

Prediksi Jumlah Kasus COVID-19 Menggunakan Teknik Sliding Windows dengan Metode BPNN

Apolinaris Edius Radho¹, Putu Sugiartawan^{*2}, Gede Agus Santiago³

^{1,2}Teknik Informatika, STMIK STIKOM Indonesia, Bali

e-mail: ¹ardianradho97@stiki-indonesia.ac.id, ^{*2}putu.sugiartawan@stiki-indonesia.ac.id,
agus.santiago@stiki-indonesia.ac.id

Abstrak

COVID-19 adalah penyakit yang menyerang sistem pernapasan yang diakibatkan oleh virus corona. Penularan COVID-19 membuat dunia menjadi resah termasuk Indonesia. Peramalan jumlah kasus COVID-19 di Indonesia perlu dilakukan karena diharapkan dapat mencegah ataupun menekan angka penyebaran jumlah kasus COVID-19 di Indonesia. Salah satu cara yang dilakukan dalam penelitian ini adalah memprediksi jumlah kasus COVID-19 menggunakan Backpropagation Neural Network dan Sliding Windows dengan memanfaatkan data riwayat harian jumlah kasus COVID-19 di Indonesia. Data yang digunakan pada penelitian ini adalah data riwayat jumlah kasus mulai dari bulan Maret tahun 2020 sampai bulan April 2021. Data riwayat penambahan jumlah kasus ini ditampilkan dengan pendekatan konsep Sliding Windows berdasarkan ukuran jendela yang digunakan yaitu 2, 3, 4 dan 5. Windows size mencerminkan jumlah hari sebagai variabel lapisan input pada arsitektur Backpropagation untuk memprediksi jumlah kasus hari berikutnya. Hasil prediksi yang diperoleh dengan nilai Sum Squared Error terkecil pada pengujian jaringan adalah 0,69 sebagai nilai akurasi hasil prediksi yang tinggi. Arsitektur jaringan dengan SSE terkecil menggunakan 3 lapisan input, 6 neuron hidden layer dan 1 lapisan output dapat menjadi sebuah pengetahuan yang dapat membantu pemerintah dalam memprediksi jumlah kasus COVID-19 di Indonesia untuk masa yang akan datang.

Kata kunci—Prediksi, COVID-19, Jaringan syaraf tiruan, Backpropagation, Sliding window.

Abstract

COVID-19 is a disease that attacks the respiratory system caused by the coronavirus. The spread of COVID-19 has made the world restless, including Indonesia. Forecasting the number of COVID-19 cases in Indonesia needs to be done because it is expected to prevent or reduce the number of COVID-19 cases in Indonesia. One of the methods used in this research is to predict the number of COVID-19 cases using Backpropagation Neural Network and Sliding Windows by utilizing daily historical data on the number of COVID-19 cases in Indonesia. The data used in this study is historical data on the number of cases from March 2020 to April 2021. The historical data for the addition of the number of cases is displayed using the Sliding Windows concept approach based on the window sizes used, namely 2, 3, 4, and 5. Windows size reflects the number of days as an input layer variable on the Backpropagation architecture to predict the number of cases the next day. The prediction results obtained with the smallest Sum Squared Error value in network testing is 0.69 as a high predictive accuracy value. The network architecture with the smallest SSE using 3 input layers, 6 hidden layer neurons, and 1 output layer can be knowledge that can help the government in predicting the number of COVID-19 cases in Indonesia in the future.

Keywords—*Prediction, COVID-19, Artificial Neural Networks, Backpropagation, Sliding window.*

1. PENDAHULUAN

Perkembangan penularan COVID-19 membuat dunia menjadi resah termasuk Indonesia yang merupakan negara dengan penduduk terbanyak keempat di dunia [1]. Menurut World Health Organization hingga 28 Desember 2020, tercatat virus corona telah merenggut sekitar 1,58 juta nyawa. Sementara di Indonesia, kasus pertama infeksi virus corona terdeteksi pada Maret 2020. Sejak saat itu Indonesia telah kehilangan 21.452 nyawa hingga 28 Desember 2020. Mudah-mudahan penularan virus mengakibatkan peningkatan terhadap kasus baru terus terjadi hingga saat ini serta meningkatnya angka kematian dan permasalahan ekonomi menjadi ancaman paling serius yang di hadapi Indonesia.

