

Perbandingan Fungsi Aktivasi Terhadap Kinerja Algoritma Neural Network Pada Klasifikasi Data Diabetes

A.Amirul Asnan Cirua*¹, Cindi², Sugiarto Cokrowibowo³

^{1,2,3}Program Studi Teknik Informatika, Universitas Sulawesi Barat

E-mail: *¹amirulasnancirua@unsulbar.ac.id, ²cindi@gmail.com,

³sugiarto.cokrowibowo@unsulbar.ac.id

Abstrak

Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja tiga fungsi aktivasi, yaitu Sigmoid biner, Sigmoid bipolar, dan TanH, dalam algoritma Backpropagation untuk klasifikasi data diabetes. Dataset yang digunakan terdiri dari 70.692 entri dengan 18 atribut, di mana 17 atribut berperan sebagai variabel input dan 1 atribut sebagai target klasifikasi (label diabetes). Evaluasi kinerja dilakukan dengan rasio pembagian data pelatihan dan pengujian sebesar 90:10. Hasil eksperimen menunjukkan bahwa fungsi aktivasi Sigmoid biner dengan arsitektur jaringan 17-5-1 menghasilkan akurasi tertinggi sebesar 76,22%, dengan nilai precision sebesar 83.07% untuk kelas 0 dan 71.63% untuk kelas 1, serta recall sebesar 66.23% untuk kelas 0. Sementara itu, fungsi aktivasi Sigmoid bipolar dengan arsitektur 17-15-1 menghasilkan akurasi sebesar 76.02%, dengan precision masing-masing sebesar 80.97% (kelas 0) dan 72.42% (kelas 1), serta recall sebesar 68.08% dan 83.97% untuk kelas 0 dan 1 secara berurutan. Fungsi aktivasi TanH dengan arsitektur yang sama juga menghasilkan akurasi sebesar 76.02%, dengan nilai precision dan recall yang identik dengan fungsi aktivasi Sigmoid bipolar. Berdasarkan hasil tersebut, dapat disimpulkan bahwa fungsi aktivasi Sigmoid biner memberikan performa akurasi terbaik dalam klasifikasi data diabetes menggunakan algoritma Backpropagation.

Kata kunci—diabetes, backpropagation, sigmoid biner, sigmoid bipolar, tanh

Abstract

This study aims to analyze and compare the performance of three activation functions Binary Sigmoid, Bipolar Sigmoid, and TanH within the Backpropagation algorithm for diabetes data classification. The dataset used consists of 70.692 entries with 18 attributes, of which 17 serve as input variables and one as the target classification attribute (diabetes label). Performance evaluation was conducted using a training and testing data split ratio of 90:10. Experimental results show that the Binary Sigmoid activation function, with a network architecture of 17-5-1, achieved the highest accuracy of 76.22%, with precision values of 83.07% for class 0 and 71.63% for class 1, and a recall of 66.23% for class 0. Meanwhile, the Bipolar Sigmoid function using a 17-15-1 architecture yielded an accuracy of 76.02%, with precision scores of 80.97% (class 0) and 72.42% (class 1), and recall values of 68.08% and 83.97% for class 0 and class 1, respectively. The TanH activation function, also implemented with a 17-15-1 architecture, produced an identical accuracy of 76.02%, with precision and recall metrics

matching those of the Bipolar Sigmoid function. Based on these results, it can be concluded that the Binary Sigmoid activation function provides the best classification performance for the diabetes dataset using the backpropagation algorithm.

Keywords— *diabetes, backpropagation, sigmoid biner, sigmoid bipolar, tanh*

1. PENDAHULUAN

Artificial Neural network (ANN) adalah sebuah sistem yang terdiri atas sekelompok unit pemroses yang dimodelkan untuk pemrosesan informasi yang meniru cara kerja sistem syaraf biologis seperti jaringan syaraf manusia. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisir *neuron* sehingga mampu mengenali pola secara efektif. *Backpropagation* merupakan salah satu algoritma pembelajaran dalam *Artificial Neural network*. ANN terdiri dari beberapa lapisan *neuron* yang saling berhubungan yang dapat belajar dari data untuk melakukan tugas seperti klasifikasi, regresi, pengenalan pola, dan prediksi[1]. Jaringan Syaraf Tiruan tersusun dari beberapa *neuron* yang saling berhubungan. *Neuron* tersebut akan mentransmisikan informasi yang diterima, menuju *neuron-neuron* yang lain. *Neuron* yang terdapat pada jaringan syaraf tiruan akan dikumpulkan pada suatu layer dan layer tersebut akan dihubungkan dengan layer-layer sebelum dan sesudahnya. Umumnya, layer pada Jaringan Syaraf Tiruan tersusun dari *Input Layer*, *Hidden layer*, dan *Output Layer*[2]. *Backpropagation* adalah sebuah metode sistematis jaringan syaraf tiruan yang menggunakan metode pembelajaran terawasi (*supervised learning*) dan banyak lapisan untuk mengubah bobot-bobot yang ada pada lapisan tersembunyi. *Backpropagation* merupakan metode pelatihan di mana sebagian dari dataset latihan digunakan sebagai input untuk jaringan, dan kemudian jaringan menghitung outputnya. Saat ada perbedaan antara hasil yang diinginkan dan nilai output yang dihasilkan, nilai kesalahan diperhitungkan. Dalam situasi ini, bobot dalam jaringan diperbarui untuk mengurangi kesalahan tersebut[3]. Metode pelatihan merupakan proses latihan mengenali data dan menyimpan pengetahuan atau informasi yang didapat ke dalam bobot-bobot[4].

