

# Optimasi Algoritma Pelatihan *Levenberg–Marquardt* Berdasarkan Variasi Nilai *Learning-Rate* dan Jumlah Neuron dalam Lapisan Tersembunyi (*Optimization of Levenberg-Marquardt Training Algorithm Based on the Variations Value of Learning-Rate and the Number of Neurons in the Hidden Layer*)

Hindayati Mustafidah<sup>1)</sup>, Amrisa Yanri Rahmadhani<sup>2)</sup>, Harjono<sup>3)</sup>

<sup>1,2,3)</sup>Teknik Informatika - Fakultas Teknik dan Sains - Universitas Muhammadiyah Purwokerto  
Jl. Raya Dukuwaluh Purwokerto 53182

<sup>1)</sup>fida.mustafidah@ump.ac.id

<sup>2)</sup>amrisayanril@gmail.com

<sup>3)</sup>pakjono@gmail.com

**Abstrak** -*Backpropagation* (BP) merupakan salah satu paradigma pembelajaran dalam jaringan syaraf tiruan yang dibangun dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron pada lapisan tersembunyi. Perubahan bobot dilakukan oleh sebuah algoritma pelatihan sehingga mendekati bobot optimal. Kinerja algoritma pelatihan dikatakan optimal dapat dilihat dari *error* yang dihasilkan (MSE). Semakin kecil MSE, semakin optimal kinerjanya. Beberapa penelitian yang dilakukan sebelumnya, diperoleh informasi bahwa algoritma pelatihan yang paling optimal berdasarkan MSE terkecil adalah algoritma *Levenberg–Marquardt* (LM) dengan rata-rata MSE 0.001 dengan tingkat pengujian  $\alpha=5\%$ . Penelitian tersebut menggunakan 10 neuron dalam lapisan tersembunyi. Keadaan optimal banyaknya neuron dalam lapisan tersembunyi yang dapat menghasilkan MSE paling kecil belum diketahui. Oleh karena itu, dalam penelitian ini dilakukan analisis untuk mengetahui jumlah neuron yang paling optimal dalam lapisan tersembunyi pada algoritma LM. Parameter jaringan yang digunakan yaitu 15 neuron dalam lapisan *input*, target *error*=0.001 ( $10^{-3}$ ) dan variasi nilai laju pembelajaran. Penelitian ini menggunakan metode campuran yaitu pengembangan program komputer dan pengujian data statistik menggunakan ANOVA. Pengembangan program komputer digunakan untuk membangkitkan data random sebagai masukan jaringan, menjalankan

algoritma LM, dan menghasilkan data keluaran. Hasil analisis menunjukkan bahwa dengan 27 neuron pada lapisan tersembunyi menghasilkan MSE terkecil yaitu  $0.0000807 \pm 0.0001220$  dengan laju pembelajaran=0.8.

**Kata-kata Kunci:** *backpropagation*, MSE, neuron, *Levenberg–Marquardt*, optimal.

*Abstract* -*Backpropagation* (BP) is one of the learning paradigms in artificial neural networks that are built with many layers to change the weights associated with neurons in hidden layers. Changing the weight is done by a training algorithm so that it approaches the optimal weight. The performance of the training algorithm is said to be optimal can be seen from the resulting error (MSE). The smaller the MSE, the more optimal the performance. Several previous studies have obtained information that the most optimal training algorithm based on the smallest MSE is the *Levenberg – Marquardt* (LM) algorithm with an average of MSE 0.001 with a test level of  $\alpha = 5\%$ . The study used 10 neurons in a hidden layer. The optimal state of the number of neurons in hidden layers that can produce the smallest MSE is unknown. Therefore, in this study an analysis was carried out to determine the most optimal number of neurons in the hidden layer of the LM algorithm. The network parameters used are 15 neurons in the input layer, target error = 0.001 ( $10^{-3}$ )

and variations in the value of the learning rate. This study uses a mixed method, namely the development of computer programs and statistical data testing using ANOVA. Development of computer programs is used to generate random data as network input, run LM algorithms, and produce output data. The results of the analysis showed that with 27 neurons in the hidden layer produces the smallest MSE was  $0.0000807 \pm 0.0001220$  with the learning rate = 0.8.

