

Analisis Diagnosa Rhinitis Alergi dan Rhinitis Vasomotor Menggunakan Jaringan Syaraf Tiruan Metode Backpropagation

Feri Setiawan¹, Lusiyaniti², Siti Julianita Siregar³

^{1,2,3} Sistem Informasi, STMIK Triguna Dharma

Email: ¹ferysetiawan13@gmail.com, ²lusiyaniti.tgd@gmail.com, ³siti.julianita18@gmail.com

Email Penulis Korespondensi: ferysetiawan13@gmail.com

Abstrak

Rhinitis adalah kondisi umum pada mukosa hidung yang ditandai dengan gejala seperti bersin, rinore, hidung tersumbat, dan rasa gatal, yang terbagi menjadi dua jenis utama, yaitu rhinitis alergi (RA) dan rhinitis vasomotor (RV). Meskipun keduanya memiliki gejala yang serupa, perbedaan patofisiologis di antara keduanya seringkali menyulitkan dalam proses diagnosa. Diagnosa yang tepat sangat penting untuk penanganan yang sesuai dan untuk menghindari komplikasi lebih lanjut. Jaringan Syaraf Tiruan (JST) dengan metode backpropagation telah terbukti efektif dalam membantu diagnosa berbagai penyakit, termasuk rhinitis, dengan mengenali pola kompleks dalam data klinis. Penelitian ini bertujuan untuk menganalisis diagnosa rhinitis alergi dan vasomotor menggunakan JST yang diimplementasikan melalui Microsoft Excel. Model JST dilatih menggunakan 20 epoch, dan hasil testing menunjukkan kemampuan model untuk mengklasifikasikan gejala dengan benar, menghasilkan 5 diagnosa untuk masing-masing jenis rhinitis. Penelitian ini menunjukkan bahwa penerapan metode backpropagation menggunakan Excel dapat meningkatkan akurasi dan efisiensi diagnosa rhinitis alergi dan vasomotor, serta memberikan alternatif yang terjangkau dan mudah diakses untuk mendukung deteksi dini kondisi ini.

Kata Kunci: Rhinitis Alergi, Rhinitis Vasomotor, Jaringan Syaraf Tiruan, Backpropagation, Diagnosa

Abstract

Rhinitis is a common condition of the nasal mucosa characterized by symptoms such as sneezing, rhinorrhea, nasal congestion, and itching. It is categorized into allergic rhinitis (AR) and vasomotor rhinitis (VR), both of which present with similar clinical symptoms, making accurate diagnosis challenging. Despite their distinct pathophysiological mechanisms, accurate diagnosis is essential for proper management and to prevent complications. This study aims to analyze the diagnosis of allergic rhinitis and vasomotor rhinitis using Artificial Neural Networks (ANN) with the backpropagation method, implemented through Microsoft Excel. The model was trained using 20 epochs, and the testing results demonstrated the model's ability to correctly classify the symptoms, yielding 5 diagnoses of allergic rhinitis and 5 diagnoses of vasomotor rhinitis. This research highlights that backpropagation-based ANN using Excel can enhance the accuracy and efficiency of rhinitis diagnosis, providing an accessible and cost-effective solution for early detection of these conditions.

Keywords: Allergic rhinitis, Vasomotor rhinitis, Artificial Neural Network, Backpropagation, Diagnosis

1. PENDAHULUAN

Rhinitis merupakan kondisi umum pada mukosa hidung yang ditandai dengan gejala bersin, rinore, hidung tersumbat, dan rasa gatal pada hidung [1]. Rhinitis terbagi menjadi beberapa jenis, di antaranya adalah rhinitis alergi (RA) dan rhinitis vasomotor (RV) [2]. Rhinitis alergi disebabkan oleh reaksi hipersensitivitas terhadap alergen seperti serbuk sari, debu, atau bulu hewan, sedangkan rhinitis vasomotor bersifat non-alergi dengan pemicu seperti perubahan suhu, kelembapan, bau menyengat, atau stres emosional [3]. Kedua kondisi ini sering kali memiliki gejala yang serupa, seperti hidung tersumbat, bersin dan rinore. Perbedaan utama antara RA dan RV terletak pada mekanisme patofisiologi, namun keduanya memiliki manifestasi klinis yang mirip, sehingga sering menyulitkan pasien dan tenaga medis dalam melakukan diagnosa awal hanya berdasarkan gejala klinis [4][5]. Diagnosa yang tepat sangat penting karena penanganan dan pengobatan kedua kondisi ini berbeda, serta untuk mencegah terjadinya komplikasi seperti sinusitis kronik dan gangguan kualitas hidup pasien. Namun, keterbatasan fasilitas kesehatan, waktu konsultasi yang terbatas, dan kesadaran masyarakat yang rendah terkait kondisi yang dialaminya menjadi kendala dalam diagnosa dini rhinitis.

