



A Neural Network Prediction of Ozone Level in Mataram City

Nurul Hikmah^{a,*}, Syamsul Bahri^b, Irwansyah^c

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Mataram, Jl. Majapahit No. 62, Mataram, 83125, Indonesia, Email: nurulhikmah4651@gmail.com^a, syamsul.math@unram.ac.id^b, irw@unram.ac.id^c

ABSTRACT

Ozone tropospher (O_3) is one of the pollutants in the environment of Mataram City, Lombok, NTB, Indonesia. Based on the data obtained from the Agency of Environment and Forestry of West Nusa Tenggara Province, ozone pollutant concentrations in Mataram City have changed unpredictably. One time pollutant concentrations increase and then decrease, but then quickly increase again significantly. Therefore, the concentrations of ozone pollutant must be monitored because its presence at certain levels can cause various negative effects human health and the environment. Changes in ozone pollutant concentrations can be identified by carrying out a method of predicting ozone pollutant levels so that a decision can be taken to prevent the negative impact of the pollutant. In this research, a backpropagation artificial neural network is used to find the model prediction of the concentration of ozone in Mataram City. The input variables that are used in this network are air temperature (x_1), wind direction (x_2), wind speed (x_3), humidity (x_4), solar radiation (x_5), concentration of NO_2 (x_6), the concentration of SO_2 (x_7) and the concentration of O_3 a day before (x_8) for the period of 6 July 2018 to 31 May 2019. The method in this study was to conduct trial and error on 60 different combinations of network architectures and parameters. Then all the network architectures performance will be compared based on the RMSE, MAPE and R^2 indicators. Based on this research, the best neural network model to predict the concentration of ozone pollutant in Mataram City is the network with architecture 8-20-1, with logsig-purelin activation function and trainlm learning function. The performance of the training model is RMSE = 0.011, MAPE = 1,043 % and $R^2 = 0,9566$. Meanwhile, the performance of the testing model is RMSE = 0.001, MAPE = 0.749 % and $R^2 = 0.497$.

Keywords : air pollution; artificial neural network; backpropagation algorithm; concentration of ozone pollutant; prediction

Diserahkan: 20-04-2022; Diterima: 30-12-2022;

Doi: <https://doi.org/10.29303/emj.v5i2.129>

* Corresponding author.

Alamat e-mail: nurulhikmah4651@gmail.com

1. Pendahuluan

Udara yang bersih merupakan komponen penting dalam kehidupan makhluk hidup, termasuk manusia. Namun, seiring perkembangan zaman, kualitas udara menurun akibat berbagai macam aktivitas manusia yang menghasilkan beragam polutan ke udara (Octavianti & Utami, 2014).

Beberapa polutan yang terkandung di udara adalah timbal (Pb), ozon (O₃) troposfer, karbon monoksida (CO), sulfur dioksida (SO₂), dan nitrogen dioksida (NO₂) (Mustafidah, dkk., 2013). Dari berbagai macam polutan tersebut, ozon troposfer adalah salah satu polutan yang ada di lingkungan Kota Mataram, Lombok, NTB, Indonesia (DLHK NTB, 2019). Walaupun secara garis besar kualitas udara di Kota Mataram masih baik (Laporan DPPU, 2018), kadar polutan ozon tetap harus dipantau karena keberadaannya dalam kadar tertentu dapat menimbulkan berbagai efek negatif baik bagi kesehatan manusia maupun bagi lingkungan (Arifien, dkk., 2012).

Berdasarkan data yang diperoleh dari laboratorium lingkungan Dinas Lingkungan Hidup dan Kehutanan Provinsi NTB di Kota Mataram nilai rata-rata kadar polutan ozon pada bulan Agustus 2018, September 2018, dan Oktober 2018 terjadi peningkatan, dengan nilai masing-masing adalah sebesar 59,4914 $\mu\text{g}/\text{m}^3$, 78,97168 $\mu\text{g}/\text{m}^3$, dan 136,4303 $\mu\text{g}/\text{m}^3$. Namun dari bulan November 2018 sampai dengan Januari 2019 terjadi penurunan kadar polutan ozon. Kemudian pada bulan Februari 2019 terjadi peningkatan lagi, dengan nilai yang sangat signifikan. Pada bulan itu, rata-rata kadar ozon di Kota Mataram mencapai nilai 948,8267 $\mu\text{g}/\text{m}^3$ (DLHK NTB, 2019).

Berdasarkan data tersebut, kadar polutan ozon mengalami perubahan yang tidak menentu. Suatu waktu kadar polutan meningkat kemudian menurun, namun dengan cepat dapat meningkat lagi secara signifikan. Oleh karena itu, untuk mengetahui perubahan kadar polutan ozon ini, suatu metode prediksi perlu dilakukan. Apabila hasil prediksi menyatakan kadar polutan meningkat, maka dapat dilakukan pengambilan keputusan dalam aksi pencegahan agar dampak negatif dari polutan udara dapat diminimalisir. Selain itu prediksi juga dapat dijadikan sebagai salah satu upaya untuk meningkatkan sistem pemantauan kualitas udara yang ada.

Penelitian yang berkaitan dengan metode prediksi sudah sangat banyak dan akan terus bertambah karena semakin kompleks permasalahan yang ada. Metode prediksi yang sering digunakan dalam penelitian adalah metode prediksi kuantitatif dengan data runtun waktu (*time series*). Metode

prediksi *time series* terbagi menjadi dua bagian yaitu model prediksi yang didasarkan pada model matematika statistik dan metode *soft computing* (kecerdasan buatan) (Gunaryati & Suhendra, 2015).

Model analisis *time series* dengan teknik statistika seperti *Autoregressive Moving Average* (ARIMA) *Box Jenkins*, memiliki keterbatasan dengan mensyaratkan data bersifat stasioner dan linier. Dalam kenyataannya, data hasil penelitian/observasi umumnya bersifat *non*-linier dan *non*-stasioner. Sejak tahun 1990, para peneliti telah mengembangkan metode berbasis *soft computing* untuk mengatasi permasalahan ini. Metode yang termasuk teknik *soft computing*, salah satunya adalah jaringan syaraf tiruan (JST). Keuntungan utama dari JST adalah kemampuan jaringan yang adaptif, *learning algorithm by self*, generalisasi dan menyelesaikan masalah *non*-linier yang kompleks dan rumit (Bahri, dkk., 2019).

Selain itu dalam Aprianto dkk., (2018), dikatakan bahwa penggunaan jaringan syaraf tiruan merupakan metode yang efektif untuk melakukan adaptasi terhadap kondisi atmosfer yang bersifat *non*-linier. JST juga mampu melakukan memorisasi data dan generalisasi data suatu kejadian, sehingga dapat diketahui kejadian yang akan datang. Kajian mengenai prediksi terhadap kualitas udara menggunakan JST *Backpropagation* telah dilakukan oleh beberapa peneliti seperti Arifien dkk. (2012) dan Aprianto dkk. (2018).

Penelitian yang dilakukan oleh Arifien dkk. (2012) adalah memprediksi kadar polutan ozon troposfer menggunakan jaringan syaraf tiruan untuk pemantauan kualitas udara di Kota Surabaya. Model JST yang digunakan adalah tipe *multivariate* dan *time series*. Hasil dari kedua model tersebut dibandingkan untuk mendapatkan model terbaik. Parameter pemilihan model didasarkan pada nilai *root mean square error* (RMSE) dan koefisien determinasi (R^2). Penelitian ini menghasilkan kesimpulan bahwa JST tipe *multivariate* lebih baik dibandingkan dengan JST tipe *time series*. Model JST terbaik untuk melakukan prediksi terhadap kadar polutan O₃ di esok hari adalah JST *multivariate* dengan arsitektur 7-20-20-1 dan fungsi aktivasi berupa *tansig* pada *input layer*, *logsig* pada *hidden layer* serta *purelin* pada *output layer*.

Selanjutnya, penelitian yang dilakukan oleh Aprianto dkk., (2018) adalah memprediksi kadar *Particulate Matter* (PM₁₀) menggunakan jaringan syaraf tiruan *backpropagation* untuk pemantauan kualitas udara di Kota Pontianak. Data yang digunakan sebagai masukan (*input*) jaringan pada penelitian ini yaitu parameter cuaca seperti curah hujan, penyinaran matahari, kelembaban udara, kecepatan angin dan temperatur udara, sedangkan

target jaringan adalah kadar PM_{10} . Arsitektur jaringan syaraf tiruan yang digunakan tersusun dari neuron sebanyak 20-20-15-15-10-1 dan menggunakan fungsi aktivasi *tansig-logsig-tansig-logsig-tansig-purelin* pada setiap lapisan jaringan. Hasil dari penelitian ini menunjukkan bahwa sebesar 93,34 % parameter cuaca dapat digunakan sebagai parameter masukan pada JST untuk memprediksi kadar PM_{10} di Kota Pontianak.

Berdasarkan kedua penelitian tersebut dan permasalahan pencemaran udara di kota Mataram, dalam penelitian ini akan digunakan metode jaringan syaraf tiruan *backpropagation* untuk menentukan model prediksi kadar polutan ozon di kota Mataram. Penelitian ini serupa dengan yang dilakukan oleh Arifien dkk. (2012), yaitu memprediksi kadar polutan ozon, namun masukan jaringan yang akan digunakan sama seperti penelitian Aprianto dkk., (2018). Pada penelitian ini selain parameter cuaca, prekursor ozon juga dijadikan sebagai masukan jaringan.

Melalui penelitian ini, diharapkan dapat memberikan hasil prediksi yang baik sehingga dapat memberikan informasi mengenai kadar polutan ozon di Kota Mataram serta pihak pengambil kebijakan akan lebih mudah dalam menentukan arah kebijakan terkait pengendalian pencemaran udara.

1. Landasan Teori

1.1. Ozon (O₃) Troposfer

Jenis ozon yang bersifat polutan dan berbahaya bagi kesehatan adalah ozon yang terdapat di lapisan troposfer. Ozon troposfer dapat terbentuk dari proses alami serta proses aktivitas manusia. Secara alami, ozon troposfer dihasilkan oleh hidrokarbon yang dilepas oleh tanaman dan tanah, serta berasal dari ozon yang turun dari lapisan stratosfer menuju permukaan bumi.

Di sisi lain, ozon troposfer juga dapat diperoleh berdasarkan reaksi kimia kompleks yang terjadi di atmosfer dari berbagai prekursornya seperti NO_x , VOCs, CH_4 , dan CO saat adanya radiasi sinar UV (Octavianti & Utami, 2014). Gas NO_2 juga merupakan salah satu polutan yang termasuk prekursor utama dalam pembentukan ozon di atmosfer (Serlina, 2020). Prekursor-prekursor ozon tersebut mempengaruhi kadar polutan ozon di atmosfer. Semakin tinggi emisi prekursor ozon tersebut, maka kadar polutan ozon yang terbentuk akan meningkat.

