

## INFERENSI TINGKAT KESALAHAN DALAM JARINGAN BACKPROPAGATION BERDASARKAN LAJU PEMAHAMAN

Hidayati Mustafidah<sup>1</sup>, Suwarsito<sup>2</sup>

(1) Teknik Informatika – Universitas Muhammadiyah Purwokerto,  
(085842817313, h.mustafidah@ump.ac.id/fida.mustafidah@gmail.com)

(2) Pendidikan Geografi – Universitas Muhammadiyah Purwokerto  
(081327108877, ito\_warsito@yahoo.co.in)

### Abstract

Backpropagation network as a form of Artificial Neural Network (ANN) has been widely applied to help solve problems in various areas of life, such as forecasting, diagnostics, and pattern recognition. Performance of ANN is determined by training algorithm. In back propagation network, there are 12 training algorithms that can be used. Determining the most optimal algorithm will be tested one indicator that is the resulting error. The smaller the error, the more optimal performance of the algorithm. This study inferred statistically against training algorithms in the backpropagation network based on the variation in the  $\eta$  using ANOVA test. The network parameter values that used are the target error = 0.001, the maximum epoch = 10000, and variations in the value of  $\eta$  are 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1. Data input is given by the randomization with a variety of many neurons in the input layer, are 5, 10, and 15, and with 1 neuron in the output layer. ANOVA test using  $\alpha = 5\%$  resulted a conclusion that the Levenberg-Marquardt is the most optimal training algorithm with an average MSE of 0.001001.

**Keywords:** backpropagation, Artificial Neural Network, training algorithm, error, Levenberg-Marquardt

### 1. Pendahuluan

Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* (ANN) merupakan salah satu komponen pembentuk model *Soft Computing* [1]. JST memberikan solusi ideal untuk permasalahan-permasalahan yang tidak dapat diformulasikan dengan mudah menggunakan algoritma [2]. Jaringan syaraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologis, yang dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologis tersebut [3]. Beberapa bahasan yang dapat dikaji pada JST lebih banyak pada aplikasi dan pengembangan algoritma-algoritma pembelajaran/pelatihan. *Backpropagation* adalah algoritma pembelajaran yang terawasi dan paling banyak digunakan, dengan lebih dari satu lapisan (*multi layer*) untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma backpropagation menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* (kesalahan) ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensialkan. Fungsi aktivasi ini harus memenuhi beberapa syarat yaitu: kontinu, terdiferensialkan dengan mudah, dan merupakan fungsi yang tidak

turun. Terdapat beberapa algoritma pelatihan yang terdapat dalam metode *backpropagation* [4], diantaranya adalah *Fletcher-Reeves Update*, *Polak-Ribière*, *Powell-Beale Restarts*, *Scaled Conjugate Gradient*, *Gradient Descent*, *Gradient Descent* dengan *Adaptive Learning Rate*, *Gradient Descent* dengan *Momentum* dan *Adaptive Learning Rate*, *Resilient Backpropagation*, *BFGS*, *One Step Secant*, dan *Levenberg-Marquardt*.

Beberapa algoritma pelatihan secara parsial telah diuji dan diterapkan dalam suatu kasus untuk menyelesaikan masalah, namun belum diuji secara menyeluruh di antara algoritma-algoritma yang ada. Algoritma pelatihan *Gradient Descent* telah diterapkan dalam menyelesaikan permasalahan 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 [5] dan menghasilkan tingkat *error* sebesar 0.0664179 dari target *error* 0,05. Algoritma pelatihan *Fletcher-Reeves Update*, *Polak-Ribière*, *Powell-Beale Restarts*, dan *Scaled Conjugate Gradient* telah diuji keoptimalannya dalam penyelesaian kasus prediksi prestasi belajar mahasiswa [6] dan dihasilkan kesimpulan bahwa dari keempat algoritma tersebut dengan  $\alpha = 5\%$  tidak berbeda keoptimalannya secara signifikan. Selain itu, beberapa algoritma pembelajaran/pelatihan juga telah diuji