**Key words:** *backpropagation, MSE, neuron, Levenberg – Marquardt, optimal.*

## I. PENDAHULUAN

Jaringan syaraf tiruan (*artificial neural network/JST*) adalah model komputasi yang terinspirasi secara biologis. JST terdiri dari beberapa elemen pengolahan (*neuron*) dan ada hubungan antara neuron [1]. Setiap pengolahan elemen 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 di antara lapisan *input* dan *output* disebut dengan lapisan tersembunyi (*hidden layer*). JST dengan pola lapisan seperti ini disebut JST dengan model *multilayer*. *Backpropagation* merupakan algoritma pembelajaran dalam JST yang terawasi (*supervised learning*) yang menggunakan model *multilayer*. Pembelajaran *backpropagation* menggunakan pola penyesuaian bobot untuk mencapai nilai kesalahan minimum antara keluaran hasil prediksi dengan keluaran yang nyata [2]. 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, Resilient Backpropagation, BFGS, One Step Secant, Levenberg-Marquardt*.

Beberapa penelitian pendahuluan terkait dengan penerapan JST adalah seperti yang dilakukan oleh [3], [4], [5], [6], [7]. Sampai dengan tahap ini baru diterapkan sebuah algoritma pelatihan untuk membantu memecahkan suatu kasus dan belum diuji algoritma pelatihan yang lain.

Berdasarkan penelitian yang dilakukan oleh [8] dalam melakukan pengujian terhadap 12 algoritma pelatihan dalam jaringan *backpropagation* diperoleh informasi bahwa algoritma *Levenberg–Marquardt* merupakan algoritma pelatihan yang paling optimal pada tingkat laju pembelajaran (*learning rate/lr*) =

0.05 dengan rata-rata *error* 0.0002196. Demikian juga hasil penelitian [9], [8], [10], [11], [12], [13] menghasilkan informasi bahwa algoritma *Levenberg–Marquardt* merupakan algoritma pelatihan yang menghasilkan *error* terkecil sebesar 0.001001 dengan lapisan masukan sebanyak 5, 10, dan 15 neuron dengan 1 neuron pada lapisan *output*. Hal ini beralasan karena algoritma *Levenberg–Marquardt* menggunakan perbaikan metode *Newton* yang diketahui sebagai metode yang sangat cepat dan akurat untuk mendapatkan *error* minimum [14]. Dalam penelitian tersebut digunakan 10 neuron dalam lapisan tersembunyi. Sementara itu, banyaknya neuron dalam lapisan tersembunyi akan sangat berpengaruh terhadap kinerja jaringan terutama dalam *error* yang dihasilkan yang berdampak pada tingkat keakuratan keluaran jaringan [15]. Secara teori, semakin banyak neuron dalam lapisan tersembunyi akan semakin akurat keluaran yang dihasilkan, namun kinerja jaringan semakin lambat.

Penentuan jumlah neuron yang digunakan dalam lapisan tersembunyi mengikuti aturan: 1) jumlah neuron dalam lapisan tersembunyi baiknya di antara ukuran lapisan masukan dan lapisan keluaran, atau 2) jumlah neuron dalam lapisan tersembunyi sebaiknya 2/3 dari ukuran lapisan masukan ditambah ukuran lapisan keluaran, atau 3) jumlah neuron dalam lapisan tersembunyi sebaiknya lebih kecil atau kurang dari dua kali ukuran neuron dalam lapisan *input* [15]. Informasi mengenai banyaknya neuron dalam lapisan tersembunyi yang paling optimal kinerjanya belum diketahui. Oleh karena itu, penelitian ini dilakukan analisis dan pengujian terhadap variasi banyaknya neuron dalam lapisan tersembunyi menggunakan algoritma pelatihan *Levenberg–Marquardt*. Karena jumlah neuron dalam lapisan input yang digunakan sebanyak 15, maka banyaknya neuron dalam lapisan tersembunyi digunakan 11, 15, 19, 23, dan 27 neuron.

