

PEMILIHAN VARIABEL YANG RELEVAN PADA ATURAN FUZZY MENGUNAKAN JARINGAN SYARAF

Sri Kusumadewi

Jurusan Teknik Informatika, Fakultas Teknologi Industri

Universitas Islam Indonesia Yogyakarta

email: cicie@fti.uii.ac.id

Abstrak

Pada kebanyakan sistem inferensi fuzzy, fungsi keanggotaan pada setiap himpunan fuzzy dibentuk melalui pendekatan terhadap fungsi-fungsi tertentu. Selain itu, sistem juga belum mengakomodasi penentuan variabel-variabel yang relevan pada sistem tersebut. Padahal adakalanya, terdapat variabel yang tidak memiliki tingkat relevansi yang cukup tinggi. Pada penelitian ini, akan dibangun suatu model aturan pada sistem inferensi fuzzy melalui proses pembelajaran jaringan syaraf dengan menggunakan algoritma backpropagation. Sistem dibangun melalui tahap, yaitu: (1) pemilihan variabel input-output dan data pelatihan; (2) proses pembelajaran pada bagian input; (3) pengelompokan data pelatihan; (4) pembelajaran jaringan syaraf yang berhubungan dengan bagian anteseden (bagian IF) pada aturan-aturan inferensi fuzzy; (5) pembelajaran jaringan syaraf yang berhubungan dengan bagian konsekuen (bagian THEN) pada aturan-aturan inferensi fuzzy; (6) penyederhanaan bagian konsekuen (bagian THEN) menggunakan metode eliminasi backward; dan (7) pengujian. Hasil penelitian terhadap data yang diberikan (4 variabel input), menghasilkan 2 aturan fuzzy dengan mengeliminasi variabel kedua untuk bagian anteseden, dan mengeliminasi variabel pertama untuk bagian konsekuen. Hasil pengujian menunjukkan SSE antara target output dengan output jaringan adalah 0,3825; dan koefisien korelasi sebesar 0,8461; yang berarti bahwa sistem inferensi fuzzy yang terbentuk berdasarkan pembelajaran jaringan syaraf memiliki validitas yang cukup tinggi

Keywords: sistem inferensi fuzzy, backpropagation, pembelajaran.

1. PENDAHULUAN

1.1 Latar Belakang

Salah kelemahan terbesar dari sistem inferensi fuzzy adalah penentuan fungsi keanggotaan dan pembangkitan fungsi pembelajaran pada aturan-aturan inferensi. Biasanya, fungsi keanggotaan pada setiap himpunan fuzzy dibentuk melalui pendekatan terhadap fungsi-fungsi tertentu (Cox, 1994), seperti: fungsi linear, sigmoid, gauss, beta, dll. Pemilihan fungsi tersebut tentunya akan berimplikasi pada tingginya tingkat subyektivitas.

Apabila dipunyai beberapa *data history* terhadap permasalahan tertentu, fungsi keanggotaan dapat dibentuk dengan terlebih dahulu melakukan *clustering* terhadap data-data tersebut. Pada sistem fuzzy, dapat digunakan fuzzy c-means (FCM) atau *subtractive clustering* untuk mendapatkan kelompok - kelompok data yang nantinya dapat difungsikan untuk membantu penentuan fungsi keanggotaan.

Permasalahan lain yang juga muncul pada pembentukan sistem inferensi fuzzy adalah penentuan variabel-variabel yang relevan. Dalam suatu sistem, adakalanya suatu variabel memiliki tingkat kepentingan yang tidak terlalu tinggi. Hal ini kemungkinan disebabkan oleh keberadaannya yang sebenarnya telah terwakili oleh variabel yang lain. Jika hal ini terjadi, maka variabel tersebut dapat dieliminasi untuk mengurangi beban komputasi dan meningkatkan akurasi hasil. Pemilihan tingkat

kepentingan variabel ini juga belum diakomodasi oleh pembentukan sistem inferensi fuzzy biasa.

Di lain pihak, algoritma pembelajaran pada jaringan syaraf memiliki kemampuan untuk melatih *data history* untuk kepentingan adaptasi. Jaringan syaraf dengan bobot-bobot yang telah terbentuk melalui proses pembelajaran terbukti mampu menyelesaikan beberapa kasus seperti: peramalan, pengenalan pola, klasifikasi, dll (Fausett, 1994). Pada kasus peramalan, jaringan syaraf dapat digunakan untuk mengestimasi nilai output apabila diberikan beberapa input. Sehingga pada kasus pencarian variabel input yang relevan untuk sistem inferensi fuzzy, jaringan syaraf dapat digunakan sebagai alat untuk meramalkan nilai output baik pada bagian anteseden (pernyataan setelah IF pada aturan fuzzy) maupun pada bagian konsekuen (pernyataan setelah THEN pada aturan fuzzy).

