

Kinerja Algoritma Pelatihan Levenberg-Marquardt dalam Variasi Banyaknya Neuron pada Lapisan Tersembunyi

(Performance of Levenberg-Marquardt Training Algorithm Based on Variations in the Number of Neurons in the Hidden Layer)

Hindayati Mustafidah¹⁾, Muhamad Zaeni Budiastanto¹⁾, Suwarsito²⁾

¹⁾Teknik Informatika - Fakultas Teknik dan Sains - Universitas Muhammadiyah Purwokerto

²⁾Geografi-Fakultas Keguruan dan Ilmu Pendidikan - Universitas Muhammadiyah Purwokerto
Jl. Raya Dukuwaluh Purwokerto 53182

fida.mustafidah@ump.ac.id

Abstrak - Algoritma pelatihan Levenberg-Marquardt (LM) merupakan algoritma pelatihan yang paling optimal daripada algoritma pelatihan lainnya ditinjau dari eror yang dihasilkan. Kondisi tersebut menggunakan 10 neuron pada lapisan tersembunyi. Banyaknya neuron dalam lapisan tersembunyi yang digunakan dalam proses pembelajaran berpengaruh pada kinerja jaringan. Sebagai kelanjutan dari penelitian sebelumnya, maka dalam penelitian ini dilakukan analisis terhadap kinerja algoritma pelatihan LM ditinjau dari epoh yang diperlukan oleh jaringan menggunakan beberapa variasi banyaknya neuron dalam lapisan tersembunyi. Epoh dipandang sebagai salah satu parameter jaringan syaraf tiruan yang digunakan sebagai tolok ukur kinerja. Tahapan penelitian yang dilakukan adalah membangun program komputer menggunakan MATLAB untuk menjalankan algoritma pelatihan LM, selanjutnya rata-rata epoh jaringan dalam 20 kali perulangan sebagai data penelitian dianalisis menggunakan ANOVA. Algoritma pelatihan LM dijalankan dengan 5, 10, dan 15 neuron pada lapisan input dengan 1 neuron pada lapisan output, dan variasi banyaknya neuron pada lapisan tersembunyi untuk masing-masing banyaknya neuron pada lapisan input. Variasi banyaknya neuron pada lapisan tersembunyi digunakan untuk menemukan kondisi optimal algoritma pelatihan yang berupa rata-rata epoh paling kecil. Hasil analisis menunjukkan bahwa kondisi optimal algoritma pelatihan LM dengan 5 neuron pada lapisan input dicapai pada penggunaan 9 neuron pada lapisan tersembunyi dengan rata-rata epoh sebesar 10.80; untuk 10 neuron pada lapisan input dicapai pada penggunaan 19 neuron pada lapisan tersembunyi dengan rata-rata epoh sebesar 21.52; dan untuk 15 neuron pada lapisan input dicapai pada penggunaan 29 neuron pada lapisan tersembunyi dengan rata-rata epoh sebesar 7.38.

Kata-kata Kunci: *backpropagation, neuron, lapisan tersembunyi, Levenberg-Marquardt, epoh.*

Abstract - The Levenberg-Marquardt (LM) training algorithm is the most optimal training algorithm than other training algorithms in terms of the resulting error. The condition uses 10 neurons in the hidden layer. The number of neurons in the hidden layer that is used in the learning process affects network performance. As a continuation of previous research, this study analyzed the performance of the LM training algorithm in terms of epoch required by the network using several variations of the number of neurons in the hidden layer. Epoch is seen as one of the parameters of artificial neural networks that are used as performance benchmarks. Stages of research carried out is to build a computer program using MATLAB to run the LM training algorithm, then the average epoch network in 20 repetitions as research data is analyzed using ANOVA. The LM training algorithm is run with 5, 10, and 15 neurons in the input layer with 1 neuron in the output layer, and variations in the number of neurons in the hidden layer for each number of neurons in the input layer. The variation of the number of neurons in the hidden layer is used to find the optimal conditions of the training algorithm in the form of the smallest average epoch. The results of the analysis showed that the optimal condition of the LM training algorithm with 5 neurons in the input layer was achieved in the use of 9 neurons in the hidden layer with an average of 10.80; for 10 neurons in the input layer achieved on the use of 19 neurons in the hidden layer with an average epoch of 21.52; and for 15 neurons in the input layer was achieved with the use of 29 neurons in the hidden layer with an average of 7.38.