Salah satu komponen penting dalam membangun model *Artificial Neural network* dengan menggunakan algoritma *Backpropagation* adalah fungsi aktivasi. Fungsi aktivasi ini memiliki peranan sangat penting dalam suatu jaringan syaraf tiruan dimana penggunaannya tergantung sesuai kebutuhan dan target yang diinginkan serta fungsi aktivasi ini yang akan menentukan besarnya bobot. Fungsi aktivasi berperan sebagai sinyal untuk menentukan output ke beberapa *neuron* lainnya. Beberapa fungsi aktivasi yang sering digunakan yaitu *sigmoid biner*, *sigmoid bipolar*, logistik dan juga *Hyperbolic Tangent Function* (Tanh)[6]. Fungsi aktivasi dalam algoritma *Backpropagation* berperan sebagai sinyal untuk menentukan keluaran ke beberapa *neuron* lainnya. Fungsi aktivasi ini akan menentukan besarnya bobot dengan menggambarkan hubungan antara tingkat aktivasi internal (*summation function*) yang mungkin berbentuk linear atau non-linear. Fungsi aktivasi akan menentukan apakah sinyal dari input *neuron* akan diteruskan atau tidak, sehingga fungsi aktivasi berfungsi untuk menentukan apakah *neuron* tersebut harus “aktif” atau tidak berdasarkan dari *weighted sum* dari input. Inputan ini akan diproses melalui suatu fungsi perambatan. Fungsi aktivasi pada algoritma *Backpropagation* harus mempunyai beberapa karakteristik penting, yaitu kontinu, dapat dibedakan, dan tidak meningkat secara monoton.

Hyperparameter pada penelitian ini juga dilakukan tuning parameter untuk mencari kombinasi parameter terbaik seperti *learning rate*. *Learning rate* adalah sebuah pengaturan dalam pelatihan jaringan syaraf yang membantu menentukan seberapa besar penyesuaian yang dilakukan pada bobot selama proses belajar. Nilai *learning rate* ini berada pada *range* nol 0 sampai 1. Jika nilai ini rendah, proses pelatihan jaringan akan berlangsung lebih pelan dan membutuhkan waktu lebih lama untuk menghasilkan hasil yang optimal. Namun jika nilai *learning rate* yang tinggi dapat mempercepat pelatihan, tapi ada kemungkinan mengurangi

tingkat akurasi hasil yang didapat[7]. Selain itu, *num of epoch* adalah hyperparameter yang menentukan seberapa banyak kali algoritma pembelajaran akan bekerja pada keseluruhan dataset pelatihan[8].

Penelitian ini menguji fungsi aktivasi pada algoritma *Backpropagation* dengan variabel uji berupa dataset yang berasal dari website *open dataset kaggle* dengan nama *diabetes_data.csv* yang terdiri dari 18 atribut dan 70.692 record. Dengan metode validasi menggunakan *confusion matrix* sebagai salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya[9].

2. METODE

Penelitian ini menggunakan metode eksperimen sebagai pendekatan utama untuk mencapai tujuan penelitian. Metode eksperimen dipilih untuk memberikan wawasan yang mendalam tentang dampak variabel independen terhadap variabel dependen dalam konteks klasifikasi data diabetes.

2.1 Pengumpulan Data

Tahapan awal yaitu melakukan pengumpulan data, termasuk mengumpulkan referensi atau landasan teori dari penelitian sebelumnya yang berkaitan dengan topik yang sedang diteliti. Dataset yang digunakan berasal dari website *open dataset kaggle* dengan nama *diabetes_data.csv*. Detail atribut terjabarkan pada tabel 1.

Tabel 1. *Detail Atribut Dataset*

Atribut	Keterangan
<i>Age</i>	: Atribut ini menunjukkan usia seseorang
<i>Sex</i>	: Atribut ini merupakan Jenis kelamin seseorang, dengan nilai 1 untuk laki-laki dan 0 untuk perempuan.
<i>HighChol</i>	: Atribut ini menunjukkan apakah seseorang memiliki kadar kolesterol tinggi (0 untuk tidak, 1 untuk ya).
<i>CholCheck</i>	: Atribut ini menunjukkan apakah seseorang telah melakukan pemeriksaan kolesterol dalam 5 tahun terakhir (0 untuk tidak, 1 untuk ya).
<i>BMI</i>	: Indeks Masa Tubuh, mengukur proporsi berat badan terhadap tinggi badan
<i>Smoker</i>	: Atribut ini menunjukkan apakah seseorang telah merokok setidaknya 100 batang rokok dalam hidupnya (0 untuk tidak, 1 untuk ya)
<i>HeartDiseaseor Attack</i>	: Atribut ini menunjukkan apakah seseorang memiliki penyakit jantung koroner atau serangan jantung (0 untuk tidak, 1 untuk ya).
<i>PhysActivity</i>	: Atribut Tingkat aktivitas fisik dalam 30 hari terakhir, tidak termasuk pekerjaan (0 untuk tidak, 1 untuk ya).
<i>Fruits</i>	: Atribut ini menunjukkan apakah seseorang mengonsumsi buah setidaknya satu kali atau lebih per hari (0 untuk tidak, 1 untuk ya)
<i>Veggies</i>	: Atribut ini menunjukkan apakah seseorang mengonsumsi sayuran setidaknya satu kali atau lebih per hari (0 untuk tidak, 1 untuk ya).
<i>HvyAlcoholConsump</i>	: Atribut ini menunjukkan apakah seseorang mengonsumsi alkohol secara berat (0 untuk tidak, 1 untuk ya)