Jaringan Syaraf Tiruan (JST) adalah pendekatan yang bertujuan meniru kemampuan sistem saraf biologis dalam memproses informasi [6]. Dalam beberapa tahun terakhir, penerapan teknologi kecerdasan buatan (AI), khususnya Jaringan Syaraf Tiruan (JST) dengan metode backpropagation, telah menunjukkan potensi besar dalam meningkatkan akurasi diagnosa berbagai penyakit, termasuk rhinitis. Ko J mengembangkan model deep learning untuk menganalisis gambar endoskopi hidung guna mendiagnosa rhinitis alergi, dengan akurasi mencapai 90,80% [7].

Seiring berkembangnya teknologi, kecerdasan buatan telah banyak dimanfaatkan dalam bidang kesehatan, termasuk untuk membantu dalam proses diagnosa penyakit secara cepat dan akurat [8]. Salah satu metode AI yang efektif untuk klasifikasi diagnosa penyakit adalah Jaringan Syaraf Tiruan (JST) dengan algoritma backpropagation karena memiliki kemampuan dalam mengenali pola kompleks dan mempelajari hubungan non-linear antar gejala untuk klasifikasi [9]. Penerapan JST dengan metode backpropagation telah digunakan untuk diagnosa berbagai penyakit, seperti klasifikasi

kanker payudara, diabetes, dan penyakit jantung, dengan tingkat akurasi yang tinggi [10]. Dalam beberapa tahun terakhir, pengembangan sistem pendukung keputusan berbasis JST pada aplikasi kesehatan semakin diminati karena dapat membantu pasien melakukan skrining mandiri serta membantu tenaga medis dalam proses pre-screening untuk diagnosa awal penyakit [11]. Pemanfaatan sistem diagnosa otomatis menggunakan JST pada kasus rhinitis alergi dan vasomotor masih terbatas, sehingga memerlukan penelitian lebih lanjut untuk membantu klasifikasi dini kedua kondisi ini berbasis gejala klinis yang dapat digunakan sebagai alat skrining pendahuluan.

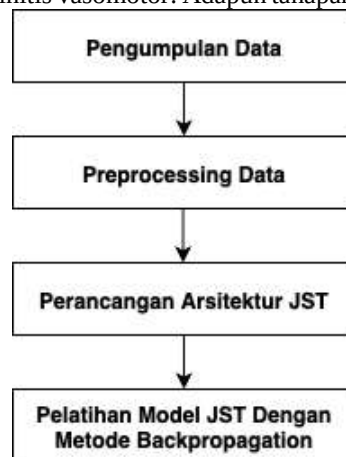
Penelitian ini bertujuan melakukan analisis diagnosa terhadap rhinitis alergi dan rhinitis vasomotor menggunakan Jaringan Syaraf Tiruan dengan metode backpropagation yang diimplementasikan melalui Microsoft Excel. Penelitian ini diharapkan mampu memberikan kontribusi penting dalam meningkatkan akurasi dan efisiensi diagnosa rhinitis dan membantu masyarakat maupun tenaga medis dalam proses deteksi awal kondisi rhinitis dengan lebih efisien dan akurat.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan metode penelitian yang terencana menjadi kerangka kerja untuk menganalisis diagnosa rhinitis alergi dan rhinitis vasomotor yang memiliki kesamaan gejala klinis sehingga membutuhkan pemrosesan data gejala secara presisi untuk proses klasifikasi diagnosa.

Dalam penelitian ini, tahapan metode penelitian akan memandu peneliti dalam mengembangkan model JST dengan metode backpropagation yang akan dilatih menggunakan data gejala pasien untuk klasifikasi diagnosa apakah menghasilkan diagnosa rhinitis alergi atau rhinitis vasomotor. Adapun tahapan metode penelitian ini seperti gambar berikut:



Gambar.1 Tahapan Metode Penelitian

2.1.1 Pengumpulan Data

Pada penelitian ini, pengumpulan data dilakukan dengan dua metode yaitu:

- Studi Pustaka**
Studi pustaka dilakukan untuk memperoleh data sekunder terkait gejala klinis rhinitis alergi dan rhinitis vasomotor yang relevan untuk dijadikan variabel input JST dan untuk mengetahui kriteria diagnosa berdasarkan panduan medis dari jurnal dan buku referensi.
- Wawancara**
Wawancara dilakukan untuk memperoleh data primer dari dokter spesialis THT terkait pengalaman klinis dalam membedakan rhinitis alergi dan vasomotor, serta gejala yang paling sering ditemukan pada pasien di lapangan. Wawancara juga dilakukan kepada pasien terkait gejala yang dialami, pemicu gejala dan riwayat diagnosa yang telah diberikan oleh dokter.

2.1.2 Preprocessing Data

Preprocessing data adalah tahapan pembersihan, transformasi, dan normalisasi data mentah sebelum digunakan dalam pelatihan model Jaringan Syaraf Tiruan (JST), agar model dapat belajar secara optimal dengan mengurangi noise, outlier, dan ketidakkonsistenan pada dataset [12].