Keywords: *backpropagation, neuron, hidden layer, Levenberg–Marquardt, epoch.*

I. PENDAHULUAN

Jaringan syaraf tiruan (JST) adalah model komputasi yang terinspirasi secara biologis, jaringan syaraf tiruan terdiri dari beberapa elemen pengolahan (*neuron*) dan ada hubungan antara neuron. Neuron–neuron tersebut akan mentransformasikan informasi yang diterima oleh neuron satu menuju neuron lainnya, hubungan ini disebut dengan bobot [1].

Salah satu model pembelajaran terawasi JST yang banyak diminati adalah model pembelajaran *backpropagation* dikarenakan banyak aplikasi yang berhasil diselesaikan dengan *backpropagation*, sebagaimana yang dilakukan oleh [2] untuk memprediksi tingkat validitas soal. Selain itu, [3] memprediksi prestasi belajar mahasiswa Program Studi Teknik Informatika Universitas Muhammadiyah Purwokerto didasarkan atas nilai-nilai dalam mata pelajaran yang diujikan dalam Ujian Nasional saat di SLTA. Ref. [4] juga menerapkan *backpropagation* untuk mengetahui tingkat kualifikasi calon siswa pada sistem informasi penerimaan siswa baru di Madrasah Aliyah Negeri 2 Banjarnegara.

Pada *backpropagation* terdapat pengolahan elemen yang membuat perhitungan berdasarkan pada jumlah masukan (*input*). Sebuah kelompok pengolahan elemen disebut *layer* atau lapisan dalam jaringan. Lapisan pertama adalah *input* dan yang terakhir adalah *output*. Lapisan diantara lapisan *input* dan *output* disebut dengan lapisan tersembunyi atau *hidden layer* [5].

Dalam model *backpropagation* terdapat 12 algoritma pelatihan yang bisa digunakan, yaitu algoritma *Fletcher-Reeves Update*, *Polak-Ribiere*, *Powell-Beale Restarts*, *Scaled Conjugate Gradient*, *Gradient Descent* dengan *Momentum* dan *Adaptive Learning Rate*, *Resilent Backpropagation*, *BFGS*, *One Step Secant*, *Levenberg-Marquardt* [6]. Pengujian terhadap 12 algoritma pelatihan tersebut telah dilakukan oleh [7], [8], [9], dan [10] untuk mendapatkan algoritma pelatihan yang paling optimal ditinjau dari *error* yang dihasilkan. Penelitian tersebut menghasilkan simpulan bahwa algoritma *Levenberg-Marquardt* merupakan algoritma pelatihan yang paling optimal pada tingkat laju pembelajaran (*learning rate/lr*) = 0.05. Dalam penelitian tersebut menggunakan 10 neuron dan belum disebutkan banyaknya neuron yang paling optimal dalam lapisan tersembunyi. Oleh karena itu diperlukan adanya penelitian untuk menentukan banyaknya neuron dalam lapisan tersembunyi ditinjau dari epoch yang

diperlukan untuk mencapai kondisi *error* paling kecil menggunakan algoritma *Levenberg–Marquardt*. Epoch menunjukkan banyaknya iterasi atau langkah yang diperlukan oleh jaringan dalam melaksanakan pelatihan hingga dipenuhi kriteria tertentu untuk berhenti [6].

II. METODE

Jenis penelitian yang digunakan adalah penelitian dengan metode campuran (*mixed-method*) yaitu penelitian pengembangan program komputer dengan pengujian kuantitatif dan kualitatif menggunakan uji statistik. Program JST dibangun menggunakan bahasa MATLAB karena praktis dalam menggunakannya [11] dan terdapat fungsi-fungsi *built-in* berupa *toolbox* JST [12]. Sementara itu, uji statistika terhadap data penelitian menggunakan program SPSS [13].

A. Variabel Penelitian

Variabel yang diteliti dalam penelitian ini menggunakan parameter-parameter jaringan dan berikut desain JST seperti pada Gambar 1.

Epoch maksimum : 1000 atau 10^3

Learning rate (LR) : 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1

Target *error* : 0.001 (10^{-3})

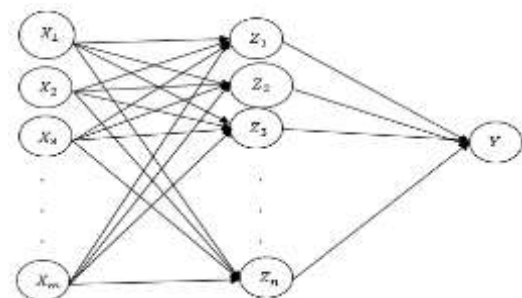
Neuron dalam lapisan *input*: 5, 10, 15

Neuron dalam lapisan *output*: 1

Neuron dalam lapisan tersembunyi: tersaji pada Tabel 1.