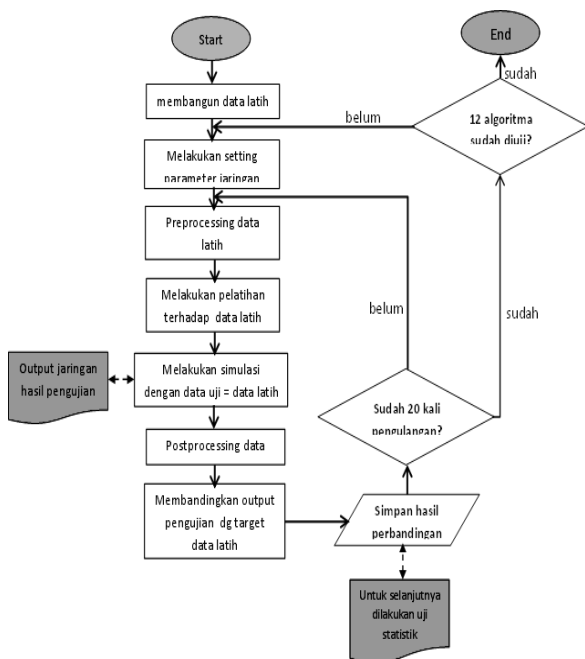
keoptimalannya oleh [7] dan [8] dengan hasil informasi bahwa dengan tingkat kepercayaan 95% diperoleh hasil bahwa algoritma *Levenberg-Marquardt* merupakan algoritma yang paling teliti dengan rata-rata error 0,0063. Sementara itu [9] juga telah menerapkan jaringan syaraf tiruan untuk memprediksi tingkat validitas soal dengan menggunakan algoritma pelatihan *Levenberg-Marquardt* dan menghasilkan pengenalan pola data dengan memiliki kecocokan sebesar 86,54% dengan tingkat error 0.00063347. Oleh karena itu, dalam penelitian ini dilakukan pengujian tingkat keoptimalan dari algoritma pelatihan ditinjau dari error jaringan yang dihasilkan.

**2. Metode**

Penelitian ini diawali dengan mengembangkan program komputer menggunakan Bahasa Pemrograman MATLAB untuk menjalankan algoritma pelatihan. Masukan program berupa data random dengan struktur 5 neuron, 10 neuron, dan 15 neuron dalam layer input, dengan 1 neuron pada layer output. Variabel penelitian digunakan sebagai berikut:

- Variabel control (nilai bervariasi): maksimum epoch (=10.000), target error (=10<sup>-3</sup>).
- Var independen (nilai bervariasi): data masukan, *learning rate* (lr)
- Var dependen/faktor: algoritma pelatihan
- Output: tingkat error yang dihasilkan oleh setiap algoritma (berdasarkan variasi input yang diberikan dan variasi variabel control dan variable independen):

Setiap algoritma dijalankan sebanyak 20 kali. Alur penelitian disajikan pada Gambar 1.



Gambar 1. Alur Penelitian

Selanjutnya dilakukan pengujian statistik inferensi terhadap *output* yang dihasilkan dan menyimpulkan. Pengujian dilakukan untuk menentukan optimalisasi dari algoritma-algoritma pelatihan tersebut dengan membandingkan tingkat *error* yang terjadi dengan tahapan langkah sebagai berikut [10]:

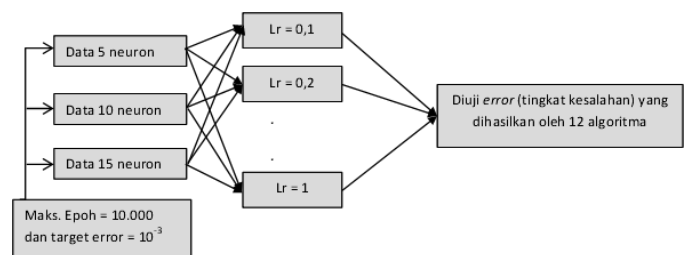
- 1) menentukan hipotesis  
 $H_0 : \mu_1 = \mu_2 = \dots = \mu_k$  (semua algoritma pelatihan bersifat homogen/memiliki tingkat kesalahan yang sama)  
 $H_1 : \text{minimal 2 mean tidak sama (ada beberapa algoritma pelatihan yang tidak homogen / tidak memiliki tingkat kesalahan yang sama)}$
- 2) menentukan nilai alpha ( $\alpha$ )
- 3) menentukan alat uji  
 Alat uji yang digunakan dalam hal ini adalah uji F seperti pada persamaan 1.

$$F_{hit} = \frac{MST}{MSE} \sim F_{k-1, N-k} \dots\dots\dots (1)$$

dimana MST = Mean Square of Treatment  
 MSE = Mean Square of Error

- 4) pengambilan kesimpulan/inferensi  
 Inferensi diambil berdasarkan nilai signifikansi yang diperoleh (sig.) dengan ketentuan  $H_0$  ditolak jika nilai sig <  $\alpha$ .

