

## **IMPLEMENTASI JARINGAN SYARAF TIRUAN UNTUK ESTIMASI KUNJUNGAN PASIEN REHABILITASI MEDIS**

**Eka Ramadhani Putra<sup>1\*</sup>, Tesa Vausia Sandiva<sup>2</sup>, Melati Rahma Suri<sup>3</sup>**

*<sup>1,2,3</sup> Program Studi Sistem Informasi, Universitas Pamulang*

*\*E-mail: [dosen03223@unpam.ac.id](mailto:dosen03223@unpam.ac.id)*

### **ABSTRAK**

Jaringan Syaraf Tiruan (JST) merupakan salah satu model kecerdasan buatan yang dirancang menyerupai sistem kerja jaringan syaraf biologis, khususnya dalam mengenali dan mempelajari pola data yang kompleks. Dalam penerapannya, algoritma Backpropagation berfungsi sebagai metode pelatihan utama untuk menyesuaikan bobot dan bias guna meningkatkan akurasi model. Penelitian ini memanfaatkan JST dengan algoritma Backpropagation untuk memprediksi jumlah kunjungan pasien rehabilitasi medis di masa depan. Data yang digunakan bersumber dari laporan tahunan Rumah Sakit Otak Dr. Drs. M. Hatta Bukittinggi selama periode 2015 hingga 2024. Proses penelitian terdiri dari enam tahap, yaitu input data, normalisasi, pelatihan, pengujian, perhitungan akurasi, dan prediksi. Pemrograman Python digunakan sebagai alat bantu pengolahan data dan pelatihan model karena memiliki pustaka yang mendukung pengembangan JST. Hasil penelitian menunjukkan bahwa model dengan struktur jaringan 3-10-1 mampu melakukan prediksi jumlah kunjungan pasien rehabilitasi medis selama lima bulan ke depan secara berturut-turut, yaitu 2547, 2506, 2463, 2482, dan 2495 pasien. Model ini menunjukkan waktu komputasi yang sangat cepat, yakni 0,001 detik, dengan rata-rata tingkat kesalahan sebesar 8,794% dan akurasi sebesar 91,706%. Dengan demikian, model JST Backpropagation ini dapat menjadi alternatif yang andal untuk membantu manajemen rumah sakit dalam merencanakan pelayanan rehabilitasi medis secara lebih efisien dan terukur.

Kata Kunci : Jaringan Syaraf Tiruan, Algoritma Backpropagation, Pasien, Prediksi, Python.

### **ABSTRACT**

*Artificial Neural Network (ANN) is one of the artificial intelligence models designed to resemble the working system of biological neural networks, particularly in recognizing and learning complex data patterns. In its application, the Backpropagation algorithm serves as the primary training method to adjust weights and biases in order to improve model accuracy. This study utilizes ANN with the Backpropagation algorithm to predict the number of future patient visits to the medical rehabilitation unit. The data used in this research was sourced from the annual reports of Dr. Drs. M. Hatta Brain Hospital in Bukittinggi, covering a 36-month period from 2020 to 2022. The research process consists of six stages: data input, normalization, training, testing, accuracy calculation, and prediction. Python programming is used as a tool for data processing and model training due to its extensive libraries that support ANN development. The results of this study indicate that a network structure of 3-10-1 is capable of predicting the number of medical rehabilitation patient visits for the next five months consecutively, with results of 2547, 2506, 2463, 2482, and 2495 patients. The model demonstrated a very fast computation time of 0.001 seconds, an average error rate of 8.794%, and an accuracy of 91.706%. Therefore, the Backpropagation-based ANN model can serve as a reliable alternative to assist hospital management in planning medical rehabilitation services more efficiently and measurably.*

*Keywords : Artificial Neural Network, Backpropagation Algorithm, Patient, Prediction, Python.*

## **PENDAHULUAN**

Perkembangan teknologi informasi berlangsung sangat pesat, salah satunya ditandai dengan meningkatnya perhatian terhadap machine learning yang kini banyak diterapkan dalam berbagai bidang seperti teknologi, bisnis, industri, kesehatan, hingga pasar saham. Machine learning digunakan untuk menyelesaikan berbagai permasalahan yang bersifat kompleks, seperti analisis citra, pengolahan data, dan deteksi objek. Tujuan utama dari teknik ini adalah memungkinkan komputer untuk belajar secara mandiri tanpa intervensi manusia serta memberikan respons yang tepat terhadap situasi tertentu. Fokus utama dalam penelitian machine learning adalah bagaimana meningkatkan performa sistem pembelajaran otomatis melalui berbagai proses pelatihan. Berdasarkan rumusan masalah dalam penelitian ini, tujuan utamanya adalah mengidentifikasi metode pelatihan, struktur jaringan, serta parameter terbaik yang sesuai untuk penerapan Jaringan Syaraf Tiruan (JST) dengan algoritma Backpropagation dalam memprediksi jumlah kunjungan pasien ke layanan rehabilitasi medis. Selain itu, penelitian ini juga bertujuan menjelaskan bagaimana penggunaan aplikasi Python dalam proses prediksi tersebut, serta menguji kombinasi metode dan parameter yang dapat menghasilkan hasil prediksi yang akurat dan stabil dalam konteks beban kunjungan rehabilitasi medis menggunakan JST Backpropagation.