TABEL I
BANYAKNYA NEURON DALAM LAPISAN TERSEMBUNYI

Neuron dalam lapisan <i>input</i>	Neuron dalam lapisan tersembunyi
5	1, 2, 3, 4, 5, 7, 9
10	6, 7, 8, 9, 10, 12, 14, 16, 19
15	8, 11, 13, 15, 17, 19, 21, 23, 27, 29



Gambar 1. Desain Jaringan Syaraf Tiruan sesuai dengan Neuron Masukan sebanyak m ($m = 5, 10, 15$) dan Neuron Lapisan Tersembunyi sebanyak n ($n = \{1, 2, 3, 4, 5, 7, 9, 6, 7, 8, 9, 10, 12, 14, 16, 19\}, \{8, 11, 13, 15, 17, 19, 21, 23, 27, 29\}$).

B. Sumber Data

Sumber data neuron masukan dan neuron target yang digunakan adalah data dokumentasi yang diambil dari penelitian [14].

C. Alur Pengembangan Program

Struktur program JST yang dibangun sebagai hasil modifikasi dari penelitian [15] dengan garis besar langkah pengembangannya yaitu membangun data latih, melakukan seting parameter jaringan, preprosesing data latih, melakukan pelatihan data latih, dan melakukan simulasi menggunakan data latih. Proses pelatihan ini diulang sebanyak 20 kali untuk setiap nilai LR dan setiap variasi banyaknya neuron dalam lapisan tersembunyi. Selanjutnya epoch yang diperoleh dilakukan uji statistik.

D. Analisis Data

Pengujian dilakukan dengan menggunakan uji statistik ANOVA dengan tahapan berikut :

1) Menentukan hipotesis

H_0 : tidak terdapat perbedaan epoch yang diperlukan untuk menghasilkan *error* terkecil pada suatu banyak neuron dalam lapisan tersembunyi.

H_1 : terdapat perbedaan epoch yang diperlukan untuk menghasilkan *error* terkecil pada suatu banyak neuron dalam lapisan tersembunyi.

2) Menentukan nilai α (α). Dalam penelitian ini digunakan $\alpha = 5\%$.

3) Pengambilan kesimpulan. Kesimpulan diambil berdasarkan nilai signifikan yang diperoleh (sig.) dengan ketentuan H_0 ditolak jika nilai $\text{sig} < \alpha$.

III. HASIL DAN PEMBAHASAN

A. Data Penelitian

Data penelitian diambil dari data *random* dengan 5, 10 dan 15 data, baik data masukan maupun data target yang berasal dari penelitian [14]. Data epoch didapatkan dengan cara melakukan *training* data penelitian menggunakan algoritma *Levenberg-Marquardt* kemudian dilakukan tabulasi data epoch.

B. Analisis Data

Hasil analisis terhadap pengujian statistik dari data penelitian disajikan per kelompok data masukan berupa 5, 10, dan 15 neuron dalam lapisan input untuk diketahui rata-rata epoch yang diperlukan oleh algoritma *Levenberg-Marquardt* untuk mencapai kondisi *error* paling kecil. Seluruh pengujian dilakukan menggunakan nilai $\alpha = 5\%$.

1) Lima (5) neuron dalam lapisan input untuk setiap nilai LR.

Hasil uji ANOVA terhadap data rata-rata epoch yang dihasilkan dari simulasi data uji untuk setiap nilai LR dan setiap variasi banyaknya neuron dalam lapisan tersembunyi (n) dengan n = 1, 2, 3, 4, 5, 7, 9 diperoleh nilai signifikansi seperti tersaji pada Tabel 2.

TABEL II
NILAI SIGNIFIKANSI HASIL UJI ANOVA UNTUK 5
NEURON DALAM LAPISAN INPUT

No.	Banyaknya neuron dalam lapisan tersembunyi	Nilai sig.
1.	1	0.970
2.	2	0.748
3.	3	0.857
4.	4	0.878
5.	5	0.416
6.	7	0.780
7.	9	0.573

Terlihat pada Tabel 2 bahwa ketujuh variasi banyaknya neuron dalam lapisan tersembunyi sebagai parameter dalam pelatihan JST menggunakan algoritma LM semuanya menghasilkan nilai signifikansi lebih besar dari α (5%). Nilai-nilai signifikansi ini berakibat penerimaan H_0 (hipotesis nol). Hal ini mengindikasikan rata-rata epoch yang diperlukan oleh algoritma LM untuk mencapai kondisi *error* paling kecil tidak berbeda secara signifikan di antara ketujuh macam banyaknya neuron dalam lapisan tersembunyi. Namun demikian, perlu dilakukan analisis deskriptif dengan hasil seperti tersaji pada Tabel 3.