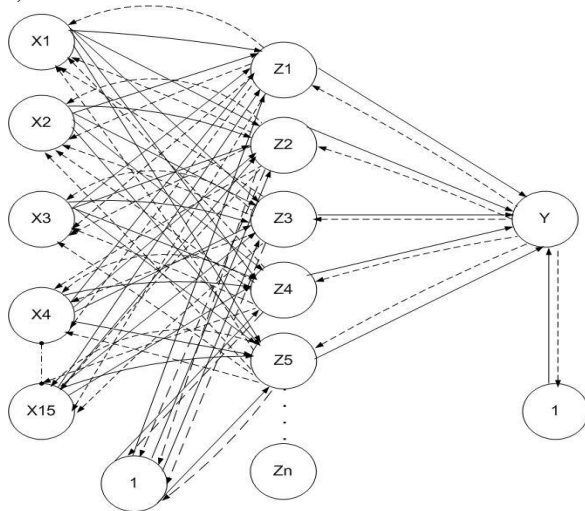
## II. METODE PENELITIAN

Penelitian ini merupakan penelitian pengembangan dengan pengujian kuantitatif dan kualitatif (menggunakan uji statistik). Pengembangan kode program dilakukan menggunakan bahasa pemrograman MATLAB karena terdapat *toolbox neural network* yang bisa dipakai untuk fungsi – fungsi yang berkaitan dengan jaringan syaraf tiruan [16]. Selanjutnya data keluaran program dianalisis menggunakan uji statistika dengan SPSS.

**A. Variabel Penelitian**

Variabel penelitian berupa parameter jaringan dengan arsitektur jaringan seperti pada Gambar 1. Parameter jaringan yang dimaksud yaitu:

- 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 masukan (X): 15
- neuron keluaran (Y) : 1
- neuron dalam lapisan tersembunyi (Z) : 11, 15, 19, 23, 27



**Gambar 1. Desain Jaringan Syaraf Tiruan dengan 15 Neuron Masukan, n Neuron pada Lapisan Tersembunyi (n = 11, 15, 19, 23, 27) dan 1 Neuron pada Lapisan Keluaran.**

**B. Sumber Data**

Sumber data neuron masukan dan neuron target yang digunakan dalam penelitian ini adalah data dokumentasi yang diambil dari penelitian [10].

**C. Desain Penelitian**

Berikut merupakan desain struktur program jaringan syaraf tiruan (JST) yang dibangun sebagai hasil modifikasi dari penelitian [17] seperti terlihat pada Gambar 2.

**D. Analisis Data**

Data keluaran jaringan yang dijalankan menggunakan algoritma pelatihan Levenberg-Marquardt (LM) diuji untuk setiap nilai lr dengan suatu banyak neuron tertentu yaitu 11, 15, 19, 23, dan 27 untuk mendapatkan error (MSE). Desain analisis data ini tersaji pada Gambar 3 dengan tahapan uji ANAVA sebagai berikut [18]:

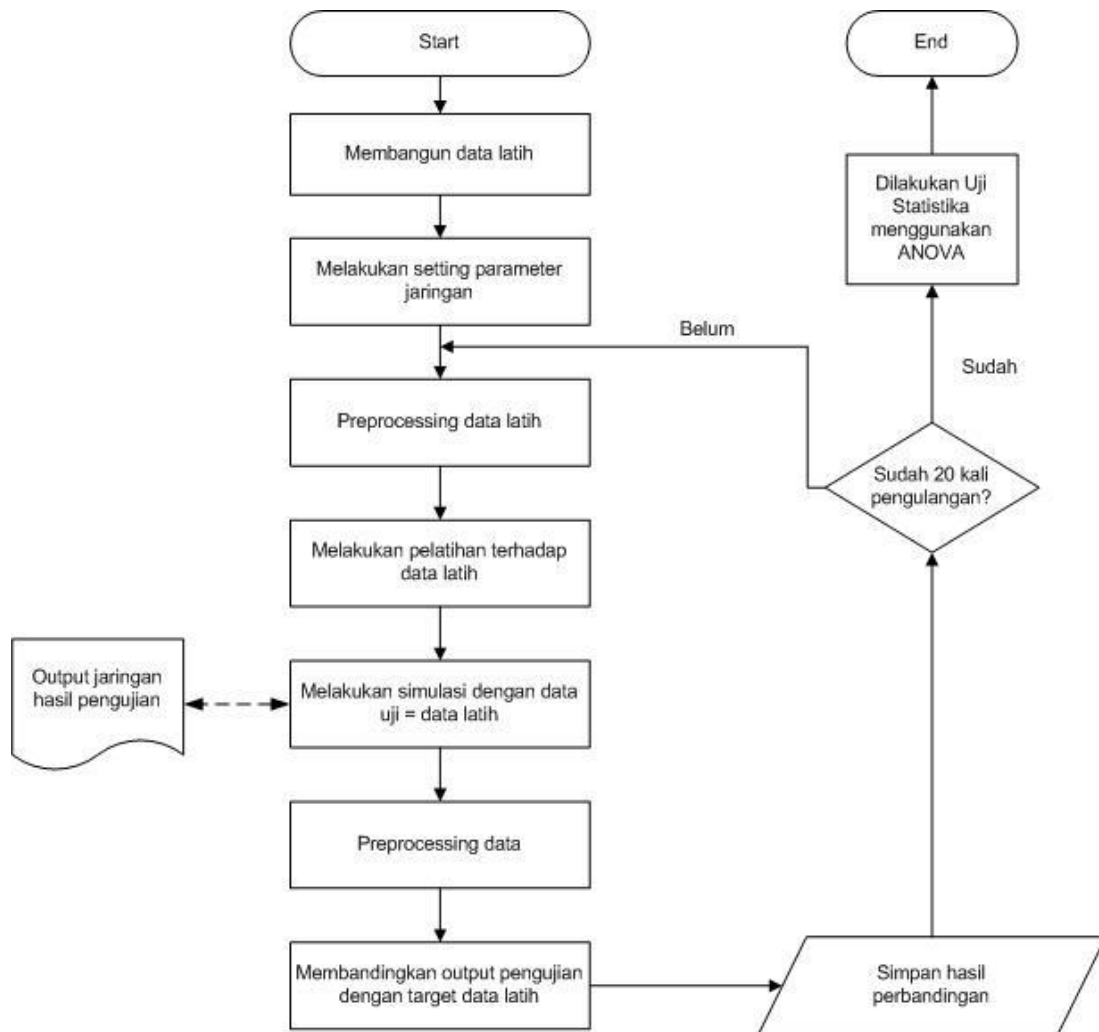
*1) Menentukan hipotesis.*

$H_0$  : tidak terdapat perbedaan error yang dihasilkan oleh algoritma LM pada setiap nilai lr untuk sejumlah neuron dalam lapisan tersembunyi.

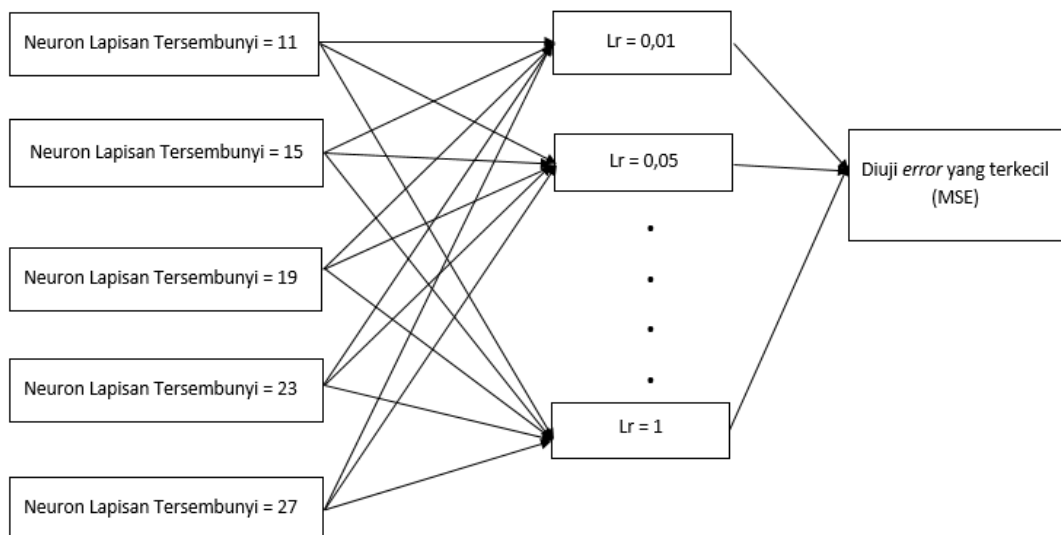
$H_1$  : terdapat perbedaan error yang dihasilkan oleh algoritma LM pada setiap nilai lr untuk sejumlah neuron dalam lapisan tersembunyi.

