



## Algoritma Fungsi Pelatihan pada *Machine Learning* berbasis ANN untuk Peramalan Fenomena Bencana

Anjar Wanto<sup>1,2</sup>, Sarjon Defit<sup>2</sup>, Agus Perdana Windarto<sup>1,2</sup>

<sup>1</sup>STIKOM Tunas Bangsa, Pematangsiantar, Indonesia

<sup>2</sup>Department of Informatics Engineering, Universitas Putra Indonesia YPTK Padang, Indonesia  
[anjarwanto@amiktunasbangsa.ac.id](mailto:anjarwanto@amiktunasbangsa.ac.id), [sarjond@yahoo.co.uk](mailto:sarjond@yahoo.co.uk), [agus.perdana@amiktunasbangsa.ac.id](mailto:agus.perdana@amiktunasbangsa.ac.id)

### Abstract

Research has been carried out with several training functions using standard backpropagation methods, One-Step Secant (OSS), and Bayesian regulation. The purpose of this study was to (i) analyze the Performance accuracy (Performance) of the standard backpropagation method and (ii) optimize the training function with the One-Step Secant (OSS) and Bayesian regulation methods to obtain comparison results of the three methods in the search for the best results implementation of disaster phenomenon forecasting data. The research method is based on quantitative methods with times-series data on disaster phenomena in Indonesia over the last ten years (2011-2020) which were analyzed using two network architecture models, namely 4-8-1 and 4-10-1. The results showed that the 4-8-1 architectural model with the Bayesian regulation training function method was able to optimize quite well through accelerating training time and resulted in a low MSE measurement, although not the lowest with an epoch value of 197 iterations and a Performance of 0.0148480766. The lowest epoch value is generated by the OSS method, but it Performs poorly. The best Performance is produced by the standard backpropagation method with the traingd training function, but the training process for achieving convergence is also too long. In general, it can be concluded that the 4-8-1 architectural model with Bayesian regulation can be used to predict (predict) the phenomenon of natural disasters in Indonesia because the training time to achieve convergence is not too long and Performs exceptionally well.

Keywords: ANN, bayesian regulation, Performance, backpropagation method, One-Step Secant.

### Abstrak

Telah dilakukan penelitian dengan beberapa fungsi pelatihan dengan metode *backpropagation* standard, *One-Step Secant* (OSS), dan *Bayesian regulation*. Tujuan penelitian ini untuk (i) menganalisis keakuratan kinerja (*Performance*) metode *backpropagation* standard dan (ii) melakukan optimasi fungsi pelatihan dengan metode *One-Step Secant* (OSS) dan *Bayesian regulation*, sehingga diperoleh hasil perbandingan dari ketiga metode tersebut dalam pencarian hasil terbaik untuk pelaksanaan data peramalan fenomena bencana. Metode penelitian berbasis metode kuantitatif dengan data *times-series* fenomena bencana di Indonesia pada kurun waktu 10 tahun terakhir (2011-2020) yang dianalisis dengan dua model arsitektur jaringan, yakni 4-8-1 dan 4-10-1. Hasil penelitian ditunjukkan, bahwa model arsitektur 4-8-1 dengan metode fungsi pelatihan *Bayesian regulation* mampu dengan optimasi cukup baik melalui pemercepatan waktu pelatihan dan dihasilkan pengukuran MSE rendah walaupun bukan paling rendah dengan nilai *epoch* sebesar 197 iterasi dan kinerja sebesar 0,0148480766. Nilai *epoch* paling rendah dihasilkan oleh metode OSS, tetapi berkinerja kurang baik. Kinerja terbaik dihasilkan oleh metode *backpropagation* standard dengan fungsi pelatihan *traingd*, tetapi proses pelatihan untuk pencapaian konvergensi juga terlalu lama. Secara umum dapat disimpulkan, bahwa model arsitektur 4-8-1 dengan *Bayesian regulation* dapat digunakan untuk peramalan (prediksi) fenomena bencana alam di Indonesia, karena waktu pelatihan untuk pencapaian konvergensi tidak terlalu lama dan berkinerja cukup baik.

Kata kunci: ANN, bayesian regulation, kinerja, metode *backpropagation*, *One-Step Secant*.

### 1. Pendahuluan

Pemanfaatan fungsi transfer ataupun fungsi pelatihan hasilkan keakuratan peramalan yang berbeda-beda,

tergantung kepada metode yang diberikan dan data terprediksi [1-5]. Pendekatan beberapa fungsi pelatihan pada *Machine Learning* berbasis ANN

(jaringan syaraf tiruan, JST) dapat digunakan untuk penyelesaian masalah komputasi yang berhubungan dengan data peramalan atau prediksi [6]. Fungsi pelatihan pada teknik *Machine Learning* dengan *ANN* metode *backpropagation* dapat dihasilkan keakuratan peramalan berbeda-beda, bergantung kepada parameter masukan dan data terprediksi. Teknik *Machine Learning* berbasis *ANN* sangat populer dan sukses diimplementasikan pada berbagai bidang penelitian, seperti bisnis, kesehatan, teknologi, maupun pendidikan [7]. Selain hal itu, JST dapat digunakan untuk penyelesaian masalah yang kompleks dengan tingkat kekompleksan tinggi [8-10], seperti jaringan saraf untuk analisis gambar dan pelacakan objek maupun analisis data *times-series* [11].

Teknik *Machine Learning* dengan penyediaan metode untuk kemungkinan pembelajaran komputer secara otomatis tanpa campur tangan manusia [12] dan untuk penyesuaian terhadap tindakan yang tepat [13], karena *Machine Learning* berfokus peningkatan kinerja sistem pembelajaran secara otomatis melalui pengalaman [14]. Masalah *Machine Learning* dapat dilihat sebagai salah satu cara untuk optimasi kinerja metode komputer berdasarkan data sebelumnya [15]. Pemanfaatan fungsi pelatihan untuk pemberian kinerja yang optimal, telah digunakan secara luas dalam penyelesaian masalah dengan tingkat kekompleksan tinggi. Uwanuakwa and Akpinar [38] dengan pembahasan tentang efisiensi prediksi kedalaman karbonasi beton dengan optimasi fungsi pelatihan metode *One-Step Secant* (OSS) sebagai alternatif dari *Levenberg-Marquardt* konvensional yang digunakan. Untuk prediksi karbonasi beton diperoleh nilai koefisien relasi (R) sebesar 0,99 dengan variasi hasil persentase 30-55% dengan arti, bahwa terdapat peningkatan nilai R secara signifikan dari yang diamati sebesar 60-80%. Selain hal itu, berdasarkan pengamatan ditunjukkan, bahwa variasi *neuron* tersembunyi antara 5-25 dihasilkan perubahan akurasi prediksi kurang signifikan, baik nilai R, MSE, maupun persentasenya [16].

Metode *backpropagation* merupakan metode berbasis gradien yang banyak digunakan untuk pelatihan jaringan saraf *feedforward* (umpan maju) [17-19]. Perkembangan metode *backpropagation* dengan beberapa fungsi, diantaranya fungsi transfer dan pelatihan yang masing-masing fungsi juga dengan banyak teknik dan metode yang dapat digunakan untuk menyelesaikan penyelesaian masalah dengan tingkat kekompleksan yang tinggi. Penggunaan fungsi transfer pada *Machine Learning* berbasis jaringan saraf, khususnya *backpropagation*, yaitu fungsi transfer *sigmoid* tangen hiperbolik (*tansig*) [20], fungsi transfer *log-sigmoid* (*logsig*) [21], dan fungsi transfer linear (*purelin*) [22], sedangkan penggunaan fungsi pelatihan pada metode *backpropagation* standar, diantaranya

*gradient descent* (*traingd*, *traingdx*, *traingda* dan *traingdm*) [23]. Fungsi pelatihan lain dapat digunakan untuk optimasi dan berpengaruh terhadap hasil komputasi, seperti *Levenberg-Marquardt* (*trainlm*) [24,25], pelatihan *batch* dengan aturan pembelajaran bobot dan bias (*trainb*) [26], *BFGS quasi-Newton* (*trainbfg*) [27], *BFGS quasi-Newton* dengan referensi adaptif control (*trainbfgc*) [28], regulasi *bayesian* (*trainbr*) [29], pelatihan *batch* bobot/bias tanpa pengawasan (*trainbu*) [30], urutan siklus bobot/bias (*trainc*) [31], pelatihan konjugasi gradien (*traingcf*, *traingcgp*, *traingcb*) [32], *One-Step Secant* (garis potong satu langkah, *trainoss*) [33], pelatihan tambahan urutan acak dengan fungsi pembelajaran (*trainr*) [34], *resilient* (*trainrp*) [35], pelatihan bobot/bias perintah acak tidak diawasi (*trainru*) [36], pelatihan urutan inkremental berurutan dengan fungsi pembelajaran (*trains*) [37], dan gradien konjugasi berskala (*trainscg*) [38].

