

Penerapan Jaringan Saraf Tiruan *Backpropagation* Dalam Memprediksi Nilai Tukar Petani

Wahyudi Ariannor¹, Muhammad Fajar Razatillah^{2*}

Program Studi Teknik Informatika, STMIK Banjarbaru
 Jl. Ahmad Yani KM 33,5 Loktabat, Banjarbaru, Indonesia
 *Email Corresponding Author: fajarfreesdom130@gmail.com

Abstrak

Nilai Tukar Petani (NTP) merupakan alat untuk mengukur kemampuan tukar produk yang dijual petani dengan produk yang dibutuhkan petani dalam produksi dan konsumsi rumah tangga. NTP khususnya pada subsektor Tanaman Pangan, seringkali berfluktuasi setiap bulannya, sehingga dipandang perlu untuk diprediksi dengan tepat, agar dapat membantu pemerintah dan pihak terkait mempersiapkan tindakan-tindakan pencegahan seperti menjaga kestabilan harga produksi pertanian dan mengendalikan harga-harga biaya usaha pertanian. Paper ini menguji penerapan Jaringan Saraf Tiruan (JST) *Backpropagation* untuk memprediksi NPT pada sub sektor Tanaman Pangan di Provinsi Kalimantan Selatan. Pengujian dilakukan menggunakan 84 data latih dan 36 data uji. Data masukan berupa data deret waktu NPT subsektor Pangan Provinsi Kalimantan Selatan selama 12 bulan sebelumnya untuk memprediksi NPT setiap bulannya selama periode 12 bulan mendatang. Hasil uji menunjukkan nilai presentase *error* (MAPE) 0,97, atau diperoleh persentase akurasi prediksi sebesar 99,03 %.

Kata Kunci: *Prediksi; Nilai Tukar Petani; Subsektor Tanaman Pangan; Jaringan Saraf tiruan; Backpropagation*

Abstract

Farmer's Exchange Rate is a tool to measure the ability to exchange products sold by farmers with products needed by farmers in household production and consumption. Farmer's Exchange Rate, especially in the Food Crops sub-sector, often fluctuates every month, so it is deemed necessary to predict accurately, in order to assist the government and related parties in preparing preventive measures such as maintaining stability in agricultural production prices and controlling agricultural costs. This paper examines the application of Backpropagation Artificial Neural Networks to predict Farmer's Exchange Rate in the Food Crops sub-sector in South Kalimantan Province. The test was carried out using 84 training data and 36 test data. The input data is in the form of time series data for the Farmer's Exchange Rate of the Food sub-sector of South Kalimantan Province for the previous 12 months to predict the Farmer's Exchange Rate every month for the next 12 month period. The test results show the percentage error value (MAPE) is 0.97, or the percentage of prediction accuracy is 99.03%.

Keywords: *Prediction; Farmer's Exchange Rate; Food Crops Subsector; Artificial Neural Networks; Backpropagation*

1. Pendahuluan

Nilai Tukar Petani (NTP) adalah salah satu alat ukur kesejahteraan petani yang umum digunakan. Kegunaan dari NTP yaitu untuk mengukur kemampuan tukar produk yang dijual petani dengan produk yang dibutuhkan petani dalam produksi dan konsumsi rumah tangga [1]. NTP Subsektor Tanaman Pangan di Provinsi Kalimantan Selatan berfluktuasi setiap bulannya. Pada bulan April 2020 NTP Subsektor Tanaman Pangan turun sebesar 99,85 atau turun 1,97 persen dibanding periode sebelumnya. Penurunan NTP Subsektor Tanaman Pangan disebabkan karena indeks yang diterima petani (It) sebesar 105,19 lebih rendah dari indeks yang dibayarkan petani (Ib) sebesar 105,34. Menurunnya NTP Subsektor Tanaman Pangan di bawah 100, berarti kesejahteraan petani tanaman pangan rendah, karena pengeluaran petani lebih besar daripada pendapatan petani. Jika kesejahteraan petani rendah, bisa membawa dampak bertambahnya tingkat kemiskinan di pedesaan sebagai salah satu penghambat pertumbuhan ekonomi

Kalimantan Selatan. Selama ini belum ada proses prediksi NTP untuk memastikan tindakan pencegahan seperti menjaga kestabilan harga produksi dan mengendalikan harga-harga biaya usaha pertanian apakah sudah pada jalur yang tepat. Atas dasar tersebut diperlukannya prediksi NTP Subsektor Tanaman Pangan guna memberikan gambaran berapa NTP Subsektor Tanaman Pangan untuk periode bulan yang akan datang sehingga dapat membantu pemerintah dan pihak terkait dalam mempersiapkan tindakan-tindakan pencegahan seperti menjaga kestabilan harga produksi pertanian dan mengendalikan harga-harga biaya usaha pertanian apabila terjadi penurunan secara signifikan dari periode sebelumnya atau NTP turun di bawah 100.

Jaringan Saaraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik mirip dengan jaringan syaraf manusia yang melakukan proses belajar melalui perubahan bobot sinapsisnya [2]. Algoritma momentum merupakan pengembangan dari algoritma *backpropagation* standar. Algoritma momentum dalam JST adalah perubahan bobot yang didasarkan atas arah gradient pola terakhir dan pola sebelumnya yang dimasukkan [3]. Penambahan momentum bertujuan untuk menghindari perubahan bobot yang signifikan, diakibatkan oleh adanya data yang sangat berbeda dengan yang lain. Parameter momentum juga dapat meningkatkan waktu pelatihan dan stabilitas dari proses pelatihan [4].

Artikel ini menyajikan model prediksi NTP Subsektor Tanaman Pangan Kalimantan Selatan menggunakan JST *Backpropagation*.

2. Tinjauan Pustaka

Tri Wardati Khusniyah dan Sutiknotelah meneliti tentang prediksi Nilai Tukar Petani menggunakan jaringan saaraf tiruan *backpropagation*. Penelitian ini menggunakan dua variabel pembentuk NTP yaitu indeks harga yang diterima petani (It) dan indeks harga yang dibayar petani (Ib) sebagai masukan jaringan. Hasil penelitian tersebut menunjukkan bahwa persentase *error* terkecil apabila jumlah node lapisan tersembunyi 7 dan nilai laju pembelajaran 0.1 dengan rata-rata *error* sebesar 0.61% atau tingkat akurasi mencapai 99.39% [5].

Endri Widada juga pernah melakukan penelitian tentang prediksi Nilai Tukar Petani (NTP) D.I. Yogyakarta menggunakan jaringan saraf tiruan dengan algoritma *backpropagation*. Tujuan dilakukannya penelitian adalah dapat mengetahui nilai prediksi NTP D.I Yogyakarta pada bulan berikutnya dalam rentang 6 bulan yang akan datang dan mengetahui tingkat akurasi. Hasil penelitian yang diperoleh dari proses training dengan jumlah epoch 10000, neuron hidden layer 10, learning rate 0.04 dan target *error* 0.00001 menghasilkan *Mean Square Error* (MSE) 0.0004. Sistem juga mampu memprediksi NTP dengan tingkat *Mean Absolut Percentage Error* (MAPE) sebesar 0.52% dengan tingkat akurasi sebesar 99.48% [6].

Penelitian lainnya tentang prediksi Nilai Tukar Petani dilakukan oleh Sandy Putra Siregar, Dedy Hartama, dan Anjar Wanto estimasi Nilai Tukar Petani Subsektor Tanaman Pangan menggunakan JST pada Provinsi Sumatera Utara. Pada penelitian ini model arsitektur jaringan dan parameter yang digunakan sangat mempengaruhi tingkat akurasi. Model arsitektur 3-28-1 menghasilkan epoch sebesar 7060 waktu yang diperlukan 01:19 detik dengan tingkat akurasi sebesar 83% [7].