Permasalahan yang ada pada machine learning dimanfaatkan sebagai pilihan dalam mengoptimalkan kinerja komputer berdasarkan data lampau. Riset ini mengulas teknik machine learning menggunakan Jaringan Syaraf Tiruan [1]. Rumah sakit adalah suatu institusi pelayanan kesehatan yang kompleks. Metode Backpropagation dan fungsi aktivasi sigmoid adalah hal yang paling penting dan merupakan metode yang digunakan untuk pelatihan multilayer neural networks feedforward [2].

JST adalah suatu sistem yang mengolah suatu data dengan meniru jalan kerja otak manusia. JST mampu merepresentasikan diri sendiri bahkan membuat suatu organisasi sendiri atau dari pengetahuan yang diterima saat belajar[3]. Salah satu keunggulan dari JST adalah kemampuan klasifikasi terhadap data yang belum diberikan pada saat pembelajaran sebelumnya [4]. Ada tiga tahap Pelatihan Backpropagation yaitu tahap maju, tahap mundur dan merubah nilai bobot agar memperkecil kesalahan yang terjadi [5]. Memperoleh inversi matriks dan menyelesaikan persamaan linier [6].

Hasil penelitian prediksi indeks harga konsumen menyimpulkan bahwa algoritma Backpropagation dapat digunakan pada prediksi indeks harga konsumen berdasarkan kelompok kesehatan. Dari uji model diperoleh model terbaik 12-70-1 dengan tingkat akurasi sebesar 92%, MSE 0,3659742 [7]. Algoritma Bayesian Regularization Backpropagation digunakan untuk memprediksi sudut gesekan internal tanah berdasarkan 145 data yang dikumpulkan dari eksperimen [8].

Penelitian Jaringan Syaraf Tiruan Backpropagation digunakan untuk memprediksi hasil tes fisik mahasiswa mencerminkan Algoritma Backpropagation

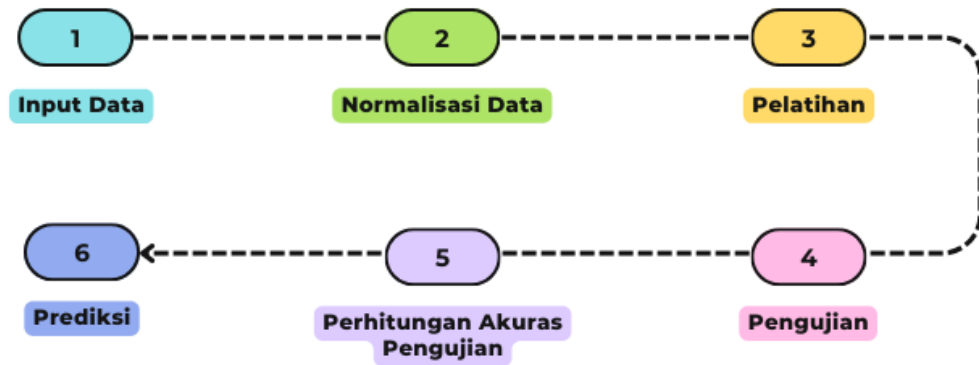
dalam memprediksi data ujian fisik dengan hasil yang berdampak pada pengembangan sistem pelatihan olahraga yang cerdas dan efektif [9]. Penelitian tentang prediksi jumlah permintaan koran menggunakan metode Jaringan Syaraf Tiruan Backpropagation mendapatkan nilai iterasi terbaik adalah 200, learning rate terbaik 0,6, pengujian data training dan data testing memperoleh nilai data training terbaik 100 dan data testing 10 [10]. Dalam implementasi Algoritma Support Vector Regression pada prediksi jumlah pengunjung pariwisata, mendapatkan hasil pengujian rata-rata nilai MAPE minimum yang dihasilkan adalah 9,16 % dan nilai MAPE terbaik adalah 6,98 % [11].

Model prediksi yang ditingkatkan untuk mengestimasi suhu Junction Transistor Bipolar Gate terisolasi (IGBT), kondisi operasional dan tingkat penuaan komponen ini dapat secara kasar dievaluasi [12]. Pengembangan model prediksi termasuk Bayesian Vector Neural Network (BVNN) yang diintegrasikan dengan Algoritma Backpropagation. Pendekatan ini memungkinkan analisis terhadap perkembangan ekonomi suatu wilayah dengan memanfaatkan fitur penting yang diekstraksi dari data ekonomi [13]. Jaringan syaraf BP umum memiliki dua kelemahan ketika digunakan untuk pelatihan peristiwa probabilitas kecil, yaitu pola tidak menjamin bahwa jumlah probabilitas semua peristiwa sama dengan satu dan kesalahan relatif antara keluaran aktual dan keluaran yang diinginkan sangat besar setelah pelatihan jaringan saraf [14].