Forecasting adalah suatu metode pendekatan dalam memprediksi kemungkinankemungkinan atas situasi pada masa yang kedepan dengan cara pengujian data yang terjadi dimasa lalu [2]. Peramalan (forecasting) digunakan untuk memprediksi sesuatu yang kemungkinan besar akan terjadi di masa mendatang sehingga tindakan yang tepat dapat dilakukan [1], [3]–[6]. Ini dapat dimanfaatkan untuk memprediksi jumlah kasus COVID-19 kedepannya. Ilmuwan di bidang matematika, aktuaria, dan ilmu data mencoba memodelkan dan memprediksi penyebaran atau kematian virus. Jaringan saraf tiruan dengan teknik Sliding Window merupakan alternatif yang digunakan untuk melakukan peramalan terhadap data berkala dari kasus, kematian, resiko dan penduduk. Menurut Zhang, G.P alasan menggunakan teknik tersebut karena hampir semua studi peramalan time series pada jaringan saraf tiruan menggunakan teknik Sliding Window. Dengan penggunaan teknik Sliding Windows akan didapatkan varian data baru (data dengan ukuran windows yang ditentukan) teknik ini melakukan pergeseran window atau jendela secara bertahap sesuai dengan ukuran window yang telah ditentukan untuk mendapatkan hasil akurasi yang paling optimal dari ukuran windows tersebut yang akan dipakai sebagai acuan untuk memprediksi jumlah kasus kedepannya.

Menurut Kusumadewi [7] Neural Network merupakan salah satu sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Beberapa penelitian yang pernah dilakukan mengenai Neural Network diantaranya tentang prediksi volume penggunaan air PDAM menggunakan metode Jaringan Syaraf Tiruan Backpropagation [8]. Neural Network digunakan untuk memprediksi harga ayam dengan medel Backpropagation Neural Network (Susanti, 2014). Neural Network digunakan untuk memprediksi curah hujan di kota medan menggunakan metode Backpropagation Neural Network (Andrian dan Ningsih, 2014) dan juga prediksi kasus COVID-19 di Indonesia menggunakan metode Backpropagation dan Fuzzy Tsukamoto yang dilakukan oleh [9].

Jaringan Syaraf Tiruan (Neural Network) menyediakan berbagai macam arsitektur jaringan dan pelatihan. Arsitektur Backpropagation merupakan salah satu arsitektur jaringan syaraf tiruan yang dapat digunakan untuk mempelajari dan menganalisis pola data masa lalu lebih tepat sehingga diperoleh keluaran yang lebih akurat (dengan kesalahan atau Error Minimum) (Oktaviani dan Afdal, 2013).

Peran utama dari data yaitu proyeksi dan prediksi, data dapat digunakan sebagai dasar untuk memprediksi keadaan di masa depan khususnya dalam proses pengambilan kebijakan. Data memiliki sifat yang sama yakni saling terkait satu sama lain (linked data) sehingga diperlukan aspek multidimensi atau melihat data dari berbagai sudut pandang. Data-data inilah yang dibutuhkan untuk kemudian digunakan sebagai dasar pengambilan kebijakan yang lebih baik untuk pemerintah [10]. Dalam penelitian ini di ambil data perkembangan jumlah kasus dari wabah COVID-19 di Indonesia melalui situs pemerintahan yaitu covid19.co.id sesuai dengan

arahan dari BPBD Provinsi Bali dengan menggunakan data dari 1 April 2020 sampai 30 April 2021.

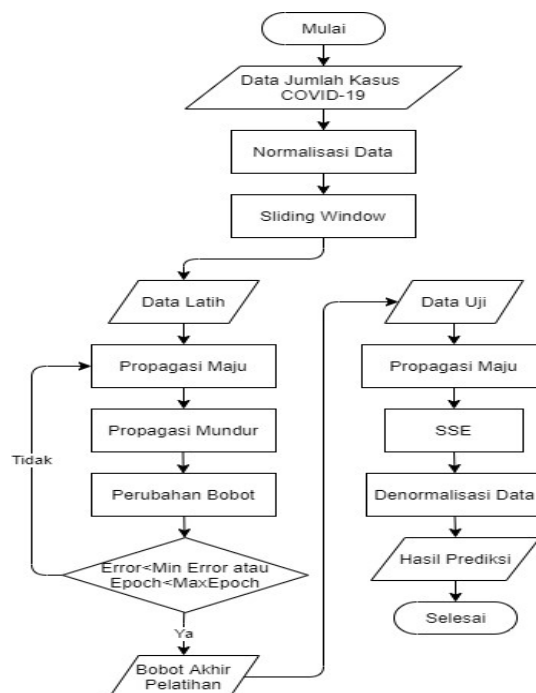
Terus bertambahnya kasus COVID-19, meningkatnya angka kematian dan permasalahan ekonomi serta pentingnya data dalam pengambil kebijakan dalam menangani penyebaran wabah COVID-19, maka tujuan penulisan ini adalah untuk membuat sistem prediksi jumlah kasus COVID-19 di Indonesia dengan metode Sliding Windows Backpropagation Neural Network sehingga dapat membantu pemerintah dalam memprediksi jumlah kasus COVID-19 di Indonesia untuk masa yang akan datang sehingga mampu mendukung pemerintah dalam menentukan kebijakan ataupun tindakan yang tepat untuk mengurangi penyebaran kasus COVID-19 di Indonesia.