Berdasarkan uraian terkait penelitian-penelitian sebelumnya, maka penelitian pada makalah ini berupa analisis pemanfaatan fungsi pelatihan untuk optimasi keakuratan dan pengukuran kinerja pada *Machine Learning* berbasis *ANN* untuk peramalan fenomena bencana alam di Indonesia. Makalah ini dengan pembahasan tentang metode fungsi pelatihan untuk penyelesaian masalah, sedangkan *dataset* bencana alam hanya digunakan untuk bantuan pembuktian dan proses pengukuran kinerja. Implementasi teknik *Machine Learning* dengan metode *backpropagation* standar sering kali dengan pemberian nilai kecepatan konvergensi yang buruk pada proses pelatihan, sehingga perlu dilakukan berbagai kombinasi fungsi pelatihan untuk pemercepatan konvergensi pelatihan jaringan syaraf, yakni dengan penggunaan metode *One-Step Secant* (OSS) dan *Bayesian Regulation*. Keakuratan dan kinerja dari metode *backpropagation* standar dianalisis dan dibandingkan terhadap metode *One-Step Secant* (OSS) dan *Bayesian regulation* yang merupakan pengembangan dari metode *backpropagation*. Tujuan penelitian ini untuk memperoleh optimasi keakuratan dan pengukuran *Performance*, sasaran penelitian berupa hasil perbandingan dari ketiga metode tersebut dalam pencarian hasil terbaik untuk pelaksanaan data peramalan fenomena bencana.

## 2. Metode Penelitian

### 2.1. Bahan Penelitian

Seperangkat data (*dataset*) penelitian untuk fungsi pelatihan berupa data *times-series* bencana alam di wilayah Indonesia pada kurun waktu 10 tahun terakhir (2011-2020) yang terdiri atas (i) putting-beliung, (ii) kebakaran hutan dan lahan, (iii) banjir, (iv) tanah longsor, (v) kekeringan, (vi) gempa bumi, (vii)

gelombang pasang dan abrasi, dan (viii) erupsi gunung berapi. Data diperoleh dari Geoportal Kebencanaan Indonesia yang bersumber dari Badan Nasional Penganggulangan Bencana (BNPB) melalui menu infografis di *website* <https://gis.bnpb.go.id/> [39]. Jenis bencana alam di Indoensia kurun waktu 2011-2020, seperti ditunjukkan pada Tabel 1.

Tabel 1. Jenis bencana alam di Indonesia (kurun waktu 2011-2020)

Jenis Bencana	Tahun			
	2011	2012	...	2020
Puting-beliung	320	543	...	880
Kebakaran hutan dan lahan	376	58	...	326
Banjir	403	581	...	1080
Tanah longsor	261	287	...	577
Kekeringan	221	263	...	29
Gempa bumi	11	12	...	16
Gelombang pasang dan abrasi	16	29	...	36
Erupsi gunung berapi	11	7	...	7
Total Bencana	1619	1780	...	2951

Tampilan infografis bencana di Indonesia tahun 2020, seperti ditunjukkan pada Gambar 1.

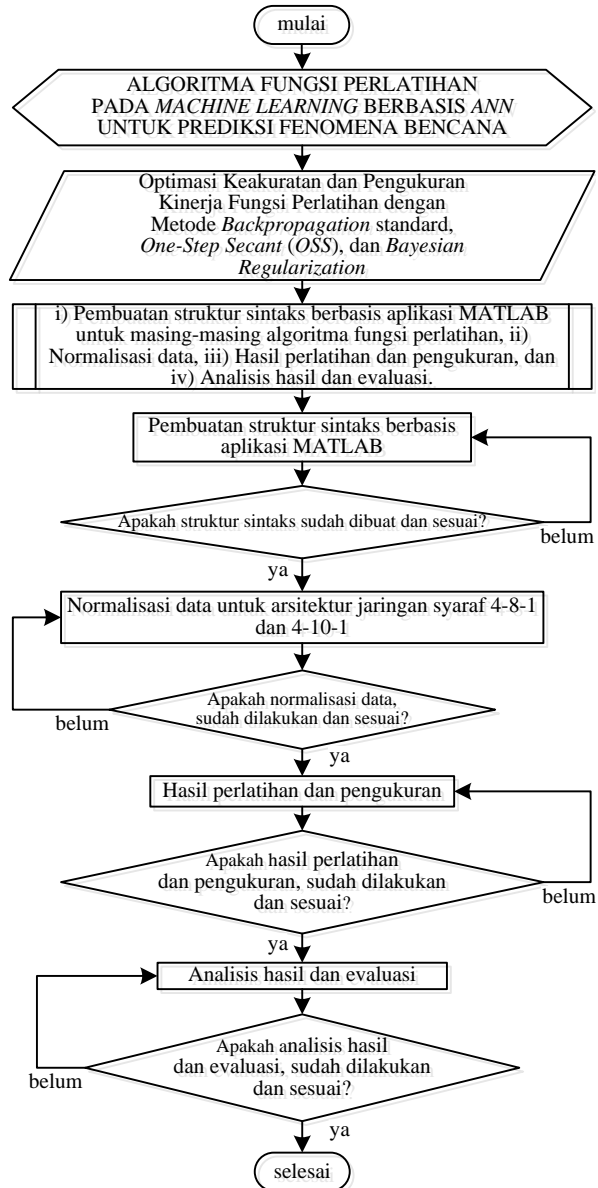


Gambar 1. Tampilan infografis bencana di Indonesia tahun 2020

Berdasarkan Gambar 1 dapat dijelaskan, bahwa informasi bencana alam di Indonesia pada tahun 2020 dengan total bencana sebanyak 2951, ditambah 1 bencana non-alam, yakni *Covid-19*. Bencana alam banjir dengan kejadian terbesar, diikuti bencana puting-beliung dan tanah longsor. Dampak bencana alam sepanjang 1 Januari sampai 31 Desember 2020, meliputi korban jiwa meninggal dunia sebanyak 370, 6.450.903 menderita dan mengungsi, 39 hilang, dan 536 luka-luka.

## 2.2. Metode Penelitian

Metode penelitian merupakan algoritma seorang peneliti dalam pelaksanaan penelitian secara keseluruhan. Diagram Alir metode penelitian, seperti ditunjukkan pada Gambar 2.



Gambar 2. Diagram alir metode penelitian

Berdasarkan Gambar 2 dapat dijelaskan, bahwa langkah pertama dalam tahapan penelitian berupa pengumpulan seperangkat data (*dataset*) penelitian, sebagaimana ditunjukkan pada Tabel 1. Langkah berikutnya berupa pemisahan *dataset* penelitian menjadi 2 kelompok, yakni untuk data pelatihan dan pengukuran kinerja. Data pelatihan berdasarkan data bencana tahun 2011-2014 (X1-X4) dengan target pelatihan tahun 2015 (Y1), sedangkan untuk data pengukuran kinerja berdasarkan data tahun 2016-2019 (X5-X8) dengan target pengukuran kinerja tahun 2020 (Y2). Tahapan selanjutnya berupa pelaksanaan normalisasi data pelatihan dan pengukuran kinerja dengan penggunaan persamaan (1) [40].

$$x' = \frac{0.8(x-a)}{b-a} + 0.1 \quad (1)$$

dengan:  $x'$  = hasil data setelah dinormalisasi, nilai 0,8 dan 0,1 merupakan nilai *default* dari persamaan normalisasi,  $x$  merupakan data target untuk dinormalisasi,  $a$  = nilai terendah dari *dataset* dan  $b$  = nilai tertinggi dari *dataset*.

