JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika

[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

PENENTUAN PARAMETER *LEARNING RATE* SELAMA PEMBELAJARAN JARINGAN SYARAF TIRUAN *BACKPROPAGATION* MENGGUNAKAN ALGORITMA GENETIKA

Christian Dwi Suhendra [a,1,*], Ade Chandra Saputra [b,2]
[a] Universitas Papua, Manokwari
[b] Universitas Palangkaraya, Palangkaraya
[1] c.suhendra@unipa.ac.id*; [2] adechandra@it.upr.ac.id;
* corresponding author

ARTICLE INFO

Keywords
Artificial Neural Network
Genetic Algorithm
Backpropagation
Local minima
SSE

ABSTRACT

One of the weakness in backpropagation Artificial neural network(ANN) is being stuck in local minima. Learning rate parameter is an important parameter in order to determine how fast the ANN Learning. This research is conducted to determine a method of finding the value of *learning rate* parameter using a genetic algorithm when neural network learning stops and the *error* value is not reached the stopping criteria or has not reached the convergence. Genetic algorithm is used to determine the value of *learning rate* used is based on the calculation of the fitness function with the *input* of the ANN weights, gradient *error*, and bias. The calculation of the fitness function will produce an *error* value of each *learning rate* which represents each candidate solutions or individual genetic algorithms. Each individual is determined by sum of squared *error* value. One with the smallest SSE is the best individual. The value of *learning rate* has chosen will be used to continue learning so that it can lower the value of the *error* or speed up the learning towards convergence.

The final result of this study is to provide a new solution to resolve the problem in the backpropagation learning that often have problems in determining the learning parameters. These results indicate that the method of genetic algorithms can provide a solution for backpropagation learning in order to decrease the value of SSE when learning of ANN has been static in large *error* conditions, or stuck in local minima.

1. Pendahuluan

Jaringan syaraf tiruan adalah sistem komputasi yang arsitekturnya dan operasinya diilhami dari pengetahuan tentang sel syaraf biologis dalam otak. Model jaringan syaraf tiruan ditunjukan dengan kemampuannya dalam emulasi, analisis, prediksi, dan asosiasi. Kemampuan yang dimiliki jaringan syaraf tiruan dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh *input* yang dimasukan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan karakteristik *input* yang diberikan kepada jaringan syaraf tiruan. Jaringan saraf memiliki daya tarik besar untuk banyak peneliti karena kedekatan mereka yang besar dengan struktur otak, karakteristik yang tidak dimiliki oleh sistem konvensional [1].

Backpropagation merupakan model *supervised learning* jaringan syaraf tiruan(JST) yang memiliki layer jamak. Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenal pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa dengan pola yang dipakai selama pelatihan[2]. Struktur arsitektur backpropagation menggunakan *multilayer perceptrons* yaitu aristektur JST yang memiliki layer input, output, dan layer tersembunyi. Metode *gradient decent*

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

digunakan oleh sebagai metode pembelajaran. Arsitektur dan metode pembelajaran yang digunakan memberi kemampuan bagi backpropagation untuk menghasilkan pembelajaran dengan akurasi yang baik.

Kemampuan backpropagation dalam mengenali pola yang diberikan membuat metode JST ini sering digunakan dalam berbagai bidang. Namun, backpropagation memiliki kekurangan dalam pembelajaran. Menurut [3] backpropagation memiliki dua kelemahan utama yaitu: kecepatan *convergence* yang buruk dan pembelajaran yang tidak stabil sehingga sering terjebak pada lokal minimum. Kelemahan backpropagation juga sering dipengaruhi oleh pemilihan parameter bobot awal yang dipilih secara random[4]. Kecepatan pembelajaran dipengaruhi oleh parameter *learning rate.* Menurut [5] untuk dapat mencapai kondisi *convergence* pemilihan *learning rate* sangat berpengaruh dalam pembelajaran, jika *learning rate* terlalu kecil maka membutuhkan waktu lama untuk mendekati error minimum, namun jika *learning rate* terlalu besar update bobot akan melampaui error minimal dan bobot akan berosilasi. Pemilihan *learning rate* dengan rentang nilai 0-1 dilakukan dengan cara manual dan tidak ada metode yang baku dalam menentukan nilai parameter yang baik. Saat JST sudah berosilasi dan terjebak pada lokal minimum maka cara yang digunakan adalah dengan menghentikan pembelajaran dan mengatur Kembali parameter JST.