Widia Istiqomah dan Mohamad Yamin Darsyahtelah melakukan penelitian tentang efektivitas metode arima dan exponential smoothing untuk meramalkan Nilai Tukar Petani di Jawa Tengah. Pada penelitian ini efektivitas model bisa dilihat dari besar nilai MSE dan MAPE pada setiap metode. Metode Arima lebih baik dalam meramalkan data Nilai Tukar Petani di Jawa Tengah karena nilai MSE dan MAPE berturut-turut sebesar 1,2487 dan 0,792402 cenderung lebih kecil dibandingkan nilai MSE dan MAPE pada metode *Exponential Smoothing* sebesar 1.91312 dan 0.93166 [8].

Penelitian yang dilakukan oleh Izza Afkarina yang menerapkan algoritma levenberg-marquardt untuk memprediksi Nilai Tukar Petani (NTP) Subsektor Tanaman Pangan. Pada penelitian ini algoritma yang diterapkan adalah *Levenberg Marquardt* (LM). Data yang digunakan adalah data indeks NTP Subsektor Tanaman Pangan Provinsi Sumatera Utara dari bulan Januari 2013 – Desember 2017. Hasil pengujian mendapatkan nilai MSE terbaik pada pembagian data 90% data latih dan 10% data uji, yaitu MSE dengan nilai 0,0044 dengan jumlah epoch 10, parameter marquardt 10 dan faktor tau 0,5 dengan *running time* 16 detik [9].

Penelitian yang pernah dilakukan oleh Irawan, Zarlis, & Nababan tentang Analisis Penambahan Nilai Momentum Pada Prediksi Produktivitas Kelapa Sawit Menggunakan *Backpropagation*. Pada penelitian ini menambahkan parameter learning rate secara adaptif pada setiap iterasi dan koefisien momentum untuk menghitung proses perubahan bobot. Hasil

penelitian menunjukkan bahwa algoritma *backpropagation* dengan penambahan nilai momentum hanya mencapai 727 *epoch* dengan nilai MSE 0,01, sedangkan algoritma *backpropagation* standar mencapai 4000 *epoch* dengan nilai MSE 0,001 tingkat akurasi sebesar 86% [10].

State of the art tulisan ini yaitu perbaikan atau modifikasi dari algoritma *backpropagation*. Perbaikan dengan penambahan momentum bertujuan untuk menghindari perubahan bobot yang signifikan, diakibatkan karena terdapat data yang sangat berbeda dengan yang lain. Momentum juga dapat meningkatkan waktu pelatihan dan stabilitas dari proses pelatihan.

2.1. Backpropagation

Backpropagation adalah salah satu model JST yang melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan [3].

Ada 3 fase pelatihan *backpropagation* yaitu propagasi maju, propagasi mundur, dan perubahan bobot. Berikut contoh algoritma pelatihan (*training*) untuk jaringan dengan satu layer tersembunyi dengan fungsi aktivasi *binary sigmoid* [3]:

Langkah 0: Inisialisasi semua bobot dengan nilai acak yang kecil.

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan 2-9.

Langkah 2: Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Fase I: Propagasi maju

Langkah 3: Tiap unit masukkan menerima sinyal masukkan dan meneruskannya ke unit tersembunyi.

Langkah 4: Hitung semua keluaran di unit tersembunyi $z_j (j = 1, 2, \dots, p)$

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \tag{1}$$

$$z_i = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \tag{2}$$

Langkah 5: Hitung semua keluaran jaringan di unit $y_k (k = 1, 2, \dots, m)$

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j w_{kj} \tag{3}$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \tag{4}$$

Fase II: Propagasi mundur

Langkah 6: Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran $y_k (k = 1, 2, \dots, m)$

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \tag{5}$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (Langkah 7).

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk mengubah bobot w_{kj}) dengan laju percepatan α

$$\Delta w_{kj} = \alpha \delta_k z_j; \tag{6}$$

($k = 1, 2, \dots, m; j = 0, 1, \dots, p$)

Langkah 7: Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi $z_j (j = 1, 2, \dots, p)$

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \tag{7}$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \tag{9}$$

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

$$\Delta v_{ij} = \alpha \delta_j x_i; \tag{10}$$

$$j = 1, 2, \dots, p; \quad i = 0, 1, \dots, n$$

Fase III: Perubahan Bobot

Langkah 8: Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (11)$$

($k = 1, 2, \dots, m$; $j = 0, 1, \dots, p$)

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ij} \quad (12)$$

($j = 1, 2, \dots, p$; $i = 0, 1, \dots, n$)

2.3. Mean Squared Error (MSE)

Perhitungan kesalahan pada keluaran jaringan digunakan untuk mengukur apakah jaringan sudah mengenali pola atau belum. Jika *error* yang diinginkan belum terpenuhi, maka jaringan dianggap belum mengenali pola. *Error* jaringan dihitung menggunakan *Mean Squared Error* (MSE). MSE adalah nilai rata-rata perbedaan yang dikuadratkan antara data aktual dengan data prediksi [14]. Rumus MSE dapat dituliskan pada persamaan sebagai berikut:

$$\text{MSE} = \frac{\sum (x_i - f_i)^2}{n} \quad (13)$$

Keterangan:

x_i : nilai aktual

f_i : nilai prediksi

n : jumlah data

2.4. Mean Absolute Percentage Error (MAPE)

Perhitungan besarnya *error* prediksi digunakan untuk mengukur akurasi persentase kesalahan dalam suatu prediksi. Persentase besarnya *error* prediksi dihitung dengan menggunakan *Mean Absolute Percentage Error* (MAPE). MAPE adalah nilai rata-rata dari keseluruhan persentase kesalahan antara data aktual dengan data prediksi [15]. Rumus MAPE dapat dituliskan pada persamaan sebagai berikut:

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \frac{|e_t|}{Y_t} \quad (14)$$

Keterangan:

e_i : *error* ke- i

n : jumlah data

2.5. Persentase Akurasi

Setelah mendapatkan persentase *error* (MAPE), maka persentase akurasi hasil prediksi dapat dihitung [13]. Rumus persentase akurasi dapat dituliskan pada persamaan sebagai berikut:

$$\text{Akurasi} = 100\% - \text{MAPE} \quad (15)$$

3. Metode Penelitian

3.1. Jenis Penelitian

Penelitian ini berfokus pada penggunaan metode komputasi yang digunakan untuk memprediksi NTP, yaitu metode JST backpropagation. Sedangkan aplikasi digunakan hanya sebagai alat pendukung penelitian yaitu melakukan perhitungan-perhitungan. Sehingga dapat dikategorikan ke dalam jenis penelitian Eksperimen.

3.2. Analisis Kebutuhan

a. Analisa Kebutuhan Masukan

Analisa kebutuhan masukan pada penelitian ini dibutuhkan data deret waktu (*time series*) NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan 12 bulan sebelumnya, sehingga untuk memprediksi NTP Subsektor Tanaman Pangan periode bulan yang akan datang, dibutuhkan NTP Subsektor Tanaman Pangan yang dimisalkan bulan ke-1, bulan ke-2, bulan ke-3, bulan ke-4, bulan ke-5, bulan ke-6, bulan ke-7, bulan ke-8, bulan ke-9, bulan ke-10, bulan ke-11, bulan ke-12 nantinya akan menjadi variabel yaitu $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}$, dan Y sebagai target. Variabel *input* dapat dilihat pada Tabel 3.1 berikut.

Tabel 1. Variabel Masukan

No	Variabel Masukan	Keterangan
1	X1	Bulan ke-1
2	X2	Bulan ke-2
3	X3	Bulan ke-3
4	X4	Bulan ke-4
5	X5	Bulan ke-5
6	X6	Bulan ke-6
7	X7	Bulan ke-7
8	X8	Bulan ke-8
9	X9	Bulan ke-9
10	X10	Bulan ke-10
11	X11	Bulan ke-11
12	X12	Bulan ke-12
13	Y	Target

b. Analisa Kebutuhan Proses

Analisa kebutuhan proses pada penelitian ini memprediksi NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan menerapkan JST *backpropagation* dengan menggunakan fungsi aktivasi sigmoid biner.

Data akan dibagi menjadi dua bagian, yaitu data latih dan data uji. Data latih yang digunakan adalah data bulan Januari 2011 sampai bulan Desember 2017. Sedangkan data uji yang digunakan adalah data bulan Januari 2018 sampai bulan Desember 2020.