Data pelatihan setelah dinormalisasi dimasukkan ke dalam aplikasi MATLAB versi 2011b untuk diproses, dilanjutkan dengan pembuatan jaringan saraf *multi layer*. Model arsitektur jaringan syaraf untuk proses pelatihan pada penelitian ini adalah 4-8-1 dan 4-10-1 (4 merupakan *input layer*, 8 dan 10 merupakan *hidden layer* dengan 8 dan 10 *neuron*, 1 merupakan *output layer*). Pembuatan jaringan syaraf *multi layer* ini, *hidden layer* digunakan fungsi transfer *logsig* (*sigmoid biner*) dan *output layer* digunakan *purelin* (fungsi linear). Tahapan ini juga dilakukan pemilihan fungsi pelatihan, dengan masing-masing data pelatihan bergantian dilatih dengan penggunaan fungsi pelatihan penurunan gradien (*traingd*) pada *backpropagation* standard, kemudian dioptimasi dengan metode fungsi pelatihan metode *One-Step Secant* (*trainoss*) dan *Bayesian regulation* (*trainbr*). Tahapan lanjutan berupa pembangkitan nilai bobot (*IW*, *LW*) dan bias (*b*). Tahapan inisialisasi parameter jaringan syaraf berdasarkan fungsi pelatihan yang digunakan, kemudian dimasukkan perintah untuk pelaksanaan proses pelatihan dan diperiksa hasil saat kinerja (*Performance*) ditemukan. Untuk kondisi dimana hasil pelatihan tercapai konvergensi, maka dilanjutkan dengan pemasukan data pengukuran yang sudah dinormalisasi, sedangkan jika hasil pelatihan belum tercapai konvergensi, maka kembali ke tahap pembuatan MLNN dan pemilihan fungsi pelatihan. Tahapan dilanjutkan dengan simulasi data ukur berdasarkan hasil pelatihan, apabila semua sudah dilakukan, tahapan akhir berupa evaluasi untuk pemeriksaan kinerja dan kemampuan algoritma fungsi pelatihan terbaik.

### 3. Hasil dan Pembahasan

#### 3.1. Metode Fungsi Pelatihan

Secara umum, masing-masing metode fungsi pelatihan pada penelitian ini diproses dengan aplikasi MATLAB versi 2011b. Struktur sintaks untuk metode *backpropagation* standar, yaitu:

##### Metode "Backpropagation Standar"

```
% Dimasukkan Data Training
% Dimasukkan Data Input
>> p=[Data pelatihan telah dinormalisasi]
% Dimasukkan Data Output Target
>> t=[Target Data pelatihan]
>> net = newff(minmax(p),[hidden layer,output layer],{'logsig','purelin'},'traingd');
% Dibangkitkan bobot dan bias
>> net.IW{1,1};
```

```
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
% Nilai metode backpropagation
>> net.trainParam.epochs = 5000;
>> net.trainParam.goal = 0.001;
>> net.trainParam.lr = 0.01;
>> net.trainParam.show = 1000;
% Dilakukan Training
>> net = train(net,p,t)
% Dilihat hasil pada saat Performance ditemukan
>> [a,Pf,Af,e,Perf] = sim(net,p,[],[],t)
% Dimasukkan Data Input (Ukur)
>> p1=[Data uji yang telah dinormalisasi]
% Dimasukkan Data Target (ukur kinerja)
>> t1=[Target Data Ukur]
% Dilakukan simulasi penggunaan data ukur berdasarkan hasil training
>> [a,Pf,Af,e,Perf] = sim(net,p1,[],[],t1)
```

Struktur sintaks untuk metode *One-Step Secant*, yaitu:

##### Metode One-Step Secant (OSS)

```
% Dimasukkan Data Training
% Dimasukkan Data Input
>> p=[Data pelatihan telah dinormalisasi]
% Dimasukkan Data Output Target
>> t=[Target data pelatihan]
>> net = newff(minmax(p),[hidden layer,output layer],{'logsig','purelin'},'trainoss');
% Dibangkitkan bobot dan bias
>> net.IW{1,1};
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
% Nilai metode One-Step Secant
>> net.trainParam.epochs = 5000;
>> net.trainParam.show = 25;
>> net.trainParam.showCommandLine = 0;
>> net.trainParam.showWindow = 1;
>> net.trainParam.goal = 0;
>> net.trainParam.time = inf;
>> net.trainParam.min_grad = 1e-6;
>> net.trainParam.max_fail = 5;
>> net.trainParam.searchFcn = 'srchcha';
% Dilakukan Training
>> net = train(net,p,t)
% Diperiksa hasil pada saat Performance ditemukan
>> [a,Pf,Af,e,Perf] = sim(net,p,[],[],t)
% Dimasukkan Data Input (Ukur kinerja)
>> p1=[Data uji yang telah dinormalisasi]
% Dimasukkan Data Target (ukur kinerja)
>> t1=[Target data uji]
% Dilakukan simulasi penggunaan data ukur kinerja berdasarkan hasil training
>> [a,Pf,Af,e,Perf] = sim(net,p1,[],[],t1)
```

Struktur sintaks untuk metode *Bayesian Regulation*, yaitu:

##### Metode Bayesian Regulation

```
% Dimasukkan Data Training
% Dimasukkan Data Input
>> p=[Data pelatihan telah dinormalisasi]
% Dimasukkan Data Output Target
>> t=[Target data pelatihan]
>> net = newff(minmax(p),[hidden layer,output layer],{'logsig','purelin'},'trainbr');
% Dibangkitkan bobot dan bias
>> net.IW{1,1};
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
% Nilai metode Bayesian Regulation
>> net.trainParam.epochs=5000;
>> net.trainParam.goal = 0;
>> net.trainParam.mu=0.005;
```

```
>> net.trainParam.mu_dec=0.1;
>> net.trainParam.mu_inc=10;
>> net.trainParam.mu_max=1e10;
>> net.trainParam.max_fail=5;
>> net.trainParam.mem_reduc=1;
>> net.trainParam.min_grad=1e-10;
>> net.trainParam.show = 25;
>> net.trainParam.showCommandLine=0;
>> net.trainParam.showWindow=1;
>> net.trainParam.time=inf;
% Dilakukan Training
>> net = train(net,p,t)
% Diperiksa hasil pada saat Performance
ditemukan
>> [a,Pf,Af,e,Perf] = sim(net,p,[],[],t)
% memasukan data Input (ukur kinerja)
>> p1=[Data ukur kinerja telah dinormalisasi]
% Dimasukkan Dta Target (ukur kinerja)
>> t1=[Target data ukur kinerja]
% Dilakukan simulasi penggunaan data ukur
kinerja berdasarkan hasil training
>> [a,Pf,Af,e,Perf] = sim(net,p1,[],[],t1)
```

### 3.2. Normalisasi Data Untuk Pelatihan dan Pengukuran Kinerja

Data bencana alam di Indonesia (berdasarkan Tabel 1), terlebih dahulu dinormalisasi dengan penggunaan persamaan (1). *Dataset* penelitian dibagi menjadi dua bagian, yakni pelatihan dan pengukuran kinerja. Data untuk pelatihan berdasarkan *dataset* fenomena bencana tahun 2011-2014 (X1-X4) dengan tahun 2015 (Y1) sebagai target pelatihan. Hasil normalisasi data untuk pelatihan, seperti ditunjukkan pada Tabel 2.

Tabel 2. Hasil normalisasi data untuk pelatihan

No.	X1	X2	X3	X4	Target (Y1)
1	0,4574	0,7105	0,6617	0,7956	0,7423
2	0,5210	0,1601	0,1363	0,2191	0,1511
3	0,5516	0,7536	0,9000	0,7616	0,6878
4	0,3905	0,4200	0,4268	0,7729	0,6640
5	0,3451	0,3928	0,1692	0,1023	0,1023
6	0,1068	0,1079	0,1125	0,1091	0,1227
7	0,1125	0,1272	0,1352	0,1193	0,1023
8	0,1068	0,1023	0,1034	0,1000	0,1045

Data pengukuran kinerja didasarkan kepada data tahun 2016-2019 sebagai X5-X8 dengan target ukur kinerja tahun 2020 sebagai Y2. Hasil normalisasi data untuk ukur kinerja, seperti ditunjukkan pada Tabel 3.

Tabel 3. Hasil normalisasi data untuk ukur kinerja

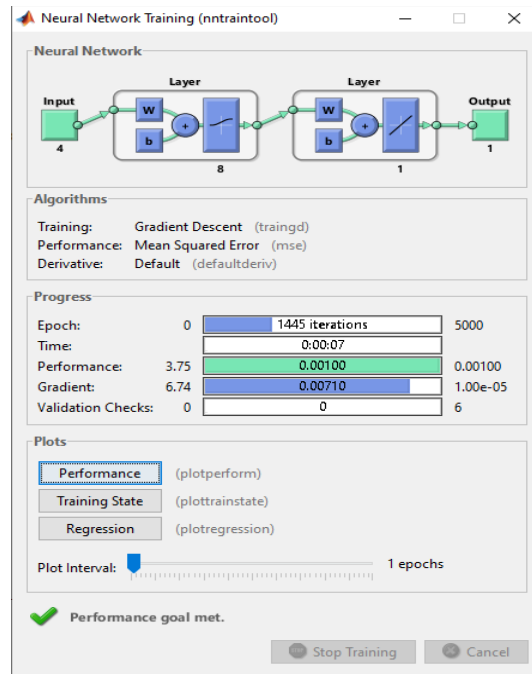
No.	X5	X6	X7	X8	Target (Y2)
1	0,4806	0,6104	0,7413	0,9000	0,6063
2	0,2002	0,1521	0,4018	0,5287	0,2854
3	0,5715	0,6636	0,6011	0,5507	0,7222
4	0,4429	0,5878	0,4528	0,5130	0,4308
5	0,1006	0,1075	0,1840	0,1678	0,1133
6	0,1023	0,1081	0,1127	0,1139	0,1058
7	0,1093	0,1029	0,1162	0,1070	0,1174
8	0,1006	0,1000	0,1301	0,1006	0,1006

### 3.3. Hasil Pelatihan dan Pengukuran Kinerja

Pelatihan dan pengukuran kinerja dilakukan dengan penggunaan aplikasi MATLAB versi 2011b dengan 2 model arsitektur jaringan, yakni 4-8-1 dan 4-10-1.