Dalam mengatasi kelemahan backpropagation berbagai penelitian telah dilakukan untuk memperoleh kombinasi parameter pembelajaran yang baik. Menurut [2] selain kombinasi parameter arsitektur, bobot awal dan bias awal yang baik pemilihan parameter *learning rate* yang sesuai dengan data yang dilatih juga menentukan dalam menghindari osilasi atau pembelajaran terjebak pada local minimum. Untuk memperoleh parameter *learning rate* yang optimal berdasarkan data latih pada penelitian ini menggunakan metode optimasi algoritma genetika. Kemampuan algoritma genetika untuk memberikan nilai optimal pada ruang pencarian solusi menjadi dasar untuk menentukan parameter *learning rate*. Penelitian yang telah dilakukan oleh [2] menggunakan algoritma genetika dalam menemukan kombinasi yang parameter aristektur, bobot, dan bias awal. Penelitian lain yang dilakukan oleh [6] bertujuan membuat learning rate dapat menyesuaikan pada setiap iterasi untuk meminimalkan error dengan menerapkan peningkatan pembelajaran JST backpropagation dengan mengoptimasi *learning rate* menggunakan variabel *learning rate* dengan cara melakukan optimasi selama update bobot. Berdasarkan permasalahan yang ada penelitian ini bertujuan untuk menentukan perubahan parameter *learning rate* secara dinamis ketika pembelajaran telah berosilasi dan terjebak pada lokal minimum sehingga pembelajaran dapat dilanjutkan tanpa harus mengulang dari awal pembelajaran JST.

2. Metodologi Penelitian

2.1. Desain Penelitian

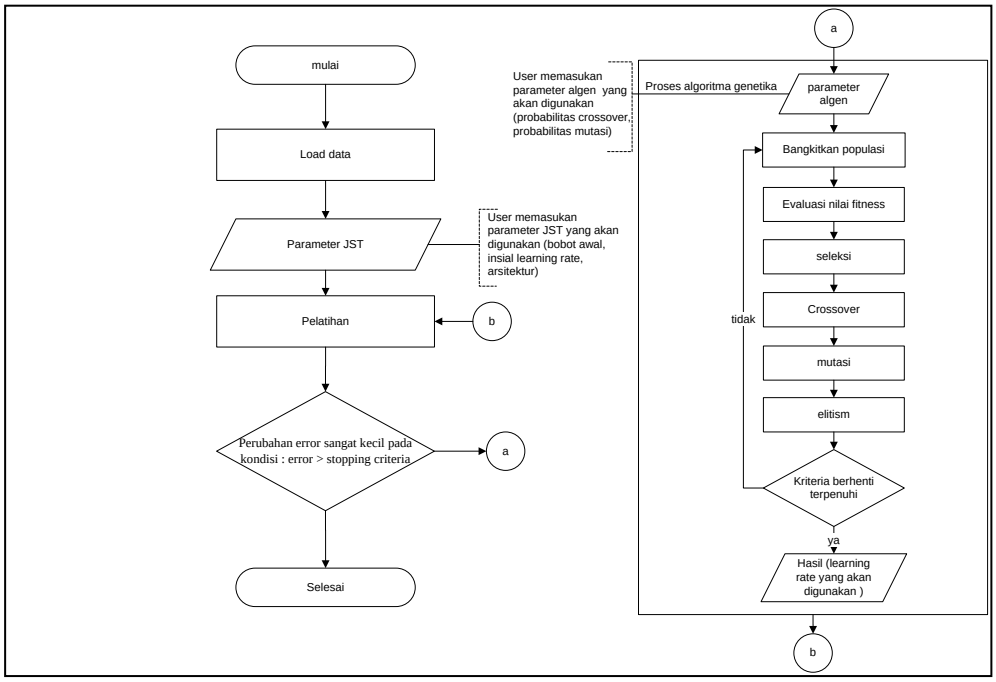
Penelitian ini akan membuat aplikasi penentuan parameter *learning rate* pada jaringan syaraf tiruan backpropagation menggunakan algoritma genetika. Penelitian dilakukan dengan membandingkan antara pembelajaran tanpa menggunakan algoritma genetika dan pembelajaran menggunakan algoritma genetika dalam menentukan parameter *learning rate* ketika terjadi osilasi. aplikasi ini dibangun menggunakan Bahasa pemrograman C#. Data yang digunakan dalam penelitian ini yaitu data data iris flower, data wine quality dan data user knowledge yang dapat diperoleh dari http://archive.ics.uci.edu/ml/. data bunga iris, dengan 4 input yaitu panjang sepal, lebar sepal, panjang petal, lebar petal dan 3 class yaitu iris setosa, iris versicolour dan iris virginica jumlah data latih adalah 151 data. Data *user knowledge modelling*, dengan 5 input dan 4 class jumlah data dengan jumlah data latih sebanyak 150 data. Data *wine quality*, dengan 11 input, dan 10 class yang merupakan kategori dari kualitas anggur dengan interval nilai dari 0-10. Jumlah data latih sebanyak 150 data.