Tabel 2. Data Latih dan data uji

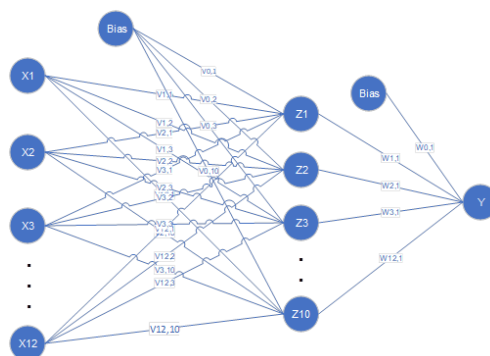
No	Masukan	Target	No	Masukan	Target
1	Januari 2011 – Desember 2011	Januari 2012	1	Januari 2018 – Desember 2018	Januari 2019
2	Februari 2011 – Januari 2012	Februari 2012	2	Februari 2018 – Januari 2019	Februari 2019
3	Maret 2011 – Februari 2012	Maret 2012	3	Maret 2018 – Februari 2019	Maret 2019
4	April 2011 – Maret 2012	April 2012	4	April 2018 – Maret 2019	April 2019
5	Mei 2011 – April 2012	Mei 2012	5	Mei 2018 – April 2019	Mei 2019
6	Juni 2011 – Mei 2012	Juni 2012	6	Juni 2018 – Mei 2019	Juni 2019
7	Juli 2011 – Juni 2012	Juli 2012	7	Juli 2018 – Juni 2019	Juli 2019
8	Agustus 2011 – Juli 2012	Agustus 2012	8	Agustus 2018 – Juli 2019	Agustus 2019
9	September 2011 – Agustus 2012	September 2012	9	September 2018 – Agustus 2019	September 2019
10	Oktober 2011 – September 2012	Oktober 2012	10	Oktober 2018 – September 2019	Oktober 2019
11	November 2011 – Oktober 2012	November 2012	11	November 2018 – Oktober 2019	November 2019
12	Desember 2011 – November 2012	Desember 2012	12	Desember 2018 – November 2019	Desember 2019
...
...
...
72	Desember 2016 – November 2017	Desember 2017	72	Desember 2019 – November 2020	Desember 2020

Sebelum masuk ke tahap proses pelatihan dan proses pengujian, data yang diberikan harus dinormalisasi terlebih dahulu. Data NTP Subsektor Tanaman Pangan akan dinormalisasi menggunakan Persamaan.

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

Kemudian melakukan inisialisasi bobot-bobot dengan nilai random atau acak yang cukup kecil. Nilai tersebut antara 0 sampai 1. Bobot ini berlaku untuk pembobotan dari *input layer* menuju *hidden layer* dan bobot dari *hidden layer* menuju *output layer*. Parameter JST yang digunakan yaitu, jumlah neuron pada *input layer* 12, jumlah neuron pada *hidden layer* 10, jumlah neuron pada *output layer* 1, alpha (laju pembelajaran) 0,3, momentum 0,8 dan toleransi *error* 0,01.

Berdasarkan jumlah neuron yang telah ditetapkan maka arsitektur JST untuk penelitian ini 12-10-1 dapat digambarkan sebagai berikut:



Gambar 1. Arsitektur JST

Berdasarkan arsitektur JST yang digunakan, maka data latih dengan target bulan Januari 2012 dapat disajikan pada tabel berikut.

Tabel 3. Data Latih dengan Target Bulan Januari 2012

Variabel Masukan	NTP Subsektor Tanaman Pangan Ternormalisasi
X1	0,7596
X2	0,7128
X3	0,7165
X4	0,7191
X5	0,8053
X6	0,7915
X7	0,7569
X8	0,7734
X9	0,7697
X10	0,8293
X11	0,8670
X12	0,9000
Y	0,8676

Nilai inisialisasi bobot V awal *input layer* menuju *hidden layer* dapat dilihat pada tabel berikut.

Tabel 4. Nilai Awal Bobot V

	1	2	3	4	5	6	7	8	9	10
0	0,17	0,23	0,75	0,17	0,67	0,45	0,55	0,22	0,66	0,61
1	0,28	0,75	0,11	0,66	0,71	0,66	0,12	0,51	0,26	0,85
2	0,15	0,73	0,54	0,49	0,67	0,74	0,39	0,73	0,48	0,47
3	0,84	0,31	0,74	0,96	0,84	0,28	0,26	0,35	0,61	0,55
4	0,51	0,23	0,58	0,48	0,76	0,85	0,39	0,88	0,87	0,72
5	0,17	0,78	0,72	0,81	0,61	0,39	0,17	0,42	0,85	0,41
6	0,68	0,84	0,93	0,92	0,73	0,73	0,23	0,73	0,47	0,30
7	0,73	0,79	0,62	0,86	0,57	0,84	0,94	0,46	0,33	0,54
8	0,49	0,87	0,11	0,34	0,30	0,63	0,58	0,80	0,20	0,65
9	0,30	0,23	0,56	0,27	0,84	0,67	0,83	0,96	0,20	0,84
10	0,63	0,26	0,34	0,68	0,79	0,62	0,61	0,90	0,31	0,63
11	0,66	0,77	0,73	0,63	0,14	0,76	0,10	0,70	0,59	0,69
12	0,66	0,21	0,63	0,59	0,63	0,82	0,55	0,59	0,51	0,66

Nilai inisialisasi bobot W awal *hidden layer* menuju *output layer* dapat dilihat pada tabel berikut.

Tabel 5. Nilai Awal Bobot W

	0	1	2	3	4	5	6	7	8	9	10
1	0,21	0,14	0,13	0,19	0,82	0,23	0,96	0,66	0,17	0,66	0,44

1. Propagasi Maju

a. Perhitungan Nilai Neuron Hidden Layer (Nilai Z)

Perhitungan nilai $Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9$, dan Z_{10} dilakukan dengan menggunakan Persamaan (2.2) dan (2.3).

Berikut adalah perhitungan untuk mencari nilai $Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9$, dan Z_{10} .