<i>GenHlth</i>	: Atribut ini mengukur persepsi umum seseorang tentang kesehatannya, dengan skala dari 1 hingga 5, di mana 1 adalah sangat baik dan 5 adalah buruk.
<i>MentHlth</i>	: Atribut ini mengukur jumlah hari dalam sebulan dimana seseorang mengalami masalah kesehatan mental, dengan skala dari 1 hingga 30 hari.
<i>PhysHlth</i>	: Atribut ini mengukur jumlah hari dalam sebulan dimana seseorang mengalami penyakit fisik atau cedera, dengan skala dari 1 hingga 30 hari.
<i>DiffWalk</i>	: Atribut ini Menunjukkan apakah seseorang memiliki kesulitan serius dalam berjalan atau naik tangga (0 untuk tidak, 1 untuk ya).
<i>Stroke</i>	: atribut ini menunjukkan menunjukkan apakah seseorang pernah mengalami stroke (0 untuk tidak, 1 untuk ya).
<i>HighBP</i>	: Atribut ini menunjukkan apakah seseorang memiliki tekanan darah tinggi (0 untuk tidak, 1 untuk ya).
<i>Diabetes</i>	: Atribut ini menunjukkan apakah seseorang memiliki diabetes (0 untuk tidak, 1 untuk ya).

Pada tabel 1, terdiri dari 18 atribut dan dengan 70.692 record data, dimana 17 atribut berperan sebagai input dan atribut ke 18 adalah target klasifikasi dengan kelas 0 dan 1. Fungsi aktivasi yang akan diuji pada penelitian ini yaitu *sigmoid biner*, *sigmoid bipolar*, dan *Hyperbolic Tangent Function* (Tanh), setiap fungsi aktivasi tersebut memiliki rumus yang berbeda.

a) *Sigmoid biner*

fungsi *sigmoid biner*, yang memiliki rentang nilai antara 0 dan 1 seperti pada persamaan (1) Berikut[10]:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

dengan turunan

$$f'(x) = f(x)(1 - f(x)) \quad (2)$$

Fungsi sigmoid memiliki nilai maksimum = 1. Maka untuk pola yang targetnya lebih dari 1, pola masukan serta keluaran harus ditransformasi sebagai akibatnya seluruh polanya memiliki *range* yang sama seperti fungsi Sigmoid yang digunakan.

b) *Sigmoid bipolar*

Fungsi *sigmoid bipolar* hampir sama dengan fungsi *sigmoid biner*, yang membedakannya adalah nilai output jaringan dari fungsi ini terletak pada interval -1 sampai 1. Fungsi *sigmoid bipolar* dirumuskan sebagai berikut [11].

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (3)$$

dengan turunan

$$f'(x) = \frac{1}{2} (1 + f(x))(1 - f(x)) \quad (4)$$

c) Tanh

Hyperbolic tangent function atau yang sering disebut tanh pada umumnya lebih cepat mencapai konvergensi dibandingkan fungsi aktivasi sigmoid. Tanh memiliki nilai output yang terletak pada interval -1 sampai 1. Rentang nilai dari fungsi aktivasi tanh lebih luas dari fungsi aktivasi *sigmoid biner*. Fungsi aktivasi tanh dirumuskan sebagai berikut:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5)$$

dengan turunan

$$f'(x) = \frac{1}{2} (1 + f(x))(1 - f(x)) \quad (6)$$

2. 2 Implementasi Backpropagation

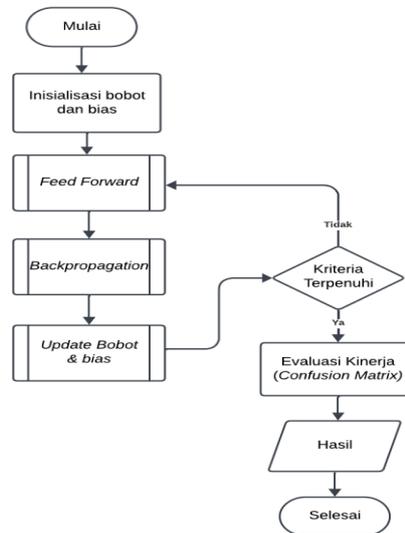
Pada tahapan ini, melakukan implementasi algoritma yang akan digunakan yaitu *backpropagation neural network* untuk melakukan klasifikasi data diabetes dengan membandingkan tiga fungsi aktivasi yang berbeda dengan menggunakan parameter parameter sebagai berikut :