Dalam proses pembelajaran JST backpropagation, langkah awal yang dilakukan adalah menentukan nilai bobot awal, arsitektur jaringan, dan juga menentukan nilai parameter *learning rate*,

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

kemudian dilakukan pembelajaran. Ketika terjadi kondisi dimana perubahan error pada JST tersebut sangat kecil namun masih jauh mendekati *error stopping criteria* membuat pembelajaran JST tersebut menjadi sangat lambat hal yang biasa dilakukan adalah mengganti nilai *learning rate* dengan tujuan untuk meminimalkan *error* sehingga mempercepat pembelajaran. Pada penelitian ini penentuan nilai *learning rate* dilakukan oleh algortima genetika ketika JST sudah mencapai kondisi tersebut dengan tujuan dapat mempercepat proses pembelajaran. Secara umum tahap – tahap penyelesaian masalah dijelaskan pada diagram alir gambar 1.



Gambar 1. Desain Penelitian

Pada Gambar 1 algoritma genetika akan mencari nilai *learning rate* yang baru ketika terjadi osilasi atau perubahan *error* yang sangat kecil namun *error* masih lebih besar daripada *stopping criteria* yang ditentukan.

2.2. Representasi Kromosom

Kromosom dalam penelitian ini ini berisi nilai dari parameter *learning rate* dengan panjang kromosom sebanyak 8 gen. representasi kromosom menggunakan pengkodean biner. Pengkodean biner ini merepresentasi bilangan ril dari range $0 < \eta < 1$ yang akan dipetakan ke bilangan integer. Jadi karena menggunakan pengkodean biner dan panjang kromosom yang diinginkan untuk mendapatkan daerah pencarian solusi sebanyak 8bit maka akan terdapat solusi sebanyak $2^8 = 256$ kromosom yang terdapat pada daerah pencarian. Representasi data yang menjadi kromosom pada algoritma genetika adalah parameter *learning rate* dengan range nilai dari [0...*current learning rate*]. *Current learning rate* adalah nilai parameter *learning rate* yang digunakan. Kemudian kromosom – kromosom tersebut di-decode ke bilangan ril dalam interval [0...*current learning rate*]. Nilai - nilai hasil konversi disubtitusi ke fungsi fitness. Representasi kromosom dapat dilihat pada Gambar 2.