Dalam penelitian ini, preprocessing data dilakukan dengan langkah-langkah berikut:

- Data Cleaning**
 - Memeriksa data yang hilang (missing values) dari hasil wawancara pasien, kuesioner, atau rekam medis.
 - Jika terdapat data kosong dengan kondisi dimana jika hanya 1 variabel hilang pada 1 sampel, dilakukan imputasi dengan modus/median dan jika lebih dari 30% variabel hilang pada satu sampel, sampel tersebut dikeluarkan dari dataset untuk menjaga kualitas data.

- Mengoreksi ketidakkonsistenan format (misalnya: “Ya” diubah menjadi 1, “Tidak” menjadi 0).
- b. Data Encoding
Data gejala dapat diubah menjadi format numerik seperti:
 - “Ya” → 1
 - “Tidak” → 0Jika terdapat variabel ordinal, seperti frekuensi bersin maka:
 - Tidak pernah = 0
 - Kadang = 0.5
 - Sering = 1
- c. Normalisasi Data
Karena JST sensitif terhadap skala data, normalisasi dilakukan agar data berada pada rentang 0–1 (Min-Max Normalization) menggunakan rumus:
$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$
Dimana:
 - X= nilai asli
 - X_{min} = nilai minimum pada variabel tersebut
 - X_{max} = nilai maksimum pada variabel tersebut
 - X' = nilai setelah normalisasi.
- d. Data Splitting
Dataset yang telah dibersihkan dan dinormalisasi akan dibagi menjadi:
 - Data latih (training data): 70%
 - Data uji (testing data): 30%

2.1.3 Perancangan Arsitektur JST

Perancangan arsitektur JST dilakukan untuk mengetahui analisis diagnosa dalam membedakan Rhinitis Alergi (RA) dan Rhinitis Vasomotor (RV) dengan menggunakan pola gejala pasien sebagai data input. Adapun tahapan dari perancangan arsitektur JST sebagai berikut:

- a. Rancangan sistem untuk menganalisis diagnosa rhinitis alergi dan rinitis vasomoor ini memiliki 7 input sebagai berikut:
 1. Seberapa sering Anda bersin dalam sehari? (x1)
 - a. Tidak bersin sama sekali → 0
 - b. Kadang - kadang 3-6 kali → 0,5
 - c. Sering lebih dari 6 kali → 1
 2. Apakah Anda mengalami hidung berair dengan cairan yang jernih (bukan kental/berwarna)? (x2)
 - a. Tidak → 0
 - b. Iya → 1
 3. Apakah Anda merasa hidung tersumbat atau sulit bernafas dari hidung?
 - a. Tidak → 0
 - b. Iya → 1
 4. Apakah mata Anda sering merasakan gatal atau ingin mengucek mata dan hidung?
 - a. Tidak → 0
 - b. Iya → 1
 5. Apakah dokter pernah mengatakan bahwa lapisan dalam hidung (mukosa) Anda mengalami pembengkakan dan warna nya pucat?
 - a. Tidak → 0
 - b. Iya → 1
 6. Apakah gejala pilek atau hidung tersumbat Anda muncul ketika udara dingin?
 - a. Tidak → 0
 - b. Iya → 1
 7. Apakah gejala hidung (bersin, pilek, hidung tersumbat) sering muncul saat Anda membersihkan rumah atau terkena debu?
 - a. Tidak → 0
 - b. Iya → 1
- b. Menentukan jumlah input neuron berdasarkan jumlah gejala klinis sebagai variabel input dimana terdapat 7 neuron seperti tabel dibawah ini:

Tabel 1. Input Layer

No	Variabel Input Gejala
1	Frekuensi Bersin
2	Hidung Berair Jernih
3	Hidung Tersumbat
4	Mata / Hidung Gatal
5	Mukosa Hidung
6	Pemicu Suhu
7	Pemicu Alergen Debu / Tungau

- c. Penentuan jumlah hidden layer, neuron dan bias dimana:
- Hidden layer 1 dimana 3 neuron, fungsi aktivitas sigmoid
 - Bias terdiri dari bias 1, bias 2, bias 3 dan bias Output
- Fungsi aktivasi sigmoid digunakan karena cocok untuk klasifikasi biner, serta dapat memperkenalkan non-linearitas pada jaringan untuk mengenali pola gejala pasien.
- d. Menentukan output layer dimana 1 neuron untuk klasifikasi biner seperti:
- 0 = Rhinitis Alergi
 - 1 = Rhinitis Vasomotor
- e. Menentukan parameter pelatihan seperti tabel berikut:

Tabel 2. Parameter Pelatihan

Parameter	Variabel Input Gejala
Learning Rate	0,5
Epoch	20
Loss Function	Mean Squared Error (MSE)
Optimizer	Stochastic Gradient Descent (SGD)
Fungsi Aktivitas	Sigmoid
Algoritma Pelatihan	Backpropagation