- | | |
|--------------------------------------|-----------------------------------------------|
| a) Pembagian rasio data | : 70 :30, 80:20, 90:10 |
| b) <i>Learning rate</i> | : 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 |
| c) Fungsi aktivasi | : <i>Sigmoid biner, Sigmoid bipolar, Tanh</i> |
| d) Input layer | : 17 |
| e) Jumlah <i>neuron Hidden layer</i> | : 5,10, 15, 20 |
| f) Jumlah <i>neuron Output layer</i> | : 1 |
| g) <i>Max epoch</i> | : 500, 1000, 1500, 2000 |
| h) <i>Target error</i> | : 0.001 |

Proses implementasi diawali dengan proses training untuk melatih model kemudian model yang telah didapatkan akan dilakukan *testing*. *Training* jaringan *Backpropagation* pada dasarnya terdiri dari tiga tahapan [2] yaitu:

- Input nilai data pelatihan sehingga diperoleh nilai output (*Feedforward*).
- Propagasi balik dari nilai error yang diperoleh (*Backpropagation*).
- Penyesuaian bobot koneksi untuk meminimalkan nilai error.

Ketiga tahapan training tersebut diulangi terus-menerus sampai mendapatkan nilai target error.



Gambar 1. Alur proses algoritma *Backpropagation*

2. 2.1 Feedforward

Tiap-tiap input ($x_i, i = 1, 2, 3, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (Hidden layer). Kemudian setiap unit tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal input terbobot [12].

$$Z_{in_j} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (7)$$

Gunakan fungsi aktivasi untuk menghitung sinyal output-nya.

$$Z_j = f(Z_in_j) \quad (8)$$

Kemudian mengirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit- unit hidden). Langkah selanjutnya Tiap-tiap hidden unit ($y_k, k = 1,2,3, \dots, m$) menjumlahkan bobot sinyal input.

$$Z_in_k = W_{0k} + \sum_{i=1}^p Z_j W_k \quad (9)$$

Kemudian gunakan fungsi aktivasi untuk menghitung sinyal output-nya.

$$Y_k = f(Y_in_k) \quad (10)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit- unit output).

2. 2.2 Backpropagation

Tiap-tiap unit output ($y_k, k = 1,2,3, \dots, m$) menerima target pola yang berhubungan dengan pola input pembelajarannya, hitung informasi error-nya:

$$\delta_k = (t_k - y_k) f'(y_k) \quad (11)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai (w_{jk}))

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (12)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai (w_{0k}))

$$\Delta w_{0k} = \alpha \delta_k \quad (13)$$

Kirimkan δ_k ini ke unit-unit yang ada di lapisan bawahnya. Selanjutnya Tiap-tiap hidden unit ($z_j, j = 1,2,3, \dots, p$) menjumlahkan input-nya (dari unit-unit yang berada pada lapisan di atasnya)

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (14)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error

$$\delta_j = \delta_in_j f'(z_j) \quad (15)$$

Kemudian hitung koreksi bobot (digunakan untuk memperbaharui v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (16)$$

Hitung juga koreksi bias (digunakan untuk memperbaharui v_{0j})

$$\Delta v_{0j} = \alpha \delta_j \quad (17)$$

2. 2.3 Update bobot dan bias

Tiap-tiap unit output ($y_k, k = 1,2,3, \dots, m$) memperbaharui bias dan bobotnya ($j = 0,1,2,3, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (18)$$

Tiap-tiap hidden unit ($z_j, j = 1,2,3, \dots, p$) memperbaiki bias dan bobotnya ($i = 0,1,2,3, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (19)$$

cek kondisi untuk berhenti ketika sudah memenuhi jumlah epoch. Hasil dari proses training tersebut adalah model yang berupa matriks bobot yang menghubungkan setiap layer. Selanjutnya proses testing untuk menguji akurasi dari model[13].

2. 2.4 Testing Model Backpropagation

Setelah nilai output yang didapatkan dari proses pelatihan yang paling mendekati target, maka bobot dan bias dari pelatihan akan disimpan dan dilakukan proses pengujian. Adapun tahapan dari Algoritma *Backpropagation*[14] yaitu:

- Langkah 1: inialisasi bobot dan bias sesuai dengan bobot yang dihasilkan pada proses pelatihan.
- Langkah 2: setiap unit ($x_i, i = 1,2,3, \dots, n$) menyebarkan sinyal input pada seluruh hidden unit.
- Langkah 3: setiap unit ($z_j, j = 1,2,3, \dots, p$) akan menghitung sinyal-sinyal input dengan bobot dan biasnya.

$$Z_in_j = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (20)$$

- Langkah 4: Setiap unit output ($y_k, k = 1,2,3, \dots, m$) akan menghitung sinyal-sinyal dan hidden unit dengan bobot dan biasnya menggunakan rumus pada persamaan (14)
- Langkah 5: menggunakan fungsi aktivasi yang telah ditentukan memperoleh sinyal output dari unit output tersebut menggunakan rumus persamaan (15).