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

kromosom	Gen								Nilai Integer
	x1	x2	x3	x4	x5	x6	x7	x8	
kromosom 1	0	0	1	1	1	1	0	1	$61_{(10)}$
kromosom 2	0	1	1	1	1	1	0	1	$125_{(10)}$
kromosom 3	1	1	1	1	0	1	0	1	$245_{(10)}$
. . .									
kromosom n	0	0	1	1	1	0	0	1	$57_{(10)}$

Gambar 2. Representasi Kromosom

Kemudian nilai integer yang diperoleh dari masing-masing kromosom dipetakan ke dalam range sesungguhnya yaitu [0...current *learning rate*] pada proses decoding menggunakan persamaaan 1.

$$\frac{1}{2^n} \times nilai\ integer\ individu \quad (1)$$

Dimana n adalah Panjang bit kromosom

Pada contoh berikut ini dengan current *learning rate* yang digunakan adalah 1 sedangkan kromosom yang digunakan sebagai berikut:

1	1	1	1	1	1	0	1

$= 125_{(10)}$

Selanjutnya menggunakan persamaan (1). merupakan nilai pemetaan yang didapatkan dari hasil pembagian antara daerah pencarian parameter *learning rate* yaitu pada $0 < \eta < 1$ dengan daerah pencarian nilai interger sebanyak panjang krosomom yaitu 8 bit $2^8 = 256$.

$$\alpha = \frac{1}{256} \times 125 = 0{,}490125.$$

Nilai parameter *learning rate* ini kemudian akan dimasukan ke persamaan (2) dan (3) untuk digunakan mencari fungsi fitness.

2.3. Fungsi Obyektif

fungsi obyektif dari permasalahan optimasi algoritma genetika pada jaringan syaraf tiruan yaitu dengan meminimalkan nilai error. Pada penilitian ini error yang digunakan adalah *Sum of square error* dengan persamaan sebagai berikut :

$$\text{Min} \quad E = \frac{1}{N} \sum_{j=1}^{N} (t_j - o_j)^2 \quad (2)$$

Dimana :

E = root means square error
t_j = output yang dihasilkan
o_j = output yang diinginkan
N = jumlah data

2.4. Fungsi Fitness

Secara umum fungsi fitness diturunkan dari fungsi obyektif pada persamaan 1. Persamaan yang digunakan untuk mengukur tingkat baik dan buruknya nilai masing-masing individu dapat dilihat pada persamaan 2.

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

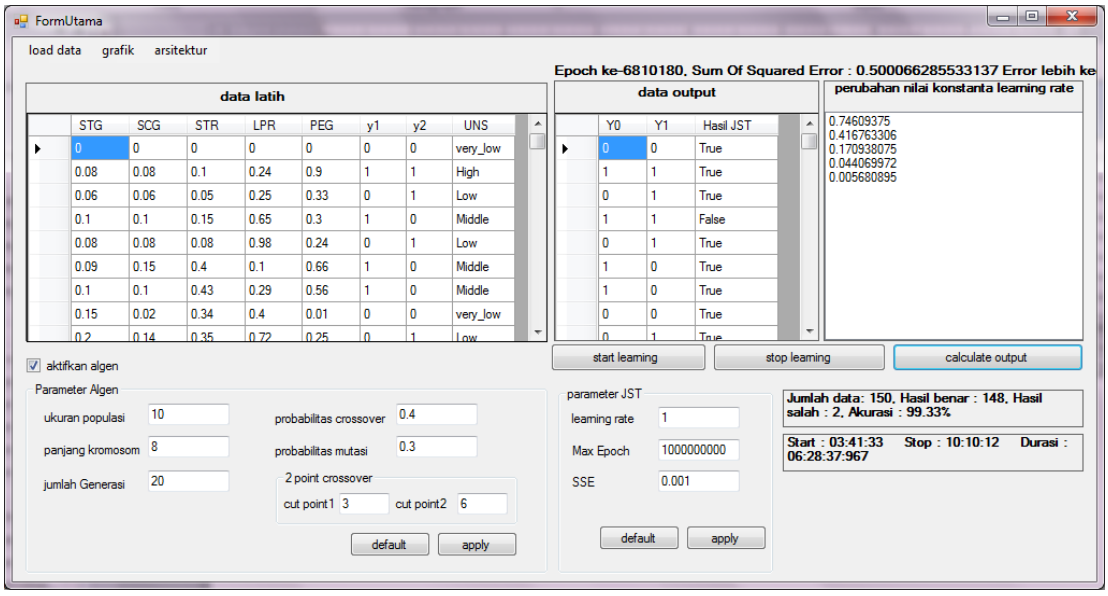
$$f(\alpha) = \frac{1}{(y_{d,k}(p) - sig\left[\sum_{i=1}^{n}\sum_{j=i}^{n} x_{(p)} \times (w + \alpha \times y_j \times \delta_k)_p - \theta\right])} \quad (2)$$