Kadar polutan ozon juga dipengaruhi oleh faktor lain yaitu kelembaban udara, tekanan udara, temperatur udara, intensitas sinar matahari, arah dan kecepatan angin. Tingginya intensitas sinar matahari yang dikombinasikan dengan tingginya tingkat emisi prekursor ozon akan memicu reaksi fotokimia

pembentukan ozon troposfer. Pembentukan ozon akan meningkat pada kondisi kelembaban udara rendah, kecepatan angin rendah, dan temperatur atmosfer yang tinggi (Ambarsari, 2015).

1.2. Analisis Time Series

Dasar pemikiran *time series* adalah pengamatan sekarang (y_t) tergantung pada satu atau beberapa pengamatan sebelumnya (y_{t-k}). Model *time series* dibuat karena secara statistik ada korelasi antar deret pengamatan sehingga dapat digunakan untuk meramalkan pada beberapa periode ke depan (Rufiyanti, 2015). Beberapa konsep yang berkaitan dengan analisis *time series* adalah fungsi autokorelasi (ACF) dan fungsi autokorelasi parsial (PACF).

Fungsi autokorelasi atau *autocorrelation function* (ACF) adalah statistik atau penduga di dalam analisis *time series*. Autokorelasi adalah hubungan *time series* dengan *time series* itu sendiri dengan selisih waktu (*lag*) 0, 1, 2 periode atau lebih (Munawaroh, 2010). Fungsi autokorelasi menunjukkan besarnya korelasi atau hubungan linier antara pengamatan pada waktu t saat sekarang dengan pengamatan pada waktu-waktu sebelumnya ($t-1, t-2, \dots, t-k$) (Rufiyanti, 2015).

Sedangkan, fungsi autokorelasi parsial atau *partial autocorrelation* (PACF) adalah ukuran korelasi antara suatu data *time series* pada saat ini, t , dengan nilai-nilai sebelumnya dengan mengabaikan nilai setelahnya adalah konstan (Makridaris dkk., 1999). Dengan kata lain, PACF digunakan untuk mengukur korelasi antara y_t dan y_{t+k} dengan mengabaikan nilai-nilai setelah y_{t+k} .

Selain menggunakan uji statistik, penentuan *time lag* yang mempengaruhi data *time series* pada waktu t , atau yang disebut *lag* signifikan, dapat diidentifikasi menggunakan plot autokorelasi. Suatu *lag* dikatakan signifikan jika autokorelasi melewati selang kepercayaan r_k (fungsi autokorelasi sampel) dengan pusat 0. Dalam sebuah plot autokorelasi, selang ini direpresentasikan berupa garis horizontal yang sejajar dengan sumbu- x .

1.3. Jaringan Syaraf Tiruan

Pada prinsipnya jaringan syaraf tiruan merupakan suatu program komputer yang dibuat berdasarkan cara kerja jaringan biologi otak manusia. Dilihat dari segi fungsi, jaringan syaraf tiruan diciptakan untuk merancang suatu komputer yang dapat difungsikan untuk melakukan proses pembelajaran dari suatu contoh kejadian. Berdasarkan struktur rancangan, jaringan syaraf tiruan merupakan suatu rancangan alat penghitung yang ditujukan untuk dapat melakukan serupa dengan

apa yang dikerjakan oleh jaringan biologi otak manusia (Sari, 2016).

Jaringan syaraf tiruan ditentukan oleh beberapa hal yaitu pola hubungan antar neuron (arsitektur jaringan), fungsi aktivasi dan metode untuk menentukan bobot penghubung (fungsi pembelajaran). Arsitektur gunanya untuk menjelaskan arah perjalanan sinyal atau data di dalam jaringan. Arsitektur jaringan syaraf tiruan yang sering digunakan antara lain adalah jaringan lapisan tunggal (*single layer network*), jaringan lapisan banyak (*multi layer network*), dan jaringan lapisan kompetitif (*recurrent network*) (Siang, 2009).

Fungsi aktivasi adalah fungsi yang menentukan suatu neuron teraktivasi atau tidak melalui proses perbandingan dengan suatu nilai ambang tertentu. Fungsi aktivasi sangat berpengaruh terhadap nilai keluaran dari jaringan itu sendiri. Umumnya neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama sehingga pada setiap lapisan yang sama neuron-neuron memiliki fungsi aktivasi yang sama (Wuryandari, 2012).

Pemilihan fungsi aktivasi disesuaikan dengan permasalahan yang akan diselesaikan. Ada beberapa fungsi aktivasi yang biasa digunakan dalam membangun jaringan syaraf tiruan, antara lain adalah fungsi undak biner (*hard limit*), fungsi undak biner dengan *threshold*, fungsi bipolar (*symmetric hard limit*), fungsi bipolar dengan *threshold*, fungsi sigmoid biner, *fungsi sigmoid bipolar*, fungsi identitas dan fungsi *saturating linear* (Siang, 2009).

Fungsi pembelajaran menjelaskan bagaimana bobot koneksi harus diubah agar pasangan *input-output* yang diinginkan dapat tercapai. Dalam setiap perubahan nilai bobot koneksi dapat dilakukan dengan berbagai cara, tergantung pada jenis metode pelatihan yang digunakan. Dengan mengatur besarnya nilai bobot ini diharapkan bahwa kinerja jaringan dalam mempelajari berbagai macam pola yang dinyatakan oleh setiap pasangan *input-output* akan meningkat (Sari, 2016).

Dalam proses pembelajaran, nilai bobot suatu neuron akan bertambah, jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, sebaliknya nilai bobot akan berkurang, jika informasi dari neuron tersebut tidak tersampaikan. Pada saat pembelajaran dilakukan terhadap himpunan *input* yang berbeda, nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Apabila nilai seimbang ini telah tercapai, maka hal ini mengindikasikan bahwa tiap-tiap *input* telah berhubungan dengan *output* yang diharapkan (Bahri, 2017).

Ada beberapa cara berlangsungnya pembelajaran atau pelatihan jaringan syaraf tiruan, yaitu pembelajaran terawasi (*supervised learning*) dan

pembelajaran tak terawasi (*unsupervised learning*) (Fausett, 1994). Pada metode pembelajaran terawasi, setiap pola yang digunakan telah diketahui *outputnya* dan terdapat sejumlah data yang dipakai untuk melatih jaringan hingga mencapai nilai bobot yang diinginkan. Algoritma jaringan syaraf tiruan yang menggunakan metode pembelajaran terawasi adalah *Hebbian* dan *Backpropagation*. Sedangkan, pada pembelajaran tak terawasi tidak diperlukan target *output*. Tujuan pembelajaran ini adalah untuk mengelompokkan neuron-neuron yang hampir sama ke dalam suatu area tertentu. Algoritma jaringan syaraf tiruan yang menggunakan metode pembelajaran tak terawasi adalah *Competitive*, *Kohonen* dan *LVQ (Learning Vector Quantization)*.

1.4. Algoritma Backpropagation

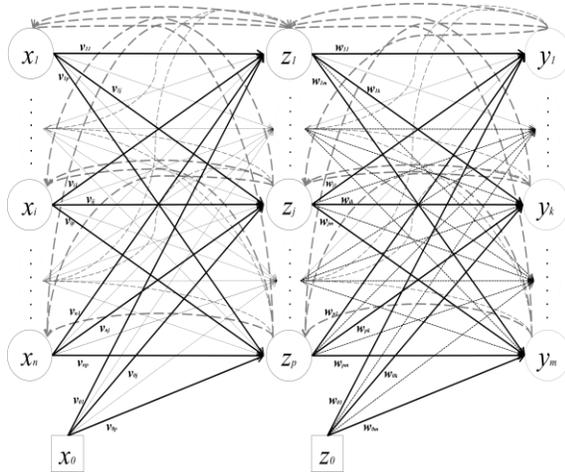
Backpropagation adalah salah satu algoritma pembelajaran terawasi pada jaringan syaraf tiruan yang paling banyak digunakan. *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola berbeda namun masih serupa dengan pola yang dipakai selama pelatihan (Siang, 2009).

Di dalam jaringan ini, setiap neuron yang berada di lapisan masukan terhubung dengan setiap neuron yang ada di lapisan tersembunyi. Setiap neuron yang ada di lapisan tersembunyi terhubung dengan setiap neuron yang ada di lapisan keluaran. Ketika jaringan ini diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju neuron pada lapisan tersembunyi untuk selanjutnya diteruskan ke neuron pada lapisan keluaran. Kemudian neuron pada lapisan keluaran ini akan memberikan respon sebagai *output* jaringan syaraf tiruan. Saat hasil *output* tidak sesuai dengan yang diharapkan, maka *output* akan disebarkan mundur (*backward*) pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan. Bobot-bobot dikoreksi supaya *errornya* dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati data yang aktual (Sari, 2016).

Gambar 2.15 adalah arsitektur jaringan yang menggunakan algoritma *backpropagation* dengan lapisan masukan x , lapisan tersembunyi z , dan lapisan keluaran y . Neuron bias ditunjukkan oleh neuron x_0 dan z_0 . Masukkan bias ini memiliki nilai yang tetap yaitu 1. Bobot v_{ij} adalah bobot garis yang menghubungkan lapisan masukan x_i dan lapisan tersembunyi z_j . Bobot v_{0j} adalah bobot garis yang menghubungkan bias di lapisan masukan ke lapisan tersembunyi. Bobot w_{jk} adalah bobot garis yang menghubungkan lapisan tersembunyi z_j dan lapisan keluaran y_k . Bobot w_{0k} adalah bobot garis yang

menghubungkan bias di lapisan tersembunyi ke lapisan keluaran.

Garis panah berwarna hitam (dan garis panah putus-putus kecil berwarna hitam) pada Gambar 2.15 menunjukkan arah perjalanan nilai masukan dari neuron di lapisan masukan ke lapisan tersembunyi sampai ke neuron pada lapisan keluaran. Sedangkan garis panah putus-putus besar berwarna abu menunjukkan arah perjalanan nilai keluaran dari neuron di lapisan keluaran pada fase *backpropagation*.



Gambar. 1 - Arsitektur Jaringan Backpropagation

Jaringan syaraf tiruan mempunyai dua istilah dalam pengolahan informasi data, yaitu pelatihan jaringan dan pengujian jaringan. Pelatihan jaringan berfungsi untuk mencapai keseimbangan antara kemampuan jaringan dalam memorisasi dan generalisasi, sedangkan pengujian jaringan dilakukan untuk melihat kecocokan antara target dan keluaran jaringan (Siang, 2009). Pelatihan jaringan menggunakan *backpropagation* meliputi tiga fase yaitu fase maju (*feedforward*), fase mundur (*backpropagation*) dan fase perubahan bobot.