1.2 Tujuan

Tujuan penelitian ini adalah membangun suatu model aturan pada sistem inferensi fuzzy melalui proses pembelajaran jaringan syaraf.

2. DASAR TEORI

Lin (1996), menggunakan jaringan syaraf untuk merealisasikan atau membangkitkan sistem inferensi fuzzy model Sugeno baik pada bagian anteseden (membangkitkan fungsi keanggotaan), maupun pada

bagian konsekuen (melakukan inferensi). Takagi & Hayashi (1991) menggunakan jaringan syaraf dengan algoritma pembelajaran backpropagation untuk membangun himpunan-himpunan fuzzy pada bagian anteseden, dan fungsi inferensi yang ada pada bagian konsekuen. Aturan inferensi yang diberikan memiliki format sebagai berikut:

$$R^s: \text{IF } x = (x_1, x_2, \dots, x_n) \text{ is } A_s \text{ THEN} \\ y_s = NN_s(x_1, x_2, \dots, x_m); \quad s = 1, 2, \dots, r. \quad (1)$$

dengan r adalah jumlah aturan inferensi, A_s adalah himpunan fuzzy pada bagian anteseden pada setiap aturan, dan $NN(\cdot)$ adalah jaringan backpropagation dengan input (x_1, x_2, \dots, x_m) dan output y_s . Jumlah variabel yang menjadi input NN sebanyak m ($m \leq n$) ditentukan dengan cara memilih model yang optimun.

Langkah-langkah pembentukan sistem inferensi fuzzy model Sugeno melalui pengendali jaringan syaraf adalah sebagai berikut:

a. Pemilihan variabel input – output dan data pelatihan.

Diberikan variabel output y dan calon variabel input x_j , $j=1,2,\dots,n$. Pada bagian ini akan ditetapkan variabel-variabel input yang relevan, x_j , $j=1,2,\dots,m$, yang berhubungan output y_i , $i=1,2,\dots,N$, dengan menggunakan jaringan backpropagation. Hal ini dilakukan dengan menggunakan metode eliminasi *backward*, dengan fungsi biaya *sum squared error* (SSE). Seleksi variabel dilakukan dengan cara mengeliminasi variabel-variabel yang tidak diperlukan, dan mempertahankan variabel-variabel yang memberikan korelasi yang cukup signifikan terhadap variabel output y_i . Misalkan terdapat N pasangan data input – output, maka kita bagi kumpulan data tersebut menjadi N_t data pelatihan (TRD) dan N_c data pengujian (CHD), sehingga $N_t + N_c = N$.

b. Pengelompokan (*clustering*) data pelatihan.

Pada bagian ini, TRD akan dibagi menjadi r kelas dengan menggunakan metode pengclusteran tertentu. Sehingga nantinya akan terdapat r buah aturan R^s , $s=1,2,\dots,r$. Pasangan input – output pada cluster ke- s (aturan ke- s) direpresentasikan sebagai (x_i^s, y_i^s) , $i=1,2,\dots,N_s$; dengan N_s adalah jumlah data yang masuk pada kelas ke- s .

c. Pembelajaran jaringan syaraf yang berhubungan dengan bagian anteseden (bagian IF) pada aturan-aturan inferensi fuzzy.

Untuk setiap vektor input pada TRD kita tentukan $m_i = (m_i^1, m_i^2, \dots, m_i^r)$, $i=1,2,\dots,N_t$; sebagai berikut:

$$m_i^k = \begin{cases} 1; & k = s \\ 0; & k \neq s \end{cases} \quad (2)$$

Kemudian derajat keanggotaan setiap data dapat diperoleh sebagai output jaringan yang telah dilatih, sebagai:

$$\mu_{A_s}(x_i) = \hat{m}_i^s; \quad i=1,2,\dots,N; \quad s=1,2,\dots,r. \quad (3)$$

d. Pembelajaran jaringan syaraf yang berhubungan dengan bagian konsekuen (bagian THEN) pada aturan-aturan inferensi fuzzy.