#1) Pelatihan dan pengukuran kinerja dengan metode *backpropagation* standar

@1.a) Jaringan syaraf model arsitektur 4-8-1  
 Tampilan hasil pelatihan terhadap model arsitektur 4-8-1 (*traingd*), seperti ditunjukkan pada Gambar 3.



Gambar 3. Pelatihan Model 4-8-1 (*traingd*)

Gambar 3 menjelaskan bahwa hasil pelatihan dengan penggunaan fungsi *traingd* pada model 4-8-1 dihasilkan *epoch* sebesar 1445 iterasi dengan waktu latih selama 7 detik.

Hasil pelatihan terhadap model arsitektur 4-8-1 (*traingd*), seperti ditunjukkan pada Tabel 4.

Tabel 4. Hasil Pelatihan Model Arsitektur 4-8-1 (*traingd*)

Target (Y1)	Epoch 1445			
	Actual	Error	SSE	Performance
0,7423	0,7254	0,0169	0,00028458	0,00099981
0,1511	0,1088	0,0423	0,00178623	
0,6878	0,7013	-0,0135	0,00018221	
0,6640	0,6727	-0,0087	0,00007618	
0,1023	0,1734	-0,0711	0,00505955	
0,1227	0,0989	0,0238	0,00056620	
0,1023	0,0992	0,0031	0,00000942	
0,1045	0,0987	0,0058	0,00003409	
			Total SSE	0,00799847
			MSE	0,00099981

Berdasarkan Tabel 4 dapat dijelaskan, bahwa hasil pelatihan dengan model arsitektur 4-8-1 dengan nilai konvergensi pada iterasi ke 1445 dengan nilai *Performance* sebesar 0,00099981.

Hasil pengukuran kinerja terhadap model arsitektur 4-8-1 (*traingd*), seperti ditunjukkan pada Tabel 5.

Tabel 5. Hasil pengukuran kinerja terhadap arsitektur jaringan syaraf 4-8-1 (*traingd*)

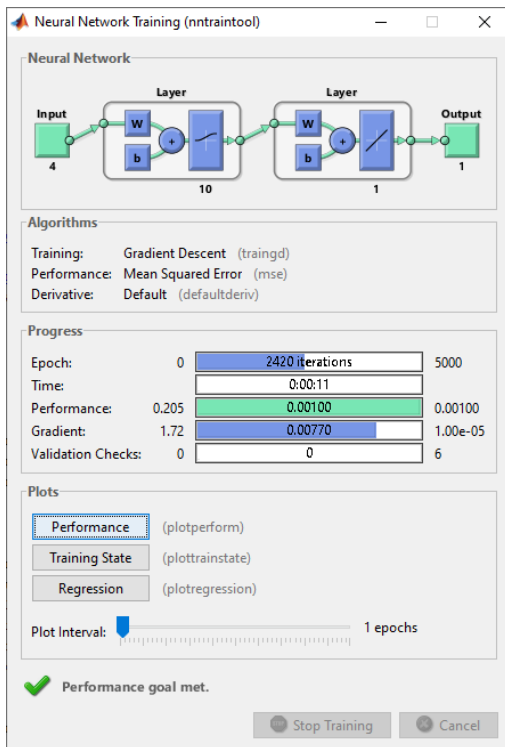
Target (Y2)	Epoch 1			
	Actual	Error	SSE	Performance
0,6063	0,7669	-0,1606	0,025792430	0,063334482
0,2854	0,1206	0,1648	0,027150115	
0,7222	0,1543	0,5679	0,322462545	
0,4308	0,7923	-0,3615	0,130700470	
0,1133	0,0999	0,0134	0,000180195	
0,1058	0,0990	0,0068	0,000046144	
0,1174	0,0989	0,0185	0,000341463	
0,1006	0,0990	0,0016	0,000002494	
Total SSE			0,506675855	
MSE			0,063334482	

Berdasarkan Tabel 5 dapat dijelaskan, bahwa nilai *Performance* diperoleh sebesar 0,063334482.

Nilai aktual pada Tabel 4 dan Tabel 5 diperoleh dari keluaran hasil pelatihan dan pengukuran kinerja. Nilai *error* diperoleh dari perhitungan Target (Y). Nilai aktual *SSE* diperoleh berdasarkan perhitungan *error* dikuadratkan ( $error^2$ ). Nilai total *SSE* diperoleh dari penjumlahan nilai *SSE* secara keseluruhan, sedangkan nilai *Performance* (*Perf*) diperoleh dari total *SSE* dibagi 8, dengan 8 merupakan jumlah baris data. Berdasarkan hasil pelatihan dan pengukuran kinerja dengan penggunaan aplikasi MATLAB dan Ms. Excel, terdapat kesesuaian (*valid*).

@1.b) Jaringan syaraf model arsitektur 4-10-1

Tampilan hasil pelatihan terhadap model arsitektur 4-10-1 (*traingd*), seperti ditunjukkan pada Gambar 4.



Gambar 4. Tampilan hasil pelatihan terhadap model arsitektur 4-10-1 (*traingd*)

Berdasarkan Gambar 4 dapat dijelaskan, bahwa hasil pelatihan dengan penggunaan fungsi *traingd* pada model 4-8-1 dihasilkan *epoch* sebesar 2420 iterasi dengan waktu latih selama 11 detik.

Hasil pelatihan terhadap model arsitektur 4-10-1 (*traingd*), seperti ditunjukkan pada Tabel 6.

Tabel 6. Hasil Pelatihan Model Arsitektur 4-10-1 (*traingd*)

Target (Y1)	Epoch 2420			
	Actual	Error	SSE	Performance
0,7423	0,7615	-0,0192	0,00036981	0,00100007
0,1511	0,1870	-0,0359	0,00129141	
0,6878	0,7029	-0,0151	0,00022797	
0,6640	0,6131	0,0509	0,00258792	
0,1023	0,0537	0,0486	0,00235900	
0,1227	0,1171	0,0056	0,00003130	
0,1023	0,1354	-0,0331	0,00109763	
0,1045	0,1105	-0,0060	0,00003553	
Total SSE			0,00800057	
MSE			0,00100007	

Berdasarkan Tabel 6 dapat dijelaskan, bahwa hasil pelatihan dengan model arsitektur 4-10-1 dengan nilai konvergensi pada iterasi ke 2420 dengan nilai *Performance* sebesar 0,00100007.

Hasil pengukuran kinerja terhadap model arsitektur 4-10-1 (*traingd*), seperti ditunjukkan pada Tabel 7.

Tabel 7. Hasil pengukuran kinerja terhadap arsitektur jaringan syaraf 4-10-1 (*traingd*)

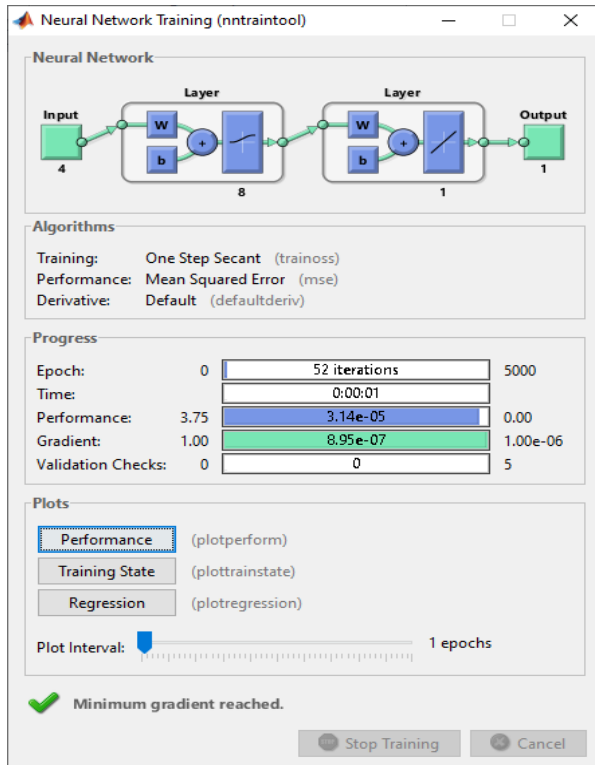
Target (Y2)	Epoch 1			
	Actual	Error	SSE	Performance
0,6063	0,8151	-0,2088	0,043597531	0,011395198
0,2854	0,3217	-0,0363	0,001319657	
0,7222	0,5990	0,1232	0,015167858	
0,4308	0,6052	-0,1744	0,030424150	
0,1133	0,1193	-0,0060	0,000035716	
0,1058	0,1202	-0,0144	0,000207564	
0,1174	0,1105	0,0069	0,000047317	
0,1006	0,1196	-0,0190	0,000361787	
Total SSE			0,091161580	
MSE			0,011395198	

Berdasarkan Tabel 7 dapat dijelaskan, bahwa nilai *Performance* diperoleh sebesar 0,011395198.