*2) Menentukan nilai alpha ( $\alpha$ ) dalam penelitian ini digunakan  $\alpha = (5\%)$ .*

*3) Pengambilan kesimpulan.* Kesimpulan diambil berdasarkan nilai signifikan yang diperoleh (sig.) dengan ketentuan  $H_0$  ditolak jika nilai sig <  $\alpha$  dan ketentuan  $H_0$  diterima jika nilai sig  $\geq \alpha$ .



**Gambar 2. Alur Pengembangan Program JST**



**Gambar 3. Desain Pengujian Statistik untuk Setiap *Learning Rate***

III.HASIL DAN PEMBAHASAN

A. Data Penelitian

Data penelitian berupa data *random* baik data masukan maupun data target seperti pada TABEL I.  $X_1$  s.d.  $X_{15}$  adalah data masukan, sedangkan Y adalah data target [10].

TABEL I. DATA MASUKAN JARINGAN DAN TARGET DENGAN 15 NEURON PADA LAPISAN TERSEMBUNYI

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
9.5013	4.0571	0.1527	5.0281	4.9655	7.0274	1.3652	2.2595
2.3114	9.3547	7.4679	7.0947	8.9977	5.4657	0.1176	5.7981
6.0684	9.1690	4.4510	4.2889	8.2163	4.4488	8.9390	7.6037
4.8598	4.1027	9.3181	3.0462	6.4491	6.9457	1.9914	5.2982
8.9130	8.9365	4.6599	1.8965	8.1797	6.2131	2.9872	6.4053
7.6210	0.5789	4.1865	1.9343	6.6023	7.9482	6.6144	2.0907
4.5647	3.5287	8.4622	6.8222	3.4197	9.5684	2.8441	3.7982
0.1850	8.1317	5.2515	3.0276	2.8973	5.2259	4.6922	7.8333
8.2141	0.0986	2.0265	5.4167	3.4119	8.8014	0.6478	6.8085
4.4470	1.3889	6.7214	1.5087	5.3408	1.7296	9.8833	4.6110
6.1543	2.0277	8.3812	6.9790	7.2711	9.7975	5.8279	5.6783
7.9194	1.9872	0.1964	3.7837	3.0929	2.7145	4.2350	7.9421
9.2181	6.0379	6.8128	8.6001	8.3850	2.5233	5.1551	0.5918
7.3821	2.7219	3.7948	8.5366	5.6807	8.7574	3.3395	6.0287
1.7627	1.9881	8.3180	5.9356	3.7041	7.3731	4.3291	0.5027

TABEL 1. (LANJUTAN)