2. METODE PENELITIAN

Proses prediksi jumlah kasus COVID-19 di Indonesia memiliki beberapa tahapan yang dilakukan dimana pada masing-masing proses terdiri dari perhitungan tersendiri dengan melibatkan beberapa algoritma.

Lebih detail rancangan prediksi ditunjukkan pada Gambar 1. Alur Proses Prediksi dengan BPNN. Pada gambar rancangan tersebut terdiri dari beberapa proses yang diantaranya:

- Normalisasi data jumlah kasus COVID-19.
- Sliding Windows untuk membentuk varian data baru.
- Pelatihan dengan menggunakan algoritma BPNN
- Pengujian data dengan menggunakan bobot terbaik
- Denormalisasi prediksi jumlah kasus COVID-19



Gambar 1 Alur Proses Prediksi dengan BPNN

2.1 Normalisasi dan Denormalisasi

Tujuan dari normalisasi data adalah untuk proses penskalaan nilai atribut dari data sehingga bisa berada pada range tertentu. Salah satu metode normalisasi yang dapat digunakan adalah Min-max normalization. Proses normalisasi tersebut merupakan metode dengan melakukan transformasi linier terhadap data aslinya. Adapun persamaan yang digunakan untuk menghitung nilai normalisasi adalah [11]:

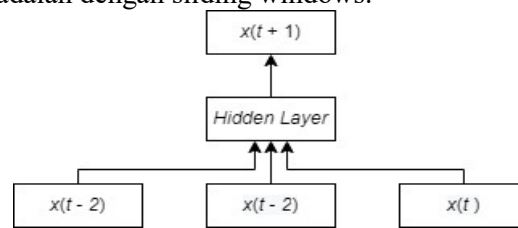
$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Dengan x' merupakan data hasil normalisasi, x_i merupakan data ke i , x_{min} merupakan data dengan nilai minimum dan x_{max} merupakan data dengan nilai maksimum. Proses denormalisasi data bertujuan untuk mengembalikan hasil keluaran jaringan agar berada pada range sebelumnya. Denormalisasi data dapat dilakukan dengan menggunakan persamaan berikut:

$$x_i = x'(x_{max} - x_{min}) + x_{min} \tag{2}$$

2. 2 Sliding Windows

Salah satu pendekatan dari model standar Jaringan Saraf Tiruan dalam melakukan prediksi data time series adalah dengan sliding windows.



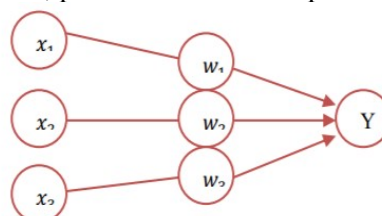
Gambar 2 Ilustrasi Sliding Windows

Teknik ini mendorong fungsi f pada Jaringan Saraf Tiruan dan menggunakan fungsi feedforward seperti pada arsitektur MLP, RBF atau korelasi cascade, dengan menggunakan satu set N tuple sebagai masukan dan satu output sebagai nilai target jaringan. Pada Gambar 2 Ilustrasi Sliding Windows menunjukkan teknik standar prediksi data time series menggunakan Sliding Windows, pada kasus tersebut menggunakan tiga tahapan waktu. Pada Gambar 2 Ilustrasi Sliding Windows rentan waktu yang digunakan adalah tiga hari, di mana tiga hari data sebelumnya digunakan untuk memprediksi hasil data kedepannya [11].

2. 2 Neural Network

Menurut M.F Andrijasa Neural Network (Jaringan Saraf Tiruan) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut. Puspitaningrum mengatakan Neural Network tercipta sebagai suatu generalisasi model matematika dari pemahaman manusia (human cognition) yang didasarkan atas asumsi pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron. Isyarat mengalir diantara sel saraf melalui suatu sambungan penghubung, setiap sambungan penghubung memiliki bobot yang bersesuaian dan setiap sel saraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya. JST ditentukan oleh 3 hal yaitu :

- a. Pola hubungan antar neuron (disebut arsitektur jaringan)
- b. Metode untuk menentukan bobot penghubung
- c. Fungsi aktivasi. Sebagai contoh, perhatikan neuron Y pada Gambar 1. Fungsi Aktivasi.



Gambar 3 Fungsi Aktivasi

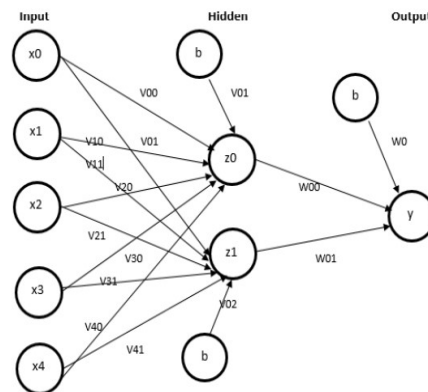
Y menerima input dari neuron X_1, X_2, X_3 dengan bobot hubungan masing-masing W_1, W_2, W_3 . Ketiga impuls neuron yang ada dijumlahkan yaitu $net = X_1 W_1 + X_2 W_2 + X_3 W_3$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(\text{net})$. Topologi jaringan yang digunakan adalah Jaringan Banyak Lapisan (*Multilayer Network*).