Dimana :

$e_k(p+1)$ = error yang dihasilkan ke iterasi P+1
$y_{d,k}(p)$ = output yang diharapkan pada iterasi ke P
x_i = input
w = bobot jaringan
δ_k = error gradient pada *output* layer
θ = threshold
α = *learning rate*
y_j = output neuron
i = indeks neuron pada *input layer*
j = indeks neuron pada *hidden layer*
k = indeks neuron pada *output layer*

3. Hasil dan Pembahasan

Pada pengujian ini menggunakan tiga data input untuk pelatihan JST dengan parameter input dan output beserta jumlah data yang berbeda – beda yaitu data *iris flower* dengan 4 input, 3 class dan jumlah data 150 data, data *modelling user knowledge* dengan 5 input, 4 class dan jumlah data adalah 150 data, dan yang terakhir adalah data *wine quality* dengan 10 input, 12 class dan jumlah data adalah 150. Dan menggunakan 2 arsitektur dalam melakukan pengujian untuk setiap data. Kondisi dimana algoritma dipakai untuk merubah learning rate adalah jika dalam 200 epoch nilai SSE tidak mengalami perubahan atau nilai SSE meningkat lebih besar dan jika dalam 300 epoch JST berosilasi. Pada gambar 2 dapat dilihat program yang akan menguji kedua metode yang dibandingkan.



Gambar 3. Aplikasi penentuan parameter *learning rate* selama pembelajaran JST *Backpropagation*

Hasil pengujian yang dilakukan pada ketiga data latih yaitu data *iris flower*, data *user knowledge*, dan data *wine quality* yaitu dengan melihat perbandingan epoch untuk mencapai kondisi konvergen. Pengujian juga dilakukan menggunakan beberapa model arsitektur.

3.1. Pengujian terhadap data *iris flower*

1) Pengujian terhadap arsitektur jumlah hidden layer 1 dan jumlah node 11

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

Tabel 1. Pengujian pada JST *backpropagation* dengan *hidden layer* 1 dan jumlah node 11 data *iris flower*.

Learning rate	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	74	74	34525
0.6	49	49	82825
0.4	49	49	116258
0.1	49	32.98666666923	112302
0.01	32.98666678	0.0000923953	39442832

Pada Tabel 1 menunjukan nilai parameter *learning rate* yang dapat memberikan pelatihan JST yang stabil sehingga konvergen adalah 0.01 dengan jumlah epoch sebanyak 39442282, sedangkan untuk nilai *learning rate* lebih dari 0.01 tidak dapat memberikan pelatihan JST yang stabil.

Tabel 2. Pengujian pada JST *backpropagation*-algortima genetika dengan *hidden layer* 1 dan jumlah node 11 data *iris flower*.