Pengujian statistik dilakukan dengan desain berikut (Gambar 2).



Gambar 2. Desain Pengujian Statistik terhadap Algoritma Pelatihan

**3. Pembahasan**

Penelitian ini merupakan pengembangan dari yang dilakukan oleh [11] yang menghasilkan simpulan algoritma *Levenberg-Marquardt* sebagai algoritma yang menghasilkan error terkecil yaitu 0,0001986858 dengan parameter-parameter target error = 0,001 (10<sup>-3</sup>), epoch maksimum = 10000, *learning rate* (lr) = 0.01, dengan 5 neuron masukan dan 1 neuron keluaran yang dibangkitkan secara random. Penelitian ini dilanjutkan dengan *learning rate* (lr) = 0,05 dan parameter jaringan yang sama dan dihasilkan algoritma *trainlm* sebagai algoritma yang paling optimal dengan rata-rata tingkat error sebesar 0,0002196 [12].

Sebagaimana dalam kedua penelitian [11] dan [12], parameter atau variabel control dalam penelitian berupa maksimum epoch (=10.000) dan target error (=10<sup>-3</sup>), sedangkan variabel independen berupa data

masukan (nilai bervariasi dengan 5, 10, dan 15) dan *learning rate* (lr) dengan nilai 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 0.01, dan 0.05. Keduabelas algoritma pelatihan (sebagai variabel dependen atau variabel faktor) digunakan untuk menguji tingkat kesalahan (*error*).

Data penelitian terdiri dari data masukan dan data target jaringan. Data masukan jaringan yang terdapat dalam lapisan masukan memiliki variasi 5 neuron, 10 neuron, dan 15 neuron. Sedangkan data target jaringan yang terdapat dalam lapisan keluaran terdiri dari 1 neuron. Semua data masukan maupun data target jaringan dibangun secara random menggunakan kode program dengan instruksi "rand". Data ini tersaji pada Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1. Data Masukan Jaringan (X) dan Target (Y) dengan Banyak Neuron *Input* =5

X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	Y
9.5013	7.6210	6.1543	4.0571	0.5789	2.0277
2.3114	4.5647	7.9194	9.3547	3.5287	1.9872
6.0684	0.1850	9.2181	9.1690	8.1317	6.0379
4.8598	8.2141	7.3821	4.1027	0.0986	2.7219
8.9130	4.4470	1.7627	8.9365	1.3889	1.9881

Tabel 2. Data Masukan Jaringan (X) dan Target (Y) dengan Banyak Neuron *Input* = 10

X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	Y
5.8279	2.0907	4.1537	2.1396	6.8333	4.5142	6.0854	0.8408	1.2105	2.3189	4.3979
4.2350	3.7982	3.0500	6.4349	2.1256	0.4390	0.1576	4.5436	4.5075	2.3931	3.4003
5.1551	7.8333	8.7437	3.2004	8.3924	0.2719	0.1635	4.4183	7.1588	0.4975	3.1422
3.3395	6.8085	0.1501	9.6010	6.2878	3.1269	1.9007	3.5325	8.9284	0.7838	3.6508
4.3291	4.6110	7.6795	7.2663	1.3377	0.1286	5.8692	1.5361	2.7310	6.4082	3.9324
2.2595	5.6783	9.7084	4.1195	2.0713	3.8397	0.5758	6.7564	2.5477	1.9089	5.9153
5.7981	7.9421	9.9008	7.4457	6.0720	6.8312	3.6757	6.9921	8.6560	8.4387	1.1975
7.6037	0.5918	7.8886	2.6795	6.2989	0.9284	6.3145	7.2751	2.3235	1.7390	0.3813
5.2982	6.0287	4.3866	4.3992	3.7048	0.3534	7.1763	4.7838	0.8487	1.7079	4.5860
6.4053	0.5027	4.9831	9.3338	5.7515	6.1240	6.9267	5.5484	9.0840	9.9430	8.6987

Tabel 3. Data Masukan Jaringan (X) dan Target (Y) dengan Banyak Neuron *Input* = 15