2.1.4 Pelatihan Model JST dengan Metode Backpropagation

Adapun tahapan pelatihan JST dengan metode backpropagation dalam menganalisis diagnosa Rhinitis Alergi (RA) dan Rhinitis Vasomotor (RV) sebagai berikut:

- Inisialisasi bobot
Bobot dan bias pada setiap neuron diinisialisasi secara random kecil (biasanya -1 hingga 1) agar JST dapat memulai proses pembelajaran.
- Forward propagation
Data input (gejala pasien) diproses layer per layer: Input layer → Hidden Layer 1 → Hidden Layer 2 → Output Layer.
Pada setiap layer: Nilai input dikalikan bobot, dijumlahkan, lalu dimasukkan ke fungsi aktivasi (sigmoid) untuk menghasilkan output pada neuron berikutnya [13].

Rumus:

$$y_j = f \left(\sum_{i=1}^n x_i \times W_{ij} + b_j \right)$$

Dimana:

- y_i = output neuron j,
 - x_i = input neuron i,
 - w_{ij} = bobot dari neuron i ke j,
 - b_j = bias neuron j,
 - f = fungsi aktivitas sigmoid.
- Backward propagation
Menghitung error output terhadap target label menggunakan fungsi loss:

$$E = \frac{1}{2} (target - output)^2$$

Gradient error dihitung dan digunakan untuk memperbarui bobot dengan metode backpropagation agar error dapat diminimalkan secara iteratif.

Pembaruan bobot:

$$\omega_{ij}^{new} = \omega_{ij}^{old} + \eta \times \delta_j + x_i$$

dimana:

- η = learning rate (misal: 0.01)
- δ_j = error neuron j.
- Iterasi epoch
Proses forward dan backward propagation diulang selama jumlah epoch (misal: 1000 iterasi) atau hingga nilai error loss mencapai nilai minimum yang ditentukan.

3. HASIL DAN PEMBAHASAN

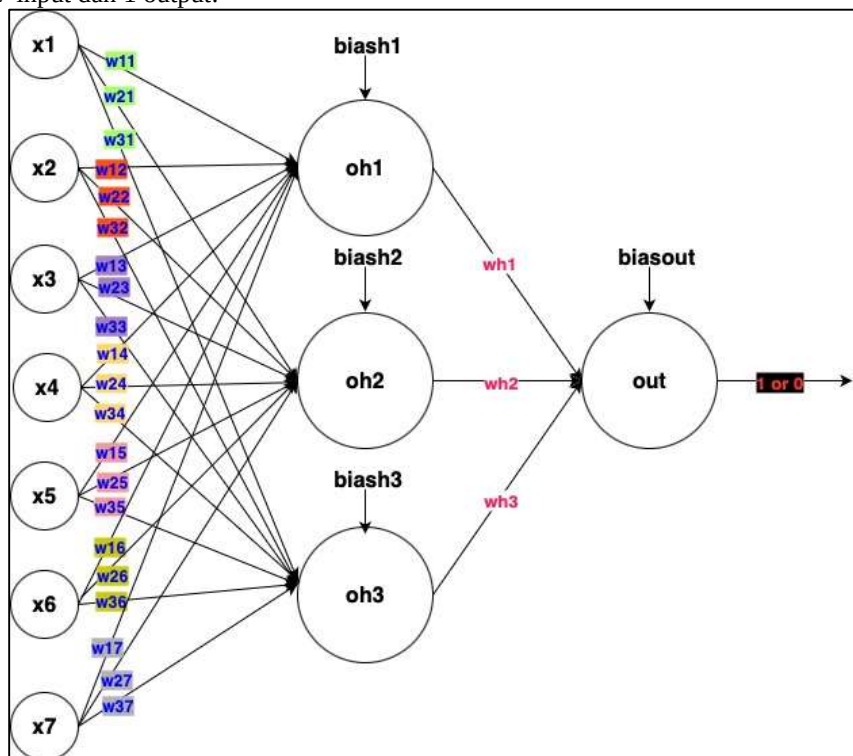
3.1 Analisis Hasil

Langkah pertama yang dilakukan adalah menentukan data input gejala. Dalam penelitian ini ada 10 pasien memiliki ciri-ciri data input gejala seperti tabel berikut:

Tabel 3. Data Input Gejala / Data Set

Data ke	Frekuensi Bersin (x1)	Hidung Berair jernih (x2)	Hidung Tersumbat (x3)	Mata Gatal/Hidung (x4)	Mukosa hidung (x5)	Pemicu Suhu (x6)	Pemicu alergen debu/tungau (x7)	Target
1	0.5	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0
5	1	1	1	0	0	1	1	1
6	1	1	1	1	0	1	1	1
7	1	1	0	0	1	0	0	0
8	0.5	0	1	0	0	1	0	0
9	0.5	0	0	1	1	0	0	1
10	1	0	0	0	0	0	0	0