$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	Y
4.1537	4.1195	4.5142	0.5758	1.2105	1.9089	9.3424	6.8732
3.0500	7.4457	0.4390	3.6757	4.5075	8.4387	2.6445	3.4611
8.7437	2.6795	0.2719	6.3145	7.1588	1.7390	1.6030	1.6603
0.1501	4.3992	3.1269	7.1763	8.9284	1.7079	8.7286	1.5561
7.6795	9.3338	0.1286	6.9267	2.7310	9.9430	2.3788	1.9112
9.7084	6.8333	3.8397	0.8408	2.5477	4.3979	6.4583	4.2245
9.9008	2.1256	6.8312	4.5436	8.6560	3.4005	9.6689	8.5598
7.8886	8.3924	0.9284	4.4183	2.3235	3.1422	6.6493	4.9025
4.3866	6.2878	0.3534	3.5325	8.0487	3.6508	8.7038	8.1593
4.9831	1.3377	6.1240	1.5361	9.0840	3.9324	0.0993	4.6077
2.1396	2.0713	6.0854	6.7564	2.3189	5.9153	1.3701	4.5735
6.4349	6.0720	0.1576	6.9921	2.3931	1.1975	8.1876	4.5069
3.2004	6.2989	0.1635	7.2751	0.4975	0.3813	4.3017	4.1222
9.6010	3.7048	1.9007	4.7838	0.7838	4.5860	8.9032	9.0161
7.2663	5.7515	5.8692	5.5484	6.4082	8.6987	7.3491	0.0558

B. Analisis Data

Data masukan dan terget dijalankan (dilatih dan disimulasikan) menggunakan algoritma LM untuk setiap nilai lr dan setiap n banyaknya neuron dalam lapisan tersembunyi (dengan n = 11, 15, 19, 23, 27) dengan masing-masing dilakukan pengulangan sebanyak 20 kali. Pelatihan dan simulasi ini

ditujukan untuk mendapatkan data MSE yang selanjutnya dianalisis menggunakan uji statistik ANAVA. Hasil analisis untuk setiap banyaknya neuron dalam lapisan tersembunyi berdasarkan nilai lr tersaji pada TABEL II.

TABEL II. PENCAPAIAN NILAI SIG. PADA PENGUJIAN ALGORITMA LM UNTUK SETIAP JUMLAH NEURON DALAM LAPISAN TERSEMBUNYI

Banyak neuron pada lapisan tersembunyi	Nilai sig.
11	.463
15	.886
19	.927
23	.232
27	.164

Berdasarkan Tabel 2 diketahui bahwa semua nilai signifikansi di atas  $\alpha$  (5%). Hal ini berarti  $H_0$  diterima sehingga disimpulkan bahwa tidak terdapat perbedaan *error* secara signifikan untuk setiap nilai *lr* pada masing-masing jumlah neuron dalam lapisan tersembunyi. Dengan kata lain, algoritma LM memiliki tingkat akurasi yang relatif sama ditinjau dari *error* dengan variasi nilai *lr* yang diberikan. Namun dalam setiap nilai *lr* yang diberikan, diperoleh data rata-rata MSE yang dapat dijadikan sebagai acuan dalam menetapkan nilai *lr* seperti pada TABEL III.