2. 4 Backpropagation Neural Network (BPNN)

Adalah salah satu algoritma multi-layer perceptron (MLP) yang memiliki dua arah yaitu maju dan mundur. Karena BPNN termasuk multilayer maka BPNN memiliki tiga layer dalam proses pelatihannya yaitu input layer, hidden layer dan output layer. Dengan adanya hidden layer maka tingkat error pada BPNN dapat diperkecil dibandingkan pada single layer. Hal ini dikarenakan fungsi hidden layer pada BPNN untuk memperbarui dan menyesuaikan bobot. Dengan adanya penyesuaian bobot ini akan didapatkan nilai bobot yang baru yang bisa diarahkan untuk mendekati target yang diinginkan.

Arsitektur algoritma BPNN terdiri dari tiga layer yaitu input layer, hidden layer dan output layer. Pada input layer tidak terjadi proses komputasi, namun terjadi pengiriman sinyal masukan ke hidden layer. Pada hidden layer dan output layer terjadi proses komputasi terhadap bobot dan bias, selain itu dilakukan perhitungan juga terhadap hasil dari hidden layer ke output layer tersebut berdasarkan fungsi aktivasi tertentu. Menurut [12] berikut merupakan contoh dari gambar arsitektur Backpropagation:



Gambar 4 Arsitektur Backpropagation

Dengan X_i , $i \in \{0, 1, 2, 3, 4\}$ berperan sebagai *neuron* pada *input layer*, Z_j , $j \in \{0, 1\}$ berperan sebagai *neuron hidden layer*, V_{ij} , $i \in \{0, 1, 2, 3, 4\}$ $j \in \{0, 1\}$ merupakan bobot yang berada di antara *input* dan *hidden*, b merupakan Bias, W_{jk} , $j \in \{0, 1\}$ $k \in \{0, 1\}$ merupakan bobot yang berada di antara *hidden* dengan *output layer* dan Y_k (y) merupakan berperan sebagai *output layer*.

Pelatihan Backpropagation meliputi 3 Tahap yaitu tahap maju, propagasi mundur, dan perubahan bobot. Algoritma pelatihan untuk jaringan dengan satu lapisan tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut:

Langkah 0: Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-8

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

Tahap I: Propagasi Maju

Langkah 3: Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.

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

$$z_{\text{net}_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (3)$$

$$z_j = f(z_{\text{net}_j}) = \frac{1}{1 + e^{-z_{\text{net}_j}}}$$

Keterangan :

z_net_j = sinyal *input* pada *hidden layer* ke - j
 v_{jo} = bias ke *hidden layer* ke - j
 v_{ji} = bobot antara *unit input layer* ke - I dan *hidden layer* ke - j
 x_i = *unit input layer* ke - i
 z_j = *unit input layer* ke - j
 i = urutan *unit input layer*
 j = urutan *unit hidden layer*
 p = jumlah maksimum unit pada *hidden layer*
 e = *bilangan euler*

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

$$y_net_k = w_{ko} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (4)$$

Keterangan :

y_net_k = sinyal masukan *output* ke - k
 w_{ko} = bias ke *hidden layer* ke - k
 w_{kj} = *output* ke - k dan *hidden layer* ke - j
 z_j = aktivasi *hidden layer* ke - j

Tahap II: Propagasi mundur

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

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k)$$

δ_k merupakan unit eror yang akan dipakai dalam perubahan bobot lapisan di bawahnya (langkah 7). Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α .

$$\Delta w_{kj} = \alpha \delta_k z_j; k = 1, 2 \dots, m; j = 0, 1, 2 \dots, p$$

Keterangan :

δ_k = faktor koreksi *error* bobot w_{kj}
 t_k = target *output* ke - k
 y_k = aktivasi *output* ke - k
 Δw_{kj} = nilai koreksi *error* bobot w_{kj}
 z_j = aktivasi *hidden layer* ke - j

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

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{kj} \quad (5)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_net_j f'(z_net_j) = \delta_net_j z_j (1 - z_j)$$

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

$$\Delta v_{ji} = \alpha \delta_j x_i; j = 1, 2 \dots, p; i = 0, 1, 2 \dots, n$$

Keterangan :

δ_net_j = jumlah delta bobot *hidden layer* ke - j
 w_{kj} = bobot antara *output* ke - k dan *hidden layer* ke - j
 δ_j = faktor koreksi bobot v_{ji}
 z_j = aktivasi *hidden layer* ke - j
 v_{ji} = nilai koreksi *error* bobot v_{ji}
 α = laju percepatan (*learning rate*)
 x_i = *unit input* ke - i

Tahap III: Perubahan bobot**Langkah 8:** Hitung semua perubahan bobot.

Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (6)$$

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

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (7)$$

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

Keterangan :

 w_{kj} (baru) = bobot baru dari *unit hidden layer* menuju *unit output layer* Δw_{kj} (lama) = bobot lama dari *unit hidden layer* menuju *unit output layer* v_{ji} (baru) = bobot baru dari *unit hidden layer* menuju *unit input layer* Δv_{ji} (lama) = bobot lama dari *unit hidden layer* menuju *unit input layer*

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka langkah 4 dan 5 harus diselesaikan. Demikian juga turunannya pada langkah 6 dan 7.

Langkah 9: Uji kondisi berhenti (akhir iterasi).**2. 4.1 Pengujian Algoritma Backpropagation**

Proses pengujian dengan algoritma Backpropagation Neural Network dapat dilakukan setelah proses pelatihan pada data latih telah mendapatkan nilai bobot akhir dan model jaringan. Nilai bobot akhir dan model jaringan ini akan digunakan untuk melakukan proses pengujian pada data uji. Pengujian Backpropagation dilakukan melalui propagasi maju, dimana langkah-langkahnya sebagai berikut:

Langkah 0: Inisialisasi semua bobot (dari hasil pelatihan)**Langkah 1:** Setiap unit input menerima sinyal input dan menyebarkan sinyal tersebut pada seluruh unit pada hidden layer**Langkah 2:** Setiap unit hidden $z_j (j = 1, 2, \dots, p)$ akan menjumlahkan sinyal-sinyal input yang sudah berbobot termasuk biasnya menggunakan rumus berikut

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (8)$$

Kemudian menggunakan fungsi aktivasi sigmoid yang telah ditentukan untuk menghitung sinyal output dari hidden unit yang bersangkutan,

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (9)$$

Langkah 3: Setiap unit output $y_k (k = 1, 2, \dots, m)$ akan menjumlahkan sinyal-sinyal input yang sudah berbobot termasuk biasnya

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (10)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari unit output yang bersangkutan:

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (11)$$

2. 4.2 Learning rate (α)

Menurut Fausett [13], learning rate adalah suatu parameter pelatihan yang digunakan untuk menghitung nilai koreksi bobot pada waktu proses pelatihan. Nilai learning rate adalah 0 hingga 1. Semakin besar nilai learning rate yang digunakan, maka semakin cepat waktu dalam proses pelatihan. Namun, jika nilai learning rate terlalu besar, maka ketelitian dalam proses pelatihan akan semakin rendah. Begitu pula apabila nilai learning semakin kecil, maka ketelitian akan semakin besar, namun membutuhkan waktu yang cukup lama.

2. 4.3 Momentum

Menurut Badieah dkk [14], momentum berfungsi untuk mempercepat pelatihan pembelajaran. Metode momentum melibatkan bobot ditambah faktor tertentu dari penyesuaian sebelumnya. Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Formula yang digunakan perubahan bobot menggunakan momentum adalah $\Delta w_{jk}(t+1) = \alpha s_k z_j + \mu \Delta w_{jk}(t)$

2. 4.4 Sum Square Error

Perhitungan *error* merupakan pengukuran kesalahan antara output aktual dan output target. Langkah yang digunakan adalah *SSE* (*Sum Square Error*) yang merupakan hasil penjumlahan nilai kuadrat *error neuron* 1 dan *neuron* 2 pada setiap lapisan *output* pada setiap data [15].

Sum Of Squares Error (*SSE*) adalah mengukur kesalahan penggunaan estimasi persamaan regresi untuk menghitung nilai variabel terikat dari sampel. Fungsi dari *SSE* dapat dilihat pada persamaan $SSE = \sum (Y_i - \hat{Y}_i)^2$. Dengan Y_i merupakan data aktual dan \hat{Y}_i merupakan data hasil prediksi [16]

2. 4.5 Stop Condition

Adalah suatu kriteria tertentu yang telah ditetapkan di awal pada proses pelatihan jaringan syaraf tiruan. Pada metode Backpropagation, proses training akan dilakukan secara berulang-ulang dan akan berhenti apabila stop condition (kondisi berhenti) telah terpenuhi. Menurut Kholis & Alam (2016) Stop condition pada metode Backpropagation dapat ditunjukkan seperti di bawah ini [12]:

- Telah mencapai batas error tertentu. Perhitungan nilai error yang digunakan adalah menggunakan *SSE* (*Sum Square Error*).
- Mencapai batas iterasi maksimum atau epoch yang telah di tentukan di awal proses pelatihan.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini data yang digunakan adalah data riwayat jumlah kasus COVID-19 di Indonesia dari bulan Maret 2020 sampai dengan bulan April 2021 yang berjumlah 393 hari. Data tersebut di bagi menjadi dua yaitu 80% digunakan sebagai data latih dan 20% digunakan sebagai data uji.