Algoritma pelatihan *backpropagation* untuk jaringan dengan n lapisan tersembunyi dan fungsi aktivasi f adalah (Fausset, 1994):

Langkah 0: Inisialisasi semua bobot dengan bilangan acak kecil.

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

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

FASE I: FASE MAJU (*Feedforward*)

Langkah 3: Tiap lapisan masukan $x_i (i = 1, 2, \dots, n)$ menerima sinyal masukan x_i dan meneruskannya ke lapisan tersembunyi.

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

$$z_{net_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (1)$$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran.

$$z_j = f(z_{net_j}) \quad (2)$$

Kirimkan sinyal tersebut ke semua neuron lapisan di atasnya (neuron-neuron *output*).

Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

Langkah 5: Hitung semua *output* di lapisan keluaran $y_k (k = 1, 2, \dots, m)$.

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (3)$$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran.

$$y_k = f(y_{net_k}) \quad (4)$$

FASE II: FASE MUNDUR (*Backpropagation*)

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

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

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

Hitung suku perubahan bobot w_{jk} (Δw_{jk}) dan perubahan bias w_{0k} (Δw_{0k}).

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (6)$$

$$\Delta w_{0k} = \alpha \delta_k \quad (7)$$

Dan kirimkan δ_k ke lapisan di bawahnya.

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

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (8)$$

Faktor δ lapisan tersembunyi:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) \quad (9)$$

Hitung perubahan bobot v_{ij} (Δv_{ij})

dan perubahan bias v_{0j} (Δv_{0j}).

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

$$\Delta v_{0j} = \alpha \delta_j \tag{11}$$

FASE III: PERUBAHAN BOBOT

Langkah 8: Hitung semua perubahan bobot.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \tag{12}$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \tag{13}$$

Langkah 9: Jika kondisi penghentian terjadi, algoritma pelatihan selesai.

Jika kondisi penghentian belum terjadi, algoritma pelatihan diulangi dari langkah 2.

Setelah tahap pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola atau tahap pengujian. Namun, dalam hal ini, hanya fase maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Secara sistematis model jaringan syaraf tiruan algoritma *backpropagation* dapat dituliskan sebagai berikut (Kusumadewi F., 2014):

$$y_k = \sum_{j=1}^p w_{jk} \cdot f[v_{0j} + \sum_{i=1}^n x_i v_{ij}] + w_{0k} \tag{14}$$

dengan

- y_k neuron keluaran
- x_i neuron masukan
- v_{ij} nilai bobot dari lapisan masukan ke lapisan tersembunyi
- v_{0j} nilai bias dari lapisan masukan ke lapisan tersembunyi
- w_{jk} nilai bobot dari lapisan tersembunyi ke lapisan keluaran
- w_{0k} nilai bias dari lapisan tersembunyi ke lapisan keluaran
- p jumlah neuron pada lapisan tersembunyi
- $f(x)$ fungsi aktivasi

Fungsi aktivasi yang dipakai dalam *backpropagation* harus memenuhi beberapa syarat, yaitu kontinu, terdiferensial, dan merupakan fungsi yang tidak turun. Fungsi aktivasi yang memenuhi ketiga syarat tersebut di antaranya adalah fungsi sigmoid biner (*logsig*), fungsi sigmoid bipolar (*tansig*), dan fungsi identitas (*purelin*) (Gunaryati & Suhendra, 2015).

Terdapat berbagai macam fungsi pembelajaran dalam algoritma jaringan syaraf tiruan *backpropagation*, yang masing-masing fungsi memiliki karakteristik dan cara kerja yang berbeda. Menurut Kusumadewi, S. (2004), fungsi

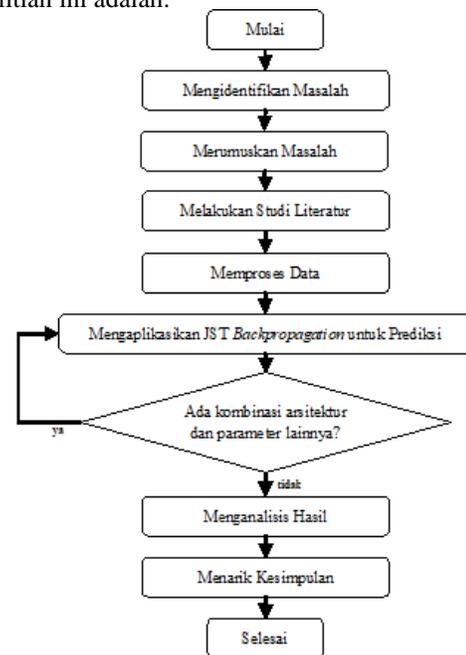
pembelajaran yang dasar ada dua macam, yaitu *incremental mode* (*adapt*) dan *batch mode*. Pada *incremental mode*, gradien dihitung dan bobot diubah setiap kali *input* dijalankan pada jaringan, sedangkan pada *batch mode*, semua *input* dijalankan pada jaringan dulu sebelum bobot diubah. Fungsi yang termasuk ke dalam *incremental mode* adalah *learngd* dan *learnghm*, sedangkan fungsi yang termasuk ke dalam *batch mode* adalah fungsi *traingd*, *traingdm*, *traingdx*, *trainlm* dan masih banyak lainnya (Kusumadewi F., 2014).

Salah satu penerapan algoritma JST *backpropagation* adalah dalam bidang prediksi. Prediksi adalah memperkirakan suatu kejadian atau peristiwa pada waktu yang akan datang berdasarkan pola data lampau yang dianalisis secara ilmiah. Model prediksi berkaitan dengan pembuatan sebuah model yang dapat melakukan pemetaan dari setiap himpunan variabel ke setiap targetnya, kemudian menggunakan model tersebut untuk memberikan nilai target pada himpunan baru yang didapat (Sari, 2016).

Langkah-langkah membangun jaringan syaraf tiruan dengan algoritma *backpropagation* untuk prediksi adalah menentukan *input* jaringan, membagi data untuk pelatihan jaringan dan pengujian jaringan, transformasi data, merancang arsitektur jaringan, memilih parameter-parameter jaringan, memilih arsitektur jaringan yang optimum, dan denormalisasi (transformasi talik) data.

2. Metode Penelitian

Langkah-langkah yang dilakukan dalam penelitian ini adalah:



Gambar. 2 - Diagram Alur Prosedur Penelitian

- 1) Mengidentifikasi Masalah
- 2) Merumuskan Masalah
- 3) Melakukan Studi Literatur

4) Memproses Data

Dalam memproses data ada beberapa hal yang dilakukan, yaitu:

a. Mengumpulkan Data

Data yang digunakan pada penelitian ini adalah data harian parameter meteorologis dan data kadar polutan di Kota Mataram periode 6 Juli 2018 sampai dengan 31 Mei 2019 yang diperoleh dari Dinas Lingkungan Hidup dan Kebersihan Provinsi NTB. Data-data tersebut terdiri dari data temperatur udara ($^{\circ}\text{C}$), arah angin ($^{\circ}$), kecepatan angin (m/s), kelembaban udara (%), intensitas sinar matahari (W/m^2), kadar polutan NO_2 ($\mu\text{g}/\text{m}^3$), kadar polutan SO_2 ($\mu\text{g}/\text{m}^3$) dan kadar polutan O_3 ($\mu\text{g}/\text{m}^3$).

b. Mendefinisikan Data *Input* dan Data Target

Data *input* adalah nilai masukan yang akan diproses menjadi nilai *output*, sedangkan data target adalah data yang dijadikan *output* atau keluaran pada jaringan. Data *input* pada penelitian ini adalah faktor meteorologis dan kadar polutan primer yang mempengaruhi kadar polutan ozon di udara ambien, sedangkan data target adalah kadar polutan ozon.

Faktor meteorologis yang dipilih sebagai data *input* adalah temperatur udara, arah angin, kecepatan angin, kelembaban udara, dan intensitas sinar matahari. Sedangkan, polutan primer atau prekursor ozon yang dipilih sebagai *input* tambahan adalah kadar polutan NO_2 dan kadar polutan SO_2 . Kedua polutan ini merupakan prekursor utama dalam pembentukan ozon.

c. Membagi Data Menjadi 2 Bagian

Data yang ada dibagi menjadi 2 bagian yaitu data untuk pelatihan dan data untuk pengujian. Data pelatihan digunakan untuk membangun jaringan, sedangkan data pengujian digunakan untuk mengukur akurasi prediksi. Pada penelitian ini akan digunakan 2 macam pembagian, dengan tujuan untuk melihat pengaruh dari pembagian data yang dilakukan. Pembagian pertama adalah 80% untuk proses pelatihan dan 20% untuk proses pengujian dan pembagian yang kedua adalah 90% untuk proses pelatihan dan 10% untuk proses pengujian.

d. Melakukan Transformasi Data

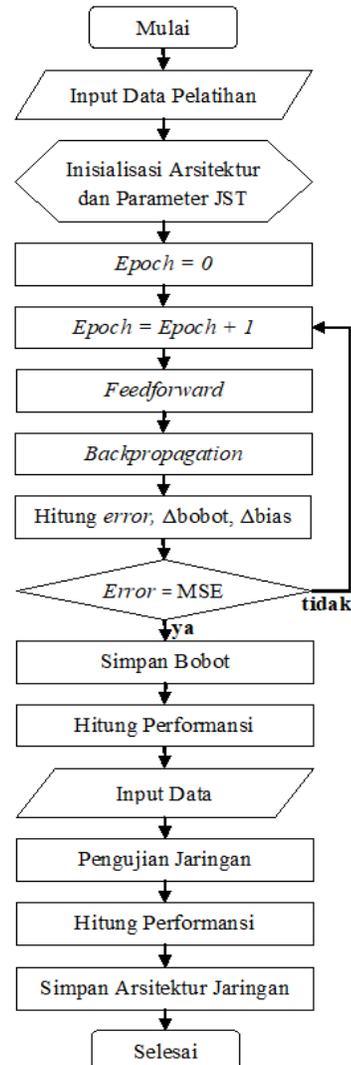
Transformasi data yang digunakan pada penelitian ini adalah transformasi ke interval

[0,1, 0,9] dikarenakan fungsi aktivasi yang digunakan merupakan fungsi sigmoid. Persamaan transformasi tersebut adalah :

$$x' = \frac{(x-x_{min}) \times 0,8}{x_{max}-x_{min}} + 0,1 \quad (15)$$

5) Mengaplikasikan JST *Backpropagation* untuk Prediksi

Prosedur pengaplikasian Jaringan Syaraf Tiruan *Backpropagation* untuk menentukan model prediksi ditunjukkan pada Gambar 3.