Penggunaan Aplikasi Python ialah komputasi yang populer digunakan dalam bidang ilmu pengetahuan dan teknik, termasuk dalam pengembangan dan penerapan JST. Python menyediakan alat dan fungsi yang kuat untuk membangun, melatih, dan menerapkan JST dengan Algoritma Backpropagation. Dalam konteks ini, Python dapat digunakan sebagai platform untuk mengimplementasikan JST dengan Algoritma Backpropagation untuk memprediksi beban kunjungan rehabilitasi medis. Maka peneliti memilih aplikasi Python untuk menyelesaikan berbagai masalah numerik, perangkat lunak ini menawarkan kemudahan dan kesederhanaan dalam menyelesaikan masalah yang berhubungan dengan vektor dan matriks.

## **METODE**

Proses kerja Algoritma Backpropagation dalam penelitian ini mencakup beberapa tahapan, mulai dari pemberian data input, normalisasi data, pelaksanaan iterasi dari awal hingga akhir, pelatihan jaringan dengan penentuan parameter yang sesuai, pencarian nilai error terkecil, hingga pengujian metode Backpropagation untuk memperoleh hasil akhir. Seluruh tahapan ini berlanjut sampai proses selesai. Alur langkah kerja Algoritma Backpropagation tersebut dapat dijelaskan secara visual pada Gambar 2.1 berikut:



Gambar 2.1 Tahapan Algoritma *Backpropagation*

## 2.1 Input Data

Pada tahap input data, langkah awal yang dilakukan adalah mendefinisikan variabel input yang akan digunakan untuk memprediksi jumlah kunjungan pasien ke layanan rehabilitasi medis. Prediksi ini menjadi dasar dalam proses pengambilan keputusan menggunakan metode Jaringan Syaraf Tiruan (JST). Penentuan variabel dilakukan dengan mempertimbangkan relevansi data terhadap fokus penelitian yang sedang dijalankan. Adapun data input yang digunakan berasal dari jumlah kunjungan pasien rehabilitasi medis selama periode tahun 2015 hingga 2024.

## 2.2 Normalisasi Data

Normalisasi merupakan tahapan dalam pengolahan data yang bertujuan untuk mengubah format data, seperti memindahkan angka dari kolom ke baris serta mengonversi bilangan bulat menjadi bilangan pecahan. Langkah ini diperlukan agar data lebih sesuai untuk proses perhitungan bobot dalam Python, yang memerlukan nilai dalam bentuk pecahan atau telah dinormalisasi. Tujuan utama dari normalisasi adalah untuk memperkecil selisih antara nilai maksimum dan minimum pada data, sehingga memudahkan proses pelatihan jaringan. Setelah proses ini, data yang telah dinormalisasi akan diinput ke dalam sistem dan dihitung menggunakan rumus (2.1) sebagaimana dijelaskan pada perhitungan berikut:

$$X_i \text{ norm} = 0,8 * \frac{x_i - \min(x)}{\max(x) - \min(x)} + 0,1 \quad (2.1)$$

dimana:

$X_i \text{ norm}$  = Nilai data ke-i setelah dinormalisasi

$x_i$  = Nilai data ke-i

$\min(x)$  = Data nilai terkecil

$\max(x)$  = Data nilai Terbesar

## 2.3 Pelatihan

Setelah proses normalisasi selesai dilakukan, tahap berikutnya adalah pelatihan jaringan. Pada tahap ini, iterasi dalam Algoritma Backpropagation dilakukan dengan melibatkan lapisan tersembunyi (hidden layer), dimulai dari

penentuan nilai awal (initial weight) dan dilanjutkan dengan pendefinisian sejumlah variabel yang diperlukan. Adapun variabel-variabel tersebut digunakan sebagai dasar dalam proses perhitungan dan penyesuaian bobot selama pelatihan berlangsung:

a. Inisialisasi parameter-parameter JST Backpropagation

Langkah pertama yang harus dilakukan sebelum melakukan proses pelatihan ialah menentukan/menginisialisasi parameter-parameter JST Backpropagation yang akan digunakan, seperti:

Jumlah neuron pada *input layer* = 3

Jumlah neuron pada *hidden layer* = 10

Jumlah neuron pada *output layer* = 1

*Learning Rate* ( $\alpha$ ) = 0.1

Jumlah Iterasi = 100

Toleransi Error = 0.1

b. Inisialisasi nilai awal bobot V dan bobot W

Penentuan nilai awal untuk bobot V, bobot W, dan bias dilakukan secara acak dalam rentang nilai antara 0 hingga 1 atau -1 hingga 1. Rentang ini dipilih agar nilai bobot awal tidak terlalu besar, sehingga proses pelatihan dapat berjalan lebih stabil. Ukuran dari bobot V dan bobot W disesuaikan dengan jumlah neuron pada masing-masing layer dalam jaringan. Nilai awal bobot tersebut dihasilkan menggunakan fungsi =RAND() dengan batas bawah 0,05 dan batas atas 0,08. Berikut ini merupakan rincian ukuran bobot V dan bobot W yang digunakan dalam penelitian ini:

bobot V =  $(3+1) \times 10 = 4 \times 10$

bobot W =  $(10+1) \times 1 = 11 \times 1$

c. Proses Perambatan Maju (Forward Propagation)