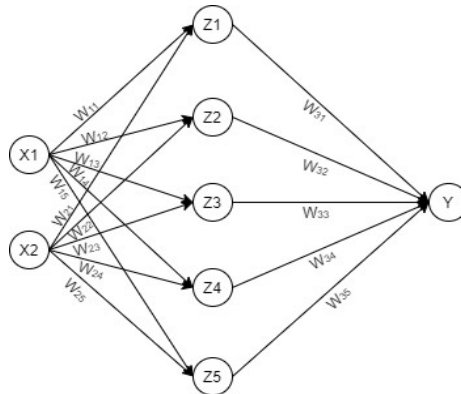
3. 1 Analisis Hasil Training dengan BPNN

Dalam sebuah Neural Network wajib melakukan proses training yang bertujuan untuk mengajarkan pola data yang diberikan sehingga jaringan bisa mengenali dan mengidentifikasi pola-pola tersebut. Dalam penelitian ini arsitektur neural network yang digunakan adalah jaringan backpropagation.

Faktor-faktor dalam sistem yang berpengaruh agar dapat menghasilkan output atau keluaran yang baik diantaranya adalah windows size, jumlah neuron, learning rate, momentum, jumlah maksimum epoch, dan target error. Pada penelitian ini training dilakukan dengan membandingkan hasil yang didapat, kemudian hasil yang terbaik akan digunakan pada proses selanjutnya.

3. 1.1 Training Arsitektur Backpropagation Neural Network Windows Size Dua

Arsitektur jaringan backpropagation dan parameter terbaik yang diperoleh dari training dengan windows size dua dan digunakan untuk pengujian ditunjukkan pada Gambar 10 dan Tabel 1.



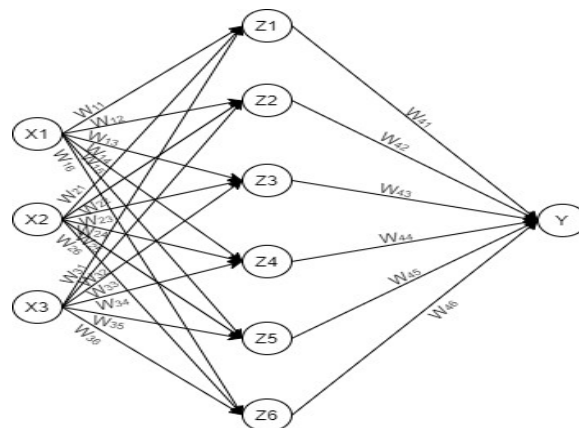
Gambar 10 Arsitektur Pengujian BPNN Windows Size Dua

Tabel 1 Parameter Proses Pengujian Menggunakan Jumlah Neuron Dengan Windows Size Dua

Parameter	Nilai	Keterangan
Input	2	X_1 dan X_2
Target Error (SSE)	0,001	Target proses berhenti
Learning Rate	0,12	Kecepatan pembelajaran sistem
Momentum	0,95	Konstanta momentum
Jumlah Hidden	1	Jumlah lapisan tersembunyi
Jumlah Neuron	5	Jumlah Neuron pada hidden
Maksimum Epoch	1.000.000	Jumlah iterasi atau pengulangan
Target Output	1	Y
Toleransi	0,1	Nilai toleransi prediksi jumlah kasus COVID-19
Transfer Function	Sigmoid Biner	
Bobot	Random (0,1)	

3. 1.3 Training Arsitektur Backpropagation Neural Network Windows Size Tiga

Arsitektur jaringan backpropagation dan parameter terbaik yang diperoleh dari training dengan windows size tiga dan digunakan untuk pengujian ditunjukkan pada Gambar 11 dan Tabel 2.



Gambar 11 Arsitektur Pengujian BPNN Windows Size Tiga

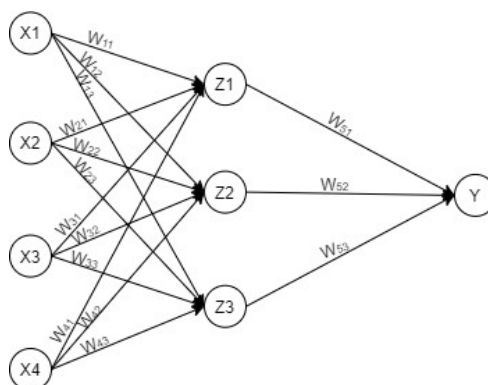
Tabel 2 Parameter Proses Pengujian Menggunakan Jumlah Neuron Dengan Windows Size Tiga

Parameter	Nilai	Keterangan
Input	3	X_1 , X_2 dan X_3
Target Error (SSE)	0,001	Target proses berhenti

Parameter	Nilai	Keterangan
<i>Learning Rate</i>	0,12	Kecepatan pembelajaran sistem
<i>Momentum</i>	0,65	Konstanta momentum
Jumlah <i>Hidden</i>	1	Jumlah lapisan tersembunyi
Jumlah <i>Neuron</i>	6	Jumlah <i>Neuron</i> pada <i>hidden</i>
Maksimum <i>Epoch</i>	500.000	Jumlah iterasi atau pengulangan
Target Output	1	Y
Toleransi	0,1	Nilai toleransi prediksi jumlah kasus COVID-19
<i>Transfer Function</i>	<i>Sigmoid Biner</i>	
Bobot	<i>Random (0,1)</i>	

3. 1.4 Training Arsitektur Backpropagation Neural Network Windows Size Empat

Arsitektur jaringan backpropagation dan parameter terbaik yang diperoleh dari training dengan windows size empat dan digunakan untuk pengujian ditunjukkan pada Gambar 12 dan Tabel 3.