Nilai aktual pada Tabel 6 dan Tabel 7 diperoleh dari keluaran hasil pelatihan dan pengukuran kinerja. Nilai aktual *SSE* diperoleh berdasarkan perhitungan *error* dikuadratkan ( $error^2$ ). Nilai total *SSE* diperoleh dari penjumlahan nilai *SSE* secara keseluruhan, sedangkan nilai *Performance* (*Perf*) diperoleh dari total *SSE* dibagi 8, dengan 8 merupakan jumlah baris data. Berdasarkan hasil pelatihan dan pengukuran kinerja dengan penggunaan aplikasi MATLAB dan Ms. Excel, terdapat kesesuaian (*valid*).

@2.a) Jaringan syaraf model arsitektur 4-8-1

Tampilan hasil pelatihan terhadap model arsitektur 4-8-1 (*trainoss*), seperti ditunjukkan pada Gambar 5.



Gambar 5. Pelatihan Model 4-8-1 (*trainoss*)

Berdasarkan Gambar 5 dapat dijelaskan, bahwa hasil pelatihan dengan penggunaan fungsi *trainoss* pada model 4-8-1 dihasilkan *epoch* sebesar 52 iterasi dengan waktu latih selama 1 detik.

Hasil pelatihan terhadap model arsitektur 4-8-1 (*trainoss*), seperti ditunjukkan pada Tabel 8.

Tabel 8. Hasil Pelatihan Model Arsitektur 4-8-1 (*trainoss*)

Target (Y1)	Epoch 52			
	Actual	Error	SSE	Performance
0,7423	0,7423	0,0000	0,00000000	0,00003121
0,1511	0,1511	0,0000	0,00000000	
0,6878	0,6878	0,0000	0,00000000	
0,6640	0,6640	0,0000	0,00000000	
0,1023	0,1023	0,0000	0,00000000	
0,1227	0,1099	0,0128	0,00016371	
0,1023	0,1101	-0,0078	0,00006132	
0,1045	0,1095	-0,0050	0,00002461	
Total SSE			0,00024965	
MSE			0,00003121	

Berdasarkan Tabel 8 dapat dijelaskan, bahwa hasil pelatihan dengan model arsitektur 4-8-1 dengan nilai konvergensi pada iterasi ke 52 dengan nilai *Performance* sebesar 0,00003121.

Hasil pengukuran kinerja terhadap model arsitektur 4-8-1 (*trainoss*), seperti ditunjukkan pada Tabel 9.

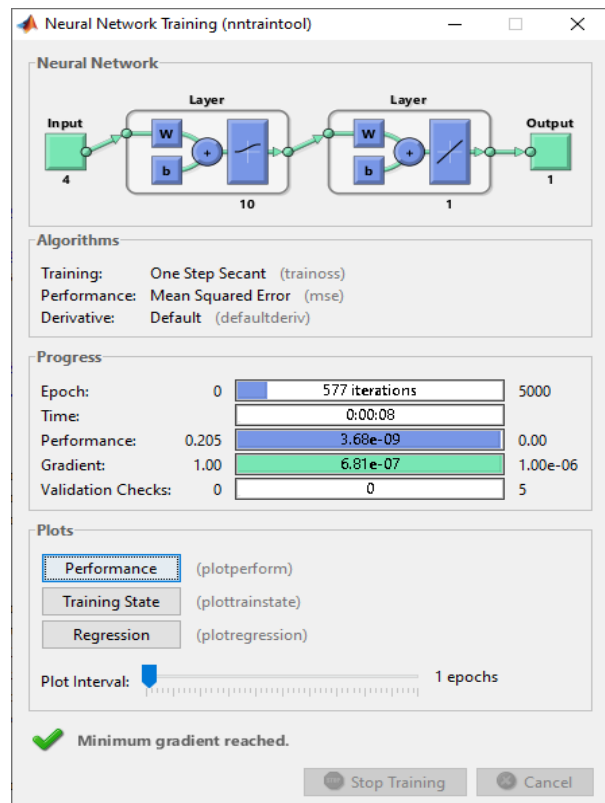
Berdasarkan Tabel 9 dapat dijelaskan, bahwa nilai *Performance* diperoleh sebesar 0,060843522.

Tabel 9. Hasil pengukuran kinerja terhadap arsitektur jaringan syaraf 4-8-1 (*trainoss*)

Target (Y2)	Epoch 1			
	Actual	Error	SSE	Performance
0,6063	0,8029	-0,1966	0,038651645	0,060843522
0,2854	0,2384	0,0470	0,002206455	
0,7222	0,1242	0,5980	0,357553598	
0,4308	0,7277	-0,2969	0,088164574	
0,1133	0,1130	0,0003	0,000000105	
0,1058	0,1102	-0,0044	0,000019422	
0,1174	0,1099	0,0075	0,000055931	
0,1006	0,1104	-0,0098	0,000096446	
Total SSE			0,486748177	
MSE			0,060843522	

Nilai aktual pada Tabel 8 dan Tabel 9 diperoleh dari keluaran hasil pelatihan dan pengukuran kinerja. Nilai *error* diperoleh dari perhitungan Target (Y). Nilai aktual *SSE* diperoleh berdasarkan perhitungan *error* dikuadratkan ( $error^2$ ). Nilai total *SSE* diperoleh dari penjumlahan nilai *SSE* secara keseluruhan, sedangkan nilai *Performance* (*Perf*) diperoleh dari total *SSE* dibagi 8, dengan 8 merupakan jumlah baris data. Berdasarkan hasil pelatihan dan pengukuran kinerja dengan penggunaan aplikasi MATLAB dan Ms. Excel, terdapat kesesuaian (*valid*).

@2.b) Jaringan syaraf model arsitektur 4-10-1  
 Tampilan hasil pelatihan terhadap model arsitektur 4-10-1 (*trainoss*), seperti ditunjukkan pada Gambar 6



Gambar 6. Pelatihan Model 4-10-1 (*trainoss*)

Berdasarkan Gambar 6 dapat dijelaskan, bahwa hasil pelatihan dengan penggunaan fungsi *trainoss* pada model 4-10-1 dihasilkan *epoch* sebesar 577 iterasi dengan waktu latih selama 8 detik.

Hasil pelatihan terhadap model arsitektur 4-10-1 (*trainoss*), seperti ditunjukkan pada Tabel 10.

Tabel 10. Hasil Pelatihan Model Arsitektur 4-10-1 (*trainoss*)

Target (Y1)	Epoch 577			
	Actual	Error	SSE	Performance
0,7423	0,7423	0,0000	0,00000000	0,0000000042
0,1511	0,1511	0,0000	0,00000000	
0,6878	0,6878	0,0000	0,00000000	
0,6640	0,6640	0,0000	0,00000000	
0,1023	0,1023	0,0000	0,00000000	
0,1227	0,1226	0,0001	0,00000001	
0,1023	0,1024	-0,0001	0,00000002	
0,1045	0,1046	-0,0001	0,00000000	
Total SSE			0,00000003	
MSE			0,0000000042	

Berdasarkan Tabel 10 dapat dijelaskan, bahwa hasil pelatihan dengan model arsitektur 4-10-1 dengan nilai konvergensi pada iterasi ke 577 dengan nilai *Performance* sebesar 0,0000000042.

Hasil pengukuran kinerja terhadap model arsitektur 4-10-1 (*trainoss*), seperti ditunjukkan pada Tabel 11.