Pada tahap ini, data input diteruskan melalui jaringan dari lapisan *input* hingga lapisan *output*. Setiap *neuron* di setiap lapisan menghitung nilai outputnya berdasarkan bobot dan bias yang ada, serta fungsi aktivasi yang digunakan seperti pada persamaan di bawah ini:

$$z_{in_j} = v_{0j} \sum_{i=1}^n x_i v_{ij} \quad (2.2)$$

$$z_j = f(z_{in_j}) \quad (2.3)$$

dimana:

$z_{in_j}$  = sinyal informasi dari *input layer* ke neuron *hidden layer* ke-j

$v_{0j}$  = Bias pada neuron *hidden layer* ke-j

$x_i$  = Neuron *input layer* ke-i

$v_{ij}$  = bobot dari *input layer* ke *hidden layer*

$z_j$  = Neuron *hidden layer* ke-j

$f(z_{in_j})$  = Fungsi aktivasi terhadap nilai  $z_{in}$  ke-j

Setelah tahap mencari nilai *neuron* pada *input* ke *hidden*, selanjutnya mencari nilai *hidden* ke *output* layer, menggunakan persamaan di bawah ini:

$$y_{in_j} = w_{0k} \sum_{j=1}^n z_j w_{jk} \quad (2.4)$$

$$y_j = f(y_{in_j}) \quad (2.5)$$

dimana:

$y_{in_j}$  = sinyal informasi dari *hidden layer* ke neuron *output layer* ke-j

$w_{0j}$  = Bias pada neuron *output layer* ke-j

$z_i$  = Neuron *hidden layer* ke-j

$w_{jk}$  = bobot dari *hidden layer* ke *output layer*

$y_j$  = Neuron *output layer* ke-j

$f(y_{in_k})$  = Fungsi aktivasi terhadap nilai  $y_{in}$  ke-k

#### d. Proses Perambatan Mundur (Backward Propagation)

Pada tahap ini, *gradien error* dihitung untuk setiap bobot dan bias dalam jaringan. Proses perambatan mundur dilakukan untuk memperbarui bobot yang menghubungkan antara *output layer* dan *hidden layer* (bobot  $W$ ). Berikut adalah persamaan yang dilakukan pada proses perambatan mundur:

$$\delta_k = (t_k - y_k) f'(y_{in_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.6)$$

dimana:

$\delta_k$  = Faktor koreksi dari neuron *output layer* ke-k

$t_k$  = Target *output* pada neuron *output layer* ke-k

$y_k$  = Neuron *output layer* ke-k.

## 2.4 Pengujian

Proses pengujian hanya melakukan proses perambatan maju atau *forward propagation*. Data yang digunakan merupakan data uji yang telah disediakan diawal. Untuk mengetahui pasien prediksi sebenarnya, maka kita harus melakukan denormalisasi terhadap hasil prediksi dengan menggunakan persamaan berikut ini:

$$y_{denorm} = (y * \max(x) - y * \min(x)) + \min(x) \quad (2.7)$$

## 2.5 Perhitungan Akurasi Pengujian

Proses pengujian diukur dengan menggunakan tingkat akurasi dan tingkat error. Tingkat akurasi menggambarkan seberapa tepat model atau sistem dalam melakukan prediksi terhadap data uji. Selanjutnya tingkat error, atau sering disebut sebagai tingkat kesalahan, mengindikasikan seberapa besar perbedaan antara hasil prediksi dari model dengan nilai sebenarnya dalam data uji dengan persamaan berikut:

$$\text{Persentase error (\%)} = \frac{\text{error}}{\text{denormalisasi}} * 100\% \quad (2.8)$$

$$\text{Persentase akurasi (\%)} = 100\% - \text{Persentase error} \quad (2.9)$$

## 2.6 Prediksi

Tahapan prediksi bertujuan untuk memperkirakan jumlah pasien pada bulan-bulan selanjutnya, di mana banyaknya bulan yang diprediksi disesuaikan dengan jumlah iterasi yang dijalankan. Pada tahap ini, proses yang dilakukan adalah *forward propagation* (perambatan maju) tanpa disertai langkah *backpropagation* (mundur). Selama iterasi berlangsung, model prediksi secara terus-menerus memperbarui dan menyempurnakan parameter-parameter jaringan guna meningkatkan tingkat akurasi hasil prediksi.

## HASIL DAN PEMBAHASAN

Hasil penelitian menunjukkan bahwa setelah tahap normalisasi, pelatihan menggunakan Algoritma *Backpropagation* memberikan peningkatan dalam akurasi prediksi model dan langkah-langkah iterasi pada lapisan tersembunyi dalam memperbarui bobot dan bias.