Pada bagian ini akan dilakukan pembelajaran jaringan syaraf pada bagian THEN dari R^s dengan input $x = (x_{i1}^s, x_{i2}^s, \dots, x_{im}^s)$ dan target output y_i^s , $i=1,2,\dots,N_s$. Selanjutnya hasil pelatihan akan diujikan pada CHD, dengan input $x = (x_{i1}^s, x_{i2}^s, \dots, x_{im}^s)$, $i=1,2,\dots,N_c$ untuk mendapatkan SSE sebagai berikut:

$$E_m^s = \sum_{i=1}^{N_c} [y_i - \mu_s(x_i)\mu_{A_s}(x_i)]^2 \quad (4)$$

dengan estimasi $\mu_s(x_i)$ diperoleh dari output jaringan. Selanjutnya error dengan pembobotan dapat dihitung sebagai berikut:

$$E_m^s = \sum_{i=1}^{N_c} \mu_{A_s}(x_i)[y_i - \mu_s(x_i)\mu_{A_s}(x_i)]^2 \quad (5)$$

e. Penyederhanaan bagian konsekuen (bagian THEN) menggunakan metode eliminasi *backward*.

Dari m variabel input yang berpartisipasi pada jaringan syaraf, tidak semuanya memiliki kontribusi yang cukup baik. Kita dapat mencoba dengan mengeliminasi salah satu variabel, dan kemudian melatih jaringan kembali untuk mendapatkan SSE (E_{m-1}^{SP}) sebagai berikut:

$$E_{m-1}^{SP} = \sum_{i=1}^{N_c} [y_i - \mu_s(\hat{x}_i)\mu_{A_s}(\hat{x}_i)]^2; \quad p=1,2,\dots,m. \quad (6)$$

dengan $x_i = (x_{i,1}; \dots; x_{i,p-1}; x_{i,p+1}; \dots; x_{im})$. Selanjutnya bandingkan **persamaan 4** dengan **persamaan 6**, jika:

$$E_m^s > E_{m-1}^{SP} \quad (7)$$

maka x_p dapat dihilangkan. Proses ini diulangi hingga, **persamaan 7** benar-benar dipenuhi.

f. Penentuan output akhir.

Nilai y_i^* akhirnya dapat diperoleh sebagai berikut:

$$y_i = \frac{\sum_{s=1}^r \mu_{AS}(x_i) \mu_s(x_i)}{\sum_{s=1}^r \mu_{AS}(x_i)}; \quad i=1,2,\dots,N \quad (8)$$

3. MODEL YANG DIUSULKAN

3.1 Pemilihan variabel input-output dan data pelatihan.

Kusumadewi (2006), memberikan 30 sampel data untuk dibentuk sistem inferensi fuzzy dalam 4 variabel input, x_1 , x_2 , x_3 , dan x_4 sebagaimana terlihat pada **Tabel 1**. Data ke-1 sampai 15 akan digunakan sebagai data pelatihan (TRD), sedangkan data ke-16 sampai 30 akan digunakan sebagai data pengujian (CHD).

Tabel 1 Data-data ada sistem.

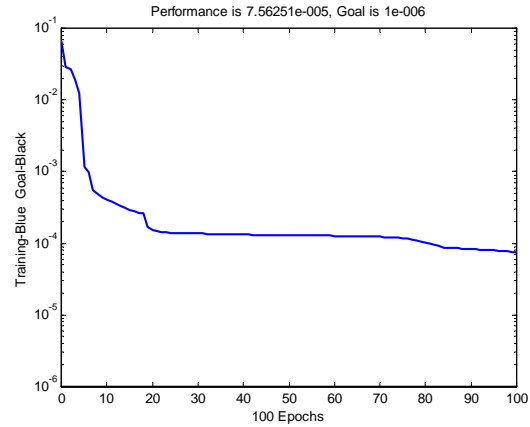
Data ke-	variabel input				target output (y)
	x_1	x_2	x_3	x_4	
1	0,52	0,22	0,31	0,43	0,5144
2	0,45	0,25	0,57	0,48	0,5588
3	0,60	0,19	0,51	0,40	0,4894
4	0,81	0,27	0,42	0,52	0,6219
5	0,25	0,23	0,64	0,46	0,5217
6	0,91	0,30	0,73	0,62	0,7269
7	0,63	0,35	0,50	0,71	0,8011
8	0,82	0,37	0,24	0,74	0,8429
9	0,90	0,46	0,35	0,90	1,0068
10	0,65	0,31	0,42	0,63	0,7225
11	0,79	0,47	0,91	0,79	0,8901
12	0,81	0,43	0,75	0,85	0,9514
13	0,23	0,21	0,82	0,52	0,5795
14	0,17	0,22	0,27	0,44	0,4936
15	0,52	0,34	0,61	0,68	0,7637
16	0,35	0,23	0,54	0,46	0,5310
17	0,12	0,37	0,44	0,67	0,7164
18	0,43	0,27	0,56	0,58	0,6573
18	0,28	0,43	0,27	0,87	0,9351
20	0,45	0,28	0,66	0,58	0,6587
21	0,14	0,16	0,44	0,33	0,3795
22	0,32	0,44	0,66	0,88	0,9480
23	0,17	0,27	0,54	0,55	0,6030
24	0,32	0,35	0,43	0,75	0,8184
25	0,41	0,28	0,65	0,56	0,6356
26	0,32	0,32	0,44	0,65	0,7184
27	0,17	0,17	0,23	0,33	0,3839
28	0,51	0,31	0,48	0,65	0,7332
29	0,65	0,42	0,45	0,89	0,9824
30	0,34	0,23	0,44	0,46	0,5303