Tabel 11. Hasil pengukuran kinerja terhadap arsitektur jaringan syaraf 4-10-1 (*trainoss*)

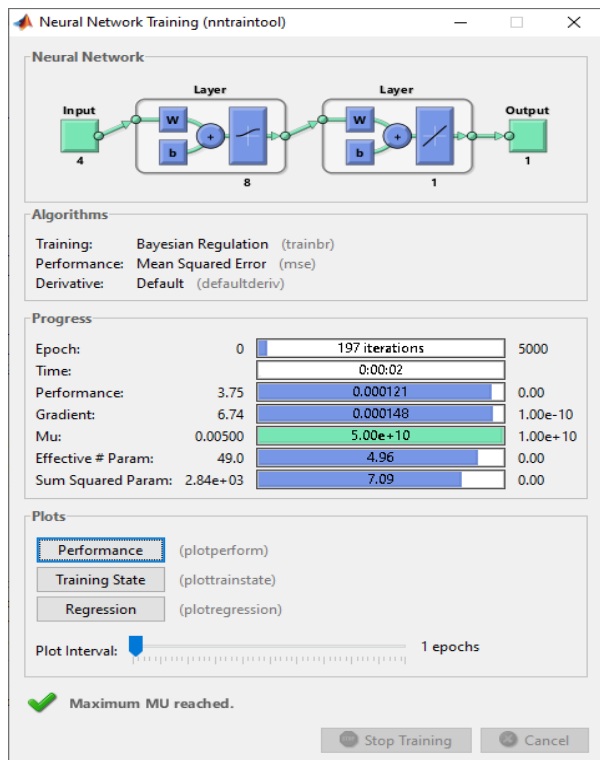
Target (Y2)	Epoch 1			
	Actual	Error	SSE	Performance
0,6063	0,8594	-0,2531	0,064059720	0,231618030
0,2854	1,5691	-1,2837	1,647955221	
0,7222	0,5691	0,1531	0,023426707	
0,4308	0,5244	-0,0936	0,008765678	
0,1133	0,4233	-0,3100	0,096085320	
0,1058	0,1498	-0,0440	0,001936625	
0,1174	0,1343	-0,0169	0,000286330	
0,1006	0,2027	-0,1021	0,010428639	
Total SSE			1,852944240	
MSE			0,231618030	

Berdasarkan Tabel 11 dapat dijelaskan, bahwa nilai *Performance* diperoleh sebesar 0,231618030.

Nilai aktual pada Tabel 10 dan Tabel 11 diperoleh dari keluaran hasil pelatihan dan pengukuran kinerja. Nilai *error* diperoleh dari perhitungan Target (Y). Nilai aktual *SSE* diperoleh berdasarkan perhitungan *error* dikuadratkan ( $error^2$ ). Nilai total *SSE* diperoleh dari penjumlahan nilai *SSE* secara keseluruhan, sedangkan nilai *Performance* (*Perf*) diperoleh dari total *SSE* dibagi 8, dengan 8 merupakan jumlah baris data. Berdasarkan hasil pelatihan dan pengukuran kinerja dengan penggunaan aplikasi MATLAB dan Ms. Excel, terdapat kesesuaian (*valid*).

#3) Pelatihan dan pengukuran kinerja dengan metode metode *bayesian regulation*

@3.a) Jaringan syaraf model arsitektur 4-8-1  
 Tampilan hasil pelatihan terhadap model arsitektur 4-8-1 (*trainbr*), seperti ditunjukkan pada Gambar 7.



Gambar 7. Pelatihan Model 4-8-1 (*trainbr*)

Berdasarkan Gambar 7 dapat dijelaskan, bahwa hasil pelatihan dengan penggunaan fungsi *trainoss* pada model 4-8-1 dihasilkan *epoch* sebesar 197 iterasi dengan waktu latih selama 2 detik.

Hasil pelatihan terhadap model arsitektur 4-8-1 (*trainbr*), seperti ditunjukkan pada Tabel 12.

Tabel 12. Hasil Pelatihan Model Arsitektur 4-8-1 (*trainbr*)

Target (Y1)	Epoch 197			
	Actual	Error	SSE	Performance
0,7423	0,7265	0,0158	0,00024868	0,00012119
0,1511	0,1471	0,0040	0,00001571	
0,6878	0,6940	-0,0062	0,00003842	
0,6640	0,6732	-0,0092	0,00008516	
0,1023	0,1099	-0,0076	0,00005822	
0,1227	0,1080	0,0147	0,00021594	
0,1023	0,1190	-0,0167	0,00027991	
0,1045	0,0993	0,0052	0,00002745	
Total SSE			0,00096950	
MSE			0,00012119	

Berdasarkan Tabel 12 dapat dijelaskan, bahwa hasil pelatihan dengan model arsitektur 4-8-1 dengan nilai konvergensi pada iterasi ke 197 dengan nilai *Performance* sebesar 0,00012119.

Hasil pengukuran kinerja terhadap model arsitektur 4-8-1 (*trainoss*), seperti ditunjukkan pada Tabel 13.



Tabel 13. Hasil pengukuran kinerja terhadap arsitektur jaringan syaraf 4-8-1 (*trainbr*)

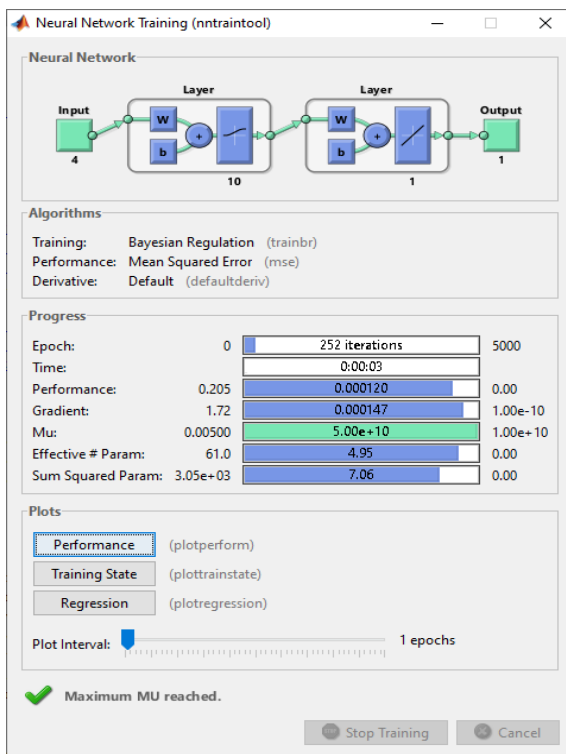
Target (Y2)	Epoch 1			
	Actual	Error	SSE	Performance
0,6063	0,7938	-0,1875	0,035156331	0,014848077
0,2854	0,4591	-0,1737	0,030181099	
0,7222	0,5005	0,2217	0,049132205	
0,4308	0,4752	-0,0444	0,001973598	
0,1133	0,1596	-0,0463	0,002141498	
0,1058	0,1128	-0,0070	0,000049099	
0,1174	0,1051	0,0123	0,000150767	
0,1006	0,1007	-0,0001	0,000000015	
Total SSE			0,118784613	
MSE			0,014848077	

Berdasarkan Tabel 9 dapat dijelaskan, bahwa nilai *Performance* diperoleh sebesar 0,014848077.

Nilai aktual pada Tabel 12 dan Tabel 13 diperoleh dari keluaran hasil pelatihan dan pengukuran kinerja. Nilai *error* diperoleh dari perhitungan Target (Y). Nilai aktual *SSE* diperoleh berdasarkan perhitungan *error* dikuadratkan ( $error^2$ ). Nilai total *SSE* diperoleh dari penjumlahan nilai *SSE* secara keseluruhan, sedangkan nilai *Performance* (*Perf*) diperoleh dari total *SSE* dibagi 8, dengan 8 merupakan jumlah baris data. Berdasarkan hasil pelatihan dan pengukuran kinerja dengan penggunaan aplikasi MATLAB dan Ms. Excel, terdapat kesesuaian (*valid*).

@3.b) Jaringan syaraf model arsitektur 4-10-1

Tampilan hasil pelatihan terhadap model arsitektur 4-10-1 (*trainbr*), seperti ditunjukkan pada Gambar 8



Gambar 8. Pelatihan Model 4-10-1 (*trainbr*)

Berdasarkan Gambar 6 dapat dijelaskan, bahwa hasil pelatihan dengan penggunaan fungsi *trainoss* pada model 4-10-1 dihasilkan *epoch* sebesar 2527 iterasi dengan waktu latih selama 3 detik.

Gambar 8 menjelaskan bahwa hasil pelatihan penggunaan fungsi *trainbr* pada model 4-10-1 menghasilkan *epoch* sebesar 2527 iterasi dengan waktu latih selama 3 detik.

Hasil pelatihan terhadap model arsitektur 4-10-1 (*trainbr*), seperti ditunjukkan pada Tabel 14.