### 3.1 Input Data

Data yang digunakan adalah jumlah pasien tersebut diperoleh dari Laporan Tahunan Rumah Sakit Otak Dr. Drs. M. Hatta Bukittinggi. Data yang diperoleh untuk penelitian ini berupa format .pdf sehingga harus dilakukan pengolahan data terlebih dahulu. Selanjutnya data tersebut ditransformasi dalam bentuk .xlsx seperti yang disajikan pada Tabel 3.1:

**Tabel 3.1 Data Jumlah Kunjungan Pasien Rawat Jalan**

<b>Tahun</b>	<b>2020</b>	<b>2021</b>	<b>2022</b>
<b>Bulan</b>	<b>(orang)</b>	<b>(orang)</b>	<b>(orang)</b>
<b>Januari</b>	2220	1859	2735
<b>Februari</b>	2085	1764	2175
<b>Maret</b>	1852	2195	2668
<b>April</b>	1006	1977	2150
<b>Mei</b>	720	1843	2333
<b>Juni</b>	1593	2238	2472
<b>Juli</b>	1905	1962	2444
<b>Agustus</b>	1729	1782	2824
<b>September</b>	1727	2161	2757
<b>Oktober</b>	1506	2449	2772
<b>November</b>	2453	2761	2282
<b>Desember</b>	2057	2553	2858

### 3.2 Normalisasi Data

Setelah proses normalisasi dilakukan menggunakan rumus (2.1), dapat menggambarkan hasil-hasil yang telah diproses normalisasi data yang melibatkan serangkaian langkah untuk memperbaiki struktur data, sehingga tabel ini menjadi titik awal yang berguna untuk menganalisis hubungan antar variabel serta pola-pola yang mungkin tersembunyi di dalamnya.

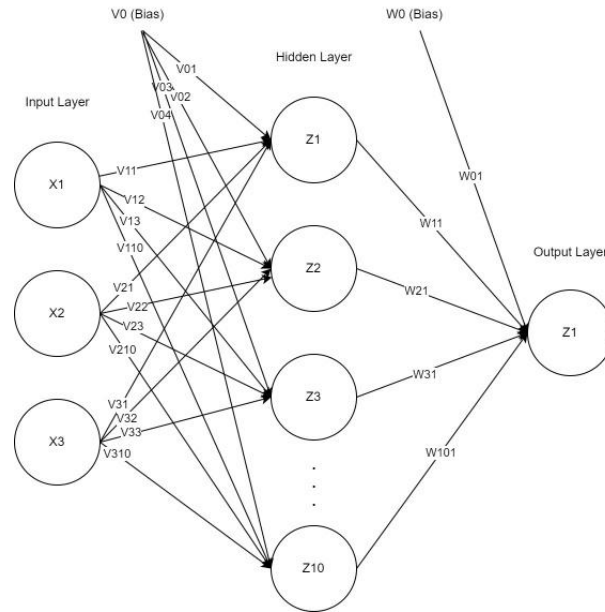
### 3.3 Pelatihan

Setelah menyelesaikan tahap normalisasi, langkah berikutnya adalah memulai proses pelatihan model. Setiap iterasi bertujuan untuk meningkatkan akurasi dan memperbarui bobot serta bias dalam lapisan tersembunyi untuk meningkatkan kemampuan prediktif model.

#### a. Inisialisasi parameter-parameter JST *Backpropagation*

Pada Tabel 3.1 dapat digambarkan bahwa data *input* terdiri atas 3 variabel *input*, variabel *output* / target dengan data sampel hasil normalisasi sebanyak 36 data. Adapun Arsitektur Jaringan yang dirancang adalah sebagai berikut:





Gambar 3.1 Arsitektur Jaringan *Backpropagation*

Jumlah neuron pada input layer akan menentukan format masukan dari JST *Backpropagation* baik itu untuk data latih ataupun data uji. Alasan mengapa jumlah *neuron* pada *output layer* bernilai 1 adalah karena target keluaran yang akan diprediksi berjumlah 1. Perubahan nilai parameter-parameter JST *Backpropagation* akan mempengaruhi kinerja JST *Backpropagation*.

b. Inisialisasi nilai awal bobot V dan bobot W

Setelah menentukan nilai awal bobot V dan bobot W menggunakan rumus , maka proses perambatan maju dan perambatan mundur dapat dilakukan. Kasus ini akan memperlihatkan contoh perhitungan matematis proses perambatan maju dan perambatan mundur untuk data latih ke-1. Berikut merupakan hasil dari penggunaan rumus RAND seperti yang disajikan pada Tabel 3.2 dan Tabel 3.3:

Tabel 3.2 Nilai Awal Bobot V

$\begin{matrix} j \\ i \end{matrix}$	1	2	3	4	5	6	7	8	9	10
0	0,056	0,058	0,073	0,056	0,065	0,073	0,067	0,076	0,074	0,058
1	0,059	0,066	0,068	0,069	0,079	0,075	0,057	0,055	0,065	0,060
2	0,055	0,056	0,056	0,071	0,068	0,057	0,059	0,072	0,062	0,059
3	0,066	0,057	0,077	0,074	0,055	0,063	0,066	0,071	0,061	0,064

Tabel 3.3 Nilai Awal Bobot W

$\begin{matrix} k \\ j \end{matrix}$	1
0	0,0601
1	0,0509
2	0,0703
3	0,0518
4	0,0569
5	0,0564
6	0,0558
7	0,0553
8	0,0547
9	0,0542
10	0,0536

c. Proses Perambatan Maju (Forward Propagation)