Berikut adalah keterangan rumus yang digunakan:

- z_inj : total sinyal masukan pada linjtasan j
- v_{0j} : nilai bobot bias input layer ke *Hidden layer*
- x_i : nilai input pada unit i
- v_{ij} : bobot tantara input unit i dan lapisan unit j
- y_ink : total sinyal masukan pada lapisan unit i
- w_{0k} : nilai bias pada *Hidden layer*
- z_j : nilai masukan pada lapisan j
- δ_k : faktor kesalahan pada unit keluaran k
- w_{jk} : bobot tantara lapisan unit j dan keluaran unit k
- tk : target data
- y_k : keluaran pada keluaran unit k
- α : *Learning rate*

2. 3. Evaluasi

Dalam klasifikasi biner, *confusion matrix* menggambarkan performa model dengan membagi hasil klasifikasi menjadi empat kategori[15] seperti pada Tabel 2 berikut:

Tabel 2. *Confussion matrix*

		Kelas Aktual	
		1	0
Kelas Prediksi	1	TP (True Positive)	FP (False Positive)
	0	FN (False Negative)	TN (True Negative)

- True Positive* (TP) adalah jumlah objek dari kelas positif yang diklasifikasikan dengan benar
- False Negative* (FN) adalah jumlah objek dari kelas positif yang salah diklasifikasikan

- sebagai kelas negatif
- c) *False Positive* (FP) adalah jumlah objek dari kelas negatif yang salah diklasifikasikan sebagai kelas positif
 - d) *True Negative* (TN) adalah jumlah objek dari kelas negatif yang diklasifikasikan dengan benar.

Beberapa ukuran evaluasi kinerja klasifikasi dapat dihitung berdasarkan *confusion matrix*, antara lain *accuracy*, *precision* dan *recall*. Nilai akurasi menggambarkan seberapa akurat dan efektifitas sistem secara keseluruhan.

a) *Accuracy*

Nilai akurasi menggambarkan seberapa akurat dan efektifitas sistem secara keseluruhan dalam mengklasifikasikan data secara benar. Perhitungan akurasi dinyatakan dalam persamaan berikut :

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (21)$$

b) *Precision*

Precision adalah ukuran ketepatan dari proses klasifikasi atau proporsi klasifikasi *Positive* dari hasil prediksi yang benar terhadap keseluruhan hasil prediksi yang bernilai *Positive*. Persamaan dari nilai *precision* adalah sebagai berikut:

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (22)$$

c) *Recall*

Recall merupakan ukuran pada *completeness* pada proses klasifikasi *positive* dari hasil prediksi yang benar (*true*) terhadap keseluruhan kelas aktual bernilai *positive*. Persamaan dari nilai *recall* adalah sebagai berikut.

$$recall = \frac{TP}{TP + FN} \times 100\% \quad (23)$$

3. HASIL DAN PEMBAHASAN

Pengujian dengan algoritma *Backpropagation neural network* pada data Diabetes yang berjumlah 70.692 *record*. Adapun *Hyperparameter* yang akan digunakan dalam pengujian ini yaitu jumlah *neuron* pada lapisan tersembunyi (*neuron in Hidden layer*), fungsi aktivasi (Aktivasi *Sigmoid biner*, *Sigmoid bipolar*, dan *Hyperbolic Tangent Function*), *learning rate*, *Max_epoch*, *Target Error* dapat dilihat pada Tabel 3.

Tabel 3. Hyperparameter Pengujian

Hyperparameter	
<i>Learning rate</i>	: 0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9
Fungsi aktivasi	: <i>Sigmoid biner</i> , <i>sigmoid bipolar</i> , <i>tanh</i>
<i>Hidden layer</i>	: 5, 10, 15, 20
<i>Max Epoch</i>	: 500,1000,1500,2000
<i>Target Error</i>	: 0.001

Pemilihan *hyperparameter* dalam pengujian model *Backpropagation Neural Network* (BPNN) bertujuan untuk menemukan konfigurasi yang menghasilkan kinerja terbaik pada dataset yang digunakan. *Learning rate* dipilih 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 untuk mengatur kecepatan pembelajaran, sementara fungsi aktivasi yang dibandingkan adalah *Sigmoid biner*, *Sigmoid bipolar*, dan *Tanh* untuk menentukan fungsi aktivasi dengan akurasi tertinggi. Setiap

fungsi aktivasi memiliki sifat unik yang memengaruhi cara jaringan mempelajari dan mengolah data. *Sigmoid biner* cocok untuk klasifikasi biner, *Sigmoid bipolar* membantu menyeimbangkan data, sementara Tanh, yang berpusat di sekitar nol, membantu dalam pelatihan jaringan yang lebih dalam. Fungsi aktivasi yang digunakan dalam *Backpropagation* neural network harus memiliki turunan, yang memungkinkan penggunaan algoritma *Backpropagation* untuk memperbarui bobot jaringan berdasarkan gradien kesalahan terhadap bobot. Jumlah *neuron* pada lapisan tersembunyi yang dipilih yaitu 5, 10, 15, dan 20 untuk mengeksplorasi kapasitas model, dan Max Epoch yang digunakan adalah 500, 1000, 1500, dan 2000 untuk memberikan cukup waktu dalam pelatihan. Target Error ditetapkan pada 0,001 untuk memastikan bahwa model mencapai tingkat kesalahan yang sangat rendah. Evaluasi model dilakukan menggunakan *Confusion Matrix*.