Learning rate **awal**	**Perubahan** ***Learning rate***	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	0.078125	49	0.000924	17121714
	0.000305176			
0.6	0.00234375	49	0.000985	18140658
	0.00009105			
0.4	0.0015625	49	0.000995	17157775
	0.000006104			
0.1	0.000390625	33.66626	0.000995	19004845
0.01	0.000039062	32.98667	0.000995	19010334

Pada tabel 2 perubahan *learning rate* yang dihasilkan oleh algoritma genetika menyebabkan pelatihan JST dapat menghasilkan global mininum sehingga mencapai toleransi error berhenti yang diinginkan.

2) Pengujian terhadap arsitektur jumlah hidden layer 2 dan node neuron 8

Tabel 3. Pengujian pada JST *backpropagation* dengan *hidden layer* 2 dan jumlah node 8 data *iris flower*.

Learning rate	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	42.5	32	41054
0.6	43	25.50561798	60633
0.4	43	32	80324

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

Learning rate	Error Awal	Error Berhenti	*Epoch*
0.1	32	31.99999876	112502
0.01	33.66626	0.000993204	45442832

Pada tabel 3 dapat dilihat nilai parameter *learning rate* yang dapat memberikan pelatihan JST yang stabil adalah 0.01 dengan jumlah epoch sebanyak 45442832, sedangkan nilai *learning rate* yang lain tidak dapat memberikan konvergensi pada pelatihan JST, dan hanya bisa berhenti pada nilai SSE yang jauh diatas nilai toleransi yang sudah ditentukan pada *stopping criteria*.

Tabel 4. Pengujian pada JST *backpropagation*-algortima genetika dengan *hidden layer* 2 dan jumlah node 8 data *iris flower*.

Learning rate awal	Perubahan *Learning rate*	Error Awal	Error Berhenti	*Epoch*
0.9	0.371322632	42.5	0.000997482648	16508149
	0.127642155			
	0.000498602			
	0.00012465			
0.6	0.0234375	43	0.000997361846	11785235
	0.000585938			
0.4	0.148375	43	0.000996648234	9910957
	0.00057983			
	0.00007248			
0.1	0.025	32	0.000998543234	8018258
	0.0625			
	0.0015625			
0.01	0.0025	33.66626	0.000989852012	17442832
	0.0003125			

Tabel 4 menunjukan setiap *learning rate* yang dimasukan akan memberikan konvergensi terhadap JST namun nilai *learning rate* yang memberikan pembelajaran yang lebih cepat adalah 0.1 dengan perubahan akhir menjadi 0.0015625 yang menghasilkan epoch sebanyak 8018258.

3.2. Pengujian terhadap data userknowledge

1) Pengujian terhadap arsitektur jumlah hidden layer 1 dan jumlah node 11

Tabel 5. Pengujian pada JST *backpropagation* dengan *hidden layer* 1 dan jumlah node 11 data *user knowledge*

Learning rate	Error Awal	Error Berhenti	*Epoch*

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

Learning rate	Error Awal	Error Berhenti	*Epoch*
0.9	79	79	61354
0.6	73.5	79	60633
0.4	73.5	79	80324
0.1	49	0.000999987	33453645
0.01	32	0.000999991	38834366

Pada pengujian terhadap data *user knowledge* dengan struktur *hidden layer* 1 dan jumlah *neuron* pada *hidden layer* 11 seperti yang ditunjukan pada tabel 5 nilai parameter *learning rate* yang dapat memberikan pembelajaran yang stabil adalah 0.1 yang mengasilkan epoch sebanyak 33453645.

Pengujian yang dilakukan dengan menggunakan algoritma genetika ditunjukan pada tabel 6. Pembelajaran JST yang dilakukan menggunakan algoritma genetika memberikan hasil yang sama dalam hal *learning rate* awal dimana nilai 0.1 memberikan pembelajaran yang stabil. Sedangkan parameter awal yang diatur lebih besar memberikan perubahan parameter dan menghasilkan pembelajaran yang lebih cepat.