Langkah awal dalam melakukan proses *Forward Propagation* ialah menentukan nilai dari neuron-neuron pada *hidden layer* (nilai Z). Perhitungan nilai Neuron Z1, Z2 dan Z3 dilakukan dengan menggunakan persamaan (2.2). Berikut adalah perhitungan untuk mencari nilai neuron Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9 dan Z10:

$$z_{in_1} = 0,0564 + (0,7924 * 0,0590 + 0,1762 * 0,0545 + 0,7610 * 0,0664 = 0,1633$$

...

$$z_{in_{10}} = 0,0576 + (0,4628 * 0,0598 + 0,6496 * 0,0592 + 0,8028 * 0,0637 = 0,1749$$

Setelah memperoleh nilai Z\_in, langkah berikutnya adalah menerapkannya ke dalam fungsi aktivasi. Fungsi aktivasi bertanggung jawab untuk mengubah nilai Z\_in menjadi keluaran yang diinginkan sesuai dengan karakteristik fungsi tersebut. Berikut tahapan fungsi aktifasi menggunakan persamaan (2.3):

$$z_1 = f(z_{in_1}) = \frac{1}{1+e^{(-0,1633)}} = \frac{1}{1+0,8493} = 0,5407$$

...

$$z_{10} = f(z_{in_{10}}) = \frac{1}{1+e^{(-0,1749)}} = \frac{1}{1+0,8395} = 0,5436$$

Langkah selanjutnya dalam melakukan proses *Forward Propagation* ialah menentukan nilai dari *neuron-neuron* pada *output layer* (nilai Y). Perhitungan nilai Neuron Y1 dilakukan dengan menggunakan persamaan (2.4). Berikut adalah perhitungan untuk mencari nilai neuron Y1:

$$y_{in_1} = 0,0601 + (0,5407 * 0,0509 + 0,5298 * 0,0703 + 0,5480 * 0,0518 + 0,5245 * 0,0569 + 0,5250 * 0,0564 + 0,5419 * 0,0558 + 0,5370 * 0,0553 + 0,5439 * 0,0547 + 0,5459 * 0,0542 + 0,5436 * 0,0536) = 0,3610$$

Setelah memperoleh nilai Y\_in, langkah berikutnya adalah menerapkannya ke dalam fungsi aktivasi. Fungsi aktivasi bertanggung jawab untuk mengubah nilai Y\_in menjadi keluaran yang diinginkan sesuai dengan karakteristik fungsi tersebut. Berikut tahapan fungsi aktifasi menggunakan persamaan (2.5):

$$y_1 = f(y_{in_1}) = \frac{1}{1 + e^{(-y_{in_1})}} = \frac{1}{1 + 2,7183^{(-0,3610)}} = 0,5893$$

d. Proses Perambatan Mundur (Backward Propagation)

Langkah awal dalam melakukan proses *Backward Propagation* ialah menghitung nilai perambatan mundur dari *output layer* ke *hidden layer*. Perhitungan pembaruan bobot W pada proses ini dilakukan dengan menggunakan persamaan (2.6) dan persamaan (2.7) dan berikut langkah dalam penerapannya:

$$\delta_1 = (0,7301 - 0,5893) * 0,5893 (1 - 0,5893) = 0,0340$$

$$\Delta w_{01} = a\delta_1 = 0,1 * 0,0340 = 0,00341$$

...

$$\Delta w_{101} = a\delta_1 z_{10} = 0,1 * 0,0340 * 0,5436 = 0,00185$$

Setelah mendapatkan bobot terbaru untuk W, maka kita lakukan tahapan pembaharuan bobot W dengan persamaan (2.8) dan berikut langkah penerapannya:

$$w_{01}(\text{baru}) = w_{01}(\text{lama}) + \Delta w_{01} = 0,0601 + 0,00341 = 0,0635$$

...

$$w_{101}(\text{baru}) = w_{101}(\text{lama}) + \Delta w_{101} = 0,0536 + 0,00185 = 0,0555$$

Setelah dilakukan perubahan nilai bobot W, dilakukan Langkah yang sama untuk mencari bobot nilai V. Maka proses pelatihan untuk data ke-1 telah selesai dilakukan. Selanjutnya proses pelatihan untuk data ke-2 dapat dilakukan dengan menggunakan cara yang sama seperti proses pelatihan pada data ke-1. Bobot yang digunakan pada proses pelatihan ke-2 adalah bobot terbaru yang didapatkan dari hasil pelatihan data ke-1. Begitu juga pada proses pelatihan ke-3, bobot yang digunakan adalah bobot terbaru yang didapatkan dari hasil pelatihan data ke-2. Dan begitu seterusnya untuk data selanjutnya.

Ketika proses pelatihan untuk semua data latih selesai dilakukan, maka proses pelatihan untuk satu siklus iterasi telah selesai dilakukan. Maka, proses perhitungan rata-rata eror atau *Mean Square Error (MSE)* dari proses pelatihan pada iterasi pertama dapat dilakukan.