Berdasarkan data input gejala/data set tersebut, berikut arsitektur jaringan analisis diagnosa rhinitis alergi dan rinitis vasomotor dengan 7 input dan 1 output:



Gambar 2. Arsitektur JST dengan 1 Hidden Layer

3.1.1 Forward Pass**Langkah 1: Menghitung setiap net pada hidden layer**

Dimana input data ke 1 terdiri dari: $x_1=0,5$; $x_2=1$; $x_3=1$; $x_4=1$; $x_5=1$; $x_6=1$; $x_7=1$; Target=1

Bobot awal hidden layer:

$w_{11} = 0,05$; $w_{12} = 0,1$; $w_{13} = 0,15$; $w_{14} = 0,2$; $w_{15} = 0,25$; $w_{16} = 0,12$; $w_{17} = 0,3$; bias1 = 0.25

$w_{21} = 0,3$; $w_{22} = 0,35$; $w_{23} = 0,4$; $w_{24} = 0,45$; $w_{25} = 0,24$; $w_{26} = 0,26$; $w_{27} = 0,3$; bias2 = 0.5

$w_{31} = 0,55$; $w_{32} = 0,6$; $w_{33} = 0,65$; $w_{34} = 0,7$; $w_{35} = 0,5$; $w_{36} = 0,56$; $w_{37} = 0,67$; bias3 = 0.75

$w_{41} =$

Bobot awal output layer:

$wh_1 = 0,8$; $wh_2 = 0,85$; $wh_3 = 0,9$

$neth_1 = w_{11} * x_1 + w_{12} * x_2 + w_{13} * x_3 + w_{14} * x_4 + w_{15} * x_5 + w_{16} * x_6 + w_{17} * x_7 + bias_1$

$neth_1 = 0,05 * 0,5 + 0,1 * 1 + 0,15 * 1 + 0,2 * 1 + 0,25 * 1 + 0,12 * 1 + 0,3 * 1 + 0,25$

$= 1,395$

$neth_2 = w_{21} * x_1 + w_{22} * x_2 + w_{23} * x_3 + w_{24} * x_4 + w_{25} * x_5 + w_{26} * x_6 + w_{27} * x_7 + bias_2$

$neth_2 = 0,3 * 0,5 + 0,35 * 1 + 0,4 * 1 + 0,45 * 1 + 0,24 * 1 + 0,26 * 1 + 0,3 * 1 + 0,5$

$= 2,65$

$neth_3 = w_{31} * x_1 + w_{32} * x_2 + w_{33} * x_3 + w_{34} * x_4 + w_{35} * x_5 + w_{36} * x_6 + w_{37} * x_7 + bias_3$

$neth_3 = 0,55 * 0,5 + 0,6 * 1 + 0,65 * 1 + 0,7 * 1 + 0,5 * 1 + 0,56 * 1 + 0,67 * 1 + 0,75$

$= 4,705$

Langkah 2: Menghitung setiap output pada hidden layer dengan menggunakan aktivasi sigmoid

$$oh_1 = \frac{1}{1 + e^{-neth_1}}$$

$$oh_1 = \frac{1}{1 + e^{-1,395}} = 0,801$$

$$oh_2 = \frac{1}{1 + e^{-neth_2}}$$

$$oh_2 = \frac{1}{1 + e^{-2,65}} = 0,934$$

$$oh_3 = \frac{1}{1 + e^{-neth_3}}$$

$$oh_3 = \frac{1}{1 + e^{-4,705}} = 0,991$$

Langkah 3: Menghitung net pada output layer

Diketahui bobot output layer dan biasout seperti berikut:

$wh_1 = 0,8$; $wh_2 = 0,85$; $wh_3 = 0,9$

biasout = 0,95

$netout = wh_1 * oh_1 + wh_2 * oh_2 + wh_3 * oh_3 + biasout$

$netout = 0,8 * 0,801 + 0,85 * 0,934 + 0,9 * 0,991 + 0,95$

$netout = 3,726$

Langkah 4: Menghitung output pada output layer dengan menggunakan fungsi aktivasi sigmoid

$$out = \frac{1}{1 + e^{-netout}}$$

$$out = \frac{1}{1 + e^{-3,726}} = 0,963$$

Langkah 5: Menghitung error pada output layer

$$E = (target - out)^2$$

$$E = (1 - 0,963)^2$$

$$E = 0,0013$$

3.1.2 Backward Pass**Langkah 6: Update bobot di output layer wh_1 , wh_2 , wh_3 dimana learning rate $\eta = 0,5$**

$$wh_1^+ = wh_1 - \eta * (-2 * (target - out)) * (out * (1 - out)) * oh_1$$

$$= 0,8 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,901$$

$$= 0,801$$

$$wh_2^+ = wh_2 - \eta * (-2 * (target - out)) * (out * (1 - out)) * oh_2$$

$$= 0,85 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,934$$

$$= \mathbf{0,851}$$

$$wh3^+ = wh3 - \eta * (-2 * (target - out)) * (out * (1 - out)) * oh1$$

$$= 0,9 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,991$$

$$= \mathbf{0,901}$$

Langkah 7: Update bias di output layer dimana learning rate $\eta = 0,5$

$$biasout^+ = biasout - \eta * (-2 * (target - out)) * (out * (1 - out))$$

$$= 0,95 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963))$$

$$= \mathbf{0,9512}$$

Langkah 8: Update bobot di hidden layer, misalkan update bobot w11, w12, w13, w14, w15, w16, w17 dimana learning rate $\eta = 0,5$