Tabel 6. Pengujian pada JST *backpropagation*-algortima genetika dengan *hidden layer* 1 dan jumlah node 11 data *user knowledge*

***Learning rate* awal**	**Perubahan *Learning rate***	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	0.8671875	79	0.00099999	1425493
	0.735076904			
	0.485265613			
	0.227468256			
	0.056867064			
0.6	0.3703125	73.5	0.00099999	8220312
	0.122955322			
	0.030738831			
0.4	0.1453125	73.5	0.00099998	7220312
	0.036328125			
0.1		49	0.00099999	30204127
0.01		32	0.00099999	40632197

2) Pengujian terhadap arsitektur jumlah hidden layer 2 dan jumlah node 8

Berbeda dengan pelatihan menggunakan struktur hidden layer 1 dan neuron 11, pada Pelatihan JST backpropagation menggunakan arsiktektur yaitu jumlah hidden layer 2 dan node tiap layer adalah 8 dapat memberikan pelatihan JST yang lebih cepat. Hal ini ditunjukan pada tabel 7 untuk *learning rate* 0.1 menghasilkan jumlah epoch sebanyak 27986 dan untuk *learning rate* 0.01

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

menghasilkan jumlah epoch sebanyak 294312 sehingga waktu komputasinya menjadi lebih sedikit. Sedangkan untuk *learning rate* yang lebih besar tidak dapat menghasilkan konvergensi pada pelatihan JST.

Tabel 7. Pengujian pada JST backpropagation dengan *hidden layer* 2 dan jumlah node 8 data user knowledge

Learning rate	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	40	79	52145
0.6	40	79	57532
0.4	40	73.5	60514
0.1	40	0.000999987	27986
0.01	40	0.000999991	294312

Tabel 8 menunjukan dengan menggunakan algoritma genetika nilai *learning rate* yang sebelumnya terhenti pada error yang besar dapat diganti dengan melihat hasil fitness tiap – tiap nilai yang memberikan error minimal. Setiap perubahan nilai *learning rate* bergantung pada bobot dan bias yang menjadi parameter dari fungsi fitness. Perubahan *learning rate* yang dihasilkan oleh algoritma genetika menyebabkan pelatihan JST dapat menghasilkan global mininum sehingga mencapai toleransi error berhenti yang diinginkan.

Tabel 8. Pengujian pada JST backpropagation-algoritma genetika dengan *hidden layer* 2 dan jumlah node 8 data *user knowledge*

Learning rate **awal**	**Perubahan** ***Learning rate***	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	0.87109375	40	0.000999991	25359
	0.656723022			
	0.361710727			
	0.124338062			
	0.031084516			
0.6	0.34453125	40	0.000999987	79180
	0.114595142			
	0.015193105			
0.4	0.159375	40	0.000999984	28835
	0.031929932			
0.1		40	0.000999984	27986
0.01		40	0.000999991	304312

3.3. Pengujian terhadap data *wine quality*

1) Pengujian terhadap arsitektur jumlah hidden layer 1 dan jumlah node 11

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

Tabel 9. Pengujian pada JST *backpropagation* dengan *hidden layer* 1 dan jumlah node 30 data *wine quality*

Learning rate	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	44.5	44.5	1541054
0.6	44.5	44.5	1530633
0.4	44.5	32	1780324
0.1	42	31.99999876	28112502
0.01	35	19.45345773	28533075

Pada hasil percobaan menggunakan data *wine quality* yang ditunjukan pada tabel 9 nilai parameter yang digunakan pada percobaan tidak dapat memberikan pembelajaran yang stabil dan tidak dapat mencapai kondisi konvergen. Sedangkan menggunakan algoritma genetika pembelajaran dapat berjalan dengan baik dan mencapai kondisi konvergen setelah dilakukan beberapa perubahan nilai *learning rate*. Tabel perubahan *learning rate* ditunjukan pada tabel 10.