TABEL III
RATA-RATA EPOH PADA SETIAP *LEARNING RATE* UNTUK 5 NEURON PADA LAPISAN *INPUT*

No	LR	Rata-rata epoh pada banyaknya neuron dalam lapisan tersembunyi						
		1	2	3	4	5	7	9
1	0.01	97.32	67.26	188.71	56.74	84.79	10.20	8.30
2	0.05	67.40	171.44	116.30	125.30	17.89	19.89	13.60
3	0.1	165.06	91.44	74.17	51.89	32.30	23.55	12.20
4	0.2	67.65	123.07	70.18	41.83	33.50	11.85	10.55
5	0.3	40.60	65.17	223.83	77.22	64.40	20.84	9.20
6	0.4	74.55	101.06	100.32	139.17	50.50	13.95	8.95
7	0.5	122.47	89.53	100.26	44.00	16.85	10.35	11.85
8	0.6	52.00	137.81	115.50	95.75	63.60	18.35	9.30
9	0.7	91.32	234.79	115.50	97.20	13.42	10.95	13.65
10	0.8	78.78	173.83	157.63	123.00	62.85	9.20	9.85
11	0.9	80.93	85.50	157.63	70.68	112.45	9.20	10.10
12	1	103.06	136.74	157.63	70.68	12.30	14.20	12.10

Data yang dicetak tebal dalam Tabel 3 adalah rata-rata epoh terkecil dalam setiap kelompok banyaknya neuron pada lapisan tersembunyi yang bersesuaian dengan tingkat LR yang digunakan. Berdasarkan data tersebut, rata-rata epoh terkecil diperoleh pada penggunaan 9 neuron dalam lapisan tersembunyi pada tingkat LR 0.01. Secara grafikal, kondisi tersebut digambarkan pada Gambar 2.



Gambar 2. Grafik rata – rata epoh terkecil berdasarkan *learning rate* pada penggunaan 5 neuron dalam lapisan *input*

Data pada Tabel 3 selanjutnya dianalisis menggunakan uji ANOVA untuk mendapatkan informasi pencapaian rata-rata epoh terkecil berdasarkan banyaknya neuron dalam lapisan tersembunyi tanpa dipengaruhi oleh nilai LR. Hasil uji ANOVA menggunakan taraf $\alpha = 5\%$, diperoleh nilai signifikansi sebesar 0.573 yang berarti tidak ada perbedaan secara signifikan rata-rata epoh yang diperlukan oleh algoritma LM berdasarkan banyaknya

neuron dalam lapisan tersembunyi. Meskipun demikian, rata-rata epoh terkecil dapat diketahui melalui analisis deskriptif dengan hasil seperti yang ditampilkan pada Gambar 3.



Gambar 3. Grafik rata – rata epoh terkecil pada penggunaan 5 neuron dalam lapisan *input* tanpa mempertimbangkan nilai *learning rate*

Gambar 3 memperlihatkan pola kurva yang mirip dengan kurva pada Gambar 2. Rata-rata epoh terkecil yang diperlukan oleh algoritma LM menggunakan 5 neuron dalam lapisan input adalah sebesar 10.8 dengan 9 neuron pada lapisan tersembunyi. Nilai ini diperoleh tanpa memandang nilai learning rate yang digunakan.

2) Sepuluh (10) neuron dalam lapisan input untuk setiap nilai LR.

Hasil analisis terhadap data rata-rata epoh yang dihasilkan berupa nilai signifikansi dari simulasi data uji untuk setiap nilai LR dan setiap variasi banyaknya n

neuron dalam lapisan tersembunyi disajikan pada Tabel 4. Nilai-nilai neuron dalam lapisan tersembunyi yang digunakan adalah 6, 7, 8, 9, 10, 12, 14, 16, dan 19.

Berdasarkan data pada Tabel 4 terlihat nilai signifikansi yang dihasilkan semua melebihi nilai α (yaitu 0.05). Hal ini menunjukkan bahwa rata-rata epoch yang diperlukan oleh algoritma LM untuk mencapai kondisi *error* paling kecil tidak berbeda secara signifikan meskipun digunakan jumlah neuron dalam lapisan tersembunyi yang berbeda-beda. Untuk mengetahui rata-rata epoch terkecil di antara penggunaan kesembilan variasi banyaknya neuron dalam lapisan tersembunyi ini, dilakukan analisis deskriptif dengan hasil seperti disajikan pada Tabel 5.