Dengan menggunakan data-data pada **Tabel 1**, akan dibangun aturan-aturan pada sistem inferensi fuzzy dengan input (bagian anteseden) x_1 , x_2 , x_3 , x_4 , dan output (konsekuen) y .

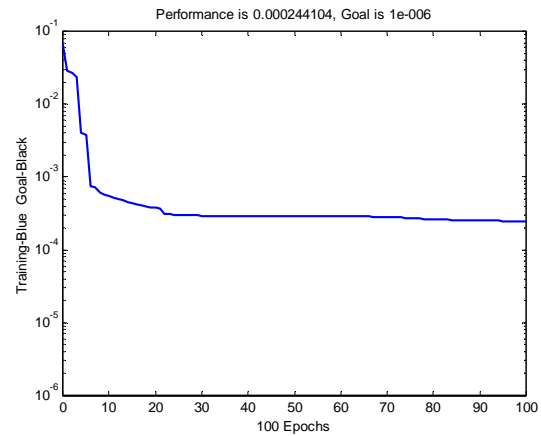
3.2 Proses pembelajaran pada bagian input

Proses pembelajaran dilakukan dengan input x_1 , x_2 , x_3 dan x_4 serta target output y menggunakan backpropagation *levenberg marquardt* (Jang, 1997) dengan 1 lapisan tersembunyi (dengan 5 neuron). Parameter-parameter: maksimum epoh = 100; laju pembelajaran = 1; dan toleransi error = 10^{-6} .

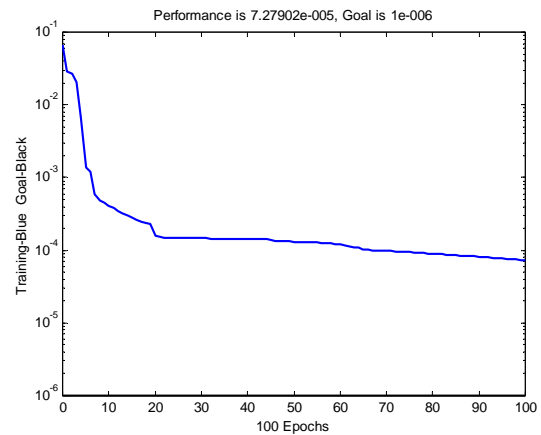
Proses pembelajaran dilakukan dengan semua input (**Gambar 1**), mengeliminasi variabel x_1 (**Gambar 2**), mengeliminasi variabel x_2 (**Gambar 3**), mengeliminasi variabel x_3 (**Gambar 4**), dan mengeliminasi variabel x_4 (**Gambar 5**).



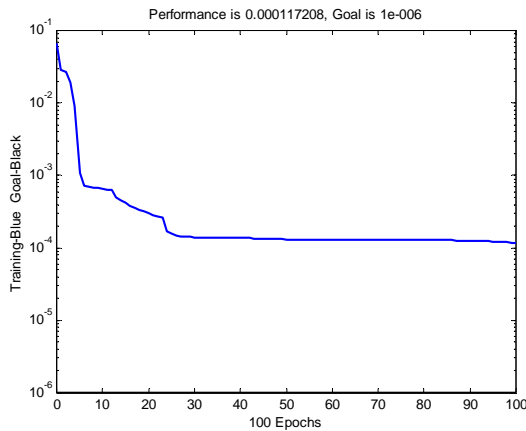
Gambar 1 Hasil pembelajaran dengan semua input.