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

$$\begin{aligned}
 z_{in1} &= 0,17 + (0,28 * 0,7596) + (0,15 * 0,7128) + (0,84 * 0,7165) + (0,51 * 0,7191) + (0,17 * \\
 &0,8053) + (0,68 * 0,7915) + (0,74 * 0,7569) + (0,49 * 0,7734) + (0,3 * 0,7697) + (0,63 * \\
 &0,8293) + (0,66 * 0,867) + (0,66 * 0,9) = 5,0055 \\
 z_{in2} &= 0,23 + (0,75 * 0,7596) + (0,73 * 0,7128) + (0,31 * 0,7165) + (0,23 * 0,7191) + (0,78 * \\
 &0,8053) + (0,84 * 0,7915) + (0,79 * 0,7569) + (0,87 * 0,7734) + (0,23 * 0,7697) + (0,26 * \\
 &0,8293) + (0,77 * 0,867) + (0,45 * 0,21) = 5,5354 \\
 z_{in3} &= 0,75 + (0,11 * 0,7596) + (0,54 * 0,7128) + (0,74 * 0,7165) + (0,58 * 0,7191) + (0,72 * \\
 &0,8053) + (0,93 * 0,7915) + (0,62 * 0,7569) + (0,11 * 0,7734) + (0,56 * 0,7697) + (0,34 * \\
 &0,8293) + (0,73 * 0,867) + (0,63 * 0,9) = 5,9419 \\
 z_{in4} &= 0,17 + (0,66 * 0,7596) + (0,49 * 0,7128) + (0,96 * 0,7165) + (0,48 * 0,7191) + (0,81 * \\
 &0,8053) + (0,92 * 0,7915) + (0,86 * 0,7569) + (0,34 * 0,7734) + (0,27 * 0,7697) + (0,68 * \\
 &0,8293) + (0,63 * 0,867) + (0,59 * 0,9) = 6,2048 \\
 z_{in5} &= 0,67 + (0,71 * 0,7596) + (0,67 * 0,7128) + (0,84 * 0,7165) + (0,76 * 0,7191) + (0,61 * \\
 &0,8053) + (0,73 * 0,7915) + (0,57 * 0,7569) + (0,3 * 0,7734) + (0,84 * 0,7697) + (0,79 * \\
 &0,8293) + (0,14 * 0,867) + (0,63 * 0,9) = 6,5622 \\
 z_{in6} &= 0,45 + (0,66 * 0,7596) + (0,74 * 0,7128) + (0,28 * 0,7165) + (0,85 * 0,7191) + (0,39 * \\
 &0,8053) + (0,73 * 0,7915) + (0,84 * 0,7569) + (0,63 * 0,7734) + (0,67 * 0,7697) + (0,62 * \\
 &0,8293) + (0,76 * 0,867) + (0,82 * 0,9) = 6,7410 \\
 z_{in7} &= 0,55 + (0,12 * 0,7596) + (0,39 * 0,7128) + (0,26 * 0,7165) + (0,39 * 0,7191) + (0,17 * \\
 &0,8053) + (0,23 * 0,7915) + (0,94 * 0,7569) + (0,58 * 0,7734) + (0,83 * 0,7697) + (0,61 * \\
 &0,8293) + (0,10 * 0,867) + (0,55 * 0,9) = 4,5956 \\
 z_{in8} &= 0,22 + (0,51 * 0,7596) + (0,73 * 0,7128) + (0,35 * 0,7165) + (0,88 * 0,7191) + (0,42 * \\
 &0,8053) + (0,73 * 0,7915) + (0,46 * 0,7569) + (0,8 * 0,7734) + (0,96 * 0,7697) + (0,9 * \\
 &0,8293) + (0,7 * 0,867) + (0,59 * 0,9) = 6,5165 \\
 z_{in9} &= 0,66 + (0,26 * 0,7596) + (0,48 * 0,7128) + (0,61 * 0,7165) + (0,87 * 0,7191) + (0,85 * \\
 &0,8053) + (0,47 * 0,7915) + (0,33 * 0,7569) + (0,2 * 0,7734) + (0,2 * 0,7697) + (0,31 * \\
 &0,8293) + (0,59 * 0,867) + (0,51 * 0,9) = 5,1150 \\
 z_{in10} &= 0,61 + (0,85 * 0,7596) + (0,47 * 0,7128) + (0,55 * 0,7165) + (0,72 * 0,7191) + (0,41 * \\
 &0,8053) + (0,3 * 0,7915) + (0,54 * 0,7569) + (0,65 * 0,7734) + (0,84 * 0,7697) + (0,63 * \\
 &0,8293) + (0,69 * 0,867) + (0,66 * 0,9) = 6,3492
 \end{aligned}$$

$$\begin{aligned}
 z_i &= f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \\
 z_1 &= f(z_{in1}) = \frac{1}{1 + e^{-z_{in1}}} = \frac{1}{1 + e^{-5,0055}} = 0,9933 \\
 z_2 &= f(z_{in2}) = \frac{1}{1 + e^{-z_{in2}}} = \frac{1}{1 + e^{-5,5354}} = 0,9961 \\
 z_3 &= f(z_{in3}) = \frac{1}{1 + e^{-z_{in3}}} = \frac{1}{1 + e^{-5,9419}} = 0,9974 \\
 z_4 &= f(z_{in4}) = \frac{1}{1 + e^{-z_{in4}}} = \frac{1}{1 + e^{-6,2048}} = 0,9980 \\
 z_5 &= f(z_{in5}) = \frac{1}{1 + e^{-z_{in5}}} = \frac{1}{1 + e^{-6,5622}} = 0,9986 \\
 z_6 &= f(z_{in6}) = \frac{1}{1 + e^{-z_{in6}}} = \frac{1}{1 + e^{-6,7410}} = 0,9988 \\
 z_7 &= f(z_{in7}) = \frac{1}{1 + e^{-z_{in7}}} = \frac{1}{1 + e^{-4,5956}} = 0,9900 \\
 z_8 &= f(z_{in8}) = \frac{1}{1 + e^{-z_{in8}}} = \frac{1}{1 + e^{-6,5165}} = 0,9985 \\
 z_9 &= f(z_{in9}) = \frac{1}{1 + e^{-z_{in9}}} = \frac{1}{1 + e^{-5,1150}} = 0,9940 \\
 z_{10} &= f(z_{in10}) = \frac{1}{1 + e^{-z_{in10}}} = \frac{1}{1 + e^{-6,3492}} = 0,9983
 \end{aligned}$$

b. Perhitungan Nilai Neuron Pada Layer (Nilai Y)

Perhitungan nilai Y dilakukan dengan menggunakan Persamaan (2.4) dan (2.5). Berikut adalah perhitungan untuk mencari nilai Y1.

$$\begin{aligned}
 y_{net_k} &= w_{0k} + \sum_{j=1}^p z_j w_{kj} \\
 y_{in1} &= 0,21 + (0,14 * 0,9933) + (0,13 * 0,9961) + (0,19 * 0,9974) + (0,82 * 0,9980) + (0,23 * \\
 &0,9986) + (0,96 * 0,9988) + (0,66 * 0,9900) + (0,17 * 0,9985) + (0,66 * 0,9940) + \\
 &(0,44 * 0,9983) = 4,6034 \\
 y_k &= f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \\
 y_1 &= f(y_{in1}) = \frac{1}{1 + e^{-z_{in10}}} = \frac{1}{1 + e^{-4,6034}} = 0,9901
 \end{aligned}$$

2. Propagasi Mundur

a. Perhitungan Propagasi Mundur dari Output Layer ke Hidden Layer

Proses ini bertujuan untuk memperbarui nilai dari bobot W. Pembaruan nilai dari bobot W pada proses ini dilakukan dengan menggunakan Persamaan (2.6), (2.7), (2.11), dan (2.13). Berikut adalah perhitungan untuk memperbarui nilai bobot W.

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_k - y_k) y_k (1 - y_k)$$

$$\delta_1 = (0,8676 - 0,9901) * 0,9901 * (1 - 0,9901) = -0,0012028$$

$$\Delta w_{kj} = \alpha \delta_k z_j$$

$$\Delta w_{kj}(t + 1) = w_{kj}(t) + \alpha \delta_k w_j + \mu(w_{kj}(t) + w_{kj}(t - 1))$$

$$\Delta w_{0,1} = (0,3 * -0,0012028) + (0 * 0,8) = -0,0003608$$

$$\Delta w_{1,1} = (0,3 * -0,0012028 * 0,9933) + (0 * 0,8) = -0,0003584$$

$$\Delta w_{2,1} = (0,3 * -0,0012028 * 0,9961) + (0 * 0,8) = -0,0003594$$

$$\Delta w_{3,1} = (0,3 * -0,0012028 * 0,9974) + (0 * 0,8) = -0,0003599$$

$$\Delta w_{4,1} = (0,3 * -0,0012028 * 0,9980) + (0 * 0,8) = -0,0003601$$

$$\Delta w_{5,1} = (0,3 * -0,0012028 * 0,9986) + (0 * 0,8) = -0,0003603$$

$$\Delta w_{6,1} = (0,3 * -0,0012028 * 0,9988) + (0 * 0,8) = -0,0003604$$

$$\Delta w_{7,1} = (0,3 * -0,0012028 * 0,9990) + (0 * 0,8) = -0,0003572$$

$$\Delta w_{8,1} = (0,3 * -0,0012028 * 0,9985) + (0 * 0,8) = -0,0003603$$

$$\Delta w_{9,1} = (0,3 * -0,0012028 * 0,9940) + (0 * 0,8) = -0,0003587$$

$$\Delta w_{10,1} = (0,3 * -0,0012028 * 0,9983) + (0 * 0,8) = -0,0003602$$

Berikut adalah nilai dari bobot W terbaru.