$$w11^+ = w11 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x1)$$

$$= 0,05 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (0,5)$$

$$= \mathbf{0,05008}$$

$$w12^+ = w12 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x2)$$

$$= 0,1 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (1)$$

$$= \mathbf{0,1001}$$

$$w13^+ = w13 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x3)$$

$$= 0,15 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (1)$$

$$= \mathbf{0,1501}$$

$$w14^+ = w14 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x4)$$

$$= 0,2 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (1)$$

$$= \mathbf{0,02001}$$

$$w15^+ = w15 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x5)$$

$$= 0,25 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (1)$$

$$= \mathbf{0,2501}$$

$$w16^+ = w16 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x6)$$

$$= 0,12 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (1)$$

$$= \mathbf{0,1201}$$

$$w17^+ = w17 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) * (x7)$$

$$= 0,3 - 0,5 * (-2 * (1 - 0,963)) * (0,963 * (1 - 0,963)) * 0,8 * (0,801 * (1 - 0,8)) * (1)$$

$$= \mathbf{0,3001}$$

$$w21^+ = w21 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh2 * (oh2 * (1 - oh2)) * (x1)$$

$$= 0,3 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (0,5)$$

$$= \mathbf{0,030003}$$

$$w_{22}^+ = w_{22} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_2 * (oh_2 * (1 - oh_2)) * (x_2)$$

$$= 0,35 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1)$$

$$= \mathbf{0,35006}$$

$$w_{23}^+ = w_{23} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_2 * (oh_2 * (1 - oh_2)) * (x_3)$$

$$= 0,4 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1)$$

$$= \mathbf{0,40006}$$

$$w_{24}^+ = w_{24} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_2 * (oh_2 * (1 - oh_2)) * (x_4)$$

$$= 0,45 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1)$$

$$= \mathbf{0,45006}$$

$$w_{25}^+ = w_{25} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_2 * (oh_2 * (1 - oh_2)) * (x_5)$$

$$= 0,24 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1)$$

$$= \mathbf{0,24006}$$

$$w_{26}^+ = w_{26} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_2 * (oh_2 * (1 - oh_2)) * (x_6)$$

$$= 0,26 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1)$$

$$= \mathbf{0,26006}$$

$$w_{27}^+ = w_{27} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_2 * (oh_2 * (1 - oh_2)) * (x_7)$$

$$= 0,3 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1)$$

$$= \mathbf{0,30006}$$

$$w_{31}^+ = w_{31} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_3 * (oh_3 * (1 - oh_3)) * (x_1)$$

$$= 0,55 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,9 * (1-0,991)) * (0,5)$$

$$= \mathbf{0,550005}$$

$$w_{22}^+ = w_{32} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_3 * (oh_3 * (1 - oh_3)) * (x_2)$$

$$= 0,6 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,9 * (1-0,991)) * (1)$$

$$= \mathbf{0,60001}$$

$$w_{33}^+ = w_{33} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_3 * (oh_3 * (1 - oh_3)) * (x_3)$$

$$= 0,65 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,9 * (1-0,991)) * (1)$$

$$= \mathbf{0,650001}$$

$$w_{34}^+ = w_{34} - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh_3 * (oh_3 * (1 - oh_3)) * (x_4)$$

$$= 0,7 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,9 * (1-0,991)) * (1)$$

$$= \mathbf{0,70001}$$

$$\begin{aligned}
 w35^+ &= w35 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh5 * (oh5 * (1 - oh5)) * (x5) \\
 &= 0,5 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,9 * (1-0,991)) * (1) \\
 &= \mathbf{0,50001}
 \end{aligned}$$

$$\begin{aligned}
 w36^+ &= w36 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh6 * (oh6 * (1 - oh6)) * (x6) \\
 &= 0,56 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,934 * (1-0,934)) * (1) \\
 &= \mathbf{0,56001}
 \end{aligned}$$

$$\begin{aligned}
 w37^+ &= w37 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh3 * (oh3 * (1 - oh3)) * (x7) \\
 &= 0,67 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,85 * (0,9 * (1-0,991)) * (1) \\
 &= \mathbf{0,670001}
 \end{aligned}$$