X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>
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 3 (lanjutan)

X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>15</sub>	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

Seperti telah disampaikan sebelumnya bahwa pengembangan kode program untuk menjalankan algoritma digunakan bahasa pemrograman MATLAB. Dalam MATLAB, untuk setiap jenis algoritma pelatihan diberikan nama fungsi yang bersesuaian, yaitu:

1. *Fletcher-Reeves Update* sebagai "traincgf"
2. *Polak-Ribiere* sebagai "traincgp"
3. *Powell-Beale Restarts* sebagai "traincgb"
4. *Scaled Conjugate Gradient* sebagai "traincsg"
5. *Gradient Descent* sebagai "traingd"
6. *Gradient Descent* dengan *Adaptive Learning Rate* sebagai "traingda"
7. *Gradient Descent* dengan *Momentum* sebagai "traingdm"
8. *Gradient Descent* dengan *Momentum* dan *Adaptive Learning Rate* sebagai "traingdx"
9. *Resilient Backpropagation* sebagai "trainrp"
10. *BFGS* sebagai "trainbfg"
11. *One Step Secant* sebagai "trainoss"
12. *Levenberg-Marquardt* sebagai "trainlm"

Tabel 4. MSE pada Data Masukan 5 Neuron dengan Tingkat lr = 0.01

No.	trainbfg	traincgb	traincgf	traincgp	traingd	traingda
1	0.0781685	0.000310694	0.000977715	0.000774416	0.000965436	0.000938196
2	0.000879153	0.000887382	0.000911287	0.000409749	0.00050124	0.000952874
3	0.000389766	8.42209e-005	3.9733e-005	7.17637e-005	0.000987908	0.000916735
4	0.000421823	0.000581543	0.000477211	0.00056505	0.000961722	0.000959623
5	2.458	0.000831864	0.000268732	0.000899921	0.000986689	0.000874886
6	2.458	0.000677702	0.000932816	0.000941192	0.000964307	0.000827337
7	0.0781685	0.000788396	0.00046224	0.000576934	0.000950927	0.000950306
8	0.000280615	0.000179958	0.000919173	0.000391876	0.00094071	0.000985391
9	0.0481914	0.0539822	0.000451758	7.82838e-005	0.000996653	0.000993256
10	0.0481914	0.000364023	0.0481944	0.000232553	0.000983699	0.000948084
11	4.43199e-005	5.00224e-005	0.000811175	0.000877537	0.000978479	0.000956031
12	0.000436739	0.000700925	0.000936189	0.000984303	0.000958102	0.000885283
13	0.0781685	0.000853109	0.000659036	0.00039815	0.000963321	0.000971769
14	2.458	0.000689615	0.000423028	0.0004295	0.000969133	0.000843659
15	0.000672696	0.000800053	0.000347741	0.000534345	0.000982076	0.000972747
16	0.000750594	0.00085553	0.000721401	0.000169477	0.000942071	0.000924166
17	2.458	0.000722605	0.000940585	0.000171294	0.000947663	0.000999746
18	0.000654728	0.000906544	0.0003173	0.000364043	0.000994409	0.000948903
19	0.000327463	8.95564e-005	0.000268001	0.00063078	0.000972943	0.000917852
20	0.0539784	0.000483212	0.000848179	0.000576648	0.000968354	0.000991755

Tabel 4 (lanjutan)