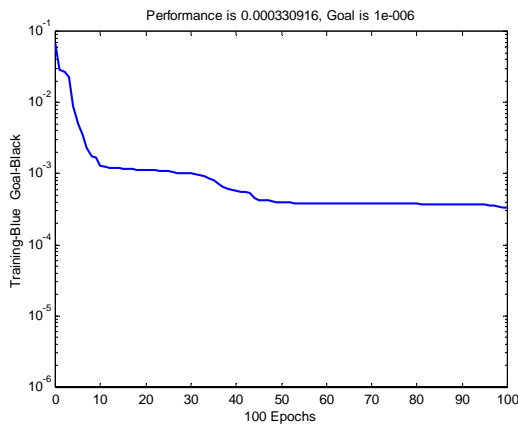
Gambar 2 Hasil pembelajaran dengan eliminasi x_1 .



Gambar 3 Hasil pembelajaran dengan eliminasi x_2 .



Gambar 4 Hasil pembelajaran dengan eliminasi x_3 .



Gambar 5 Hasil pembelajaran dengan eliminasi x_4 .

SSE dihitung terhadap semua data (pelatihan dan pengujian) dan mendapatkan nilai seperti terlihat pada Tabel 2. Dari sini terlihat bahwa, SSE yang terjadi pada saat variabel x_2 dieliminasi cukup kecil, sehingga apabila variabel x_2 dihilangkan, tidak terlalu berpengaruh. Sehingga variabel input yang akan berpartisipasi pada bagian anteseden (bagian IF) adalah x_1 , x_3 , dan x_4 saja.

Tabel 2 Tabel SSE bagian input.

	SSE
Semua variabel digunakan	0,0057
Variabel x_1 dieliminasi	0,0155
Variabel x_2 dieliminasi	0,0045
Variabel x_3 dieliminasi	0,0090
Variabel x_4 dieliminasi	0,0354

3.3 Pengelompokan data pelatihan

Data pelatihan (TRD) dikelompokkan (*cluster*) sebanyak jumlah aturan yang diinginkan. Misalkan diinginkan sebanyak 2 aturan, maka TRD akan dibagi menjadi 2 kelas (2 aturan). Algoritma

pengelompokan FCM (Pal, 2005) (Ross, 2005) (Yan, 1994) digunakan untuk proses tersebut. Dari hasil *clustering*, diperoleh nilai keanggotaan setiap data pada setiap cluster, dan kecenderungan suatu data masuk ke suatu cluster seperti terlihat pada Tabel 3.

Tabel 3 Hasil clustering TRD.

Data ke-	Matriks Partisi		Kecenderungan masuk cluster	
	$\mu_1(x_i)$	$\mu_2(x_i)$	Kelas-1	Kelas-2
1	0,8334	0,1666	*	
2	0,9832	0,0168	*	
3	0,8435	0,1565	*	
4	0,3705	0,6295		*
5	0,9137	0,0863	*	
6	0,1755	0,8245		*
7	0,1149	0,8851		*
8	0,2043	0,7957		*
9	0,1594	0,8406		*
10	0,2959	0,7041		*
11	0,2105	0,7895		*
12	0,1232	0,8768		*
13	0,7864	0,2136	*	
14	0,8318	0,1682	*	
15	0,3903	0,6097		*

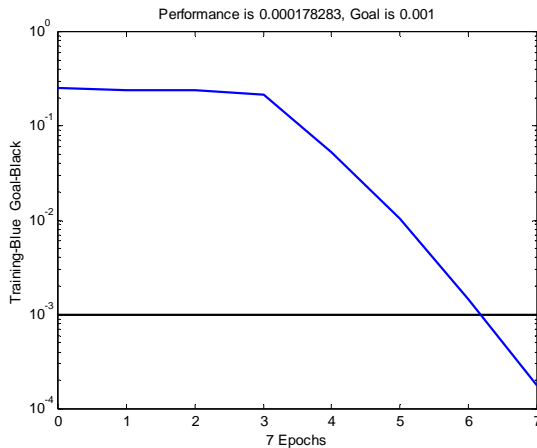
3.4 Pembelajaran jaringan syaraf yang berhubungan dengan bagian anteseden (bagian IF) pada aturan-aturan inferensi fuzzy

Untuk setiap vektor input pada TRD harus ditentukan terlebih dahulu nilai keanggotaan m_i , berdasarkan persamaan 2, hasilnya terlihat pada Tabel 4.

Tabel 4 Nilai keanggotaan setiap cluster (m).