TABEL III. RATA – RATA MSE UNTUK SETIAP NILAI LR

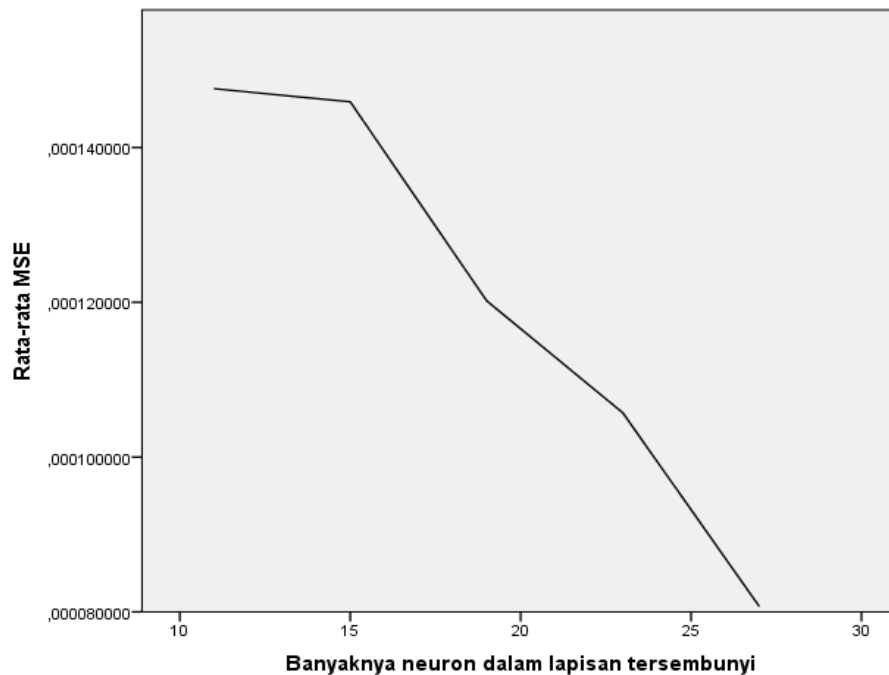
lr	Banyaknya neuron dalam lapisan tersembunyi				
	11	15	19	23	27
0.01	0.0003062±	0.0002576±	0.0001202±	0.0001992±	0.0001213±
	0.0003646	0.0002821	0.0001397	0.0002651	0.0002012
0.05	0.0001476±	0.0002455±	0.0002306±	0.0003255±	0.0003056±
	0.0002252	0.0003201	0.0002697	0.0003179	0.0003473
0.1	0.0003004±	0.0001737±	0.0001898±	0.0002255±	0.0001821±
	0.0003457	0.0002402	0.0002221	0.0002205	0.0002611
0.2	0.0001917±	0.0002282±	0.0002344±	0.0002532±	0.0003214±
	0.0002294	0.0003128	0.0002792	0.0002622	0.0003455
0.3	0.0002777±	0.0001459±	0.0002470±	0.0001730±	0.0002474±
	0.0002638	0.0002675	0.0003049	0.0002450	0.0002703
0.4	0.0001882±	0.0002167±	0.0002376±	0.0001057±	0.0002666±
	0.0002692	0.0002410	0.0003538	0.0001241	0.0003095
0.5	0.0002090±	0.0002110±	0.0001932±	0.0001609±	0.0001802±
	0.0002921	0.0002484	0.0002724	0.0002423	0.0002255
0.6	0.0003614±	0.0001560±	0.0002456±	0.0001246±	0.0001720±
	0.0003247	0.0002427	0.0002981	0.0002175	0.0002945
0.7	0.3155141±	0.0002150±	0.0001590±	0.0002237±	0.0002668±
	1.4102518	0.0002896	0.0002527	0.0003299	0.0002962
0.8	0.0305879±	0.0002302±	0.0002164±	0.0001384±	0.0000807±
	0.1358577	0.0002437	0.0002659	0.0001992	0.0001220
0.9	0.0002570±	0.0002843±	0.0001566±	0.0002445±	0.0002453±
	0.0003033	0.0003156	0.0002381	0.0002339	0.0002890
1	0.0002711±	0.0001539±	0.0002114±	0.0002356±	0.0001857±
	0.0002568	0.0002407	0.0002691	0.0002563	0.0002429

TABEL III memperlihatkan rata-rata MSE untuk setiap nilai *lr* dari masing-masing jumlah neuron dalam lapisan tersembunyi. Rangkuman masing-masing MSE terkecil ini disajikan pada TABEL IV.

Data pada TABEL IV menunjukkan bahwa *error* (MSE) terkecil pada banyaknya neuron lapisan tersembunyi terdapat pada neuron 27 dengan nilai *lr* = 0.8 dengan rata-rata 0.0000807± 0.0001220. Secara grafikal, keadaan ini digambarkan pada Gambar 4.