Langkah 9: Update bias di hidden layer, misalkan update bobot pada bias1 dimana learning rate $\eta = 0,5$

$$\begin{aligned}
 biash1^+ &= biash1 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh1 * (oh1 * (1 - oh1)) \\
 &= 0,25 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,9 * (0,8 * (0,801 * (1-0,801))) \\
 &= \mathbf{0,2501}
 \end{aligned}$$

$$\begin{aligned}
 biash2^+ &= biash2 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh2 * (oh2 * (1 - oh2)) \\
 &= 0,5 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,9 * (0,85 * (0,934 * (1-0,934))) \\
 &= \mathbf{0,50006}
 \end{aligned}$$

$$\begin{aligned}
 biash3^+ &= bias3 - \eta * (-2 * (target - out)) * (out * (1 - out)) * wh3 * (oh3 * (1 - oh3)) \\
 &= 0,75 - 0,5 * (-2 * (1-0,963)) * (0,963 * (1-0,963)) * 0,9 * (0,991 * (0,9 * (1-0,9))) \\
 &= \mathbf{0,75001}
 \end{aligned}$$

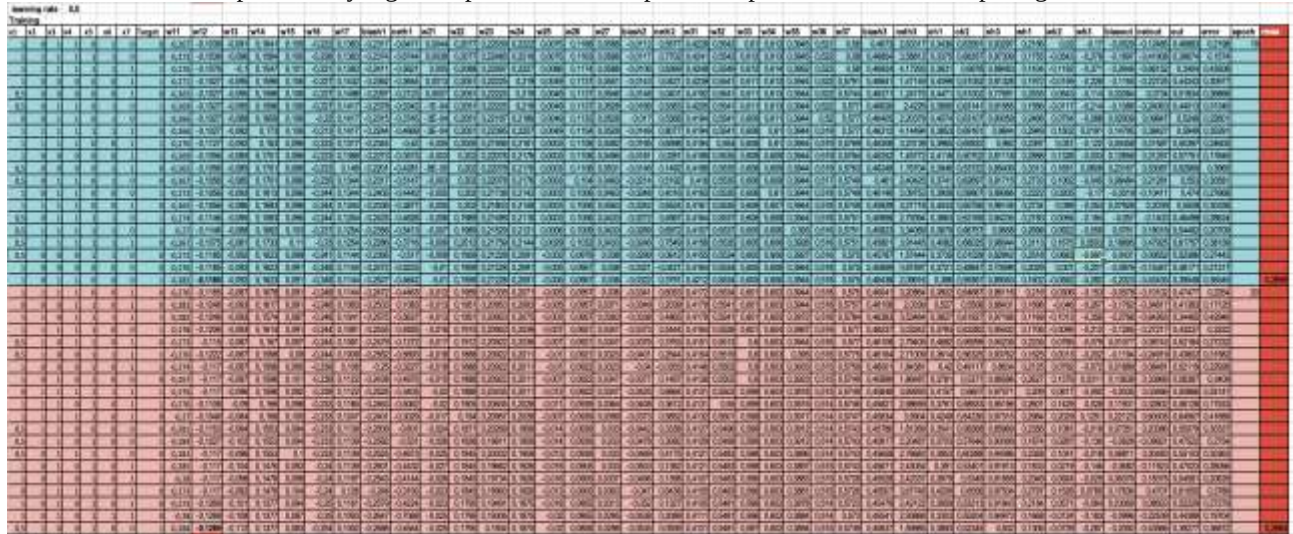
Perhitungan ini terus berlanjut hingga input data terakhir. Hal ini disebut bahwa telah menyelesaikan 1 epoch. Pada analisis rhinitis alergi dan rinitis vasomotor ini, perhitungan dilakukan sebanyak 20 epoch agar error semakin mengecil dan prediksi semakin akurat. Setelah model selesai dilatih sampai epoch ke 20, maka dilakukan evaluasi menggunakan data testing dengan 10 input data gejala yang sudah ditentukan. Adapun hasil dari perhitungan analisis diagnosa rhinitis alergi dan rhinitis vasomotor seperti tabel dibawah ini:

Tabel 4. Hasil Perhitungan

Date ke	Output JST (Error)	Hasil Diagnosa
1	0,598301051944104	Rhinitis Alergi
2	0,729531943556264	Rhinitis Alergi
3	0,280455150057877	Rhinitis Vasomotor
4	0,121735898314537	Rhinitis Vasomotor
5	0,596145568005725	Rhinitis Alergi
6	0,755433299774431	Rhinitis Alergi
7	0,0311673320394343	Rhinitis Vasomotor
8	0,177498907801729	Rhinitis Vasomotor
9	0,542424388224808	Rhinitis Alergi
10	0,0541557091121891	Rhinitis Vasomotor

3.2 Implementasi Menggunakan Microsoft Excel

Dari hasil metode analisis rhinitis alergi dan rinitis vasomotor, dilakukan implementasi perhitungan menggunakan Microsoft Excel. Adapun data yang ditampilkan dari hasil prediksi epoch ke 19 dan 20, seperti gambar berikut:



Gambar 3. Implentasi Hasil Prediksi Epoch 19 dan 20

Dari tabel diatas akan didapatkan perubahan bobot dan bias. Bobot dan bias yang baru akan digunakan untuk pengujian dari 10 data input gejala yang telah ditentukan. Adapun hasilnya menggunakan implentasi microsoft excel adalah sebagai berikut:



Gambar 4. Hasil Implementasi Testing 10 Data Uji

4. KESIMPULAN

Dari hasil implementasi excel JST dengan metode Backpropagation menggunakan 20 epoch untuk melatih model dapat memprediksi sesuai dengan target dimana RMSE semakin kecil dari epoch ke 1-20. Hasil testing didapatkan pada 10 data input gejala dengan memberikan hasil 5 diagnosa rhinitis alergi dan 5 diagnosa rhinitis vasomotor. Analisis ini menunjukkan bahwa metode backpropagation memiliki kemampuan untuk mengidentifikasi kedua jenis dengan benar.

DAFTAR PUSTAKA

- [1] A. NurJumala, N. A. Prasetyo, and H. W. Utomo, "Sistem Pakar Diagnosis Penyakit Rhinitis Menggunakan Metode Forward Chaining Berbasis Web," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 1, pp. 69–78, Feb. 2022, doi: 10.30865/jurikom.v9i1.3791.
- [2] A. E. Kornblith *et al.*, "Predictability and stability testing to assess clinical decision instrument performance for children after blunt torso trauma," *PLOS Digital Health*, vol. 1, no. 8, Aug. 2022, doi: 10.1371/journal.pdig.0000076.
- [3] S. M. Kim and C. O. Park, "Does Monthly Dupilumab Therapy Maintain its Clinical Efficacy in Moderate-to-Severe Atopic Dermatitis?," *Allergy Asthma Immunol Res*, vol. 13, no. 5, p. 681, 2021, doi: 10.4168/aaair.2021.13.5.681.
- [4] S. M. Nur Husna, H. T. T. Tan, N. Md Shukri, N. S. Mohd Ashari, and K. K. Wong, "Allergic Rhinitis: A Clinical and Pathophysiological Overview," Apr. 07, 2022, *Frontiers Media S.A.* doi: 10.3389/fmed.2022.874114.
- [5] D. Ananda Safira, A. Nabila Fauziani, A. Aushafia Nisa, P. Ristyaning Ayu Sangging, and R. Himayani, "Rhinitis Vasomotor: Etiologi, Prevalensi, Faktor Resiko, dan Tatalaksana," 2023.
- [6] S. J. Siregar, A. I. Lubis, and E. F. Ginting, "Penerapan Neural Network Dalam Klasifikasi Citra Permainan Batu Kertas Gunting dengan Probabilistic Neural Network," *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 3, pp. 420–425, Dec. 2021, doi: 10.47065/bits.v3i3.1143.
- [7] J. Ko, M. H. Kang, and Y. J. Jun, "Deep learning-based allergic rhinitis diagnosis using nasal endoscopy images," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-10087-x.

- [8] Y. Edwar, J. Sanoto, and S. Artikel, “Jaringan Syaraf Tiruan Mendeteksi Penyakit Pneumonia Infeksi Saluran Pernafasan Akut Dengan Algoritma Backpropagation,” *INNOVATION IN RESEARCH OF INFORMATICS*, vol. 4, no. 2, pp. 42–49, 2021, [Online]. Available: <http://innovatics.unsil.ac.id>
- [9] D. Andini, “Jaringan Syaraf Tiruan Untuk Klasifikasi Penyakit Demam Menggunakan Algoritma Backpropagation,” *Bulletin of Artificial Intelligence*, vol. 2, no. 1, pp. 86–99, 2023, [Online]. Available: <https://journal.grahamitra.id/index.php/buai>
- [10] Leema N., K. H. Nehemiah, Elgin Christo V. R., and Kannan A., “Evaluation of Parameter Settings for Training Neural Networks Using Backpropagation Algorithms,” *International Journal of Operations Research and Information Systems*, vol. 11, no. 4, pp. 62–85, Oct. 2020, doi: 10.4018/IJORIS.2020100104.
- [11] C. Permatasari *et al.*, “Klasifikasi Penyakit Demam Menggunakan Jaringan Syaraf Tiruan Algoritma Backpropagation,” 2021.
- [12] F. Kusuma Wardhana, N. Sumaya Jati, B. Radityo Seto, F. Mustika Wati, Y. Ayu Yuliana, and R. Kusumastuti, “PENERAPAN BACKPROPAGATION JARINGAN SARAF TIRUAN UNTUK PREDIKSI DIABETES MENGGUNAKAN DATASET PIMA INDIANS,” *SEMINAR NASIONAL AMIKOM SURAKARTA (SEMNAS)*, pp. 331–344, 2024.
- [13] A. P. Windarto *et al.*, “Jaringan Saraf Tiruan Algoritma Prediksi & Implementasi”.