Data ke-	Kecenderungan masuk cluster		Nilai keanggotaan	
	Kelas-1	Kelas-2	m_{S1}	m_{S2}
1	0,8334	0,1666	1	0
2	0,9832	0,0168	1	0
3	0,8435	0,1565	1	0
4	0,3705	0,6295	0	1
5	0,9137	0,0863	1	0
6	0,1755	0,8245	0	1
7	0,1149	0,8851	0	1
8	0,2043	0,7957	0	1
9	0,1594	0,8406	0	1
10	0,2959	0,7041	0	1
11	0,2105	0,7895	0	1
12	0,1232	0,8768	0	1
13	0,7864	0,2136	1	0
14	0,8318	0,1682	1	0
15	0,3903	0,6097	0	1

Selanjutnya jaringan syaraf dengan input x_{i1} , x_{i3} , x_{i4} dan target output m_i , kita latih dengan menggunakan jaringan backpropagation *levenberg marquardt* dengan 1 lapisan tersembunyi (dengan 3 neuron). Parameter-parameter: maksimum epoch = 100; laju pembelajaran = 1; dan toleransi error = 10^{-3} . Proses pembelajaran terlihat pada Gambar 6.



Gambar 6 Hasil pembelajaran pada setiap cluster.

Kemudian derajat keanggotaan, μ_{AS} , setiap data dapat diperoleh sebagai output dari jaringan yang telah dilatih, sebagaimana terlihat pada **Tabel 5**.

Tabel 5 Nilai keanggotaan pada bagian IF (μ_{AS}) TRD & CHD.

Data ke-	Nilai keanggotaan	
	μ_{A1}	μ_{A2}
1	0,9884	0,0116
2	0,9859	0,0141
3	0,9905	0,0095
4	0,0332	0,9668
5	0,9998	0,0002
6	0,0078	0,9922
7	0,0091	0,9909
8	0,0055	0,9945
9	0,0047	0,9953
10	0,0180	0,9820
11	0,0054	0,9946
12	0,0049	0,9951
13	0,9992	0,0008
14	0,9999	0,0001
15	0,0238	0,9762
16	0,9989	0,0011
17	0,8891	0,1109
18	0,5522	0,4478
19	0,0083	0,9917
20	0,5151	0,4849
21	1,0000	0,0000
22	0,0079	0,9921
23	0,9985	0,0015
24	0,0285	0,9715
25	0,8393	0,1607
26	0,3052	0,6948
27	1,0000	0,0000
28	0,0392	0,9608
29	0,0050	0,9950
30	0,9988	0,0012

3.5 Pembelajaran jaringan syaraf yang berhubungan dengan bagian konsekuen (bagian THEN) pada aturan-aturan inferensi fuzzy

Pada bagian ini akan dilakukan pembelajaran jaringan syaraf pada bagian THEN dari R^S dengan

input $x = (x_{i1}^S, x_{i2}^S, \dots, x_{im}^S)$ dan target output y_i^S , $i=1,2,\dots,15$; $s=1,2$. TRD yang akan dilatih pada R^1 (NN_1) seperti terlihat pada **Tabel 6**, sedangkan TRD yang akan dilatih pada R^2 (NN_2) seperti terlihat pada **Tabel 7**.

Tabel 6 Data-data yang akan dilatih pada R^1 (NN_1).

Data ke-	variabel input			target output (y)	Berasal dari data ke-
	x_1	x_3	x_4		
1	0,52	0,31	0,43	0,5144	1
2	0,45	0,57	0,48	0,5588	2
3	0,60	0,51	0,40	0,4894	3
4	0,25	0,64	0,46	0,5217	5
5	0,23	0,82	0,52	0,5795	13
6	0,17	0,27	0,44	0,4936	14

Tabel 7 Data-data yang akan dilatih pada R^2 (NN_2).

Data ke-	variabel input			target output (y)	Berasal dari data ke-
	x_1	x_3	x_4		
1	0,81	0,42	0,52	0,6219	4
2	0,91	0,73	0,62	0,7269	6
3	0,63	0,50	0,71	0,8011	7
4	0,82	0,24	0,74	0,8429	8
5	0,90	0,35	0,90	1,0068	9
6	0,65	0,42	0,63	0,7225	10
7	0,79	0,91	0,79	0,8901	11
8	0,81	0,75	0,85	0,9514	12
9	0,52	0,61	0,68	0,7637	15

Selanjutnya hasil pelatihan akan diujikan pada CHD, dengan input $x = (x_{i1}^S, x_{i2}^S, \dots, x_{im}^S)$, $i=1,2,\dots,15$ untuk mendapatkan SSE berdasarkan **persamaan 4**.