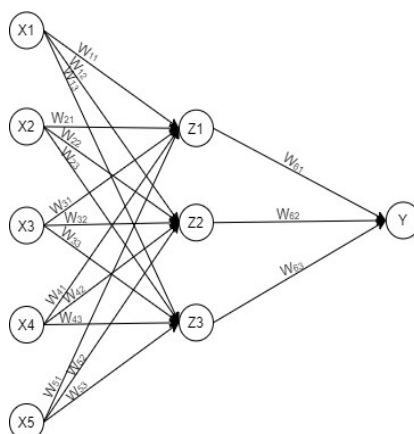
Gambar 12 Arsitektur Pengujian BPNN Windows Size Empat

Tabel 3 Parameter Proses Pengujian Menggunakan Jumlah Neuron Dengan Windows Size Empat

Parameter	Nilai	Keterangan
Input	4	X_1, X_2, X_3 dan X_4
Target Error (SSE)	0,001	Target proses berhenti
<i>Learning Rate</i>	0,08	Kecepatan pembelajaran sistem
<i>Momentum</i>	0,65	Konstanta momentum
Jumlah <i>Hidden</i>	1	Jumlah lapisan tersembunyi
Jumlah <i>Neuron</i>	3	Jumlah <i>Neuron</i> pada <i>hidden</i>
Maksimum <i>Epoch</i>	1.000.000	Jumlah iterasi atau pengulangan
Target Output	1	Y
Toleransi	0,1	Nilai toleransi prediksi jumlah kasus COVID-19
<i>Transfer Function</i>	<i>Sigmoid Biner</i>	
Bobot	<i>Random (0,1)</i>	

3. 1.5 Training Arsitektur Backpropagation Neural Network Windows Size Lima

Arsitektur jaringan backpropagation dan parameter terbaik yang diperoleh dari training dengan windows size lima dan digunakan untuk pengujian ditunjukkan pada Gambar 13 dan Tabel 4.



Gambar 13 Arsitektur Pengujian BPNN Windows Size Lima

Tabel 4 Parameter Proses Pengujian Menggunakan Jumlah Neuron Dengan Windows Size Lima

Parameter	Nilai	Keterangan
Input	5	X ₁ , X ₂ , X ₃ , X ₄ , dan X ₅
Target Error (SSE)	0,001	Target proses berhenti
Learning Rate	0,08	Kecepatan pembelajaran sistem
Momentum	0,85	Konstanta momentum
Jumlah Hidden	1	Jumlah lapisan tersembunyi
Jumlah Neuron	3	Jumlah Neuron pada hidden
Maksimum Epoch	1.000.000	Jumlah iterasi atau pengulangan
Target Output	1	Y
Toleransi	0,1	Nilai toleransi prediksi jumlah kasus COVID-19
Transfer Function	Sigmoid Biner	
Bobot	Random (0,1)	

3. 1 Analisis Hasil Pengujian dengan BPNN

Proses pengujian data jumlah kasus COVID-19 menggunakan algoritma Backpropagation Neural Network dimana arsitektur jaringan backpropagation dan parameter yang akan digunakan pada pengujian disesuaikan dengan hasil terbaik yang telah diperoleh dari proses training yang telah dilakukan dan jumlah data yang digunakan dalam proses pengujian adalah sebanyak 20% vektor data.

Berdasarkan pengujian yang telah dilakukan terhadap keempat windows size, diperoleh nilai prediksi dengan akurasi dan SSE terkecil adalah pengujian dengan menggunakan windows size tiga dengan tingkat akurasinya sebesar 17% dan nilai SSE nya sebesar 0,69.

4. KESIMPULAN

Penelitian ini dilakukan dengan menggunakan data historis deret waktu jumlah kasus COVID-19 di Indonesia selama satu tahun dengan pendekatan sliding window dimana windows size atau ukuran jendela akan mewakili jumlah variabel lapisan input pada jaringan backpropagation. Windows size atau ukuran jendela dan perbandingan data latih dan data uji sangat mempengaruhi hasil prediksi jaringan backpropagation dan nilai sse. Beberapa kombinasi ukuran jendela dengan perbandingan data yang digunakan diperoleh nilai sse terkecil yaitu 0,69 sebagai nilai akurasi hasil prediksi yang tinggi dengan arsitektur jaringan tiga lapisan input, enam lapisan tersembunyi dan 1 lapisan output dengan perbandingan 80% data latih dan 20% data uji. Model jaringan dengan nilai sse terkecil merupakan hasil prediksi yang tinggi dan diharapkan dapat membantu pemerintah dalam memprediksi jumlah kasus COVID-19 di

Indonesia untuk masa yang akan datang sehingga mampu mendukung pemerintah dalam menentukan kebijakan ataupun tindakan yang tepat untuk mengurangi penyebaran kasus COVID-19 di Indonesia.