Tabel 10. Pengujian pada JST backpropagation - algortima genetika dengan *hidden layer* 1 dan jumlah node 30 data *wine quality*

Learning rate **awal**	**Perubahan** ***Learning rate***	**Error Awal**	**Error Berhenti**	***Epoch***
0.9	0.02734375	74	0.000999991	38358231
	0.000106812			
0.6	0.004065	44.5	0.000998544	30263897
	0.000254			
	0.000064			
0.4	0.015625	44.5	0.000999891	30215037
	0.000390625			
	0.000097656			
0.1	0.000390625	42	0.000989939	30115323
	0.000097656			
0.01	0.0025	35	0.000989939	28523976
	0.000625			
	0.000156			
	0.000039			
0.001	0.00025		0.000989939	27115323
	0.0000625			

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 14 No 2]
[Agustus 2020]

4. Kesimpulan

Penggantian learning rate hanya dilakukan ketika pembelajaran JST sudah statis pada suatu nilai SSE tertentu yang masih jauh dari toleransi error yang ditetapkan atau belum konvergen. Setiap nilai learning rate yang dihasilkan oleh algoritma genetika sangat bergantung pada bobot dan error gradient sebagai parameter input untuk menentukan nilai fitness dari setiap individu. Setiap penggantian nilai learning rate tidak menentukan bahwa learning rate tersebut adalah sebuah nilai yang optimal karena setiap perubahan bobot bisa saja membuat nilai error kembali meningkat. Berdasarkan hasil percobaan yang didapatkan dengan membandingkan kedua metode, metode penggantian nilai learning rate menggunakan algoritma genetika menghasilkan pembelajaran yang lebih baik dibandingkan dengan pembelajaran JST backpropagation tanpa modifikasi. Kecepatan pembelajaran selain bergantung pada learning rate bergantung pula pada pemilihan arsitektur JST. Metode penggantian nilai learning rate dapat dipakai pada data yang kompleks biasanya dengan jumlah input banyak dan dengan berbagai jumlah parameter output yang memungkinkan untuk bisa berosilasi sedangkan untuk jenis data seperti data XOR metode ini tidak dapat dipakai karena pembelajaran JST tidak akan pernah berosilasi.

Pengembangan aplikasi dapat dilakukan dengan cara menggunakan momentum dalam fungsi fitness kemungkinan dapat memberikan hasil yang lebih baik tanpa harus bergantung pada jenis arsitektur JST yang diterapkan. Sebuah metode pengoptimalan yang melibatkan karakteristik data latih, arsiktektur JST dan learning rate dapat memberikan sebuah solusi untuk mengatasi masalah penentuan parameter yang cocok untuk sebuah pelatihan JST.

Daftar Pustaka

[1] Rojas, R., 1996, Neural Networks: A Systematic Introduction, New York, Springer.

[2] Suhendra, C. D., & Wardoyo, R. (2015). Penentuan Arsitektur Jaringan Syaraf Tiruan Backpropagation (Bobot Awal dan Bias Awal) Menggunakan Algoritma Genetika. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *9*(1), 77.

[3] Nawi,N,M., Khan, A., Rehman, M, Z., 2013, A New Backpropagation Neural Network Optimized with Cuckoo Search Algorithm, *Computational Science and Its Applications-ICCSA*, volume 7971, pp 413-426.

[4] Chang, Y., Lin, J., Sieh, J., Abbod, M,F., 2012, Optimization the Initial Weights of Artificial Neural Networks via Genetic Algorithm Applied to Hip Bone Fracture Prediction, Taiwan, *Hindawi Publishing Corporation Advances in Fuzzy Systems*, 951247, 2012, 9.

[5] Nuzly, H., 2006 Particel Swarm Optimization For Neural Network Learning Enhancment, *Tesis*, Faculty of Computer Science and Information System Univesiti Teknologi Malaysia.

[6] Yamamoto, K., Koakutsu, S., Okamoto, T. and Hirata, H. (2011), Fast backpropagation learning using optimization of learning rate for pulsed neural networks. Electron. Comm. Jpn., 94: 27-34.

https://doi.org/10.47111/JTI