3.6 Penyederhanaan bagian konsekuen (bagian THEN) menggunakan metode eliminasi backward.

Jaringan syaraf NN_1 kemudian dilatih dengan menggunakan jaringan backpropagation *levenberg marquardt* dengan 1 lapisan tersembunyi (dengan 5 neuron). Parameter-parameter: maksimum epoch = 1000; laju pembelajaran = 1; dan toleransi error = 10^{-6} . Selanjutnya kita akan melakukan eliminasi *backward* untuk variabel input. Kemudian melatih jaringan kembali untuk mendapatkan MSE (E_{m-1}^{SP}) berdasarkan **persamaan 6**.

- Pengujian pada NN_1 :

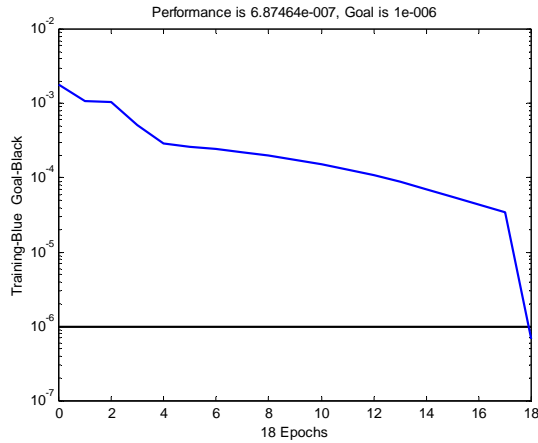
Dengan melakukan beberapa eliminasi pada variabel input, dan dengan menggunakan **persamaan 6** diperoleh SSE seperti terlihat pada **Tabel 8**. Proses pembelajaran dilakukan dengan semua input (**Gambar 7**), mengeliminasi variabel x_1 (**Gambar 8**), mengeliminasi variabel x_3 (**Gambar 9**), dan mengeliminasi variabel x_4 (**Gambar 10**).

Tabel 8 Tabel SSE untuk NN_1 iterasi pertama.

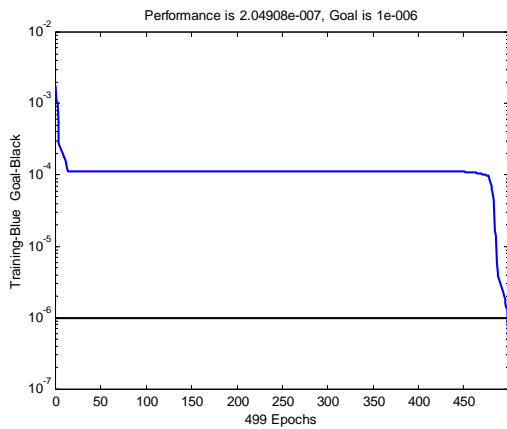
	SSE
Semua variabel digunakan	6,4595

Variabel x_1 dieliminasi	6,2289
Variabel x_3 dieliminasi	6,6843
Variabel x_4 dieliminasi	11,7457

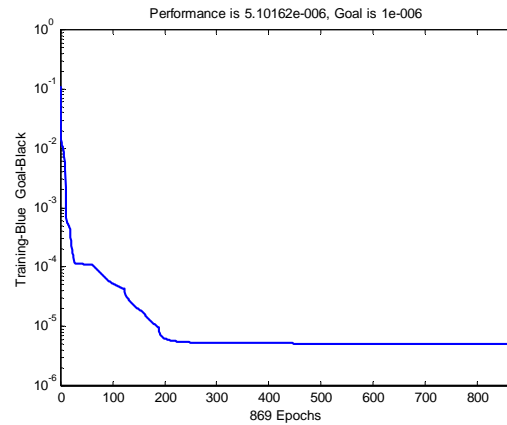
Karena nilai SSE dengan mengeliminasi variabel x_1 lebih kecil daripada SSE dengan menggunakan semua variabel, maka variabel x_1 dapat dihilangkan.



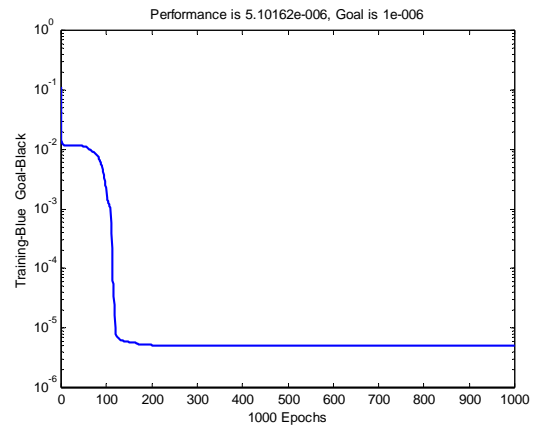
Gambar 7 Hasil pembelajaran dengan semua variabel pada NN_1 .