TABEL IV
NILAI SIGNIFIKANSI HASIL UJI ANOVA UNTUK 10 NEURON DALAM LAPISAN INPUT

No.	Banyaknya neuron dalam lapisan tersembunyi	Nilai sig.
1.	6	0.129
2.	7	0.483
3.	8	0.376
4.	9	0.642
5.	10	0.650
6.	12	0.103
7.	14	0.129
8.	16	0.475
9.	19	0.964

TABEL V
RATA-RATA EPOCH PADA SETIAP LEARNING RATE UNTUK 10 NEURON PADA LAPISAN INPUT

No	LR	Rata-rata epoch pada banyaknya neuron dalam lapisan tersembunyi								
		6	7	8	9	10	12	14	16	19
1	0.01	227.31	99.63	79.21	61.84	75.75	25.20	35.70	23.35	22.15
2	0.05	158.50	197.74	48.00	39.95	40.79	41.05	38.20	23.90	21.75
3	0.1	266.78	130.72	33.45	66.89	36.70	38.89	26.20	24.00	20.25
4	0.2	141.11	100.78	52.50	84.58	39.80	27.50	24.95	54.20	22.60
5	0.3	308.89	100.33	67.56	98.61	58.65	47.90	25.00	26.50	20.90
6	0.4	59.63	54.50	117.06	85.70	50.68	34.00	25.25	25.40	22.55
7	0.5	56.24	76.06	119.70	31.83	31.33	36.30	23.55	30.30	22.65
8	0.6	188.76	139.00	121.50	61.42	42.58	24.45	23.75	24.50	22.95
9	0.7	172.24	97.76	92.88	46.30	51.47	41.80	24.95	23.50	18.55
10	0.8	122.76	160.50	156.47	93.79	83.21	33.60	31.35	20.85	20.70
11	0.9	166.14	123.89	104.21	38.68	48.79	124.35	22.30	30.65	20.45
12	1	165.76	69.11	162.05	57.00	29.63	29.25	29.90	26.65	22.70

Sebagaimana terlihat pada Tabel 5, setiap kelompok neuron dalam lapisan tersembunyi menghasilkan data rata-rata epoch yang berbeda. Rata-rata epoch terkecil untuk setiap penggunaan sejumlah neuron dalam lapisan tersembunyi, dituliskan dengan angka yang dicetak tebal. Secara keseluruhan, rata-rata epoch terkecil diperoleh pada penggunaan 19 neuron dalam lapisan tersembunyi pada tingkat LR 0.7. Kondisi ini digambarkan pada Gambar 4.



Gambar 4. Grafik rata – rata epoch terkecil berdasarkan *learning rate* pada penggunaan 10 neuron dalam lapisan input

Analisis selanjutnya menggunakan uji ANOVA terhadap data pada Tabel 5. Analisis ini bertujuan untuk mendapatkan informasi pencapaian rata-rata epoch terkecil berdasarkan banyaknya neuron dalam lapisan tersembunyi tanpa mempertimbangkan nilai LR.

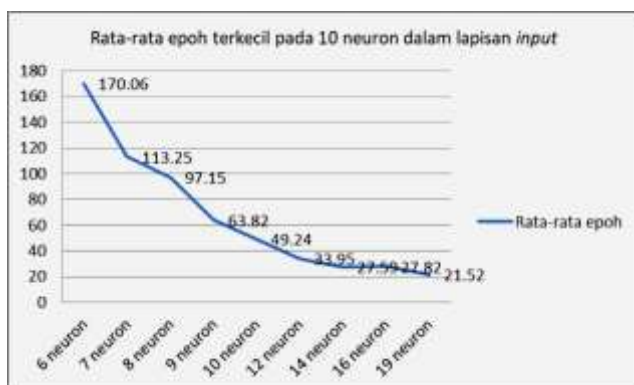
Dengan menggunakan taraf $\alpha = 5\%$, diperoleh nilai signifikansi sebesar 0.000 yang berarti terdapat perbedaan secara signifikan rata-rata epoch yang diperlukan oleh algoritma LM berdasarkan banyaknya

neuron dalam lapisan tersembunyi. Oleh karena itu, diperlukan analisis perbandingan ganda (*multiple comparison analysis/MCA*) menggunakan uji Duncan. Hasil MCA disajikan pada Tabel 6.