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

$$w_{0,1}(\text{baru}) = 0,0003584 + 0,21 = 0,0996$$

$$w_{1,1}(\text{baru}) = -0,0003584 + 0,14 = 0,2496$$

$$w_{2,1}(\text{baru}) = -0,0003594 + 0,13 = 0,2496$$

$$w_{3,1}(\text{baru}) = -0,0003599 + 0,19 = 0,2496$$

$$w_{4,1}(\text{baru}) = -0,0003601 + 0,82 = 0,2496$$

$$w_{5,1}(\text{baru}) = -0,0003603 + 0,23 = 0,2496$$

$$w_{6,1}(\text{baru}) = -0,0003604 + 0,96 = 0,2496$$

$$w_{7,1}(\text{baru}) = -0,0003572 + 0,66 = 0,2496$$

$$w_{8,1}(\text{baru}) = -0,0003603 + 0,17 = 0,2496$$

$$w_{9,1}(\text{baru}) = -0,0003587 + 0,66 = 0,2496$$

$$w_{10,1}(\text{baru}) = -0,0003602 + 0,44 = 0,2496$$

Tabel berikut ini memperlihatkan nilai dari bobot W terbaru.

Tabel 6. Bobot W Terbaru

	0	1	2	3	4	5	6	7	8	9	10
1	0,2	0,1	0,1	0,2	0,8	0,2	1	0,658	0,171	0,657	0,444

b. Perhitungan Perambatan Mundur dari Hidden Layer ke Input Layer

Proses ini bertujuan untuk memperbarui nilai dari bobot V. Pembaruan bobot V pada proses ini dilakukan dengan menggunakan Persamaan (2.8), (2.9), (2.10), (2.12), dan (2.14). Berikut adalah perhitungan untuk memperbarui nilai bobot V.

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj}$$

$$\delta_{in1} = -0,0012028 * 0,14 = -0,0001732$$

$$\delta_{in2} = -0,0012028 * 0,13 = -0,0001600$$

$$\delta_{in3} = -0,0012028 * 0,19 = -0,0002285$$

$$\delta_{in4} = -0,0012028 * 0,82 = -0,0009899$$

$$\delta_{in5} = -0,0012028 * 0,23 = -0,0002803$$

$$\delta_{in6} = -0,0012028 * 0,96 = -0,0011499$$

$$\delta_{in7} = -0,0012028 * 0,66 = -0,0007914$$

$$\delta_{in8} = -0,0012028 * 0,17 = -0,0002057$$

$$\delta_{in9} = -0,0012028 * 0,66 = -0,0007902$$

$$\delta_{in10} = -0,0012028 * 0,44 = -0,0005340$$

$$\delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j)$$

$$\delta_1 = -0,0001732 * 0,9933 * (1 - 0,9933) = -0,0000011$$

$$\delta_2 = -0,0001600 * 0,9961 * (1 - 0,9961) = -0,0000006$$

$$\delta_3 = -0,0002285 * 0,9974 * (1 - 0,9974) = -0,0000006$$

$$\delta_4 = -0,0009899 * 0,9980 * (1 - 0,9980) = -0,0000020$$

$$\delta_5 = -0,0002803 * 0,9986 * (1 - 0,9986) = -0,0000004$$

$$\delta_6 = -0,0011499 * 0,9988 * (1 - 0,9988) = -0,0000014$$

$$\delta_7 = -0,0007914 * 0,9990 * (1 - 0,9990) = -0,0000078$$

$$\delta_8 = -0,0002057 * 0,9985 * (1 - 0,9985) = -0,0000003$$

$$\delta_9 = -0,0007902 * 0,9940 * (1 - 0,9940) = -0,0000047$$

$$\delta_{10} = -0,0005340 * 0,9983 * (1 - 0,9983) = -0,0000009$$

$$\Delta v_{ij} = \alpha \delta_j x_i$$

$$\Delta v(t + 1) = v_{ji}(t) + \alpha \delta_j x_i + \mu(v_{ji}(t) + v_{ji}(t - 1))$$

$$\Delta v_{0,1} = (0,3 * -0,0000011) + (0 * 0,5) = -0,0000003$$

$$\Delta v_{1,1} = (0,3 * -0,0000011 * 0,7596) + (0 * 0,5) = -0,0000003$$

$$\Delta v_{2,1} = (0,3 * -0,0000011 * 0,7128) + (0 * 0,5) = -0,0000003$$

$$\Delta v_{3,1} = (0,3 * -0,0000011 * 0,7165) + (0 * 0,5) = -0,0000003$$

.....dst

$$\Delta v_{12,1} = (0,3 * -0,0000011 * 0,9000) + (0 * 0,5) = -0,0000003$$

$$\Delta v_{0,2} = (0,3 * -0,0000006) = -0,0000002$$

$$\Delta v_{1,2} = (0,3 * -0,0000006 * 0,7596) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{2,2} = (0,3 * -0,0000006 * 0,7128) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{3,2} = (0,3 * -0,0000006 * 0,7165) + (0 * 0,5) = -0,0000001$$

.....dst

$$\Delta v_{12,2} = (0,3 * -0,0000006 * 0,9000) + (0 * 0,5) = -0,0000002$$

$$\Delta v_{0,3} = (0,3 * -0,0000006) = -0,0000002$$

$$\Delta v_{1,3} = (0,3 * -0,0000006 * 0,7596) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{2,3} = (0,3 * -0,0000006 * 0,7128) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{3,3} = (0,3 * -0,0000006 * 0,7165) + (0 * 0,5) = -0,0000001$$

.....dst

$$\Delta v_{12,3} = (0,3 * -0,0000006 * 0,9000) + (0 * 0,5) = -0,0000002$$

$$\Delta v_{0,4} = (0,3 * -0,0000020) = -0,0000006$$

$$\Delta v_{1,4} = (0,3 * -0,0000020 * 0,7596) + (0 * 0,5) = -0,0000005$$

$$\Delta v_{2,4} = (0,3 * -0,0000020 * 0,7128) + (0 * 0,5) = -0,0000004$$

$$\Delta v_{3,4} = (0,3 * -0,0000020 * 0,7165) + (0 * 0,5) = -0,0000004$$

.....dst

$$\Delta v_{12,4} = (0,3 * -0,0000020 * 0,9000) + (0 * 0,5) = -0,0000005$$

$$\Delta v_{0,5} = (0,3 * -0,0000004) = 0,0000001$$

$$\Delta v_{1,5} = (0,3 * -0,0000004 * 0,7596) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{2,5} = (0,3 * -0,0000004 * 0,7128) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{3,5} = (0,3 * -0,0000004 * 0,7165) + (0 * 0,5) = -0,0000001$$

.....dst

$$\Delta v_{12,5} = (0,3 * -0,0000004 * 0,9000) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{0,6} = (0,3 * -0,0000014) = -0,0000004$$

$$\Delta v_{1,6} = (0,3 * -0,0000014 * 0,7596) + (0 * 0,5) = -0,0000003$$

$$\Delta v_{2,6} = (0,3 * -0,0000014 * 0,7128) + (0 * 0,5) = -0,0000003$$

$$\Delta v_{3,6} = (0,3 * -0,0000014 * 0,7165) + (0 * 0,5) = -0,0000001$$

.....dst

$$\Delta v_{12,6} = (0,3 * -0,0000014 * 0,9000) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{0,7} = (0,3 * -0,0000078) = -0,0000023$$

$$\Delta v_{1,7} = (0,3 * -0,0000078 * 0,7596) + (0 * 0,5) = -0,0000018$$

$$\Delta v_{2,7} = (0,3 * -0,0000078 * 0,7128) + (0 * 0,5) = -0,0000017$$

$$\Delta v_{3,7} = (0,3 * -0,0000078 * 0,7165) + (0 * 0,5) = -0,0000017$$

.....dst

$$\Delta v_{12,7} = (0,3 * -0,0000078 * 0,9000) + (0 * 0,5) = -0,0000021$$

$$\Delta v_{0,8} = (0,3 * -0,0000003) = -0,0000001$$

$$\Delta v_{1,8} = (0,3 * -0,0000003 * 0,7596) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{2,8} = (0,3 * -0,0000003 * 0,7128) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{3,8} = (0,3 * -0,0000003 * 0,7165) + (0 * 0,5) = -0,0000001$$