No.	traingdm	traingdx	trainlm	trainoss	trainrp	traingcg
1	0.000990477	0.000918051	2.76285e-006	2.458	0.000970461	0.000942185
2	0.000986231	0.000903277	1.91709e-005	0.000955535	0.000827962	0.000706939
3	0.000981085	0.000818023	0.000157904	0.00095813	0.000670791	0.000229741
4	0.000938916	0.000984472	0.000719061	0.000528402	0.000366767	0.000725441
5	0.000991027	0.000749829	2.34459e-006	2.458	0.000574537	0.000997388
6	0.000993436	0.000836939	0.000120145	0.000956981	0.000837113	0.000688001
7	0.000956134	0.000963453	5.45649e-006	2.458	0.000873104	0.000384876
8	0.000994875	0.000974023	0.000153304	0.000646223	0.000784764	0.000936661
9	0.000990672	0.000968763	0.000516044	0.000908489	0.000591384	0.000350462
10	0.000990539	0.000881465	0.000231007	0.0539784	0.000480217	0.00072911
11	0.000983341	0.000855156	0.000180418	2.458	0.000764007	0.000597558
12	0.000954641	0.000970021	0.000660536	0.000685945	0.000407189	0.000798862
13	0.000943529	0.000946187	0.000103428	0.00099052	0.000822803	0.000472698
14	0.000993679	0.000990289	0.000351304	2.458	0.000867431	0.000977186
15	0.000960715	0.000977001	0.000212775	0.000933286	0.000874552	0.000673188
16	0.000947285	0.000948982	3.97761e-005	0.000767107	0.00065295	0.000616146
17	0.00098698	0.000686511	0.000287332	0.000516164	0.000979903	0.00099715
18	0.000919398	0.00081974	2.91726e-006	0.0781685	0.00045106	0.000538651
19	0.000992221	0.000944422	0.000122912	0.00072297	7.80285e-005	0.000592311
20	0.00097243	0.000989885	8.51178e-005	0.000758746	0.000878979	0.000986976

Sebelum diuji perbedaan *error* yang dihasilkan oleh setiap algoritma pelatihan menggunakan uji ANOVA, MSE yang dihasilkan jaringan untuk masing-masing algoritma pelatihan diuji normalitasnya. Hasil yang diperoleh adalah bahwa data MSE baik untuk data 5, 10, maupun 15 dan variasi nilai *lr* adalah tidak normal. Dengan demikian, perlu dilakukan normalisasi data, yaitu dengan dilakukan operasi logaritma10. Uji ANOVA dilakukan terhadap data MSE yang telah dinormalisasi. Uji ini dimaksudkan untuk mengetahui algoritma pelatihan yang paling optimal yaitu yang menghasilkan *error* terkecil. Nilai signifikansi yang dihasilkan untuk masing-masing nilai *lr* dan banyaknya neuron masukan, semuanya bernilai 0.00. Sesuai dengan hipotesis yang diajukan, kesimpulan yang diambil adalah bahwa kedua belas algoritma pelatihan memiliki MSE yang berbeda. Setelah dilakukan uji Duncan, dihasilkan bahwa algoritma pelatihan Levenberg Marquardt (*trainlm*) memiliki rata-rata MSE terkecil dengan tingkat  $\alpha = 5\%$ . Hasil MSE dari *trainlm* disajikan pada Tabel 5.

Tabel 5. Hasil MSE Algoritma Pelatihan Trainlm untuk Setiap Banyak Data Masukan dan Tingkat *lr*

No	Data masukan = 5		Data masukan = 10		Data masukan = 15	
	<i>lr</i>	MSE	<i>lr</i>	MSE	<i>lr</i>	MSE
1	0.01	.00019868580	0.01	.000185121492	0.01	.000284530950
2	0.05	.00017938318	0.05	.0002577988209	0.05	.000248148088
3	0.1	.00018655020	0.1	.000126952176	0.1	.000168452917
4	0.2	.000147757748	0.2	.000185485482	0.2	.000276679704
5	0.3	.000198685800	0.3	.009731951112	0.3	.000246095458
6	0.4	.000192124870	0.4	.000265669001	0.4	.000249492183
7	0.5	.00020367686	0.5	.009751811074	0.5	.000246095458
8	0.6	.00010916826	0.6	.000185485482	0.6	.000249492183
9	0.7	.000210012926	0.7	.009731951112	0.7	.000163407430
10	0.8	.000198685800	0.8	.000265695786	0.8	.000246095458
11	0.9	.000116184938	0.9	.000126209131	0.9	.000249492183
12	1	.00020367686	1	.000294526060	1	.000163407430

Berdasarkan Tabel 5, diperoleh harga rata-rata MSE untuk setiap data random yaitu untuk data random 5 diperoleh rata-rata MSE = 0.000179, untuk data random 10 diperoleh rata-rata MSE = 0.002592, dan untuk data random 15 diperoleh rata-rata MSE = 0.000233. Hal ini bisa dikatakan bahwa algoritma *trainlm* memberikan rata-rata MSE sebesar 0.001001 dengan parameter kontrol maksimal epoch = 10.000 dan target error 0.001.