### 3.4 Pengujian

Proses pengujian hanya melakukan proses perambatan maju atau *forward propagation*. Berikut adalah tahapan yang dilakukan untuk pengujian keluaran dari data uji. Pada kasus ini, perhitungan proses prediksi akan dilakukan terhadap data uji bulan ke-13 sampai ke-15, seperti pada Tabel 3.6 :

Tabel 3. 2 Data Uji

X1	X2	X3	Target
0,7924	0,1762	0,7610	0,7301

Selanjutnya dilakukan perhitungan semua nilai output di hidden layer seperti pengerjaan matematis di bawah ini:

$$z_{in_1} = 0,0564 + (0,7924 * 0,0590 + 0,1762 * 0,0545 + 0,7610 * 0,0664 = 0,1633$$

$$z_{in_2} = 0,0583 + (0,7301 * 0,0661 + 0,1000 * 0,0561 + 0,1282 * 0,0569 = 0,1194$$

...

$$z_{in_{10}} = 0,0576 + (0,4628 * 0,0598 + 0,6496 * 0,0592 + 0,8028 * 0,0637 = 0,1749$$

Setelah mendapatkan hasil nilai output di hidden layer, lalu dilakukan perhitungan untuk menentukan fungsi aktivasi seperti di bawah ini:

$$y_{in_1} = 0,0601 + (0,5407 * 0,0509 + 0,5298 * 0,0703 + 0,5480 * 0,0518 + 0,5245 * 0,0569 + 0,5250 * 0,0564 + 0,5419 * 0,0558 + 0,5370 * 0,0553 + 0,5439 * 0,0547 + 0,5459 * 0,0542 + 0,5436 * 0,0536) = 0,3610$$

Setelah memperoleh nilai  $Y_{in}$ , langkah berikutnya adalah menerapkannya ke dalam fungsi aktivasi. Fungsi aktivasi bertanggung jawab untuk mengubah nilai  $Y_{in}$  menjadi keluaran yang diinginkan sesuai dengan karakteristik fungsi tersebut. Berikut tahapan fungsi aktifasi menggunakan persamaan (2.5):

$$y_1 = f(y_{in_1}) = \frac{1}{1 + e^{(-y_{in_1})}} = \frac{1}{1 + 2,7183^{(-0,3610)}} = 0,5893$$

Langkah terakhir dari pengujian ialah langkah yang dilakukan merupakan tahapan sebelum melakukan prediksi. Hasil denormalisasi merupakan nilai prediksi yang dari jumlah pasien sesuai target uji pada pola data yang telah ditetapkan. Untuk mengetahui pasien prediksi sebenarnya, maka kita harus melakukan denormalisasi terhadap hasil prediksi dengan menggunakan persamaan berikut ini:

$$y_{denorm} = (0,5893 * 2453 - 0,5893 * 720) + 720 = 2028,263374$$

### 3.5 Perhitungan Akurasi Pengujian

Tahapan awal dalam mencari tingkat akurasi adalah mencari error terlebih dahulu, berikut perhitungan untuk mencari persentase error:

$$\text{Persentase error (\%)} = \frac{478,7366}{2028,2633} * 100 = 8,2936\%$$

Setelah mendapatkan tingkat persentase error, selanjutnya dilakukan perhitungan untuk mencari tingkat akurasi, seperti persamaan di bawah ini:

$$\text{Persentase akurasi (\%)} = 100\% - 8,2936 = 91,7064$$

### 3.6 Prediksi

Proses prediksi hanya melakukan proses perambatan maju atau forward propagation. Berikut adalah tahapan yang dilakukan untuk prediksi keluaran dari data prediksi. Pada kasus ini, perhitungan proses prediksi akan dilakukan terhadap data uji bulan ke-34 sampai ke-36. Setelah memperoleh nilai  $Y_{in}$ , langkah berikutnya adalah menerapkannya ke dalam fungsi aktivasi. Fungsi aktivasi bertanggung jawab untuk mengubah nilai  $Y_{in}$  menjadi keluaran yang diinginkan sesuai dengan karakteristik fungsi tersebut. Berikut tahapan fungsi aktivasi menggunakan persamaan (2.5):

$$y_1 = f(y_{in1}) = \frac{1}{1 + e^{(-y_{in1})}} = \frac{1}{1 + 2,7183^{(-0,3802)}} = \frac{1}{1 + 0,6837} = 0,5939$$

Langkah terakhir dari prediksi ialah langkah yang dilakukan merupakan normalisasi. Hasil denormalisasi merupakan nilai prediksi yang dari jumlah pasien sesuai target uji pada pola data yang telah ditetapkan. Untuk mengetahui pasien prediksi sebenarnya, maka kita harus melakukan denormalisasi terhadap hasil prediksi dengan menggunakan persamaan berikut ini:

$$y_{denorm} = (0,5939 * 2858 - 0,5893 * 2150) + 2150 = 2570,502751$$

Jumlah pasien pada bulan mendatang adalah sebanyak 2570 orang. Hasil ini merupakan prediksi yang dihasilkan dari perhitungan diatas. Prediksi ini menyiratkan jumlah pasien rehabilitasi medis yang diperkirakan dalam kurun waktu satu bulan ke depan dan untuk melakukan prediksi 5 bulan kedepan maka dilakukan itersi sebanyak 5 kali.

