. Pengujian pengaruh nilai batas *error* dilakukan dengan menggunakan arsitektur jaringan 12-5-1, *learning rate* = 0,1 dan batas *error* dari 0,0001 sampai dengan 0,0015. Pada Tabel VIII merupakan hasil pengujian pengaruh nilai batas *error* dengan menggunakan fungsi aktivasi ReLU – *Linear*.

TABEL VIII. HASIL PENGUJIAN NILAI BATAS *ERROR* PADA FUNGSI AKTIVASI ReLU - *LINEAR*

No	Batas Error	Epoch	MSE Pengujian	MAPE Pengujian
1	0,0001	604	0,0364	54,16%
2	0,0002	557	0,0347	53,59%
3	0,0003	524	0,0342	53,71%
4	0,0004	501	0,0331	53,25%
5	0,0005	481	0,0325	53,10%
6	0,0006	465	0,0317	53,08%
7	0,0007	453	0,0324	53,65%
8	0,0008	442	0,0312	53,06%
9	0,0009	432	0,0318	53,47%
10	0,0010	428	0,0314	53,35%
11	0,0011	421	0,0316	53,47%
12	0,0012	416	0,0309	53,04%
13	0,0013	410	0,0311	53,59%
14	0,0014	406	0,0311	53,49%
15	0,0015	404	0,0314	54,48%

Berdasarkan Tabel VIII dapat disimpulkan bahwa nilai batas *error* = 0,0012 merupakan yang terbaik karena menghasilkan MSE dan MAPE terkecil pada tahap pengujian. Pada Gambar 6 merupakan grafik hasil pengujian pengaruh nilai batas *error* pada fungsi aktivasi ReLU – *Linear*.



Gambar 6. Grafik hasil pengujian pengaruh nilai batas *error* dengan fungsi aktivasi ReLU – *Linear*.

4.2.4. Validasi Backpropagation

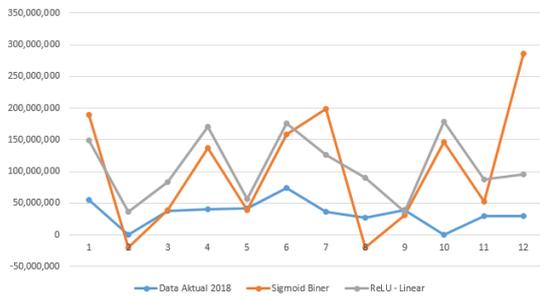
Pada tahap ini dilakukan peramalan data pada tahun 2018 dengan menggunakan bobot-bobot dengan menggunakan variabel terbaik berdasarkan hasil pengujian sistem yang telah dilakukan dengan menggunakan fungsi aktivasi ReLU - *Linear*. Pada Tabel IX merupakan perbandingan hasil validasi dengan menggunakan fungsi aktivasi ReLU – *Linear* dan fungsi aktivasi *Sigmoid Biner* yang sudah didapatkan sebelumnya pada Tabel V.

TABEL IX PERBANDINGAN HASIL VALIDASI DENGAN MENGGUNAKAN FUNGSI AKTIVASI ReLU – *LINEAR* DAN SIGMOID

No	Data Input 2018	Sigmoid Biner	ReLU - Linear
1	55.623.525	185.620.016	147.221.120
2	49.500	-18.639.762	35.416.868
3	37.925.880	77.668.640	80.769.896
4	40.510.456	169.522.848	170.257.168
5	41.392.078	35.958.744	55.815.640
6	73.611.411	137.864.880	175.732.528
7	36.691.259	219.132.336	123.330.320
8	27.526.174	-15.101.917	85.696.888
9	39.206.347	84.789.128	43.512.856
10	733.763	179.705.168	175.836.704
11	29.533.386	26.020.544	85.139.672
12	29.452.168	256.743.264	99.765.592

Berdasarkan Tabel IX bisa dilihat bahwa hasil validasi dengan menggunakan fungsi aktivasi ReLU – *Linear* menghasilkan hasil peramalan yang bernilai positif semua sedangkan hasil peramalan dengan menggunakan fungsi aktivasi *Sigmoid Biner* terdapat beberapa hasil peramalan yang bernilai negatif. Berdasarkan hal tersebut, maka fungsi aktivasi ReLU – *Linear* lebih bagus untuk digunakan pada kasus peramalan jumlah nilai ekspor di provinsi NTB dibandingkan dengan menggunakan fungsi aktivasi *Sigmoid Biner*. Pada Gambar 7 merupakan grafik perbandingan hasil validasi dengan menggunakan

fungsi aktivasi ReLU – Linear dan fungsi aktivasi *Sigmoid Biner*.