.....dst

$$\Delta v_{12,8} = (0,3 * -0,0000003 * 0,9000) + (0 * 0,5) = -0,0000001$$

$$\Delta v_{0,9} = (0,3 * -0,0000004) = -0,0000014$$

$$\Delta v_{1,9} = (0,3 * -0,0000004 * 0,7596) + (0 * 0,5) = -0,0000011$$

$$\Delta v_{2,9} = (0,3 * -0,0000004 * 0,7128) + (0 * 0,5) = -0,0000010$$

$$\Delta v_{3,9} = (0,3 * -0,0000004 * 0,7165) + (0 * 0,5) = -0,0000010$$

.....dst

$$\Delta v_{12,9} = (0,3 * -0,0000004 * 0,9000) + (0 * 0,5) = -0,0000013$$

$$\Delta v_{0,10} = (0,3 * -0,0000009) = -0,0000003$$

$$\Delta v_{1,10} = (0,3 * -0,0000009 * 0,7596) + (0 * 0,5) = -0,0000002$$

$$\Delta v_{2,10} = (0,3 * -0,0000009 * 0,7128) + (0 * 0,5) = -0,0000002$$

$$\Delta v_{3,10} = (0,3 * -0,0000009 * 0,7165) + (0 * 0,5) = -0,0000002$$

.....dan seterusnya

$$\Delta v_{12,10} = (0,3 * -0,0000009 * 0,9000) + (0 * 0,5) = -0,0000001$$

Berikut adalah nilai dari bobot V terbaru.

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

$$v_{0,1}(\text{baru}) = 0,17 + (-0,0000003) = 0,173$$

$$v_{1,1}(\text{baru}) = 0,23 + (-0,0000003) = 0,284$$

$$v_{2,1}(\text{baru}) = 0,15 + (-0,0000002) = 0,146$$

$$v_{3,1}(\text{baru}) = 0,84 + (-0,0000002) = 0,838$$

.....dst

$$v_{12,1}(\text{baru}) = 0,66 + (-0,0000003) = 0,663$$

$$v_{0,2}(\text{baru}) = 0,23 + (-0,0000002) = 0,230$$

$$v_{1,2}(\text{baru}) = 0,75 + (-0,0000001) = 0,747$$

$$v_{2,2}(\text{baru}) = 0,73 + (-0,0000001) = 0,731$$

$$v_{3,2}(\text{baru}) = 0,31 + (-0,0000001) = 0,313$$

.....dst

$$v_{12,2}(\text{baru}) = 0,21 + (-0,0000002) = 0,212$$

$$v_{0,3}(\text{baru}) = 0,75 + (-0,0000002) = 0,750$$

$$v_{1,3}(\text{baru}) = 0,11 + (-0,0000001) = 0,106$$

$$v_{2,3}(\text{baru}) = 0,54 + (-0,0000001) = 0,539$$

$$v_{3,3}(\text{baru}) = 0,74 + (-0,0000001) = 0,736$$

.....dst

$$v_{12,3}(\text{baru}) = 0,63 + (-0,0000002) = 0,634$$

$$v_{0,4}(\text{baru}) = 0,17 + (-0,0000006) = 0,173$$

$$v_{1,4}(\text{baru}) = 0,66 + (-0,0000005) = 0,664$$

$$v_{2,4}(\text{baru}) = 0,49 + (-0,0000004) = 0,493$$

$$v_{3,4}(\text{baru}) = 0,96 + (-0,0000004) = 0,956$$

.....dst

$$v_{12,4}(\text{baru}) = 0,59 + (-0,0000005) = 0,589$$

$$v_{0,5}(\text{baru}) = 0,67 + (-0,0000001) = 0,673$$

$$v_{1,5}(\text{baru}) = 0,71 + (-0,0000001) = 0,712$$

$$v_{2,5}(\text{baru}) = 0,67 + (-0,0000001) = 0,672$$

$$v_{3,5}(\text{baru}) = 0,84 + (-0,0000001) = 0,845$$

.....dst

$$v_{12,5}(\text{baru}) = 0,63 + (-0,0000001) = 0,633$$

.....

.....

$$v_{0,10}(\text{baru}) = 0,61 + (-0,0000003) = 0,611$$

$$v_{1,10}(\text{baru}) = 0,85 + (-0,0000002) = 0,853$$

$$v_{2,10}(\text{baru}) = 0,47 + (-0,0000002) = 0,469$$

$$v_{3,10}(\text{baru}) = 0,55 + (-0,0000002) = 0,547$$

.....dst

$$v_{12,10}(\text{baru}) = 0,66 + (-0,0000001) = 0,663$$

Tabel berikut ini memperlihatkan nilai dari bobot V terbaru.

Tabel 7. Bobot V Terbaru

	1	2	3	4	5	6	7	8	9	10
0	0,173	0,230	0,750	0,173	0,673	0,445	0,553	0,222	0,663	0,611
1	0,284	0,747	0,106	0,664	0,673	0,662	0,115	0,506	0,257	0,853
2	0,146	0,731	0,539	0,493	0,672	0,738	0,391	0,732	0,482	0,469
3	0,838	0,313	0,736	0,956	0,845	0,280	0,262	0,355	0,610	0,547
4	0,512	0,233	0,576	0,482	0,755	0,852	0,390	0,875	0,874	0,725
5	0,174	0,783	0,718	0,812	0,612	0,393	0,165	0,420	0,848	0,414
6	0,681	0,844	0,933	0,920	0,726	0,726	0,235	0,731	0,475	0,297
7	0,735	0,792	0,622	0,864	0,567	0,844	0,945	0,457	0,333	0,539
8	0,492	0,871	0,110	0,338	0,304	0,633	0,582	0,803	0,198	0,652
9	0,302	0,233	0,561	0,272	0,841	0,673	0,827	0,957	0,203	0,842
10	0,631	0,257	0,335	0,681	0,790	0,622	0,609	0,899	0,313	0,630
11	0,664	0,773	0,730	0,626	0,135	0,762	0,102	0,696	0,592	0,686
12	0,663	0,212	0,634	0,589	0,633	0,822	0,549	0,595	0,505	0,663

Perhitungan nilai MSE dilakukan dengan menggunakan Persamaan.

$$MSE = \frac{\sum(x_i - f_i)^2}{n}$$

$$MSE = \frac{\sum(0,9901 - 0,8676)^2}{1} = \frac{0,1224814}{1} = 0,0150017$$

Karena nilai MSE lebih kecil dari toleransi *error* yang sudah ditetapkan yaitu 0,01 maka pelatihan akan dihentikan dengan satu iterasi saja.

3.3. Metode Pemilihan Sampel

Metode pemilihan sampel yang digunakan dalam penelitian ini adalah Quota Sampling dimana pengambilan sampel dilakukan sebanyak jumlah yang telah ditentukan oleh peneliti.

Data yang digunakan dalam penelitian ini adalah data bulanan NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan bulan Januari 2011 sampai Desember 2020 yang diperoleh dari situs resmi Badan Pusat Statistik (BPS) Kalimantan Selatan dengan jumlah 120 data. Data NTP Subsektor Tanaman Pangan dapat dilihat pada berikut.