3. 1. Hyperparameter terbaik rasio data 70:30

Pengujian *Hyperparameter* pada rasio data 70:30 untuk fungsi aktivasi *sigmoid biner*, fungsi aktivasi *sigmoid bipolar* dan fungsi aktivasi tanh menggunakan jumlah *neuron* hidden layer sebanyak 5, 10, 15, dan 20, learning rate sebanyak 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan jumlah Max_epoch sebanyak 500,1000,1500, dan 2000. Adapun hasil pengujian terbaik pada fungsi aktivasi *sigmoid biner* dapat dilihat pada Tabel 4.

Tabel 4. *Hyperparameter* rasio 70:30

<i>Fungsi aktivasi</i>	<i>Neuron hidden layer</i>	<i>Learning rate</i>	<i>Max_epoch</i>	<i>Akurasi</i>
<i>Sigmoid Biner</i>	15	0.1	1000	75.45%
<i>Sigmoid Bipolar</i>	15	0.1	500	75.12%
Tanh	10	0.1	1500	75.15%

Berdasarkan hasil pengujian diatas didapatkan hasil akurasi terbaik untuk setiap fungsi aktivasi yaitu :

- Untuk *sigmoid biner* sebesar 75. 45% dengan precision untuk kelas 0 sebesar 76.43% dan kelas 1 sebesar 74.54%, untuk recall kelas 0 sebesar 73.59% dan kelas 1 sebesar 77.30%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000.
- Untuk fungsi aktivasi *sigmoid bipolar* sebesar 75.12% dengan precision untuk kelas 0 sebesar 75.17% dan kelas 1 sebesar 75.08% untuk recall kelas 0 sebesar 75.03% dan kelas 1 sebesar 75.22%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 15, learning rate 0.1, max_epoch 500.
- Untuk fungsi aktivasi Tanh sebesar 75.15% dengan precision untuk kelas 0 sebesar 75.19% dan kelas 1 sebesar 75.11% untuk recall kelas 0 sebesar 75.07% dan kelas 1 sebesar 75.23%.

3. 2. Hyperparameter terbaik rasio data 80:20

Pengujian *Hyperparameter* pada rasio data 80:20 untuk fungsi aktivasi *sigmoid biner*, fungsi aktivasi *sigmoid bipolar* dan fungsi aktivasi tanh menggunakan jumlah *neuron* hidden layer sebanyak 5, 10, 15, dan 20, learning rate sebanyak 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan jumlah Max_epoch sebanyak 500,1000,1500, dan 2000. Adapun hasil pengujian terbaik yaitu pada fungsi aktivasi *sigmoid biner* dapat dilihat pada Tabel 5.

Tabel 5. *Hyperparameter* rasio 80:20

<i>Fungsi aktivasi</i>	<i>Neuron hidden layer</i>	<i>Learning rate</i>	<i>Max_epoch</i>	<i>Akurasi</i>
------------------------	----------------------------	----------------------	------------------	----------------

<i>Sigmoid biner</i>	10	0.9	1500	76.11%
<i>Sigmoid bipolar</i>	10	0.1	500	75.27%
<i>Tanh</i>	10	0.1	1000	75.28%

Berdasarkan hasil pengujian diatas didapatkan hasil akurasi terbaik untuk setiap fungsi aktivasi yaitu

- Untuk *sigmoid biner* sebesar 76.11% dengan precision untuk kelas 0 sebesar 76.11% dan kelas 1 sebesar 72.59%, untuk recall kelas 0 sebesar 70.60% dan kelas 1 sebesar 77.84%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 10, learning rate 0.9, max_epoch 1500.
- Untuk fungsi aktivasi *sigmoid bipolar* sebesar 75.27% dengan precision untuk kelas 0 sebesar 76.62% dan kelas 1 sebesar 74.06% untuk recall kelas 0 sebesar 72.74% dan kelas 1 sebesar 77.81%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 10, learning rate 0.1, max_epoch 500.
- Untuk fungsi aktivasi *Tanh* sebesar 75.28% dengan precision untuk kelas 0 sebesar 76.66% dan kelas 1 sebesar 74.04% untuk recall kelas 0 sebesar 72.70% dan kelas 1 sebesar 77.86%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 10, learning rate 0.1, max_epoch 1000.

3. 3. Hyperparameter terbaik rasio data 90:10

Pengujian Hyperparameter pada rasio data 90:10 untuk fungsi aktivasi *sigmoid biner*, fungsi aktivasi *sigmoid bipolar* dan fungsi aktivasi *tanh* menggunakan jumlah *neuron* hidden layer sebanyak 5, 10, 15, dan 20, learning rate sebanyak 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan jumlah Max_epoch sebanyak 500,1000,1500, dan 2000. Adapun hasil pengujian terbaik yaitu pada fungsi aktivasi *sigmoid biner* dapat dilihat pada Tabel 6.