Gambar 7. Grafik perbandingan hasil validasi fungsi aktivasi *Sigmoid Biner* dengan ReLU - *Linear*

4.3. Peramalan

Pada tahap peramalan ini dilakukan dengan dua skenario yaitu meramalkan jumlah nilai ekspor di provinsi NTB dan meramalkan jumlah nilai ekspor per komoditi. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *Sigmoid Biner* dan fungsi aktivasi ReLU – *Linear*. Fungsi aktivasi *Sigmoid Biner* tetap digunakan meskipun menghasilkan hasil validasi yang bernilai negatif dengan tujuan untuk membandingkan pola data hasil peramalan jumlah nilai ekspor di provinsi NTB dengan menggunakan fungsi aktivasi *Sigmoid Biner* dan ReLU – *Linear*. Variabel yang digunakan masing – masing fungsi aktivasi adalah variabel yang dihasilkan berdasarkan pengujian sistem yang telah dilakukan dengan menggunakan masing – masing fungsi aktivasi.

4.3.1. Permalan Jumlah Nilai Ekspor di Provinsi NTB

Pada skenario peramalan ini akan diramalkan jumlah nilai ekspor di provinsi NTB pada tahun 2019 dengan menggunakan data *input* pada tahun 2018 dengan menggunakan fungsi aktivasi *Sigmoid Biner* dan ReLU - *Linear*. Pada Tabel X merupakan hasil peramalan jumlah nilai ekspor pada tahun 2019.

Berdasarkan Tabel X bisa dilihat bahwa pada fungsi aktivasi *Sigmoid Biner* menghasilkan hasil peramalan yang bernilai negatif. Penyebab hasil peramalan yang bernilai negatif ini sudah dijelaskan pada tahap validasi dengan menggunakan fungsi aktivasi *Sigmoid Biner*. Pada Gambar 8 merupakan grafik hasil peramalan jumlah nilai ekspor di provinsi NTB pada tahun 2019 sesuai dengan Tabel X.

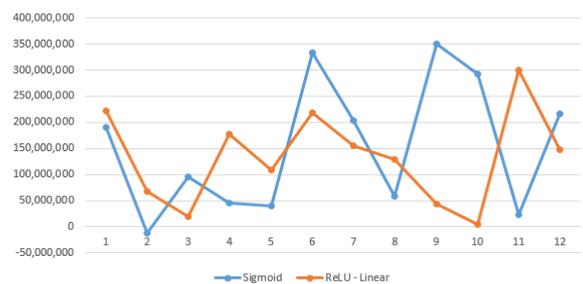
4.3.2. Peramalan Jumlah Nilai Ekspor Per Komoditi

Pada skenario peramalan ini dilakukan dengan menggunakan komoditi Gerabah, Kerajinan Buah Kering, Mutiara dan Konsentrat Tembaga. Komoditi tersebut digunakan pada skenario ini karena data

jumlah nilai ekspor komoditi tersebut ada setiap tahunnya. Vairabel - variabel yang digunakan adalah variabel yang terbaik berdasarkan hasil pengujian sistem masing – masing komoditi, karena jika menggunakan variabel – variabel berdasarkan hasil pengujian sistem pada skenario meramalkan jumlah nilai ekspor total belum tentu bagus jika diterapkan pada skenario peramalan jumlah nilai ekspor per komoditi ini. Pada Tabel XI merupakan variabel terbaik hasil pengujian sistem per komoditi dengan fungsi aktivasi *Sigmoid Biner* dan pada Tabel XII merupakan variabel terbaik hasil pengujian sistem dengan menggunakan fungsi aktivasi ReLU – *Linear*.

TABEL X. HASIL PERAMALAN JUMLAH NILAI EKSPOR DI PROVINSI NTB PADA TAHUN 2019.

No	Data Aktual 2018	Peram. 2019 Sigmoid Biner	Peram. 2019 ReLU - Linear
1	55.623.525	190.859.632	222.144.048
2	49.500	-11.642.485	67.473.528
3	37.925.880	96.318.288	20.163.722
4	40.510.456	46.288.328	176.782.112
5	41.392.078	39.141.232	108.551.848
6	73.611.411	334.063.456	217.359.584
7	36.691.259	203.145.584	155.091.504
8	27.526.174	58.803.016	129.791.768
9	39.206.347	349.832.224	43.927.732
10	733.763	292.650.752	4.587.466
11	29.533.386	23.026.430	300.569.728
12	29.452.168	216.622.864	147.184.752



Gambar 8. Grafik hasil peramalan jumlah nilai ekspor di provinsi NTB pada tahun 2019.