## KESIMPULAN

Prediksi yang dihasilkan melalui perhitungan Jaringan Syaraf Tiruan (JST) menggunakan algoritma Backpropagation dengan arsitektur jaringan 3-10-1, kemudian diuji menggunakan aplikasi berbasis Python, menghasilkan estimasi jumlah pasien rehabilitasi medis untuk lima bulan ke depan. Hasil prediksi tersebut mencakup bulan Januari 2025 sebanyak 2547 pasien, Februari 2025 sebanyak 2506 pasien, Maret 2025 sebanyak 2463 pasien, April 2025 sebanyak 2482 pasien, dan Mei 2025 sebanyak 2495 pasien. Perkiraan ini memberikan gambaran mengenai jumlah kunjungan yang diperkirakan terjadi selama periode tersebut, sehingga dapat dijadikan acuan oleh penyedia layanan kesehatan dalam merancang strategi perencanaan serta distribusi sumber daya secara efektif. Dengan demikian, hasil prediksi ini berpotensi memberikan kontribusi

penting dalam mendukung pengambilan keputusan yang tepat dalam manajemen layanan kesehatan. Model JST dengan pola arsitektur 3-10-1 ini juga menunjukkan performa yang cukup baik, dengan tingkat akurasi prediksi sebesar 91,706% dan tingkat kesalahan (error) sebesar 8,294%.

#### **DAFTAR PUSTAKA**

- Mardianto, I., & Pratiwi, D. (2018). Sistem deteksi penyakit pengeroposan tulang dengan metode jaringan syaraf tiruan backpropagation dan representasi ciri dalam ruang eigen. *CommIT (Communication and Information Technology) Journal*, 2(1), 69–80.
- Setiawan, D., Putri, R. N., & Suryanita, R. (2019). Perbandingan algoritma genetika dan backpropagation pada aplikasi prediksi penyakit autoimun. [Artikel tidak dipublikasikan atau tidak ada nama jurnal].
- Widodo, W., Rachman, A., & Amelia, R. (2014). Jaringan syaraf tiruan prediksi penyakit demam berdarah dengan menggunakan metode backpropagation. [Artikel tidak dipublikasikan atau tidak ada nama jurnal].
- Kafil, M. (2019, November). Penerapan metode K-Nearest Neighbors untuk prediksi penjualan berbasis web pada Boutiq Dealove Bondowoso. [Artikel tidak dipublikasikan atau tidak ada nama jurnal].
- Lestari, D. A., & Ginting, B. S. (2020). Jaringan saraf tiruan untuk memprediksi jumlah pasien rawat jalan bagi pengguna narkoba menggunakan metode backpropagation (Studi kasus: Kantor BNN Kota Binjai). [Artikel tidak dipublikasikan atau tidak ada nama jurnal].
- Nisak, U. K., & Cholifah. (2020). Statistik di fasilitas pelayanankesehatan (Vol. I). UMSIDA Press.
- Salimu, S. A., & Yunus, Y. (2020, December). Prediksi tingkat kedatangan wisatawan asing menggunakan metode backpropagation (Studi kasus: Kepulauan Mentawai). *Jurnal Informatika Ekonomi dan Bisnis*, 2(4), 98–103. <https://doi.org/10.37034/infeb.v2i4.50>
- Nguyen, T. A., Ly, H. B., & Pham, B. T. (2020). Backpropagation neural network-based machine learning model for prediction of soil friction angle. *Mathematical Problems in Engineering*, 2020, Article ID 8845768. <https://doi.org/10.1155/2020/8845768>

- Ma, Z., & Wang, Y. (2022). Analysis and prediction of body test results based on improved backpropagation neural network algorithm. *Advances in Multimedia*, 2022, Article ID 1701687. <https://doi.org/10.1155/2022/1701687>
- Aulya, N. (2022, September). Prediksi kunjungan wisata Kota Payakumbuh menggunakan metode jaringan syaraf tiruan backpropagation. *Jurnal Informatika Ekonomi dan Bisnis*, 4(4). <https://doi.org/10.37034/infeb.v4i4.157>
- Syahfitri, D., Windarto, A. P., & Fauzan, M. (2020). Peningkatan nilai akurasi prediksi algoritma backpropagation (Kasus: Jumlah pengunjung tamu pada hotel berbintang di Sumatera Utara). [Artikel tidak dipublikasikan atau tidak ada nama jurnal].
- Dou, Y. (2021). An improved prediction model of IGBT junction temperature based on backpropagation neural network and Kalman filter. *Complexity*, 2021, Article ID 5542889. <https://doi.org/10.1155/2021/5542889>
- Zhang, Q., Yan, L., Hu, R., Li, Y., & Hou, L. (2022). Regional economic prediction model using backpropagation integrated with Bayesian vector neural network in big data analytics. *Computational Intelligence and Neuroscience*, 2022, Article ID 1438648. <https://doi.org/10.1155/2022/1438648>
- Liao, Y., Miao, Z., & Yang, C. (2021). Probabilistic prediction of unsafe event in air traffic control department based on the improved backpropagation neural network. *Mathematical Problems in Engineering*, 2021, Article ID 9982723. <https://doi.org/10.1155/2021/9982723>