Gambar. 3 - Diagram Alur Proses Membentuk Model Prediksi Menggunakan JST

a. Merancang Arsitektur Jaringan

Langkah pertama untuk melakukan prediksi adalah merancang arsitektur jaringan yang akan digunakan, yaitu menentukan jumlah lapisan

tersembunyi dan neuron jaringan yang akan digunakan. Setiap lapisan memiliki jumlah neuron yang berbeda-beda. Pada penelitian ini jumlah neuron pada lapisan masukan dan lapisan keluaran sudah ditetapkan yaitu sebanyak 8 neuron masukan dan 1 neuron keluaran, sedangkan jumlah neuron tersembunyi dan jumlah lapisan tersembunyi ditentukan secara *trial* dan *error*.

Selain menentukan jumlah neuron dan lapisan, ditentukan juga fungsi aktivasi dan algoritma pembelajaran yang akan digunakan. Fungsi aktivasi yang digunakan pada penelitian ini ada 3 macam yaitu fungsi sigmoid biner (*logsig*), fungsi sigmoid bipolar (*tansig*), dan fungsi identitas (*purelin*). Pada penelitian ini, fungsi aktivasi ke lapisan tersembunyi akan divariasikan menggunakan fungsi aktivasi *logsig* atau fungsi *tansig*, sedangkan fungsi aktivasi ke lapisan keluaran ditetapkan menggunakan fungsi *purelin*. Selanjutnya, fungsi pembelajaran yang digunakan pada penelitian ini ada dua macam yaitu fungsi *trainlm* dan *traingdx*, dengan tujuan untuk melihat pengaruh dari kedua fungsi pembelajaran ini terhadap performansi jaringan syaraf tiruan.

b. Menentukan Parameter-Parameter Jaringan

Selanjutnya ditentukan parameter pelatihan jaringan yang diperlukan berdasarkan fungsi pembelajaran yang digunakan. Parameter jaringan yang diatur jika menggunakan fungsi pembelajaran *traingdx* adalah maksimum iterasi (*epoch*), kinerja tujuan (*goal performance*), laju pembelajaran (*learning rate*) dan momentum (*mc*). Selanjutnya, parameter jaringan yang diatur jika menggunakan fungsi pembelajaran *trainlm* adalah maksimum iterasi (*epoch*), kinerja tujuan (*goal performance*), parameter Levenberg-Marquardt (*mu*) dan nilai maksimum *mu* (*mu_max*).

Nilai-nilai parameter ini ditentukan secara *trial* dan *error* dengan membatasi menggunakan hanya beberapa nilai saja. Nilai *max_epoch* yang digunakan adalah sebesar 1000 dan 3000, target *error* sebesar 1×10^{-5} dan 0, *learning rate* sebesar 0,001 dan 0,1, momentum sebesar 0,5 dan 0,9, *mu* sebesar 0,001 dan 0,1, serta nilai *mu_max* sebesar 1×10^5 dan 1×10^{10} . Penentuan nilai-nilai ini didasarkan pada nilai-nilai parameter yang sering digunakan pada banyak penelitian yang sudah ada sebelumnya. Sedangkan nilai parameter-parameter lainnya yang tidak disebutkan dibiarkan tetap sesuai dengan nilai *default* pada Matlab.

c. Melakukan Pelatihan Jaringan

Selanjutnya dilakukan pelatihan jaringan menggunakan data pelatihan. Tahap pelatihan ini merupakan proses pengenalan pola-pola data yang telah ditransformasi dengan tujuan untuk

mendapatkan nilai bobot-bobot yang dapat memetakan antara data *input* dengan data target yang diinginkan. Bobot tersebut diubah tiap satu putaran dan dilakukan berulang kali sampai mencapai batas pelatihan. Hasil dari tahap pelatihan yaitu berupa pembaruan bobot jaringan yang nantinya akan digunakan dalam tahap pengujian jaringan.

d. Melakukan Pengujian Jaringan

Proses selanjutnya adalah melakukan pengujian jaringan. Proses pengujian dilakukan dengan menggunakan data pengujian, dimana data tersebut berbeda dengan data pelatihan. Pengujian dilakukan untuk melihat kinerja jaringan dalam generalisasi atau mengenali pola terhadap data baru yang diberikan. Proses pelatihan dan pengujian jaringan dilakukan terhadap semua kombinasi nilai parameter-parameter yang ada dan mencobanya pada tiap arsitektur jaringan yang sudah dirancang.

6) Menganalisis Hasil

Setelah melakukan pelatihan dan pengujian pada semua kombinasi arsitektur dan parameter jaringan yang ditentukan, dibandingkan performansi tiap jaringan. Kinerja jaringan dalam *backpropagation* biasanya didasarkan pada nilai *Mean Square Error* (*MSE*). Namun pada penelitian ini kinerja jaringan atau performansi JST didasarkan pada nilai *Root Mean Square Error* (*RMSE*), *Mean Absolute Percentage Error* (*MAPE*), koefisien korelasi (*R*) dan koefisien determinasi (R^2), persamaannya adalah sebagai berikut:

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2 = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2 \quad (16)$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2} \quad (17)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{(t_i - y_i)}{t_i} \right| \times 100\% \quad (18)$$

$$R = \frac{\sum_{i=1}^n (t_i - \mu_t)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^n (t_i - \mu_t)^2} \sqrt{\sum_{i=1}^n (y_i - \mu_y)^2}} \quad (19)$$

$$R^2 = \frac{\sum_{i=1}^n (y_i - \mu_y)^2 - \sum_{i=1}^n (t_i - y_i)^2}{\sum_{i=1}^n (y_i - \mu_y)^2} \quad (20)$$

dengan

t_i	target / data aktual
y_i	hasil keluaran / hasil prediksi
n	jumlah data
μ_t	rata-rata target
μ_y	rata-rata <i>output</i>

Jaringan dengan nilai RMSE atau MAPE terendah pada tahap pengujian dan mempunyai nilai yang tidak beda jauh dengan pelatihan, serta memiliki nilai koefisien korelasi dan koefisien determinasi terbesar (mendekati nilai 1), dipilih sebagai jaringan yang optimum yang akan digunakan untuk prediksi. Berdasarkan jaringan yang optimum tersebut dibentuk model prediksi kadar polutan ozon menggunakan persamaan (14).

7) Menarik Kesimpulan

Langkah ini merupakan langkah terakhir dari penelitian. Menarik kesimpulan disesuaikan dengan rumusan masalah yang disusun dan didasarkan pada hasil analisis dan pembahasan dari penelitian yang sudah dilakukan.

3. Hasil dan Pembahasan

3.1. Data Penelitian

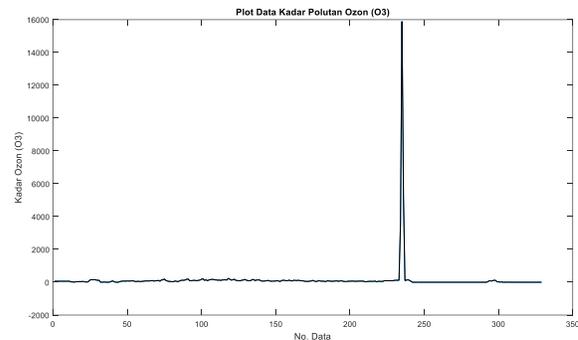
Data yang digunakan untuk melakukan prediksi pada penelitian ini adalah data harian pemantauan parameter meteorologis dan kadar polutan yang ada di kota Mataram pada periode 6 Juli 2018 sampai dengan 31 Mei 2019. Data tersebut diperoleh dari laboratorium lingkungan Dinas Lingkungan Hidup dan Kehutanan Provinsi NTB. Data-data tersebut terdiri dari temperatur udara ($^{\circ}\text{C}$), arah angin ($^{\circ}$), kecepatan angin (m/s), kelembaban udara (%), intensitas sinar matahari (W/m^2), kadar polutan NO_2 ($\mu\text{g}/\text{m}^3$), kadar polutan SO_2 ($\mu\text{g}/\text{m}^3$) dan kadar polutan O_3 ($\mu\text{g}/\text{m}^3$).

3.2. Penentuan Input Jaringan

Data *input* pada penelitian ini adalah faktor meteorologis dan kadar polutan primer yang mempengaruhi kadar polutan ozon di udara ambien. Faktor meteorologis yang dipilih sebagai data *input* pada penelitian ini adalah temperatur udara, arah angin, kecepatan angin, kelembaban udara, dan intensitas sinar matahari. Sedangkan, polutan primer atau prekursor ozon yang dipilih sebagai *input* tambahan adalah kadar polutan NO_2 dan kadar polutan SO_2 . Kedua polutan ini merupakan prekursor utama dalam pembentukan ozon.

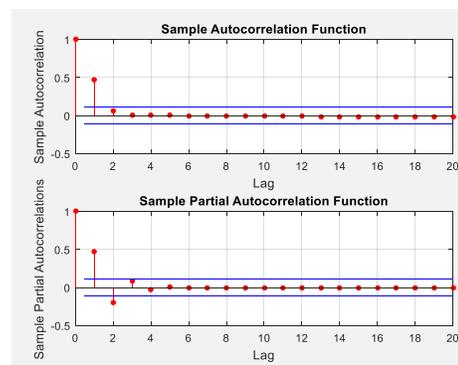
Namun selain faktor-faktor tersebut, berdasarkan konsep *time series*, kadar polutan ozon (O_3) tentunya juga dipengaruhi oleh kadar polutan ozon pada hari-hari sebelumnya. Penentuan data ozon berapa hari sebelumnya yang mempengaruhi kadar polutan ozon (variabel *lag*), dilakukan dengan membuat plot autokorelasi (ACF) dan plot autokorelasi parsial (PACF) dari data kadar polutan ozon.

Sebelumnya akan dibuat plot data dari kadar polutan ozon yang digunakan dan hasilnya ditunjukkan pada Gambar 4. Berdasarkan pada gambar ini, diketahui bahwa terdapat data pencilon di data ke 235, 236 dan 237, artinya pada hari ini nilai kadar ozon jauh lebih tinggi daripada hari-hari lainnya. Nilai ini bertepatan dengan tanggal 25, 26 dan 27 Februari 2019. Peningkatan kadar ozon pada waktu-waktu ini dapat dipengaruhi oleh berbagai hal misalnya peningkatan aktivitas manusia yang menghasilkan polutan ke udara.