Tabel 6. *Hyperparameter* rasio 90:10

<i>Fungsi aktivasi</i>	<i>Neuron hidden layer</i>	<i>Learning rate</i>	<i>Max_epoch</i>	<i>Akurasi</i>
<i>Sigmoid biner</i>	5	0.1	2000	76.22%
<i>Sigmoid bipolar</i>	15	0.1	1000	76.02%
<i>Tanh</i>	15	0.1	1000	76.02%

Berdasarkan hasil pengujian diatas didapatkan hasil akurasi terbaik untuk setiap fungsi aktivasi yaitu :

- Untuk *sigmoid biner* sebesar 76.22% dengan precision untuk kelas 0 sebesar 81.90% dan kelas 1 sebesar 72.26%, untuk recall kelas 0 sebesar 67.33% dan kelas 1 sebesar 85.12%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 5, learning rate 0.1, max_epoch 2000.
- Untuk fungsi aktivasi *sigmoid bipolar* sebesar 76.01% dengan precision untuk kelas 0 sebesar 80.91% dan kelas 1 sebesar 72.45% untuk recall kelas 0 sebesar 68.09% dan kelas 1 sebesar 83.93%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000.
- Untuk fungsi aktivasi *Tanh* sebesar 76.01% dengan precision untuk kelas 0 sebesar 80.91% dan kelas 1 sebesar 72.45% untuk recall kelas 0 sebesar 68.09% dan kelas 1 sebesar 83.93%. Didapatkan dari pengujian menggunakan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000.

Berdasarkan hasil pengujian *hyperparameter* untuk setiap rasio data 70:30, 80:20, dan 90:10 didapatkan hasil terbaik untuk masing-masing fungsi aktivasi. Untuk fungsi aktivasi *sigmoid biner* akurasi tertinggi sebesar 76.22% didapatkan pada rasio data 90:10 dengan *neuron* hidden layer 5, learning rate 0.1, max_epoch 2000. Untuk fungsi aktivasi *sigmoid bipolar* akurasi

tertinggi sebesar 76.02% didapatkan pada rasio data 90:10 dengan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000. Untuk fungsi aktivasi Tanh akurasi tertinggi sebesar 76.02% didapatkan pada rasio data 90:10 dengan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000.

3. 4. Perbandingan akurasi setiap fungsi aktivasi

Arsitektur terbaik ini didapatkan berdasarkan hasil pengujian setiap fungsi aktivasi di semua rasio data yang digunakan Dimana rasio data yang digunakan adalah 70:30, 80:20, dan 90:10 dengan 3 fungsi aktivasi yaitu fungsi aktivasi *sigmoid biner*, fungsi aktivasi *sigmoid bipolar*, dan fungsi aktivasi Tanh.

Tabel 4. Arsitektur terbaik terhadap rasio data

Fungsi Aktivasi	Neuron hidden layer	Learning rate	Max_epoch	Akurasi
<i>Sigmoid biner</i>	5	0.1	2000	76.22%
<i>Sigmoid bipolar</i>	15	0.1	1000	76.02%
<i>Tanh</i>	15	0.1	1000	76.02%

Berdasarkan hasil pengujian hyperparameter untuk setiap rasio data 70:30, 80:20, dan 90:10 didapatkan hasil terbaik untuk masing-masing fungsi aktivasi. Untuk fungsi aktivasi *sigmoid biner* akurasi tertinggi sebesar 76.22% didapatkan pada rasio data 90:10 dengan *neuron* hidden layer 5, learning rate 0.1, max_epoch 2000. Untuk fungsi aktivasi *sigmoid bipolar* akurasi tertinggi sebesar 76.02% didapatkan pada rasio data 90:10 dengan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000. Untuk fungsi aktivasi Tanh akurasi tertinggi sebesar 76.02% didapatkan pada rasio data 90:10 dengan *neuron* hidden layer 15, learning rate 0.1, max_epoch 1000. Dari pengujian tersebut didapatkan hasil akurasi tertinggi untuk semua fungsi aktivasi didapatkan pada rasio data 90:10

4. KESIMPULAN

Berdasarkan tahapan yang dilakukan dalam mengklasifikasi Data Diabetes menggunakan Algoritma *Backpropagation Neural Network*, penulis menyimpulkan bahwa algoritma ini mampu melakukan klasifikasi untuk kasus diabetes berdasarkan pengujian kinerja yang dilakukan maka Hasil perbandingan akurasi dengan rasio terbaik pada 90:10 dengan tiga jenis fungsi aktivasi *sigmoid biner* dengan arsitektur 17-5-1 didapatkan akurasi sebesar 76.22% dengan precision sebesar 83.07% untuk kelas 0 dan 71.63% untuk kelas 1, recall didapatkan hasil sebesar 66.23% untuk kelas 0, untuk fungsi aktivasi *sigmoid bipolar* dengan arsitektur 17-15-1 didapatkan hasil akurasi sebesar 76.02% dengan precision kelas 0 dan 1 sebesar 80.97% dan 72.42% dan recall untuk kelas 0 dan 1 berturut-turut sebesar 68.08% dan 83.97%, sedangkan untuk akurasi dengan fungsi aktivasi TanH dengan arsitektur 17-15-1 didapatkan hasil sebesar 76.02% dengan precision sebesar 80.97% untuk kelas 0 dan 72.42% untuk kelas 1 dengan nilai recall sebesar 68.08% untuk kelas 0 dan 83.97%. Berdasarkan hasil akurasi tersebut maka disimpulkan fungsi aktivasi *sigmoid biner* memberikan nilai akurasi terbaik pada dataset diabetes.