Tabel 8. Data NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan

Bulan	NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan									
	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Januari	106,70	108,72	105,31	96,82	101,67	103,28	97,82	95,39	95,08	102,60
Februari	105,82	107,74	106,01	97,57	103,06	104,12	98,25	96,86	95,43	102,31
Maret	105,89	107,77	105,46	98,97	104,74	102,38	96,84	95,64	93,06	101,82
April	105,94	107,32	104,80	98,97	101,94	100,38	95,86	92,98	91,98	99,85
Mei	107,56	107,75	105,00	99,06	102,31	98,96	95,56	91,69	93,26	101,56
Juni	107,30	107,37	105,49	98,76	102,78	98,12	95,41	91,66	93,39	102,68
Juli	106,65	106,87	103,92	97,88	101,87	98,30	94,98	91,50	92,85	101,79
Agustus	106,96	106,38	102,31	97,41	101,66	97,51	94,80	92,14	92,93	100,79
September	106,89	106,66	102,61	97,89	101,59	97,08	94,71	93,74	93,48	101,51
Oktober	108,01	106,24	103,27	98,57	101,87	98,14	95,26	93,65	92,90	101,87
November	108,72	106,34	96,46	98,26	101,89	98,03	94,47	93,88	93,88	102,10
Desember	109,34	106,01	95,56	98,97	102,66	97,23	94,30	93,99	95,94	103,24

4. Pembahasan dan Pengujian

4.1.1. Pengujian

Pengujian akan dilakukan dengan meliputi empat pengujian yaitu pengujian jumlah data latih dan data uji, pengujian jumlah neuron pada *hidden layer*, pengujian laju pembelajaran (*alpha*), dan pengujian momentum terhadap persentase *error* (MAPE) dan persentase akurasi.

a. Pengujian Jumlah Data Latih dan Data Uji

Pengujian ini dilakukan dengan melihat pengaruh jumlah data latih dan data uji yang digunakan terhadap persentase *error* (MAPE) dan persentase akurasi. Pada pengujian ini parameter JST yang digunakan akan bernilai tetap, namun jumlah data latih dan data uji yang digunakan akan berubah-ubah pada setiap percobaan. Adapun jumlah data latih dan data uji yang digunakan pada percobaan ini adalah data latih 72 dan data uji 48, data latih 84 dan data uji 36, dan data latih 96 dan data uji 24. Berikut adalah parameter JST lainnya yang digunakan dalam pengujian ini jumlah neuron pada *input layer* 12, jumlah neuron pada *hidden layer* 10, jumlah neuron pada *output layer* 1, laju pembelajaran (*alpha*) 0,3, toleransi *error* 0,001, jumlah iterasi 1000, dan momentum 0,8.

Tabel 9. Hasil Pengujian Jumlah Data Latih dan Data Uji

No	Jumlah Data Latih dan Data Uji	Waktu Komputasi	MSE	MAPE	Akurasi (%)
1	Data Latih 72 dan Data Uji 48	134,397	0,00304672	1,59	98,41
2	Data Latih 84 dan Data Uji 36	151,414	0,00284669	0,97	99,03
3	Data Latih 96 dan Data Uji 24	179,6	0,00253593	1,46	98,54

Pada tabel hasil pengujian jumlah data latih dan data uji, terdapat percobaan jumlah data latih dan data uji yang terbaik yaitu data latih 84 dan data uji 36 dengan waktu komputasi 151,414 detik, nilai MSE 0,00284669, nilai MAPE 0,97, dan akurasi 99,03 %. Dari berbagai percobaan dengan jumlah data latih dan data uji yang berbeda-beda dapat disimpulkan bahwa semakin besar jumlah data latih dan data uji yang digunakan maka waktu komputasi pada proses pelatihan cenderung semakin lama. Begitu juga dengan tingkat akurasi hasil pengujian yang didapatkan cenderung meningkat atau semakin tinggi.

b. Pengujian Jumlah Neuron Pada Hidden Layer

Pengujian ini dilakukan dengan melihat pengaruh perubahan jumlah neuron pada *hidden layer* terhadap persentase *error* (MAPE) dan persentase akurasi. Pada pengujian ini nilai dari parameter jumlah neuron pada *hidden layer* akan berubah-ubah sedangkan nilai parameter JST lainnya bernilai tetap. Adapun parameter jumlah neuron pada *hidden layer* yang digunakan pada percobaan ini adalah 5, 10, dan 15. Berikut adalah parameter JST lainnya yang digunakan dalam pengujian ini jumlah neuron pada *input layer* 12, jumlah neuron pada *output layer* 1, laju pembelajaran (α) 0,3, toleransi *error* 0,001, jumlah iterasi 1000, dan momentum 0,8.

Tabel 10. Hasil Pengujian Jumlah Neuron Pada Hidden Layer

No	Jumlah Neuron Pada Hidden Layer	Waktu Komputasi	MSE	MAPE	Akurasi (%)
1	5	85,945	0,0024963	1,01	98,99
2	10	151,414	0,00284669	0,97	99,03
3	15	221,161	0,0029896	0,99	99,01

Pada tabel hasil pengujian jumlah neuron pada *hidden layer*, terdapat percobaan jumlah neuron pada *hidden layer* yang terbaik yaitu 10 dengan waktu komputasi 151,414 detik, nilai MSE 0,00284669, nilai MAPE 0,97, dan akurasi 99,03 %. Dari berbagai percobaan dengan jumlah neuron pada *hidden layer* yang berbeda-beda dapat disimpulkan bahwa semakin besar jumlah neuron pada *hidden layer* maka waktu komputasi yang dibutuhkan pada proses pelatihan akan cenderung semakin lama. Begitu juga dengan tingkat akurasi hasil pengujian yang didapatkan akan cenderung semakin tinggi seiring dengan meningkatnya jumlah neuron pada *hidden layer*.

c. Pengujian Laju Pembelajaran (Alpha)

Pengujian ini dilakukan dengan melihat pengaruh perubahan laju pembelajaran (α) terhadap persentase *error* (MAPE) dan persentase akurasi. Pada pengujian ini nilai parameter laju pembelajaran (α) akan berubah-ubah sedangkan nilai parameter JST lainnya bernilai tetap. Adapun laju pembelajaran (α) yang digunakan pada percobaan ini adalah 0,3, 0,5, dan 0,8. Berikut adalah parameter JST lainnya yang digunakan dalam pengujian ini jumlah neuron pada *input layer* 12, jumlah neuron pada *hidden layer* 10, jumlah neuron pada *output layer* 1, toleransi *error* 0,001, jumlah iterasi 1000, dan momentum 0,8.

Tabel 11. Hasil Pengujian Laju Pembelajaran (Alpha)

No	Nilai Laju Pembelajaran (Alpha)	Waktu Komputasi	MSE	MAPE	Akurasi (%)
1	0,3	151,414	0,00284669	0,97	99,03
2	0,5	159,296	0,00249667	1,12	98,88
3	0,8	150,945	0,00215124	2,05	97,95

Pada tabel hasil pengujian laju pembelajaran (α), terdapat percobaan laju pembelajaran (α) yang terbaik yaitu 0,3 dengan waktu komputasi 151,414 detik, nilai MSE 0,00284669, nilai MAPE 0,97, dan akurasi 99,03 %. Dari berbagai percobaan dengan laju pembelajaran (α) yang berbeda-beda dapat disimpulkan bahwa semakin besar nilai laju pembelajaran yang digunakan maka nilai MSE pada proses pelatihan akan semakin kecil. Begitu

juga dengan tingkat akurasi hasil pengujian yang didapatkan akan semakin kecil seiring dengan bertambahnya nilai parameter laju pembelajaran (alpha).

d. Pengujian Momentum

Pengujian ini dilakukan dengan melihat pengaruh perubahan momentum terhadap persentase *error* (MAPE) dan persentase akurasi. Pada pengujian ini nilai parameter momentum akan berubah-ubah sedangkan nilai parameter JST lainnya bernilai tetap. Adapun momentum yang digunakan pada percobaan ini adalah 0,3, 0,5, dan 0,8. Berikut adalah parameter JST lainnya yang digunakan dalam pengujian ini jumlah neuron pada *input layer* 12, jumlah neuron pada *hidden layer* 10, jumlah neuron pada *output layer* 1, laju pembelajaran (alpha) 0,3, toleransi *error* 0,001, dan jumlah iterasi 1000.