TABEL IV. RATA – RATA MSE TERKECIL BERDASARKAN LR PADA SETIAP BANYAK NEURON LAPISAN TERSEMBUNYI

Banyaknya Neuron dalam Lapisan Tersembunyi	lr	Rata-rata MSE
Neuron 11	0.05	0.0001476± 0.0002252
Neuron 15	0.3	0.0001459± 0.0002675
Neuron 19	0.01	0.0001202± 0.0001397
Neuron 23	0.4	0.0001057± 0.0001241
Neuron 27	0.8	0.0000807± 0.0001220



**Gambar 4. Deskripsi visual dari MSE algoritma pelatihan LM Berdasarkan Banyaknya Neuron dalam Lapisan Tersembunyi**

Gambar 4 memperlihatkan bahwa semakin banyak neuron dalam lapisan tersembunyi, maka semakin kecil error yang dihasilkan oleh jaringan khususnya pada algoritma pelatihan LM. Hal ini sesuai dengan konsep teori yang disampaikan dalam [15] dan sebelumnya oleh [19] bahwa semakin banyak neuron dalam lapisan tersembunyi, akan semakin teliti keluaran yang dihasilkan oleh jaringan, namun waktu yang dibutuhkan semakin lama.

#### IV. PENUTUP

Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan bahwa penggunaan 27 neuron pada lapisan tersembunyi menyebabkan algoritma pelatihan LM menghasilkan error paling kecil. Dengan kata lain algoritma LM dengan model 15-27-1 memiliki kinerja optimal ditinjau dari error yang dihasilkan yaitu sebesar  $0.0000807 \pm 0.0001220$ .

Penelitian ini perlu dikembangkan lagi untuk dianalisis keoptimalan dari algoritma pelatihan LM ditinjau dari kecepatan algoritma dalam menghasilkan keluaran dengan didasarkan atas error yang paling kecil.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada 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] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu, 2003.
- [3] H. Harjono and D. Aryanto, "Application of Artificial Neural Networks to Predict Student Achievement Study," *SAINTEKS*, vol. 5, no. 2, 2009.
- [4] H. Mustafidah, D. Aryanto, and D. K. Hakim, "Uji Optimalisasi Algoritma Pelatihan Conjugate Gradient pada Jaringan Syaraf Tiruan," in *Prosiding SENATEK, ISBN: 978-602-14355-0-2, 21 September 2013*, 2013, p. B-9-1.
- [5] H. Mustafidah, D. K. Hakim, and S. Sugiyanto, "Tingkat Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan (Studi Kasus Prediksi Prestasi Belajar Mahasiswa)," *JUITA (Jurnal Inform., vol. II, no. 3, pp. 159 – 166, 2013.*

- [6] F. Wibowo, S. Sugiyanto, and H. Mustafidah, "Tingkat Ketelitian Pengenalan Pola Data pada Algoritma Pelatihan Perbaikan Metode Batch Mode dalam Jaringan Syaraf Tiruan," *JUITA (Jurnal Inform.*, vol. II, no. 4, pp. 259 – 264, 2013.
- [7] 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.
- [8] H. Mustafidah and S. Suwarsito, "Uji Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan," in *Prosiding Seminar Nasional SENATKOM 2015*, 2015, pp. 243–248.
- [9] S. Suwarsito and H. Mustafidah, "Knowledge Representation in the Determination of Fish Under Water Quality," in *International Conference of Result and Community Services*, 2016, no. August, pp. 157–164.
- [10] 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.
- [11] H. Mustafidah and S. Suwarsito, "Correlation Analysis between Error Rate of Output and Learning Rate in Backpropagation Network," in *International Summit on Knowledge Advancements (ISKA 2017)*, 26-27 July 2017, 2017.
- [12] H. Mustafidah and S. Suwarsito, "Correlation Analysis Between Error Rate of Output and Learning Rate in Backpropagation Network," *Adv. Sci. Lett.*, vol. 24, no. 12, pp. 9182–9185, 2018.
- [13] 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.
- [14] S. Kusumadewi, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Yogyakarta: Graha Ilmu, 2004.
- [15] J. Heaton, *Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning*, 1.0. Chesterfield, USA: Heaton Research Inc., 2015.
- [16] F. A. Irawan, *Buku Pintar Pemograman Matlab*. Yogyakarta: MediaKom - Andi, 2012.
- [17] 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.
- [18] T. Taniredja and H. Mustafidah, *Penelitian Kuantitatif (Sebuah Pengantar)*. Bandung: ALFABETA, 2011.
- [19] J. Heaton, *Introduction to Neural Networks with C#*, Second. St. Louis: Heaton Research, Inc., 2008.