DAFTAR PUSTAKA

- [1] A. K. Rachmawati and S. D. Miasary, "Peramalan Penyebaran Jumlah Kasus Covid19 Provinsi Jawa Tengah dengan Metode ARIMA," *Zeta - Math J.*, vol. 6, no. 1, pp. 11–16, 2020, doi: 10.31102/zeta.2021.6.1.11-16.
- [2] F. Ahmad, "PENENTUAN METODE PERAMALAN PADA PRODUKSI PART NEW GRANADA BOWL ST Di PT . X Determine the actual and actual production plan is the main thing for the organization to avoid large losses in calculating the amount of production , PT . This research is to det,," *J. Integr. Sist. Ind.*, vol. 7, no. 1, pp. 31–39, 2020.
- [3] P. Sugiartawan, R. Pulungan, and A. K. Sari, "Prediction by a Hybrid of Wavelet Transform and Long-Short-Term-Memory Neural Network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 2, 2017.
- [4] P. Sugiartawan, S. Hartati, and A. Musdholifah, "Tourist Visits Prediction with Fully Recurrent Neural Network," *Int. Conf. Inf. Technol. Eng. Sci. its Appl.*, 2018.
- [5] P. Sugiartawan, I. D. K. R. Rustina, and R. W. Saleh Insani, "E-Government Media Informasi Alat Kelengkapan Dewan Provinsi Bali dan Media Diskusi Berbasis Website," *J. Sist. Inf. dan Komput. Terap. Indones.*, vol. 1, no. 2, pp. 75–86, 2018, doi: 10.33173/jsikti.17.
- [6] I. Pratistha, I. P. A. Mahadewa, and P. Sugiartawan, "Sistem Pendukung Keputusan Kelompok Pemilihan E-Commerce Menggunakan Metode Profile Matching dan BORDA," *J. Sist. Inf. dan Komput. Terap. Indones.*, vol. 1, no. 1, pp. 13–24, 2018.
- [7] I. Halimi and W. A. Kusuma, "Prediksi Indeks Harga Saham Gabungan (IHSG) Menggunakan Algoritma Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 4, no. 1, p. 24, 2018, doi: 10.26418/jp.v4i1.25384.
- [8] B. Satria, "Prediksi Volume Penggunaan Air PDAM Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 2, no. 3, pp. 674–684, 2018, doi: 10.29207/resti.v2i3.575.
- [9] F. Siskus and D. Arianto, "Prediksi Kasus Covid-19 Di Indonesia Menggunakan Metode Backpropagation Dan Fuzzy Tsukamoto," *J. Teknol. Inf.*, vol. 4, no. 1, pp. 120–127, 2020.
- [10] R. Wiguna, A. Sutisnawati, and D. Lyesmaya, "Analisis Proses Pembelajaran Siswa Berbasis Online di Kelas Rendah Pada Masa Pandemic Covid-19," *J. perseda*, vol. III, no. 2, pp. 75–79, 2020.
- [11] P. Sugiartawan, A. A. J. P. Permana, and P. I. Prakoso, "Forecasting Kunjungan Wisatawan Dengan Long Short Term Memory (LSTM)," *J. Sist. Inf. dan Komput. Terap. Indones.*, vol. 1, no. 1, pp. 43–52, 2018.
- [12] M. F. Almas and B. D. Setiawan, "Implementasi Metode Backpropagation untuk Prediksi Harga Batu Bara," 2018.
- [13] G. Guntoro, "Supervisi Pengawas dan Kepemimpinan Kepala Sekolah sebagai Stimulus dalam Meningkatkan Kinerja Guru," *J. Ilm. Iqra'*, vol. 14, no. 1, p. 64, 2020, doi: 10.30984/jii.v14i1.1100.
- [14] Guntoro, L. Costaner, and Lisnawita, "Prediksi Jumlah Kendaraan di Provinsi Riau Menggunakan Metode Backpropagation," 2019.
- [15] D. F. Azam, D. E. Ratnawati, and P. P. Adikara, "Prediksi Harga Emas Batang Menggunakan Feed Forward Neural Network Dengan Algoritme Genetika," 2018.
- [16] F. R. Lumbanraja, I. H. B. Sitepu, D. Kurniawan, and Aristoteles, "Prediksi Jumlah Penderita Penyakit Tuberkulosis di Kota Bandar Lampung Menggunakan Metode SVM (Support Vector Machine)," 2020.