Tabel 14. Hasil Pelatihan Model Arsitektur 4-10-1 (*trainbr*)

Target (Y1)	Epoch 252			
	Actual	Error	SSE	Performance
0,7423	0,7266	0,0157	0,00024553	0,00012038
0,1511	0,1471	0,0040	0,00001571	
0,6878	0,6940	-0,0062	0,00003842	
0,6640	0,6731	-0,0091	0,00008333	
0,1023	0,1098	-0,0075	0,00005671	
0,1227	0,1080	0,0147	0,00021594	
0,1023	0,1190	-0,0167	0,00027991	
0,1045	0,0993	0,0052	0,00002745	
Total SSE			0,00096300	
MSE			0,00012038	

Berdasarkan Tabel 14 dapat dijelaskan, bahwa hasil pelatihan dengan model arsitektur 4-10-1 dengan nilai konvergensi pada iterasi ke 252 dengan nilai *Performance* sebesar 0,00012038.

Hasil pengukuran kinerja terhadap model arsitektur 4-10-1 (*trainbr*), seperti ditunjukkan pada Tabel 15.

Tabel 15. Hasil Pengujian Model Arsitektur 4-10-1 (*trainbr*)

Target (Y2)	Epoch 1			
	Actual	Error	SSE	Performance
0,6063	0,7944	-0,1881	0,035381692	0,014875502
0,2854	0,4587	-0,1733	0,030042277	
0,7222	0,5001	0,2221	0,049309692	
0,4308	0,4748	-0,0440	0,001938218	
0,1133	0,1595	-0,0462	0,002132253	
0,1058	0,1128	-0,0070	0,000049099	
0,1174	0,1051	0,0123	0,000150767	
0,1006	0,1007	-0,0001	0,000000015	
Total SSE			0,119004012	
MSE			0,014875502	

Berdasarkan Tabel 15 dapat dijelaskan, bahwa nilai *Performance* diperoleh sebesar 0,014875502.

Nilai aktual pada Tabel 14 dan Tabel 15 diperoleh dari keluaran hasil pelatihan dan pengukuran kinerja. Nilai *error* diperoleh dari perhitungan Target (Y). Nilai aktual *SSE* diperoleh berdasarkan perhitungan *error* dikuadratkan ( $error^2$ ). Nilai total *SSE* diperoleh dari penjumlahan nilai *SSE* secara keseluruhan, sedangkan nilai *Performance* (*Perf*) diperoleh dari total *SSE* dibagi 8, dengan 8 merupakan jumlah baris data. Berdasarkan hasil pelatihan dan pengukuran kinerja dengan penggunaan aplikasi MATLAB dan Ms. Excel, terdapat kesesuaian (*valid*).

3.4. Analisis Hasil dan Evaluasi

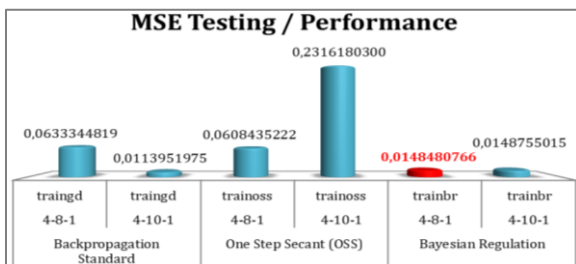
Berdasarkan hasil pelatihan dan pengujian penggunaan aplikasi Matlab 2011b dan Ms. Excel, tahapan selanjutnya adalah menganalisis hasil dan mengevaluasi untuk memperoleh model arsitektur dan fungsi pelatihan terbaik.

Tabel 16. Hasil Analisis fungsi pelatihan yang digunakan

Model	Metode	Fungsi Training	Epoch (Iterasi)	MSE Testing / Performance
4-8-1	Backpropagation Standard	traingd	1445	0,0633344819
	One-Step Secant (OSS)	trainoss	52	0,0608435222
	Bayesian regulation	trainbr	197	0,0148480766
4-10-1	Backpropagation Standard	traingd	2420	0,0113951975
	One-Step Secant (OSS)	trainoss	577	0,2316180300
	Bayesian regulation	trainbr	252	0,0148755015

Berdasarkan Tabel 16 dapat dijelaskan, bahwa perbandingan terhadap masing-masing metode fungsi pelatihan dan pengukuran kinerja yang telah dilakukan. Berdasarkan analisis, terdapat 2 nilai MSE Testing (*Performance*) dengan nilai terendah, yaitu (i) model arsitektur 4-10-1 berbasis metode backpropagation standar dengan epoch 2420 iterasi dan nilai MSE 0,0113951975 dan (ii) model arsitektur 4-8-1 dengan metode Bayesian regulation dengan epoch sebesar 197 iterasi dan nilai MSE Testing/Performance sebesar 0,0148480766.

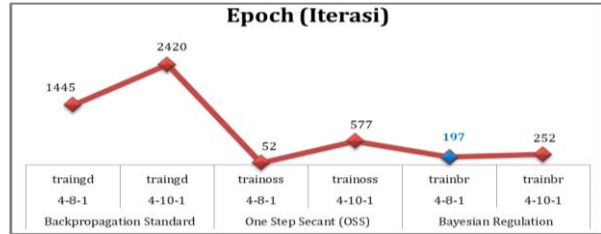
Metode Bayesian regulation terpilih menjadi fungsi pelatihan terbaik, walaupun dengan nilai MSE nomor dua terendah dibandingkan backpropagation standar, karena mampu dengan fungsi pelatihan terhadap jaringan syaraf lebih optimal. Tampilan nilai MSE Testing/Performance, seperti ditunjukkan pada Gambar 9.



Gambar 9. Tampilan nilai MSE Testing/Performance

Berdasarkan Gambar 9 ditunjukkan nilai MSE Testing/Performance dari masing-masing model arsitektur dan fungsi pelatihan yang digunakan. Tampilan nilai Epoch, seperti ditunjukkan pada Gambar 10.

Berdasarkan Gambar 10 ditunjukkan nilai Epoch dari masing-masing model arsitektur dan fungsi pelatihan yang digunakan dengan fungsi pelatihan dengan metode bayesian regulation dan model arsitektur jaringan 4-8-1 sebagai pilihan terbaik.



Gambar 10. Tampilan nilai Epoch

4. Kesimpulan

Berdasarkan hasil dan pembahasan, maka disimpulkan sesuai tujuan dan sasaran penelitian. Fungsi pelatihan dengan metode Bayesian regulation dapat digunakan dan dimanfaatkan untuk melakukan peramalan/prediksi bencana alam di Indonesia, karena waktu pelatihan untuk pencapaian konvergensi tidak terlalu lama dan Performance yang dihasilkan cukup baik dibandingkan dengan penggunaan metode backpropagation standar maupun metode One-Step Secant (OSS). Hal itu berdasarkan analisis, bahwa fungsi pelatihan bayesian regulation (trainbr) mampu untuk pelatihan jaringan syaraf dengan tingkat keoptimalan lebih baik, yaitu dihasilkan nilai epoch dan Performance pelatihan yang lebih cepat dan MSE pengukuran kinerja yang cukup baik walaupun fungsi pelatihan dengan metode backpropagation standar dengan nilai MSE pengukuran kinerja lebih kecil, tetapi backpropagation standar perlu waktu yang cukup lama untuk pelatihan jaringan syaraf, terbukti dengan penunjukan nilai epoch lebih besar dibandingkan dengan kedua metode lain.

Daftar Rujukan

- [1] H. K. Ghrilahre and R. K. Prasad, "Prediction of Thermal Performance of Unidirectional Flow Porous Bed Solar Air Heater with Optimal Training Function Using Artificial Neural Network," *Energy Procedia*, vol. 109, pp. 369–376, 2017.
- [2] E. Siregar, H. Mawengkang, E. B. Nababan, and A. Wanto, "Analysis of Backpropagation Method with Sigmoid Bipolar and Linear Function in Prediction of Population Growth," *Journal of Physics: Conference Series*, vol. 1255, no. 1, p. 012023, 2019.
- [3] M. Tyrtaiou, A. Papaleonidas, A. Elenas, and L. Iliadis, "Accomplished Reliability Level for Seismic Structural Damage Prediction Using Artificial Neural Networks," in *Proceedings of the 21st EANN (Engineering Applications of Neural Networks) 2020 Conference. EANN 2020. Proceedings of the International Neural Networks Society*, 2020, vol. 2, pp. 85–98.
- [4] B. Febriadi, Z. Zamzami, Y. Yunefri, and A. Wanto, "Bipolar function in backpropagation algorithm in predicting Indonesia's coal exports by major destination countries," *IOP Conference Series: Materials Science and Engineering*, vol. 420, no. 1, p. 012087, 2018.
- [5] N. Nasution, A. Zamsuri, L. Lisnawita, and A. Wanto, "Polak-Ribiere updates analysis with binary and linear function in determining coffee exports in Indonesia," *IOP Conference Series: Materials Science and Engineering*, vol. 420, no. 1, pp. 1–9, 2018.
- [6] A. Dolara, F. Grimaccia, S. Leva, M. Mussetta, and E. Ogliari, "Comparison of training approaches for photovoltaic forecasts by means of machine learning," *Applied Sciences (Switzerland)*, vol. 8, no. 2, 2018.