TABEL VI
HASIL UJI DUNCAN PADA SETIAP BANYAKNYA NEURON DALAM LAPISAN TERSEMBUNYI PADA 10 NEURON LAPISAN INPUT

Duncan	Neuron dalam lapisan tersembunyi	N	Subset for alpha = 0.05				
			1	2	3	4	5
Duncan ^{a,b}	19 neuron	240	21.52				
	14 neuron	240	27.59	27.59			
	16 neuron	240	27.82	27.82			
	12 neuron	237	33.95	33.95			
	10 neuron	229		49.24	49.24		
	9 neuron	228			63.82		
	8 neuron	224				97.15	
	7 neuron	208				113.25	
	6 neuron	198					170.06
	Sig.		.333	.087	.207	.163	1.000

Berdasarkan data pada Tabel 6 terlihat bahwa penggunaan 19 neuron dalam lapisan tersembunyi, algoritma LM membutuhkan rata-rata epoch paling sedikit yaitu sebesar 21.52. Secara grafik, kecenderungan hubungan antara banyaknya neuron dalam lapisan tersembunyi dengan rata-rata epoch yang dibutuhkan tanpa memperhatikan nilai LR tersaji pada Gambar 5.



Gambar 5. Grafik rata –rata epoch terkecil pada penggunaan 10 neuron dalam lapisan input tanpa mempertimbangkan nilai *learning rate*

Kurva pada Gambar 5 memperlihatkan pola hubungan antara banyaknya neuron dalam lapisan tersembunyi dengan rata-rata epoch yang diperlukan. Semakin banyak neuron dalam lapisan tersembunyi, rata-rata epoch yang diperlukan memiliki kecenderungan semakin sedikit.

3) Lima belas (15) neuron dalam lapisan input untuk setiap nilai LR

Uji ANOVA terhadap data rata-rata epoch yang dihasilkan dari simulasi data uji untuk setiap nilai LR dan setiap variasi banyaknya neuron dalam lapisan tersembunyi dengan 15 neuron pada lapisan input disajikan dalam Tabel 7. Variasi banyaknya neuron dalam lapisan tersembunyi yang digunakan adalah 8, 11, 13, 15, 17, 19, 21, 23, 27, dan 29.

TABEL VII
NILAI SIGNIFIKANSI HASIL UJI ANOVA UNTUK 15 NEURON DALAM LAPISAN INPUT

No.	Banyaknya neuron dalam lapisan tersembunyi	Nilai sig.
1.	8	0.817
2.	11	0.227
3.	13	0.156
4.	15	0.793
5.	17	0.785
6.	19	0.535
7.	21	0.104
8.	23	0.074
9.	27	0.409
10.	29	0.918

Tabel 7 memperlihatkan bahwa kesepuluh variasi banyaknya neuron dalam lapisan tersembunyi yang digunakan oleh algoritma LM semuanya menghasilkan nilai signifikansi di atas nilai α (5%). Hal ini berarti rata-rata epoch yang diperlukan oleh algoritma LM untuk mencapai kondisi *error* paling kecil tidak berbeda secara signifikan di antara kesepuluh macam banyaknya neuron dalam lapisan tersembunyi. Meskipun demikian, perlu diketahui rata-rata epoch terkecilnya. Oleh karena

itu, langkah selanjutnya dilakukan analisis deskriptif dengan hasil seperti tersaji pada Tabel 8.

TABEL VIII
RATA-RATA EPOH SETIAP *LEARNING RATE* UNTUK 15 NEURON PADA LAPISAN *INPUT*

No	LR	Rata-rata epoh									
		8	11	13	15	17	19	21	23	27	29
1	0.01	87.68	23.80	20.15	14.20	11.60	11.95	10.95	8.45	8.10	7.45
2	0.05	89.60	20.00	15.05	13.15	13.60	12.80	9.80	8.85	8.70	7.30
3	0.1	53.00	28.35	16.70	14.40	14.35	11.05	10.40	9.65	8.05	7.30
4	0.2	41.79	27.80	14.25	13.35	12.25	12.65	12.90	10.00	8.55	7.60
5	0.3	48.21	14.15	18.80	13.55	12.95	12.00	12.15	10.45	9.25	7.10
6	0.4	37.59	19.70	13.20	12.60	11.95	12.85	9.60	11.15	7.60	7.85
7	0.5	48.35	54.80	14.55	17.75	12.75	11.35	9.95	9.95	8.70	7.30
8	0.6	88.30	42.25	15.65	13.65	13.15	12.15	11.50	11.00	7.55	7.30
9	0.7	102.85	21.75	18.25	14.85	13.70	10.75	11.70	9.20	7.35	6.95
10	0.8	92.15	14.47	17.75	12.55	12.45	11.75	10.95	10.50	8.05	7.95
11	0.9	78.22	49.55	22.20	13.35	12.80	10.55	10.65	10.25	7.90	7.00
12	1	39.05	14.20	33.95	18.25	13.85	11.75	11.30	10.30	8.10	7.40

Data yang dicetak tebal dalam Tabel 8 adalah rata-rata epoh terkecil dalam setiap kelompok banyaknya neuron pada lapisan tersembunyi. Berdasarkan data tersebut, rata-rata epoh terkecil diperoleh pada penggunaan 29 neuron dalam lapisan tersembunyi pada tingkat LR 0.7. Visualisasi data Tabel 8 disajikan pada Gambar 6.