#### 4. Kesimpulan

Kesimpulan dari hasil penelitian ini adalah algoritma pelatihan dalam jaringan backpropagation yang memiliki error terkecil (paling optimal) pada tingkat model parameter jaringan target error = 0.001, maksimum epoch = 10000, dengan variasi nilai

*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, dan 1 dengan 5, 10, dan 15 neuron data masukan jaringan adalah algoritma *trainlm*. Uji yang digunakan dalam penentuan algoritma optimal ini adalah uji ANOVA dengan tingkat  $\alpha = 0.05$  (5%). Algoritma *trainlm* memberikan rata-rata MSE sebesar 0.001001. Tingkat optimalisasi algoritma pelatihan dalam jaringan backpropagation tidak hanya bisa diketahui melalui MSE yang dihasilkan jaringan, tetapi dapat diketahui juga dari tingkat kesesuaian atau kecocokan pola data antara data target dengan data keluaran jaringan. Oleh karena itu, disarankan penelitian ini dilanjutkan dengan uji kecocokan antara pola data antara data target dengan data keluaran jaringan.

#### Ucapan Terima Kasih

1. DP2M Dirjen DIKTI Departemen Pendidikan Nasional melalui Kopertis Wilayah VI yang telah memberikan dana dalam pelaksanaan penelitian ini.
2. Rektor Universitas Muhammadiyah Purwokerto yang telah memberi berbagai dorongan dan kemudahan dalam penelitian ini.
3. Ketua LPPM yang telah memberi persetujuannya, sehingga penelitian ini berjalan dengan baik.
4. Dekan Fakultas Teknik yang telah memberi kesempatan dan fasilitas dalam penelitian ini.

#### Daftar Pustaka

- [1] S. Kusumadewi dan S. Hartati, *Neuro-Fuzzy: Integrasi Sistem Fuzzy dan Jaringan Syaraf*, Yogyakarta: Graha Ilmu, 2006.
- [2] M.T., Jones, *Artificial Intelligence A Systems Approach*, New Delhi: Infinity Science Press LLC, 2008.
- [3] J.J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*, Yogyakarta: ANDI, 2009.
- [4] S. Kusumadewi, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*, Yogyakarta: Graha Ilmu, 2004.
- [5] Harjono dan D. Aryanto, "Application of Artificial Neural Networks to Predict Student Achievement Study", *SAINTEK*, ISSN: 1411-2558, Vol. 5 No. 2, April 2009.
- [6] H. Mustafidah, D. Aryanto, D.K. Hakim, "Uji Optimalisasi Algoritma Pelatihan Conjugate Gradient pada Jaringan Syaraf Tiruan" dalam *Prosiding SENATEK*, ISBN: 978-602-14355-0-2 halaman B.9-1, 21 September 2013.
- [7] H. Mustafidah, D.K. Hakim, S. Sugiyanto, "Tingkat Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan (Studi Kasus Prediksi Prestasi Belajar Mahasiswa)", *JUITA*, ISSN: 2086-9398 Vol. II Nomor 3, halaman 159 – 166, Mei 2013.
- [8] F. Wibowo, S. Sugiyanto, H. Mustafidah, "Tingkat Ketelitian Pengenalan Pola Data pada Algoritma Pelatihan Perbaikan Metode Batch Mode dalam Jaringan Syaraf Tiruan", *Jurnal Ilmiah Nasional "JUITA"*, ISSN 2086-9398, Volume II No.4, halaman 259 – 264, November 2013.
- [9] H. Mustafidah, S. Hartati, R. Wardoyo, A. Harjoko, "Prediction of Test Items Validity Using Artificial Neural Network", in *Proceeding International Conference on Education, Technology, and Science (NETS) 2013 University of Muhammadiyah Purwokerto*, "Improving The Quality Of

Education To Face The Impact Of Technology”, December 28th, 2013.

- [10] T. Taniredja dan H. Mustafidah, Penelitian Kuantitatif (Sebuah Pengantar), Bandung: Alfabeta, 2011.
- [11] H. Mustafidah, and Suwarsito, “Error Rate Testing of Training Algorithm in Back Propagation Network”, International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-5 Issue-4, pp. 46 – 50, September 2015.
- [12] H. Mustafidah, dan Suwarsito, “Uji Keoptimalan Algoritma Pelatihan pada Jaringan Syaraf Tiruan”, dalam Prosiding Seminar Nasional SENATKOM 2015 di Padang, ISSN : 2460-4690, halaman 243-248, 23 Oktober 2015.