- [7] H. Wang, R. Czerminski, and A. C. Jamieson, "Neural Networks and Deep Learning," in *The Machine Age of Customer Insight*, P. Einhorn, M. Löffler, M., de Bellis, E., Herrmann, A. and Burghartz, Ed. Emerald Publishing Limited, 2021, pp. 91–101.
- [8] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.
- [9] B. S. Rem *et al.*, "Identifying quantum phase transitions using artificial neural networks on experimental data," *Nature Physics*, vol. 15, no. 9, pp. 917–920, 2019.
- [10] R. Novickis, D. J. Justs, K. Ozols, and M. Greitans, "An Approach of Feed-Forward Neural Network," *Electronics*, vol. 9, no. 12, p. 2193, 2020.
- [11] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, "Machine learning for active matter," *Nature Machine Intelligence*, vol. 2, no. 2, pp. 94–103, 2020.
- [12] C. T. Chen and G. X. Gu, "Generative Deep Neural Networks for Inverse Materials Design Using Backpropagation and Active Learning," *Advanced Science*, vol. 7, no. 5, pp. 1–10, 2020.
- [13] R. García-Ródenas, L. J. Linares, and J. A. López-Gómez, "Memetic algorithms for training feedforward neural networks: an approach based on gravitational search algorithm," *Neural Computing and Applications*, vol. 33, pp. 2561–2588, 2020.
- [14] E. Yan, J. Song, C. Liu, J. Luan, and W. Hong, "Comparison of support vector machine, back propagation neural network and extreme learning machine for syndrome element differentiation," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2453–2481, 2020.
- [15] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.
- [16] I. D. Uwanakwa and P. Akpinar, "Investigations on the Influence of Variations in Hidden Neurons and Training Data Percentage on the Efficiency of Concrete Carbonation Depth Prediction with ANN," *Advances in Intelligent Systems and Computing*, vol. 1095, pp. 958–965, 2020.
- [17] S. Alsammarräie and N. K. Hussein, "A New Hybrid Grasshopper Optimization - Backpropagation for Feedforward Neural Network Training," *Tikrit Journal of Pure Science*, vol. 25, no. 1, pp. 118–127, 2020.
- [18] E. Bas, E. Egrioglu, and U. Yolcu, "A hybrid algorithm based on artificial bat and backpropagation algorithms for multiplicative neuron model artificial neural networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–9, 2020.
- [19] I. T. Sui Kim, V. Sethu, S. K. Arumugasamy, and A. Selvarajoo, "Fenugreek seeds and okra for the treatment of palm oil mill effluent (POME) – Characterization studies and modeling with backpropagation feedforward neural network (BFNN)," *Journal of Water Process Engineering*, vol. 37, no. 101500, pp. 1–16, 2020.
- [20] I. C. Afolabi, S. I. Popoola, and O. S. Bello, "Modeling pseudo-second-order kinetics of orange peel-paracetamol adsorption process using artificial neural network," *Chemometrics and Intelligent Laboratory Systems*, vol. 203, no. 104053, pp. 1–47, 2020.
- [21] Isha, A. S. Chaudhary, and D. K. Chaturvedi, "Effects of Activation Function and Input Function of ANN for Solar Power Forecasting," in *Lecture Notes in Networks and Systems*, vol. 94, M. L. Kolhe, S. Tiwari, M. C. Trivedi, and K. K. Mishra, Eds. Springer, 2020, pp. 329–342.
- [22] A. Panyafong, N. Neamsorn, and C. Chaichana, "Heat load estimation using Artificial Neural Network," *Energy Reports*, vol. 6, pp. 742–747, 2020.
- [23] K. Kumar, V. Singh, and T. Roshni, "Efficacy of hybrid neural networks in statistical downscaling of precipitation of the Bagmati river basin," *Journal of Water and Climate Change*, vol. 11, no. 4, pp. 1302–1322, 2020.
- [24] M. Žic, V. Subotić, S. Pereverzyev, and I. Fajfar, "Solving CNLS problems using Levenberg-Marquardt algorithm: A new fitting strategy combining limits and a symbolic Jacobian matrix," *Journal of Electroanalytical Chemistry*, vol. 866, no. 114171, pp. 1–9, 2020.
- [25] J. Bilski, B. Kowalczyk, A. Marchlewska, and J. M. Zurada, "Local Levenberg-Marquardt Algorithm for Learning Feedforward Neural Networks," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, no. 4, pp. 299–316, 2020.
- [26] N. L. W. S. R. Ginantra, M. A. Hanafiah, A. Wanto, R. Winanjaya, and H. Okprana, "Utilization of the Batch Training Method for Predicting Natural Disasters and Their Impacts," *IOP Conf. Series: Materials Science and Engineering*, vol. 1071, no. 012022, pp. 1–7, 2021.
- [27] R. Jayaseelan, G. Pandulu, and G. Ashwini, "Neural Networks for the Prediction of Fresh Properties and Compressive Strength of Flowable Concrete," *Journal of Urban and Environmental Engineering*, vol. 13, no. 1, pp. 183–197, 2019.
- [28] H. Espitia, I. Machon, and H. Lopez, "Control of a Microturbine Using Neural Networks," in *Communications in Computer and Information Science*, vol. 1052, J. C. Figueroa-García, M. Duarte-González, S. Jaramillo-Isaza, A. D. Orjuela-Cañon, and Y. D.-G. (Eds.), Eds. Springer, 2019, pp. 202–213.
- [29] D. Gong, J. Feng, W. Xiao, and S. Sun, "Spectral Reconstruction Based on Bayesian regulation Neural Network," in *Smart Innovation, Systems and Technologies*, vol. 179, R. Kountchev, S. Patnaik, J. Shi, and M. N. Favorskaya, Eds. Springer, 2019, pp. 77–85.
- [30] U. G. Inyang, E. E. Akpan, and O. C. Akinyokun, "A Hybrid Machine Learning Approach for Flood Risk Assessment and Classification," *International Journal of Computational Intelligence and Applications*, vol. 19, no. 2, pp. 1–20, 2020.
- [31] T. Afriliansyah and Z. Zulfahmi, "Prediction of Life Expectancy in Aceh Province by District City Using the Cyclical Order Algorithm," *International Journal of Information System & Technology*, vol. 3, no. 2, pp. 268–275, 2020.
- [32] G. S. Rao, S. S. Rani, and B. P. Rao, "Computed Tomography Medical Image Compression using Conjugate Gradient," *2019 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 169–173, 2019.
- [33] Q. H. Nguyen *et al.*, "A Novel Hybrid Model Based on a Feedforward Neural Network and One Step Secant Algorithm for Prediction of Load-Bearing Capacity of Rectangular Concrete-Filled Steel Tube Columns," *Molecules*, vol. 25, no. 15, pp. 1–26, 2020.
- [34] M. Zandieh, A. Azadeh, B. Hadadi, and M. Saberi, "Application of Artificial Neural Networks for Airline Number of Passenger Estimation in Time Series State," *Journal of Applied Sciences*, vol. 9, no. 6, pp. 1001–1013, 2009.
- [35] A. Perera, H. Azamathulla, and U. Rathnayake, "Comparison of different artificial neural network (ANN) training algorithms to predict the atmospheric temperature in Tabuk, Saudi Arabia," *Journal MAUSAM*, vol. 71, no. 2, pp. 233–244, 2020.
- [36] C. Perez, *Big Data and Deep Learning Examples with Matlab*. Lulu Press, Inc, 2020.
- [37] P. Parulian *et al.*, "Analysis of Sequential Order Incremental Methods in Predicting the Number of Victims Affected by Disasters," *Journal of Physics: Conference Series*, vol. 1255, no. 1, p. 012033, 2019.
- [38] C. K. Arthur, V. A. Temeng, and Y. Y. Ziggah, "Performance Evaluation of Training Algorithms in Backpropagation Neural Network Approach to Blast-Induced Ground Vibration Prediction," *Ghana Mining Journal*, vol. 20, no. 1, pp. 20–33, 2020.
- [39] BNPB, "Infografis Bencana," *Geoportal Kebencanaan Indonesia*, 2021. [Online]. Available: <https://gis.bnpb.go.id/>. [Accessed: 17-Mar-2021].
- [40] S. Setti and A. Wanto, "Analysis of Backpropagation Algorithm in Predicting the Most Number of Internet Users in the World," *JOIN (Jurnal Online Informatika)*, vol. 3, no. 2, pp. 110–115, 2018.