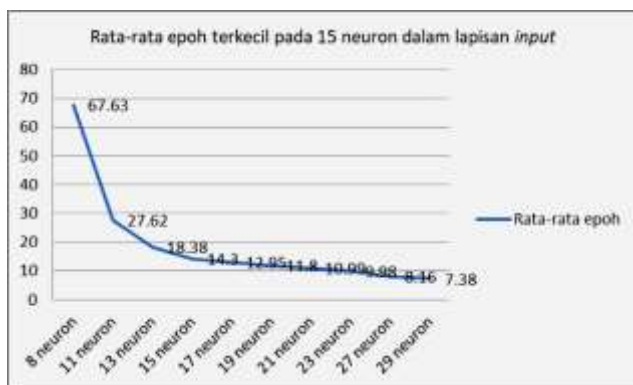


Gambar 6. Grafik rata – rata epoh terkecil berdasarkan *learning rate* pada penggunaan 15 neuron dalam lapisan *input*

Berdasarkan data pada Tabel 8, selanjutnya dilakukan uji ANOVA untuk mendapatkan informasi rata-rata epoh terkecil berdasarkan banyaknya neuron dalam lapisan tersembunyi tanpa melihat nilai LR. Pengujian ini menggunakan taraf $\alpha = 5\%$ dan diperoleh nilai signifikansi sebesar 0.000 yang berarti terdapat perbedaan secara signifikan rata-rata epoh yang diperlukan oleh algoritma LM berdasarkan banyaknya neuron dalam lapisan tersembunyi. Untuk mengetahui pada kondisi berapa neuron dalam lapisan tersembunyi yang memerlukan rata-rata epoh paling sedikit, dilakukan uji komparasi ganda atau MCA menggunakan uji Duncan. Hasil uji disajikan pada Tabel 9 dengan visualisasi pada Gambar 7.

TABEL IX
HASIL UJI DUNCAN PADA SETIAP BANYAKNYA NEURON DALAM LAPISAN TERSEMBUNYI PADA 15 NEURON
DALAM LAPISAN INPUT

Duncan	Neuron dalam lapisan tersembunyi	N	Subset for alpha = 0.05			
			1	2	3	4
Duncan ^{a,b}	29 neuron	240	7.38			
	27 neuron	240	8.16			
	23 neuron	240	9.98	9.98		
	21 neuron	240	10.99	10.99		
	19 neuron	240	11.80	11.80		
	17 neuron	240	12.95	12.95		
	15 neuron	240	14.30	14.30		
	13 neuron	240		18.38		
	11 neuron	239			27.62	
	8 neuron	232				67.63
Sig.			.167	.085	1.000	1.000



Gambar 7. Grafik rata – rata epoch terkecil pada penggunaan 15 neuron dalam lapisan input tanpa mempertimbangkan nilai *learning rate*

Hampir sama dengan pola grafik pada Gambar 5, Gambar 7 juga memperlihatkan pola penurunan jumlah rata-rata epoch dengan bertambahnya jumlah neuron dalam lapisan tersembunyi. Terkait kinerja algoritma LM dengan penggunaan 15 neuron dalam lapisan input telah dianalisis oleh [16] dan menghasilkan MSE terkecil sebesar 0.0000807 ± 0.0001220 dengan laju pembelajaran = 0.8.

IV. PENUTUP

Berdasarkan hasil uji statistik terhadap data epoch, dapat disimpulkan bahwa banyaknya neuron yang paling optimal dalam lapisan tersembunyi yang ditinjau dari epoch yang diperlukan untuk mencapai kondisi *error* paling kecil pada neuron masukan 5 adalah 9 neuron dengan rata-rata epoch 10.80 pada laju

pembelajaran = 0.01. Pada 10 neuron dalam lapisan input, kondisi optimal pada penggunaan 19 neuron dalam lapisan tersembunyi dengan rata-rata epoch 21.52 pada laju pembelajaran = 0.7. Sementara itu, pada 15 neuron dalam lapisan input, diperoleh rata-rata epoch sebesar 7.38 pada laju pembelajaran = 0.7 yang dicapai pada 29 neuron dalam lapisan tersembunyi. Pengujian ini menggunakan taraf $\alpha = 5\%$.