TABEL XI. VARIABEL TERBAIK HASIL PENGUJIAN SISTEM PER KOMODITI DENGAN FUNGSI AKTIVASI *SIGMOID BINER*

No	Komoditi	Ars. Jaringan	Learning Rate	Batas Error
1	Gerabah	12-4-1	0,3	0,0003
2	K. B. Kering	12-4-1	0,3	0,0002
3	Mutiara	12-4-1	0,3	0,0002
4	Konsentrat Tembaga	12-4-1	0,3	0,0015

TABEL XII. VARIABEL TERBAIK HASIL PENGUJIAN SISTEM PER KOMODITI DENGAN FUNGSI AKTIVASI RELU - LINEAR

No	Komoditi	Ars. Jaringan	Learning Rate	Batas Error
1	Gerabah	12-25-1	0,1	0,0002
2	K. B. Kering	12-20-1	0,1	0,0002
3	Mutiara	12-25-1	0,1	0,0002
4	Konsentrat Tembaga	12-5-1	0,1	0,0012

Selanjutnya pada Tabel XIII merupakan hasil peramalan per komoditi dengan menggunakan fungsi aktivasi *Sigmoid Biner* dan pada Tabel XIV merupakan hasil peramalan dengan menggunakan fungsi aktivasi ReLU - *Linear*.

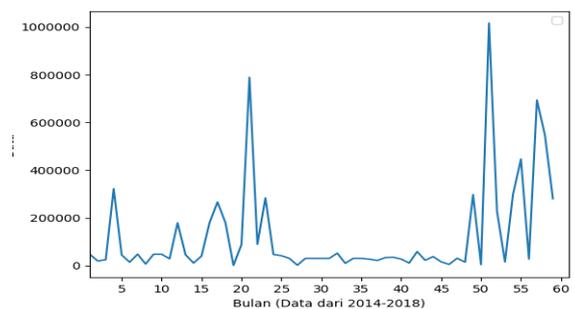
TABEL XIII. HASIL PERAMALAN NILAI EKSPOR PER KOMODITI PADA TAHUN 2019 DENGAN FUNGSI AKTIVASI SIGMOID BINER

No	Gerabah	K. B. Kering	Mutiara	K.Tembaga
1	18.723	8.110	-55.993	165.154.736
2	25.811	970	-92.432	129.919.280
3	31.227	4.208	-62.385	70.734.136
4	31.434	22.709	-35.237	173.811.616
5	24.774	22.197	44.039	242.488.976
6	15.778	16.959	198.910	309.076.384
7	10.903	18.304	122.432	83.368.928
8	10.592	42.709	68.168	149.965.712
9	13.656	52.289	-26.716	344.901.536
10	14.088	23.362	-50.408	42.881.688
11	4.766	14.936	366.867	69.247.968
12	9.062	21.194	962.188	162.349.648

TABEL XIV. HASIL PERAMALAN NILAI EKSPOR PER KOMODITI PADA TAHUN 2019 DENGAN FUNGSI AKTIVASI RELU - LINEAR

No	Gerabah	K. B. Kering	Mutiara	K.Tembaga
1	13.765	8.047	-189.743	165.154.736
2	11.439	6.696	-205.499	129.919.280
3	19.018	9.018	152.871	70.734.136
4	17.769	23.719	-297.034	173.811.616
5	15.605	19.209	-691.810	242.488.976
6	11.554	9.505	-522.058	309.076.384
7	4.399	11.034	62.080	83.368.928
8	6.224	50.030	-432.378	149.965.712
9	12.040	73.959	-1.591.301	344.901.536
10	14.149	24.016	-2.055.340	42.881.688
11	22.970	19.032	-422.365	69.247.968
12	12.488	24.806	-815.755	162.349.648

Berdasarkan hasil peramalan pada Tabel XI dan Tabel XII bisa dilihat bahwa hasil peramalan jumlah nilai ekspor komoditi Mutiara untuk masing – masing fungsi aktivasi sama – sama menghasilkan nilai negatif. Hal ini dikarena pola data jumlah nilai ekspor komoditi Mutiara sangat tidak stabil sehingga dapat menghasilkan hasil peramalan yang bernilai negatif. Pada Gambar 9 merupakan grafik pola data nilai ekspor mutiara yang menyebabkan hasil peramalan bernilai negatif.