Gambar. 4 - Plot Data Kadar Polutan Ozon (O_3)

Penentuan variabel *lag* kadar polutan ozon sebagai *input* jaringan dilakukan dengan melihat *lag-lag* yang signifikan pada plot autokorelasi (ACF) atau plot autokorelasi parsial (PACF). Suatu *lag* dikatakan signifikan jika autokorelasi melewati garis selang kepercayaan. Plot ACF dan PACF data kadar polutan ozon ditunjukkan pada Gambar 5.



Gambar. 5 - Plot ACF dan PACF Data Kadar Ozon (O_3)

Berdasarkan plot ACF, *lag* yang signifikan adalah *lag-1*, sedangkan pada plot PACF *lag* yang signifikan adalah *lag-1* dan *lag-2*. Hal ini berarti variabel *lag-1* dan variabel *lag-2* mempengaruhi kadar polutan ozon. Dengan kata lain, kadar ozon 1 hari (atau 2 hari) sebelumnya mempengaruhi kadar

polutan ozon saat ini. Oleh karena itu kedua variabel *lag* ini dapat digunakan sebagai *input* pada jaringan untuk memprediksi kadar polutan ozon. Namun berdasarkan Arifien (2012), *input* dengan interval $(t - 1)$ akan memiliki hasil prediksi yang lebih akurat dibandingkan dengan *input* yang memiliki interval $(t - 2)$. Oleh karena itu, pada penelitian ini hanya variabel *lag-1* (y_{t-1}) yang akan dijadikan variabel masukan.

Dengan demikian, pada penelitian ini akan digunakan 8 variabel sebagai *input* dan 1 variabel sebagai *output* (target). Pada Tabel 2 ditunjukkan variabel-variabel yang dijadikan *input* dan *output* (target) pada penelitian ini.

Tabel 2 – Variabel Input dan Variabel Target Jaringan.

Input (Masukan)	Output (Target)
Suhu udara (x_1) waktu ke $t - 1$	Konsentrasi O ₃ (y) waktu ke t
Arah angin (x_2) waktu ke $t - 1$	
Kecepatan angin (x_3) waktu ke $t - 1$	
Kelembaban udara (x_4) waktu ke $t - 1$	
Radiasi matahari (x_5) waktu ke $t - 1$	
Konsentrasi NO ₂ (x_6) waktu ke $t - 1$	
Konsentrasi SO ₂ (x_7) waktu ke $t - 1$	
Konsentrasi O ₃ <i>lag-1</i> (x_8) waktu ke $t - 1$	

3.3. Transformasi Data

Sebelum data digunakan pada jaringan, data harus ditransformasi terlebih dahulu agar memudahkan pencarian. Transformasi data yang digunakan pada penelitian ini adalah transformasi data ke interval $[0,1, 0,9]$. Interval ini dipilih disebabkan fungsi aktivasi yang digunakan pada penelitian ini merupakan fungsi sigmoid yang memiliki *range* $[0, 1]$. Namun dikarenakan fungsi sigmoid adalah fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1, maka dari itu dipilih untuk menggunakan interval yang lebih kecil. Berdasarkan nilai maksimum dan nilai minimum pada data penelitian, yaitu 15889,417 dan -0.0217 serta persamaan transformasi data pada persamaan (15), diperoleh persamaan transformasi data sebagai berikut:

$$\text{Data hasil transformasi} = \frac{(\text{Data penelitian} + 0.0217) \times 0.8}{15889.4387} + 0,1 \quad (21)$$

3.4. Pembagian Data

Data yang diperoleh ada sebanyak 330 data untuk masing-masing variabel data. Data tersebut kemudian dibagi ke dalam dua bagian, yaitu satu bagian untuk proses pelatihan dan satu bagian lagi untuk proses pengujian. Pada penelitian ini akan digunakan 2

macam pembagian, dengan tujuan untuk melihat pengaruh dari pembagian data yang dilakukan. Pertama, 80 % (263 data) untuk proses pelatihan dan 20 % (66 data) untuk proses pengujian, dan yang kedua 90 % (296 data) untuk proses pelatihan dan 10 % (33 data) untuk proses pengujian.

3.5. Pencarian Model JST untuk Prediksi

Secara teori, pada metode JST tidak ada aturan pasti untuk menentukan arsitektur dan parameter JST yang optimal untuk melakukan suatu prediksi. Namun, pada penelitian ini jumlah neuron pada lapisan masukan dan lapisan keluaran sudah ditetapkan yaitu sebanyak 8 neuron masukan dan 1 neuron keluaran, sedangkan jumlah neuron tersembunyi dan jumlah lapisan tersembunyi ditentukan secara *trial* dan *error*.

Dasar penentuan jumlah neuron tersembunyi yang digunakan pada penelitian ini adalah hasil dari penelitian yang dilakukan oleh Arifien, dkk. (2012) dan Aprianto, dkk. (2018). Hal ini dikarenakan kedua penelitian ini juga memprediksi kadar polutan menggunakan faktor meteorologi dan kadar polutan primer. Dalam kedua penelitian ini, arsitektur jaringan terbaik yang diperoleh adalah jaringan dengan jumlah neuron pada lapisan tersembunyi pertamanya sebanyak 20 neuron. Maka dari itu, jumlah neuron tersembunyi pada lapisan tersembunyi pertama yang digunakan pada penelitian ini sebanyak 20 neuron.

Selanjutnya, untuk melihat pengaruh jumlah neuron tersembunyi terhadap performansi jaringan maka akan digunakan juga jaringan dengan 15 neuron tersembunyi. Sedangkan untuk neuron tersembunyi yang lebih dari 20, setelah dilakukan beberapa kali percobaan untuk beberapa jaringan, dihasilkan performansi jaringan yang kurang baik. Oleh karena itu neuron dengan lebih dari 20 tidak digunakan sebagai arsitektur jaringan yang akan diujicobakan. Kemudian, untuk melihat pengaruh jumlah lapisan tersembunyi terhadap performansi jaringan maka akan digunakan jaringan dengan 2 lapisan tersembunyi dan 3 lapisan tersembunyi, dimana neuron pada masing-masing lapisan tersembunyi bervariasi antara 15 dan 20, dimana nilai ini merupakan penentuan nilai secara *random*.

Fungsi aktivasi yang digunakan dari lapisan masukan ke lapisan tersembunyi akan divariasikan menggunakan fungsi aktivasi sigmoid biner (*logsig*) atau fungsi sigmoid bipolar (*tansig*), sedangkan fungsi aktivasi ke lapisan keluaran ditetapkan menggunakan fungsi identitas (*purelin*). Hal ini juga didasarkan pada hasil penelitian yang dilakukan oleh Arifien, dkk. (2012) dan Aprianto, dkk. (2018). Fungsi pembelajaran yang digunakan pada penelitian ini ada dua macam yaitu fungsi *trainlm* dan *traingdx*.

Kedua fungsi pembelajaran ini merupakan dua fungsi pembelajaran yang terbaik. Penggunaan dua macam fungsi pembelajaran ini dilakukan dengan tujuan untuk melihat pengaruh dari kedua fungsi pembelajaran ini terhadap performansi jaringan syaraf tiruan. Dengan demikian, arsitektur jaringan yang digunakan pada penelitian ini ditampilkan pada Tabel 4.

Sedangkan nilai parameter pelatihan yang digunakan merupakan nilai parameter yang umumnya digunakan pada pelatihan jaringan syaraf tiruan. Nilai parameter-parameter tersebut ditampilkan pada Tabel 5. Parameter-parameter pada Tabel 5 dikombinasikan dan diujicobakan pada tiap arsitektur jaringan pada Tabel 4. Seluruh percobaan ini dilakukan dengan tujuan untuk mencari arsitektur JST yang paling optimal untuk prediksi. Setelah semua percobaan dilakukan, kemudian semua hasil terbaik pada masing-masing arsitektur JST dibandingkan. Hasil pelatihan dan pengujian jaringan yang dihasilkan oleh tiap arsitektur jaringan ada pada Tabel 6 (terlampir).

Tabel 4 - Arsitektur JST yang diujicobakan.

No.	Jaringan	Neuron	F. Aktivasi
1.	JST 1	[8 15 1]	[Tansig Purelin]
2.	JST 2	[8 15 1]	[Logsig Purelin]
3.	JST 3	[8 20 1]	[Tansig Purelin]
4.	JST 4	[8 20 1]	[Logsig Purelin]
5.	JST 5	[8 20 15 1]	[Logsig Tansig Purelin]
6.	JST 6	[8 20 15 1]	[Logsig Logsig Purelin]
7.	JST 7	[8 20 15 1]	[Tansig Logsig Purelin]
8.	JST 8	[8 20 20 1]	[Logsig Tansig Purelin]
9.	JST 9	[8 20 20 1]	[Logsig Logsig Purelin]
10.	JST 10	[8 20 20 1]	[Tansig Logsig Purelin]
11.	JST 11	[8 20 15 20 1]	[Logsig Tansig Logsig Purelin]
12.	JST 12	[8 20 15 20 1]	[Logsig Tansig Tansig Purelin]
13.	JST 13	[8 20 15 20 1]	[Tansig Logsig Logsig Purelin]
14.	JST 14	[8 20 15 20 1]	[Tansig Logsig Tansig Purelin]
15.	JST 15	[8 20 15 20 1]	[Tansig Tansig Logsig Purelin]

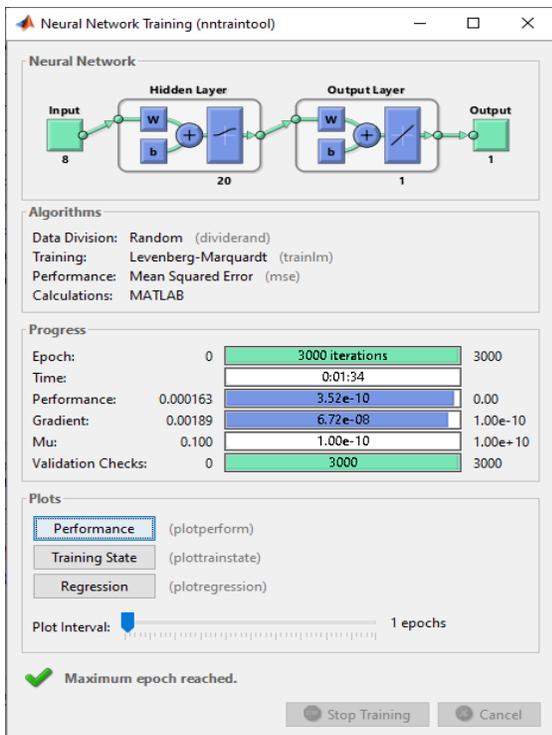
Tabel 5 - Parameter JST yang diujicobakan.