REFERENSI

- [1] N. K. Maulidia, "Penerapan artificial neural network pada proses klasifikasi curah hujan di Jawa Tengah," B.S. thesis, Dept. of Mathematics, Fac. of Science and Technology, UIN Maulana Malik Ibrahim, Malang, Indonesia, 2023. [Online]. Available: <http://etheses.uin-malang.ac.id/id/eprint/58140>

- [2] A. Santoso and S. Hansun, "Prediksi IHSG dengan *Backpropagation* Neural Network," *Jurnal Edik Informatika*, vol. 1, no. 2, pp. 1–5, Dec. 2021, doi: 10.32821/edik.v1i2.115.
- [3] R. Dina, "Optimasi *Backpropagation* Neural Network Menggunakan Metode Algoritma Genetika Dalam Memprediksi Jumlah Pengangguran," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 1, no. 2, pp. 104–109, Nov. 2019, doi: 10.30865/json.v1i2.1557.
- [4] F. A. Hizham, Y. Nurdiansyah, and D. M. Firmansyah, "Implementasi Metode *Backpropagation* Neural Network (BNN) dalam Sistem Klasifikasi Ketepatan Waktu Kelulusan Mahasiswa (Studi Kasus: Program Studi Sistem Informasi Universitas Jember)," *Berkala Sainstek*, vol. 6, no. 2, pp. 97–101, Oct. 2018, doi: 10.19184/bst.v6i2.9254.
- [5] M. E. Ervina, R. Silvi, and I. R. N. Wisisono, "Peramalan Jumlah Penumpang Kereta Api di Indonesia dengan Resilient Back-Propagation (Rprop) Neural Network," *Jurnal Matematika "MANTIK"*, vol. 4, no. 2, pp. 90–99, Nov. 2018, doi: 10.15642/mantik.2018.4.2.90-99.
- [6] G. Guntoro, L. Costaner, and L. Lisnawita, "Prediksi Jumlah Kendaraan di Provinsi Riau Menggunakan Metode *Backpropagation*," *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, vol. 14, no. 1, pp. 50–54, Feb. 2019, doi: 10.30872/jim.v14i1.1745.
- [7] B. Julianto, K. T. Nugroho, T. Maharani, and D. F. Nur, "Pengaruh Jumlah Epoch Dan Step Per Epoch Terhadap Performa Mask-Rcnn Pada Deteksi Objek Tanda Tangan," *Journal of Electrical, Electronic, Mechanical, Informatic, and Social Applied Science*, vol. 2, no. 1, pp. 7–16, Feb. 2023, doi: 10.20473/jeemisas.v2i1.45524.
- [8] N. Hadianto, H. B. Novitasari, and A. Rahmawati, "Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network," *Jurnal Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 163–170, Sep. 2019, doi: 10.33480/pilar.v15i2.658.
- [9] A. Rifa'i, "Prediksi Inflasi Indonesia Berdasarkan Fuzzy Ann Menggunakan Algoritma Genetika," *Jurnal ELTIKOM*, vol. 5, no. 1, pp. 12–24, Apr. 2021, doi: 10.31961/eltikom.v5i1.215.
- [10] J. R. Prabowo, R. Santoso, and H. Yasin, "Implementasi Jaringan Syaraf Tiruan *Backpropagation* dengan Algoritma Conjugate Gradient untuk Klasifikasi Kondisi Rumah (Studi Kasus di Kabupaten Cilacap Tahun 2018)," *Jurnal Gaussian*, vol. 9, no. 1, pp. 41–49, Feb. 2020, doi: 10.14710/j.gauss.v9i1.27522.
- [11] M. S. Wibawa, "Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan," *Jurnal Sistem dan Informatika (JSI)*, vol. 11, no. 2, pp. 167–174, Dec. 2017, doi: 10.30646/jsi.v11i2.281.
- [12] J. Prasetya, S. Budi, and S. Kacung, "Prediksi Harga Emas Menggunakan Jaringan Syaraf Tiruan *Backpropagation* Hyperparameter Tuning," *J. Inform. Teknol. dan Sains*, vol. 7, pp. 768–776, 2025.
- [13] R. E. Utama and E. H. Parmadi, "Penerapan Jaringan Syaraf Tiruan *Backpropagation* untuk Klasifikasi Akreditasi Sekolah Menengah Pertama," *Semin. Nas. Teknol. Sains*, vol. 3, no. 1, pp. 45–52, 2024, doi: 10.29407/stains.v3i1.4133.
- [14] H. Novianas, S. P. Wirman, and N. Fitrya, "Algoritma *Backpropagation* Neural Network pada Sistem Optik Pendeteksi Kontaminan Boraks dan Formalin pada Makanan," vol. 14, no. 1, pp. 61–67, 2025.
- [15] N. N. Sabilla, M. Fatchan, and A. Aguswin, "Mengoptimalkan Klasifikasi Sinyal Electroencephalogram Untuk Mendeteksi Kejang Epilepsi Dengan *Backpropagation* Neural Network," vol. 9, no. 2, pp. 3501–3509, 2025.