Tabel 12. Hasil Pengujian Momentum

No	Nilai Momentum	Waktu Komputasi	MSE	MAPE	Akurasi (%)
1	0,3	126,065	0,00439961	1,61	98,39
2	0,5	153,474	0,00500263	2,31	97,69
3	0,8	151,414	0,00284669	0,97	99,03

Pada tabel hasil pengujian momentum, terdapat percobaan momentum yang terbaik yaitu 0,8 dengan waktu komputasi 151,414 detik, nilai MSE 0,00284669, nilai MAPE 0,97, dan akurasi 99,03 %. Dari berbagai percobaan dengan momentum yang berbeda-beda dapat disimpulkan semakin besar nilai momentum yang digunakan maka waktu komputasi yang dibutuhkan pada proses pelatihan akan cenderung semakin lama. Begitu juga dengan tingkat akurasi hasil pengujian yang didapatkan akan semakin tinggi seiring dengan bertambahnya nilai parameter momentum.

4.1.2 Hasil Prediksi (Pengujian)

Dari hasil pengujian didapatkan parameter terbaik adalah data latih 84 dan data uji 36, jumlah neuron pada *hidden layer* 10, toleransi *error* 0,001, laju pembelajaran (alpha) 0,3, jumlah iterasi 1000, dan momentum 0,8. Sehingga menghasilkan nilai MAPE terkecil yaitu 0,97 dan akurasi 99,03 %. Berikut adalah tabel hasil prediksi NTP Subsektor Tanaman Pangan bulan Januari 2019 sampai Desember 2020 berdasarkan data uji.

Tabel 13. Hasil Prediksi NTP

No	Tahun	Bulan	Keluaran JST	Keluaran Sebenarnya	Error
1	2019	Januari	94,36	95,08	0,72
2	2019	Februari	95,22	95,43	0,21
3	2019	Maret	94,76	93,06	1,7
4	2019	April	92,63	91,98	0,65
5	2019	Mei	91,94	93,26	1,32
6	2019	Juni	93,01	93,39	0,38
7	2019	Juli	93,18	92,85	0,33
8	2019	Agustus	93,07	92,93	0,14
9	2019	September	93,32	93,48	0,16
10	2019	Oktober	93,52	92,9	0,62
11	2019	November	92,58	93,88	1,3
12	2019	Desember	93,91	95,94	2,03
13	2020	Januari	97,52	102,6	5,08
14	2020	Februari	102,87	102,31	0,56
15	2020	Maret	101,75	101,82	0,07
16	2020	April	101,7	99,85	1,85
17	2020	Mei	100,23	101,56	1,33
18	2020	Juni	102,04	102,68	0,64
19	2020	Juli	100,67	101,79	1,12
20	2020	Agustus	100,72	100,79	0,07
21	2020	September	101,03	101,51	0,48
22	2020	Oktober	103,05	101,87	1,18
23	2020	November	101,51	102,1	0,59
24	2020	Desember	102,77	103,24	0,47

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|e_i|}{Y_i} = 0,97$$

Akurasi = 100% - MAPE = 99,03

Berdasarkan hasil prediksi (pengujian) untuk prediksi NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan menerapkan JST *backpropagation* dapat membantu dalam memprediksi NTP Subsektor Tanaman Pangan dan kemudian dibandingkan dengan data

sebenarnya yang telah ada selama ini dengan hasil akurasi 99,03%. *Error* terkecil terdapat pada bulan Maret dan agustus dengan nilai *error* 0,07. Sedangkan nilai *error* terbesar terdapat pada bulan Januari dengan nilai 5,08. Namun jika dirata-ratakan, nilai *error* hanya sebesar 0,97.

5. Simpulan

Algoritma JST *backpropagation* berhasil melakukan prediksi NTP Subsektor Tanaman Pangan Provinsi Kalimantan Selatan untuk periode bulan yang akan datang dengan tingkat akurasi yang tinggi yaitu 99,03 %. Dari hasil pengujian didapatkan parameter terbaik adalah data latih 84 dan data uji 36, jumlah neuron pada *hidden layer* 10, toleransi *error* 0,001, laju pembelajaran (α) 0,3, jumlah iterasi 1000, dan momentum 0,8. Sehingga menghasilkan nilai MAPE terkecil yaitu 0,97 dan akurasi 99,03 %.

Daftar Referensi

- [1] Badan Pusat Statistik Provinsi Kalimantan Selatan, *Nilai Tukar Petani Provinsi Kalimantan Selatan 2019*. Banjarbaru: Badan Pusat Statistik Provinsi Kalimantan Selatan, 2019.
- [2] A. Hermawan, *Jaringan Saraf Tiruan, Teori, dan Aplikasi*. Surabaya: Penerbit Andi, 2006.
- [3] J. . Siang, *Jaringan Saraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: Penerbit Andi, 2009.
- [4] B. Bahar, S.A. Yahya, "Penerapan Algoritma Backpropagation Untuk Prediksi Kebutuhan Air Bersih pada PDAM Intan Banjar", *Progresif: Jurnal Ilmiah Komputer*, vol. 15, no. 1, pp. 51-60, 2019.
- [5] T.W. Khusniyah, S. Sutikno. "Prediksi Nilai Tukar Petani Menggunakan Jaringan Syaraf Tiruan Backpropagation". *Science Journal Informatics*, Vol. 3, No. 1, pp. 11–18, 2016.
- [6] E. Widada. "Prediksi Nilai Tukar Petani (NTP) D.I. Yogyakarta Menggunakan Jaringan Saraf Tiruan Dengan Algoritma Backpropagation". Thesis. Yoyakarta: Postgraduate Universitas Teknologi Yogyakarta; 2018.
- [7] S. P. Siregar, D. Hartama, A. Wanto. "Estimasi Nilai Tukar Petani Subsektor Tanaman Pangan Menggunakan JST pada Provinsi Sumatera Utara". *Proceedings National Workshop on Riset Information Science*. Yogyakarta, vol. 1, pp. 369, 2019.
- [8] W. Istiqomah and M. Y. Darsyah, "Efektivitas Metode Arima Dan Exponential Smoothing Untuk Meramalkan Nilai Tukar Petani Di Jawa Tengah Effectiveness of the Arima Method and Exponential Smoothing to Predict Farmer Exchange Rates in Central Jawa", *Proceedings National Wokrshop Mahasiswa Unimus*, Vol. 1, pp. 343–350, 2018.
- [9] I. Afkarina. Penerapan Algoritma Levenberg-Marquardt Untuk Memprediksi Nilai Tukar Petani (NTP) Subsektor Tanaman Pangan. Skripsi. Riau: Sarjana S1 Universitas Islam Negeri Sultan Syarif Kasim Riau; 2020.
- [10] E. Irawan, M. Zarlis, E. B. Nababan. "Analisis Penambahan Nilai Momentum Pada Prediksi Produktivitas Kelapa Sawit Menggunakan Backpropagation", *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, Vol. 1, No. 2, pp. 84–89, 2017.
- [11] S. P. Siregar and A. Wanto. "Analysis of Artificial Neural Network Accuracy Using Backpropagation Algorithm In Predicting Process (Forecasting)", *IJISTECH (International Journal Information System Technology*, vol. 1, no. 1, pp. 34, 2017.
- [12] K. Nugroho. "Model Analisis Prediksi Menggunakan Metode Fuzzy Time Series". *Infokam*, Vol. 12, No. 1, pp. 46–50, 2016.
- [13] D. Jauhari, A. Himawan, C. Dewi. "Prediksi Distribusi Air PDAM Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation Di PDAM Kota Malang", *Jurnal Teknologi Informai dan Ilmu Komputer*, Vol. 3, No. 2, pp. 83, 2016.
- [14] R. U. Kuswana, G. Abdillah, and A. Komarudin. "Prediksi Realisasi Penerimaan Pajak Bumi dan Bangunan di Pemerintah Kabupaten Bandung Barat Menggunakan Jaringan Syaraf Tiruan Backpropagation", *Proceedings of Seminar Nasional Sains dan Teknologi (SENSASI)*, Medan, Vol. 1, pp. 580–585, 2019.
- [15] M. M. Bethatian and R. Amri. "Permodelan Jaringan Saraf Tiruan Menggunakan Metode Backpropagation Untuk Prediksi Beban Listrik Di Sumatera Bagian Tengah". *Jurnal Online Mahasiswa Fakultas Teknik Universitas Riau*, vol. 6, pp. 1–8, 2019.