No.	Parameter	Nilai 1	Nilai 2
1.	Pembagian Data	8 : 2	9 : 1
2.	Algoritma Pembelajaran	<i>Trainidx</i>	<i>trainlm</i>
3.	<i>Max_Epoch = Max_Fail</i>	1000	3000
4.	<i>Goal_Error</i>	0	1×10^{-5}
5.	<i>Min_Grad</i>	1×10^{-7}	1×10^{-10}
6.	Learning Rate (<i>Lr</i>)	0,001	0,1
7.	Momentum (<i>Mc</i>)	0,5	0,9
8.	Nilai Mu (<i>Mu</i>)	0,001	0,1
9.	<i>Mu_max</i>	1×10^5	1×10^{10}

Berdasarkan Tabel 6 (terlampir), terlihat bahwa model jaringan terbaik untuk memprediksi kadar polutan ozon yang didapatkan pada penelitian ini adalah jaringan dengan 1 lapisan tersembunyi dengan jumlah neuron 8-20-1, yang berarti 8 neuron pada lapisan masukan, 20 neuron pada lapisan tersembunyi dan 1 neuron pada lapisan keluaran. Fungsi aktivasi yang digunakan pada lapisan masukan ke lapisan tersembunyi adalah fungsi sigmoid biner (*logsig*), sedangkan fungsi aktivasi pada lapisan tersembunyi ke lapisan keluaran adalah fungsi identitas (*purelin*). Pada jaringan ini fungsi pembelajaran yang digunakan adalah fungsi *trainlm* dengan pembagian data 80 % (263 data untuk masing-masing variabel) untuk pelatihan dan 20 % (66 data untuk masing-masing variabel) untuk pengujian.

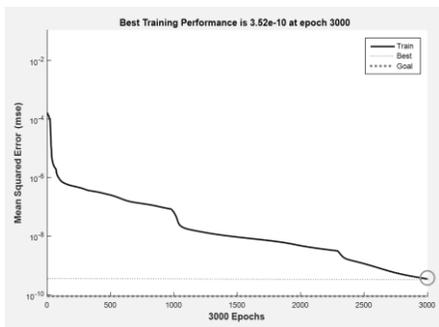
Jaringan dengan arsitektur tersebut dipilih menjadi jaringan terbaik karena menghasilkan nilai RMSE dan MAPE pengujian yang terkecil serta nilai *R* dan *R*² pengujian terbesar. Nilai RMSE dan MAPE yang kecil menunjukkan bahwa nilai antara hasil prediksi JST dan data aktual tidak berbeda jauh. Nilai *R* dan *R*² yang mendekati nilai 1 menunjukkan bahwa antara nilai target dan *output* jaringan memiliki korelasi yang baik.

Gambar 6 menunjukkan arsitektur jaringan dan konfigurasi yang digunakan dalam proses pelatihan. Dalam gambar tersebut ditunjukkan hasil *training* dengan *performance error* MSE sebesar $3,52 \times 10^{-10}$ berhenti pada iterasi (*epoch*) ke 3000 dalam waktu 1 menit 34 detik.



Gambar. 6 - Hasil Pelatihan Jaringan

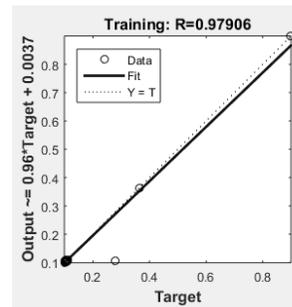
Selanjutnya, hasil grafik perubahan MSE yang terjadi pada setiap iterasi pelatihan ditunjukkan pada Gambar 7. Berdasarkan Gambar 7 terlihat bahwa nilai MSE pada proses *training* mengalami penurunan selama iterasinya. Hal ini menunjukkan bahwa proses pelatihan sudah berjalan bagus, karena nilai MSE-nya mendekati target *error* yang ditentukan yaitu sebesar 0.



Gambar. 7 - Hasil Performance Training

Selanjutnya, Gambar 8 merupakan tampilan grafik regresi proses *training* pada JST *backpropagation*. Gambar 8 menunjukkan grafik hubungan antara hasil *output* pada proses pelatihan dengan target yang ditentukan. Tingkat keakuratan pelatihan ditunjukkan dengan nilai R yang sebesar 0,97906. Nilai R yang hampir mendekati nilai 1

menunjukkan bahwa proses *training* berjalan dengan sangat baik.



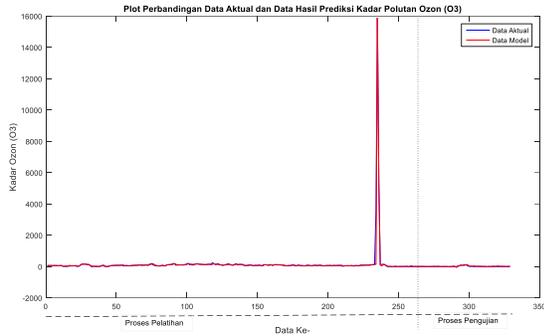
Gambar. 8 - Plot Regresi Hasil Training

Nilai *output* yang dihasilkan oleh jaringan masih dalam interval [0,1, 0,9] dikarenakan proses transformasi sebelumnya. Oleh karena itu dilakukan transformasi balik (denormalisasi) agar data sesuai dengan *range* data asli yang digunakan. Tabel 7 menunjukkan perbandingan data target (aktual) dan data *output* (model) jaringan pada tanggal 6-9 Juli 2018, sedangkan Gambar 9 menunjukkan plot perbandingan antara data target dan data *output* untuk data secara keseluruhan.

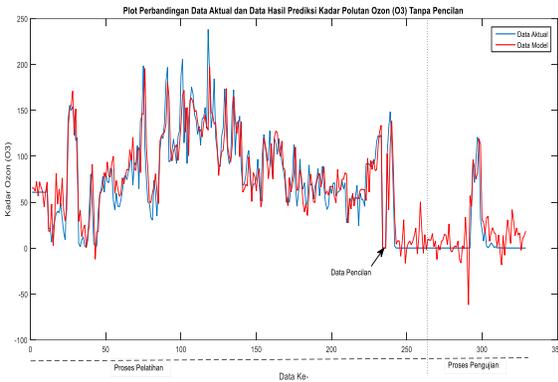
Tabel 7 - Contoh Perbandingan Data Aktual dan Data Model.

No.	Data Aktual	Data Model
1	60.69412	65.17617
2	60.71698	65.37983
3	60.65236	62.08243
4	60.73944	72.29874

Berdasarkan Tabel 7 terlihat bahwa data yang dihasilkan model cukup mendekati data aktual dengan *error* yang terjadi di antara 2 – 12. Secara umum, berdasarkan Gambar 9, pemodelan dengan arsitektur terbaik yang diperoleh pada penelitian ini memperlihatkan hasil yang sudah benar. Hal ini ditunjukkan oleh plot data model yang sangat mendekati plot data aktual. Namun, dikarenakan terdapat data pencilan pada data ke 235, 236 dan 237 maka perbedaan nilai data aktual dengan data model tidak dapat terlihat dengan jelas. Oleh karena itu dibuat Gambar 10 untuk menunjukkan plot perbandingan antara data aktual dan data model setelah data pencilan tersebut dikecualikan.



Gambar. 9 - Plot Data Aktual vs Data Hasil Prediksi Kadar Ozon (O₃)

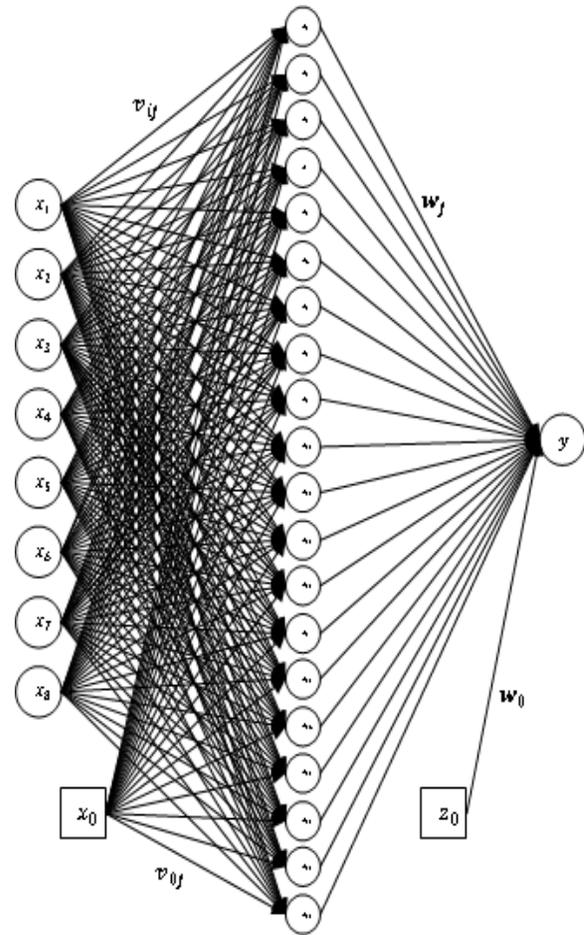


Gambar. 10 - Plot Data Aktual (merah) vs Data Hasil Prediksi Kadar Ozon (biru) Tanpa Data Pencilan

Berdasarkan Gambar 10 tersebut, pada proses pelatihan dapat terlihat dengan jelas bahwa grafik merah hampir mendekati grafik biru. Hal ini menunjukkan bahwa data hasil prediksi model yang ada telah mampu mengikuti pola data aktual dengan *error* yang cukup kecil. Pada proses pengujian, grafik hasil prediksi juga terlihat hampir mendekati grafik data aktual. Hal ini menunjukkan, pada proses pengujian model juga dapat menghasilkan hasil prediksi yang baik.

3.6. Pembahasan

Berdasarkan hasil yang didapatkan, model jaringan terbaik untuk melakukan prediksi kadar polutan ozon (O₃) adalah jaringan dengan arsitektur 8-20-1 dengan fungsi aktivasi sigmoid biner (*logsig*) pada lapisan masukan ke lapisan tersembunyi dan fungsi aktivasi *purelin* pada lapisan tersembunyi ke lapisan keluaran. Gambar 11 menunjukkan arsitektur jaringan terbaik yang sudah diperoleh.