Gambar 9. Pola data nilai ekspor komoditi Mutiara.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian sistem dan analisis penelitian yang dilakukan terhadap studi kasus peramalan jumlah nilai ekspor di provinsi NTB dengan algoritma Jaringan Saraf Tiruan metode *Backpropagation* maka dapat ditarik kesimpulan sebagai berikut :

1. Arsitektur jaringan 12-4-1, nilai *learning rate* = 0,3 dan batas *error* = 0,0015 merupakan yang terbaik pada pengujian sistem dengan menggunakan fungsi aktivasi *Sigmoid Biner* karena menghasilkan nilai MSE sebesar 0,0161 dan MAPE sebesar 30,53%, terkecil pada tahap pengujian.
2. Arsitektur jaringan 12-5-1, nilai *learning rate* = 0,1 dan batas *error* = 0,0012 merupakan yang terbaik pada pengujian sistem dengan menggunakan fungsi aktivasi ReLU pada *hidden layer* dan *Linear* pada *output layer* karena menghasilkan nilai MSE sebesar 0,0309 dan MAPE sebesar 53,04%, terkecil pada tahap pengujian.
3. Jumlah *neuron* pada *hidden layer* dapat mempengaruhi keakurasian hasil peramalan. Semakin banyak jumlah *neuron hidden layer* yang digunakan belum tentu dapat meningkatkan akurasi hasil peramalan.
4. Penerapan algoritma jaringan saraf tiruan metode *backpropagation* pada penelitian ini kurang cocok

digunakan karena pola data jumlah nilai ekspor di provinsi NTB sangat tidak stabil sehingga nilai MAPE yang dihasilkan pada pengujian sistem tidak ada yang bernilai lebih kecil dari 20%. Menurut Zainun [11], nilai MAPE yang mempunyai kinerja bagus berada diantara 10% - 20% sedangkan untuk yang bekinerja sangat bagus berada dibawah 10%.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan maka terdapat saran sebagai berikut jika ingin menerapkan algoritma Jaringan Saraf Tiruan metode *Backpropagation* kedalam suatu studi kasus :

1. Sebaiknya diuji coba lagi untuk menggunakan jumlah *neuron* pada *hidden layer* mulai dari yang sedikit hingga banyak lalu bandingkan mana yang terbaik.
2. Jika menggunakan fungsi aktivasi *Sigmoid Biner* sebaiknya uji dulu nilai *learning rate* yang akan digunakan, jangan gunakan yang terlalu besar karena dapat menghasilkan bobot yang kurang teliti meskipun dapat mempercepat proses pelatihan, sedangkan jika menggunakan fungsi aktivasi ReLU – *Linear* jangan gunakan *learning rate* terlalu besar karena dapat menyebabkan *dying* RELU.

DAFTAR PUSTAKA

- [1] T. Rauf, "Info Publik," 2018. [Daring]. Tersedia pada : <http://infopublik.id/read/280102/nilai-ekspor-ntb-periode-juni-2018-naik.html#>. [Diakses : 30-Jan-2019].
- [2] B. Siswoyo, "Analisis dan Penerapan Teknologi Jaringan Saraf Tiruan pada Model Peramalan Industri Perbankan," *JATI : Jurnal Teknologi dan Informasi UNIKOM*, vol. 5, no. 4, hal. 33-46, 2014.
- [3] J. R. Saragih, M. B. S. Saragih, dan A. Wanto, "Analisis Algoritma Backpropagation dalam Prediksi Nilai Ekspor (Juta USD)," *Jurnal Pendidikan Teknologi dan Kejuruan*, vol. 15, no. 2, hal. 254-264, 2018.
- [4] D. P. Anggriningrum, P. Hendikawati, dan Z. Abidin, "Perbandingan Prediksi Harga Saham dengan Menggunakan Jaringan Syaraf Tiruan Backpropagation dan ARIMA," *Unnes Journal of Mathematics*, vol. 2, no. 2, hal. 105-109, 2013.
- [5] I. P. Sutawinaya, I. N. G. A. Astawa, dan N. K. D. Hariyanti, "Perbandingan Metode Jaringan Saraf Tiruan pada Peramalan Curah Hujan," *Jurnal Logic*, vol. 17, no. 2, hal. 92-97, 2017.
- [6] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*, Jakarta: Andi, 2009.
- [7] M. N. D. Sawitri, I. W. Sumarjaya, dan N. K. T. Tastrawati, "Peramalan Menggunakan Metode Backpropagation Neural Network," *E-Jurnal Matematika*, vol. 7, no. 3, hal. 264-270, 2018.
- [8] R. J. Hyndman, dan A. B. Koehler, "Another Look at Measures of Forecast Accuracy," *International Journal of Forecasting*, vol. 22, hal. 679-688, 2006.
- [9] C. F. Wang, "Toward Data Science," 2019. [Daring]. Tersedia pada : <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>. [Diakses : 7-Jul-2019]
- [10] D. Liu, "Medium," 2017. [Daring]. Tersedia pada : <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>. [Diakses : 7-Jul-2019]
- [11] N. Y. B. Zainun, I. A. Rahman, dan M. Eftekhari, "Forecast Low-Cost Housing Demand in Johor Bahru, Malaysia Using Artificial Neural Network (ANN)," *Journal of Mathematics Research*, vol. 2, no. 1, hal. 14-19, 2010.