Berdasarkan informasi yang diperoleh dari hasil penelitian ini terkait kinerja dari algoritma pelatihan Levenberg-Marquardt, dapat digunakan sebagai dasar pengembangan sistem berbasis jaringan syaraf tiruan khususnya metode *backpropagation*. Untuk lebih melengkapi informasi mengenai kinerja algoritma pelatihan dalam metode *backpropagation*, disarankan dilakukan penelitian lebih lanjut untuk mengetahui waktu *training* yang paling optimal.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada KEMENRISTEKDIKTI melalui Lembaga Penelitian dan Pengabdian pada Masyarakat (LPPM) Universitas Muhammadiyah Purwokerto yang telah memberikan dana dan fasilitas dalam melaksanakan penelitian ini.

DAFTAR PUSTAKA

- [1] S. Shanmuganathan and S. Samarasinghe, *Artificial Neural Network Modelling*, vol. 628. Springer International Publishing, 2016.
- [2] H. Mustafidah, S. Hartati, R. Wardoyo, and A. Harjoko, "Prediction of Test Items Validity Using Artificial Neural Network," in *Proceeding International*

- Conference on Education, Technology, and Science (NETS) 2013, "Improving The Quality Of Education To Face The Impact Of Technology". December 28th, 2013, 2013.*
- [3] H. Mustafidah, D. K. Hakim, and S. Sugiyanto, "Tingkat Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan (Studi Kasus Prediksi Prestasi Belajar Mahasiswa) Optimization Level of Training Algorithms in Neural Network (Case Studies of Student Learning Achievement Predictions)," *JUITA*, vol. II, no. 3, pp. 159–166, 2013.
 - [4] D. C. Febrianto and H. Mustafidah, "Penerapan Jaringan Syaraf Tiruan dengan Metode Pembelajaran Backpropagation untuk Mengetahui Tingkat Kualifikasi Calon Siswa pada Sistem Informasi Penerimaan Siswa Baru di MAN 2 Banjarnegara," *JUITA (Jurnal Inform., vol. II, no. 3, pp. 189–197, 2013.*
 - [5] A. Hermawan, *Jaringan Saraf Tiruan (Teori dan Aplikasinya)*. Yogyakarta: Andi Offset, 2006.
 - [6] S. Kusumadewi, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Yogyakarta: Graha Ilmu, 2004.
 - [7] H. Mustafidah, S. Hartati, R. Wardoyo, and A. Harjoko, "Selection of Most Appropriate Backpropagation," *Int. J. Comput. Trends Technol.*, vol. 14, no. 2, pp. 92–95, 2014.
 - [8] H. Mustafidah and S. Suwarsito, "Error Rate Testing of Training Algorithm in Back Propagation Network," *Int. J. Soft Comput. Eng.*, vol. 5, no. 4, pp. 46 – 50, 2015.
 - [9] H. Mustafidah and S. Suwarsito, "Uji Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan," in *Prosiding Seminar Nasional SENATKOM 2015*, 2015, pp. 243–248.
 - [10] H. Mustafidah and H. Harjono, "Korelasi Tingkat Kesalahan dan Epoch dalam Jaringan Backpropagation," in *Prosiding SEMNASTIKOM 2017, 3 November 2017, ISBN: 978-602-50434-0-6*, 2017, pp. 55–61.
 - [11] E. Paulus and Y. Nataliani, *GUI Matlab*. Yogyakarta: ANDI, 2007.
 - [12] F. A. Irawan, *Buku Pintar Pemograman Matlab*. Yogyakarta: MediaKom - Andi, 2012.
 - [13] T. Taniredja and H. Mustafidah, *Penelitian Kuantitatif (Sebuah Pengantar)*. Bandung: ALFABETA, 2011.
 - [14] H. Mustafidah and S. Suwarsito, "Model Parameter Jaringan Syaraf Tiruan untuk Pemilihan Algoritma Pelatihan Jaringan Backpropagation yang Paling Optimal," Purwokerto, Central Java, Indonesia, 2015.
 - [15] H. Mustafidah and S. Suwarsito, "Testing Design of Neural Network Parameters in Optimization Training Algorithm," in *International Conference of Result and Community Services, 6th August 2016*, 2016, p. THN. 139-146.
 - [16] H. Mustafidah, A. Y. Rahmadhani, and H. Harjono, "Optimasi Algoritma Pelatihan Levenberg–Marquardt Berdasarkan Variasi Nilai Learning-Rate dan Jumlah Neuron dalam Lapisan Tersembunyi," *JUITA J. Inform.*, vol. 7, no. 1, pp. 55–62, 2019.