Gambar. 11 - Arsitektur Jaringan Terbakik

Model jaringan syaraf tiruan *backpropagation* tersebut secara sistematis dapat ditulis sebagai berikut:

$$y = w_0 + \sum_{j=1}^{20} w_j \cdot \frac{1}{1 + e^{-(v_{0j} + \sum_{i=1}^8 x_i v_{ij})}} \quad (22)$$

dengan:

- y *output* di lapisan keluaran
- x_i nilai-nilai dari variabel masukan
- w_j nilai bobot dari lapisan tersembunyi ke lapisan keluaran
- w_0 nilai bobot bias di lapisan tersembunyi ke lapisan keluaran
- v_{ij} nilai bobot dari lapisan masukan ke lapisan tersembunyi
- v_{0j} nilai bobot bias di lapisan masukan ke lapisan tersembunyi

Berdasarkan model jaringan pada persamaan (17), variabel y menyatakan kadar polutan ozon pada hari ke- t , sedangkan variabel x_i adalah variabel-

variabel masukan pada jaringan. Variabel masukan tersebut adalah temperatur udara, arah angin, kecepatan angin, kelembaban udara, intensitas sinar matahari, kadar polutan NO_2 , kadar polutan SO_2 dan kadar polutan ozon (O_3) satu hari sebelumnya. Selanjutnya, berdasarkan data model dan data aktual kadar O_3 dihitung tingkat performansi jaringan pada proses pelatihan dan pengujian. Pada proses pelatihan, jaringan menghasilkan nilai RMSE sebesar 0,011, MAPE sebesar 1,043 %, dan koefisien korelasi sebesar 0,979. Nilai MAPE ini menunjukkan bahwa tingkat akurasi jaringan dalam proses pelatihan adalah sebesar 98,957 %. Nilai koefisien determinasi pelatihan, yaitu 0,9566, memberikan informasi bahwa 95,6 % hasil prediksi pelatihan jaringan telah mampu menjelaskan variabilitas pada data aktual.

Sementara pada proses pengujian, jaringan menghasilkan nilai RMSE sebesar 0,001, nilai MAPE sebesar 0,749 %, nilai koefisien korelasi sebesar 0,772, dan nilai koefisien determinasi sebesar 0,4972. Nilai MAPE ini menunjukkan bahwa tingkat akurasi jaringan dalam proses pengujian adalah sebesar 99,251 %, sedangkan nilai koefisien determinasi pengujian jaringan menunjukkan bahwa 49,72 % hasil prediksi pengujian jaringan telah mampu menjelaskan variabilitas pada data aktual.

Berdasarkan koefisien korelasi hasil pelatihan dan pengujian jaringan terlihat bahwa korelasi jaringan yang dihasilkan berkategori "Sangat Kuat". Hasil korelasi ini menunjukkan bahwa bobot-bobot jaringan telah berhasil merespon data pada pelatihan dan pengujian jaringan. Selanjutnya, berdasarkan koefisien determinasi yang diperoleh jaringan secara keseluruhan yaitu sebesar 0,9565 menunjukkan bahwa 95,65 % variabel meteorologis dan kadar polutan primer dapat dijadikan sebagai masukan jaringan syaraf tiruan untuk memprediksi kadar polutan ozon (O_3).

Pada penelitian ini terdapat beberapa faktor yang ingin dilihat bagaimana pengaruhnya terhadap performansi jaringan dalam melakukan prediksi, di antaranya adalah jumlah neuron dan lapisan tersembunyi, pembagian data, dan fungsi pembelajaran. Selain itu, faktor lainnya yang dapat disimpulkan mempengaruhi performansi jaringan adalah variabel prediktor atau masukannya.

Pada penelitian ini jaringan dengan 1 lapisan tersembunyi saja dapat menghasilkan performansi jaringan yang lebih baik dibandingkan dengan arsitektur lain yang memiliki jumlah neuron dan lapisan tersembunyi yang lebih banyak. Hal ini membuktikan bahwa pendapat Arifien, dkk. (2012) yang mengatakan semakin banyak jumlah lapisan tersembunyi maka hasil prediksi akan semakin baik tidak terbukti. Berdasarkan hal tersebut, pendapat yang ada untuk memperkirakan jumlah neuron dan

lapisan pada JST belum tentu menghasilkan jaringan yang optimal, dan oleh karena itu sebaiknya tetap dilakukan *trial* dan *error* yang lebih banyak sehingga ditemukan arsitektur jaringan yang dapat menghasilkan hasil prediksi yang lebih baik. Seperti halnya neuron tersembunyi, pada metode JST tidak ada yang bisa memastikan berapa nilai parameter-parameter jaringan yang seharusnya digunakan untuk proses pelatihan JST. Oleh karena itu, proses *trial* dan *error* harus terus dilakukan hingga jaringan mencapai nilai performansi yang optimum.

Pada penelitian ini digunakan dua macam pembagian data dalam melakukan pelatihan jaringan. Namun, berdasarkan penelitian yang sudah dilakukan, belum dapat diambil kesimpulan bagaimana masing-masing pembagian data yang digunakan (8: 2 dan 9: 1) mempengaruhi performansi jaringan syaraf tiruan. Namun secara umum pelatihan menggunakan pembagian data 9:1 membutuhkan waktu yang lebih lama dibandingkan menggunakan pembagian data 8:2. Hal ini dipengaruhi oleh jumlah data pelatihan yang lebih banyak pada pembagian 9:1, sehingga diperlukan waktu yang lebih banyak untuk melakukan pelatihan.

Pada penelitian ini juga digunakan dua macam fungsi pembelajaran yang berbeda yaitu *trainlm* dan *traingdx*. Berdasarkan hasil penelitian yang sudah diperoleh, dapat disimpulkan bahwa penggunaan fungsi pembelajaran *trainlm* menghasilkan performansi JST yang lebih baik dibandingkan fungsi pembelajaran *traingdx*. Namun secara umum, pelatihan menggunakan fungsi *trainlm*, membutuhkan waktu pelatihan yang lebih lama dibandingkan menggunakan fungsi *traingdx*. Hal ini sesuai dengan Fagustina (2014) yang menyatakan bahwa fungsi *trainlm* merupakan fungsi atau metode tercepat untuk *training feedforward neural network* berukuran besar (sampai ratusan bobot), namun membutuhkan kapasitas memori yang lebih besar dibandingkan dengan algoritma lainnya. Kapasitas memori dalam hal ini adalah waktu pelatihan jaringan.

Faktor berikutnya yang mempengaruhi performansi jaringan dalam melakukan prediksi adalah variabel yang dijadikan sebagai prediktor, baik dari segi jumlah maupun pemilihan variabel masukan itu sendiri (Arifien dkk., 2012). Jumlah prediktor yang lebih banyak akan menghasilkan kinerja jaringan yang lebih baik dikarenakan informasi dari nilai/keadaan sebelumnya akan semakin banyak dan pola perubahan dapat dengan mudah dipetakan oleh jaringan. Selain itu pemilihan variabel masukan yang tepat juga dapat mempengaruhi performansi jaringan dalam melakukan prediksi. Pemilihan variabel lain yang lebih mempengaruhi kadar polutan ozon

kemungkinan akan dapat menghasilkan prediksi yang lebih baik.

Arsitektur dan nilai parameter optimum yang didapatkan pada penelitian ini hanyalah sebatas pada lingkup percobaan yang sudah dilakukan. Oleh karena itu belum dapat dikatakan sebagai hasil yang paling baik mengingat masih banyak kemungkinan kombinasi arsitektur dan parameter jaringan berbeda yang dapat diujicobakan dan akan bisa jadi memberikan performansi jaringan yang lebih baik daripada yang dihasilkan pada penelitian ini.

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan disimpulkan bahwa model jaringan syaraf tiruan *backpropagation* terbaik yang diperoleh untuk memprediksi kadar polutan ozon (O_3) di Kota Mataram pada hari berikutnya adalah jaringan dengan 8 neuron pada lapisan *input*, 20 neuron pada lapisan tersembunyi dan 1 neuron pada lapisan keluaran dengan fungsi aktivasi sigmoid biner (*logsig*) pada lapisan masukan ke lapisan tersembunyi dan fungsi aktivasi *purelin* pada lapisan tersembunyi ke lapisan keluaran, serta menggunakan nilai bobot-bobot akhir yang dicantumkan pada lampiran. Performa model pelatihan berdasarkan indikator RMSE, MAPE, R dan R^2 berturut-turut sebesar 0,011, 1,043 %, 0,979 dan 0,9566. Sedangkan, performa model pengujian berdasarkan indikator RMSE, MAPE, R dan R^2 berturut-turut sebesar 0,001, 0,749 %, 0,0772, dan 0,4972.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Dr. Marwan, S.Si., M.Si. selaku Ketua Program Studi Matematika FMIPA Universitas Mataram, dan juga bersama Ni Wayan Switrayni, S.Pd., M.Si. atas masukan dan saran dalam penyempurnaan penelitian ini.

DAFTAR PUSTAKA

- Ambarsari, N., 2015, Efek Radikal *Hidroxy* (OH) dan *Nitric Oxide* (NO) dalam Reaksi Kimia Ozon di Atmosfer, *Berita Dirgantara* 16(2): 47-54.
- Aprianto, Y., Nurhasanah, dan Sanubary, I., 2018, Prediksi Kadar Partikulat (PM_{10}) untuk Pemantauan Kualitas Udara Menggunakan Jaringan Syaraf Tiruan Studi Kasus Kota Pontianak, *POSITRON* 8(1): 15-20.
- Arifien, N. F., Arifin, S., dan Widjiantoro, B. L., 2012, Prediksi Kadar Polutan Menggunakan Jaringan Syaraf Tiruan (JST) untuk Pemantauan Kualitas Udara di Kota Surabaya, Jurusan Teknik Fisika Fakultas Teknologi Industri Institut Teknologi Surabaya, Surabaya.
- Bahri, S., 2017, Desain dan Evaluasi Performa Model *Wavelet Neural Network* untuk Pemodelan *Time Series*, Disertasi, Program Studi S3 Ilmu Matematika Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada, Yogyakarta.
- Bahri, S., Syamsuddin, dan Hadijati, M., 2019, Model *Wavelet Neural Network* untuk Pemodelan Polusi Udara di Kota Mataram Lombok Nusa Tenggara Barat Indonesia, Dipresentasikan pada Seminar Nasional Matematika, Statistika, dan Aplikasinya, Terbitan I, 5 Oktober 2019, Samarinda, Indonesia.
- Direktorat Jenderal Pengendalian Pencemaran dan Kerusakan Lingkungan, 2018, Laporan Tahunan Direktorat Pengendalian Pencemaran Udara, Kementerian Lingkungan Hidup dan Kehutanan Republik Indonesia.
- Dinas Lingkungan Hidup dan Kehutanan Provinsi NTB, 2019, Data Parameter Meteorologis dan Konsentrasi Polutan di Kota Mataram, Mataram.
- Fagustina, A., Palgunasi Y. S., dan Wiharto, 2014, Pengaruh Fungsi Pembelajaran Terhadap Kinerja Pelatihan Jaringan Syaraf Tiruan *Backpropagation* Studi Kasus Indeks Harga Saham Gabungan di Bursa Efek Indonesia, Jurusan Informatika Universitas Sebelas Maret, Surakarta.
- Fausett, L., 1994, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice Hall, USA.
- Gunaryati, A., dan Suhendra, A., 2015, Perbandingan Antara Metode Statistika dan Metode *Neural Network* Pada Model Peramalan Indeks Harga Perdagangan Besar, *Jurnal Teknologi dan Rekayasa* 20(1): 23-35.
- Kusumadewi, Felasufah, 2014. Peramalan Harga Emas Menggunakan *Feedforward Neural Network* dengan Algoritma *Backpropagation*, Skripsi, Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta, Yogyakarta.
- Makridaris, S., Wheelwright, S. C. dan McGee, V. E., 1999, *Metode dan Aplikasi Peramalan*, Edisi Kedua Jilid Satu, Alih Bahasa Hari Suminto, Binarupa Aksara, Jakarta.
- Munawaroh, Siti, 2010, Analisis Model Arima *Box Jenkins* pada Data Fluktuasi Harga Emas, Skripsi, Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim, Malang.
- Mustafidah, H., Hakim, D. K., dan Sugiyanto, S., 2013, Tingkat Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan (Studi Kasus Prediksi

- Prestasi Belajar Mahasiswa), *JUITA* 2(3): 159-166.
- Octavianti, T. dan Utami, S. T. B., 2014, Faktor Risiko Kejadian Gangguan Pernapasan Akibat Ozon (O_3) Udara Ambien di Kecamatan Jagakarsa Tahun 2014, Fakultas Kesehatan Masyarakat Universitas Indonesia, Jakarta.
- Rufiyanti, D. E., 2015, Implementasi Jaringan Syaraf Tiruan *Backpropagation* dengan *Input* Model ARIMA untuk Peramalan Harga Saham, Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang, Semarang.
- Sari, Y. N., 2016, Jaringan Syaraf Tiruan *Backpropagation* untuk Memprediksi Luas Area Serangan Hama pada Tanaman Bawang, Skripsi, Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Semarang, Semarang.
- Serlina, Yega, 2020, Pengaruh Faktor Meteorologi Terhadap Konsentrasi NO_2 di Udara Ambien (Studi Kasus Bundaran Hotel Indonesia DKI Jakarta), *Serambi Engineering* V(3): 1228-1235.
- Siang, J. J., 2009, Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB, ANDI, Yogyakarta.
- Wuryandari, M. D., 2012, Perbandingan Metode Jaringan Syaraf Tiruan *Backpropagation* dan *Learning Vector Quantization* pada Pengenalan Wajah, *Jurnal Komputer dan Informatika (KOMPUTA)* 1(1).

Lampiran

Tabel 6 - Performansi Pelatihan dan Pengujian Jaringan

Jaringan	Bagian	Pembelajaran	Pelatihan				Pengujian			
			RMSE	MAPE	R	R ²	RMSE	MAPE	R	R ²
JST1	80 20	<i>trainlm</i>	0.004	0.978	0.998	0.995	0.002	1.028	0.621	0.381
JST1	90 10	<i>trainlm</i>	0.003	0.946	0.997	0.995	0.002	1.368	0.557	0.256
JST1	80 20	<i>traingdx</i>	0.044	4.004	0.594	-4.068	0.006	4.339	0.224	0.020
JST1	90 10	<i>traingdx</i>	0.042	4.273	0.589	-3.973	0.003	2.266	0.548	0.283
JST2	80 20	<i>trainlm</i>	0.023	1.378	0.917	0.841	0.005	2.051	0.225	0.045
JST2	90 10	<i>trainlm</i>	0.022	1.932	0.917	0.842	0.007	3.454	0.136	0.001
JST2	80 20	<i>traingdx</i>	0.048	4.851	0.401	-7.409	0.016	13.369	-0.079	-2.288
JST2	90 10	<i>traingdx</i>	0.046	4.673	0.374	-6.840	0.006	5.488	0.010	-5.268
JST3	80 20	<i>trainlm</i>	0.013	1.371	0.973	0.926	0.027	5.191	-0.336	-0.043
JST3	90 10	<i>trainlm</i>	0.011	2.039	0.974	0.947	0.004	2.940	0.289	-0.032
JST3	80 20	<i>traingdx</i>	0.043	4.630	0.644	-3.776	0.009	6.291	0.275	-0.146
JST3	90 10	<i>traingdx</i>	0.040	5.049	0.632	-3.602	0.004	3.748	0.388	-1.326
JST4	80 20	<i>trainlm</i>	0.011	1.043	0.979	0.9566	0.001	0.749	0.772	0.4972
JST4	90 10	<i>trainlm</i>	0.010	0.993	0.979	0.957	0.001	0.903	0.770	0.337
JST4	80 20	<i>traingdx</i>	0.047	3.017	0.460	-7.412	0.006	5.837	0.050	-1.399
JST4	90 10	<i>traingdx</i>	0.045	3.306	0.459	-7.312	0.007	6.067	-0.279	-7.385
JST5	80 20	<i>trainlm</i>	0.003	1.377	0.998	0.996	0.007	3.122	0.260	0.025
JST5	90 10	<i>trainlm</i>	0.003	1.407	0.998	0.995	0.009	4.601	0.113	-0.058
JST5	80 20	<i>traingdx</i>	0.048	5.381	0.401	-7.146	0.008	7.518	0.264	-0.976
JST5	90 10	<i>traingdx</i>	0.046	5.593	0.399	-6.960	0.008	7.017	0.473	-2.794
JST6	80 20	<i>trainlm</i>	0.004	1.727	0.998	0.995	0.006	3.957	0.154	-0.022
JST6	90 10	<i>trainlm</i>	0.004	1.858	0.997	0.994	0.008	4.838	0.195	-0.080
JST6	80 20	<i>traingdx</i>	0.048	4.070	0.400	-5.596	0.008	6.633	0.116	-0.139
JST6	90 10	<i>traingdx</i>	0.046	2.952	0.369	-8.363	0.005	4.799	-0.240	-9.449
JST7	80 20	<i>trainlm</i>	0.002	1.143	0.999	0.999	0.010	6.699	0.132	-0.020
JST7	90 10	<i>trainlm</i>	0.004	2.005	0.996	0.993	0.004	3.311	0.126	-1.562
JST7	80 20	<i>traingdx</i>	0.048	3.624	0.434	-6.946	0.005	3.674	0.153	-0.255
JST7	90 10	<i>traingdx</i>	0.045	3.604	0.434	-6.854	0.005	3.901	-0.061	-0.658
JST8	80 20	<i>trainlm</i>	0.010	3.121	0.983	0.965	0.074	40.064	0.066	-0.367
JST8	90 10	<i>trainlm</i>	0.011	4.666	0.974	0.947	0.009	6.779	-0.029	-0.257
JST8	80 20	<i>traingdx</i>	0.049	4.451	0.343	-7.944	0.008	6.824	0.019	-2.179
JST8	90 10	<i>traingdx</i>	0.047	4.771	0.342	-7.839	0.008	6.987	-0.060	-5.647
JST9	80 20	<i>trainlm</i>	0.004	1.856	0.997	0.994	0.010	4.121	0.199	0.034
JST9	90 10	<i>trainlm</i>	0.004	1.885	0.997	0.994	0.013	5.855	0.206	0.029
JST9	80 20	<i>traingdx</i>	0.048	3.627	0.409	-7.707	0.006	5.391	-0.014	-0.509
JST9	90 10	<i>traingdx</i>	0.046	3.830	0.407	-7.564	0.006	5.099	-0.228	-1.187
JST10	80 20	<i>trainlm</i>	0.006	1.252	0.998	0.982	0.111	57.411	0.031	-0.266
JST10	90 10	<i>trainlm</i>	0.010	2.037	0.981	0.963	0.008	5.239	-0.024	-0.477

JST10	80 20	<i>traingdx</i>	0.048	3.805	0.420	-6.721	0.009	6.997	-0.070	-0.824
JST10	90 10	<i>traingdx</i>	0.045	4.274	0.413	-6.542	0.007	5.979	-0.254	-2.816
JST11	80 20	<i>trainlm</i>	0.002	0.861	1.000	0.999	0.004	2.309	0.223	0.019
JST11	90 10	<i>trainlm</i>	0.002	0.862	0.999	0.999	0.005	3.701	0.111	-0.044
JST11	80 20	<i>traingdx</i>	0.050	4.666	0.339	-8.348	0.008	7.291	0.026	-3.084
JST11	90 10	<i>traingdx</i>	0.009	8.924	-0.208	-4.234	0.047	4.794	0.340	-8.286
JST12	80 20	<i>trainlm</i>	0.023	2.466	0.919	0.667	0.105	55.385	-0.023	-0.363
JST12	90 10	<i>trainlm</i>	0.009	1.554	0.985	0.970	0.004	2.546	0.249	0.042
JST12	80 20	<i>traingdx</i>	0.011	3.280	0.977	0.952	0.122	63.836	-0.045	-0.376
JST12	90 10	<i>traingdx</i>	0.012	4.324	0.968	0.935	0.006	5.293	0.334	-1.125
JST13	80 20	<i>trainlm</i>	0.006	2.138	0.993	0.986	0.009	5.698	-0.158	-0.093
JST13	90 10	<i>trainlm</i>	0.007	2.540	0.991	0.982	0.007	5.144	0.030	-0.198
JST13	80 20	<i>traingdx</i>	0.009	2.330	0.987	0.973	0.004	3.528	-0.074	-1.259
JST13	90 10	<i>traingdx</i>	0.008	2.479	0.988	0.976	0.004	3.387	-0.184	-3.279
JST14	80 20	<i>trainlm</i>	0.003	0.159	0.998	0.996	0.005	3.158	0.246	0.042
JST14	90 10	<i>trainlm</i>	0.003	1.021	0.998	0.996	0.006	4.318	0.044	-0.075
JST14	80 20	<i>traingdx</i>	0.011	2.751	0.979	0.955	0.007	5.491	-0.078	-0.782
JST14	90 10	<i>traingdx</i>	0.010	2.917	0.978	0.955	0.008	6.569	-0.301	-1.367
JST15	80 20	<i>trainlm</i>	0.002	1.131	0.999	0.998	0.007	3.210	0.102	0.000
JST15	90 10	<i>trainlm</i>	0.002	1.226	0.999	0.998	0.009	4.436	0.035	-0.014
JST15	80 20	<i>traingdx</i>	0.050	4.531	0.327	-8.639	0.010	8.785	0.015	-2.675
JST15	90 10	<i>traingdx</i>	0.047	4.962	0.323	-8.483	0.010	9.096	-0.258	-12.295