Gambar 8 Hasil pembelajaran dengan eliminasi x_1 pada NN_1 .



Gambar 9 Hasil pembelajaran dengan eliminasi x_3 pada NN_1 .



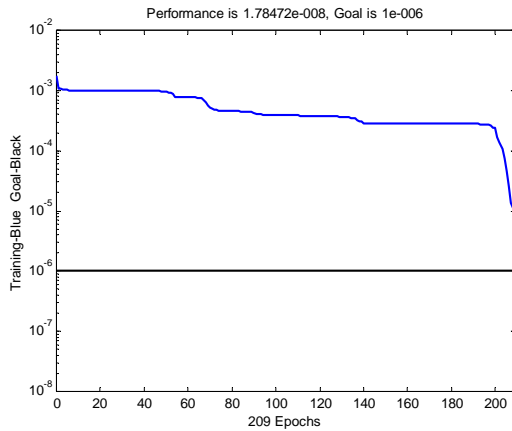
Gambar 10 Hasil pembelajaran dengan eliminasi x_4 pada NN_1 .

Kemudian, dengan setelah x_1 dihilangkan, NN_1 dilatih kembali dengan mengeliminasi x_3 atau x_4 . Nilai SSE dapat dilihat pada **Tabel 9**. Proses pembelajaran dilakukan dengan mengeliminasi variabel x_3 (**Gambar 11**), dan mengeliminasi variabel x_4 (**Gambar 12**).

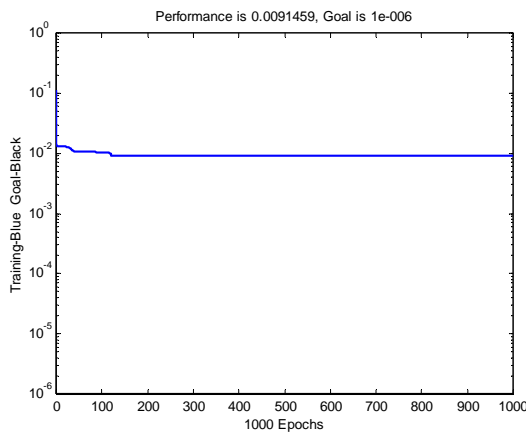
Tabel 9 Tabel SSE untuk NN_1 iterasi kedua.

	SSE
Variabel x_3 dieliminasi	9,9826
Variabel x_4 dieliminasi	8,0614

Karena nilai SSE dengan mengeliminasi variabel x_3 maupun variabel x_4 tidak ada yang lebih kecil daripada 6,4595 (menggunakan semua variabel, lihat **Tabel 9**), maka tidak ada variabel input yang dapat dihilangkan.



Gambar 11 Hasil pembelajaran dengan eliminasi x_3 pada NN_1 .



Gambar 12 Hasil pembelajaran dengan eliminasi x_4 pada NN_1 .

Sehingga format untuk aturan pertama R^1 adalah:

$$R_1: \text{IF } x = (x_1, x_3, x_4) \text{ THEN } y_1 = NN_1(x_3, x_4).$$

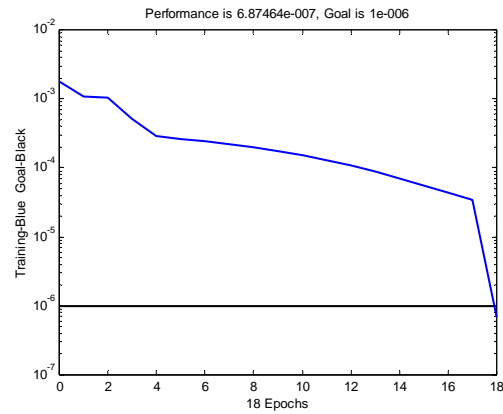
- Pengujian pada NN_2 :

Jaringan syaraf NN_2 dilatih dengan menggunakan jaringan backpropagation *levenberg marquardt* dengan 1 lapisan tersembunyi (dengan 5 neuron). Parameter-parameter: maksimum epoch = 1000; laju pembelajaran = 1; dan toleransi error = 10^{-6} . Selanjutnya, dengan melakukan beberapa eliminasi pada variabel input, dan dengan menggunakan persamaan 6 diperoleh SSE seperti terlihat pada Tabel 10. Proses pembelajaran dilakukan dengan semua input (Gambar 13), mengeliminasi variabel x_1 (Gambar 14), mengeliminasi variabel x_3 (Gambar 15), dan mengeliminasi variabel x_4 (Gambar 16).

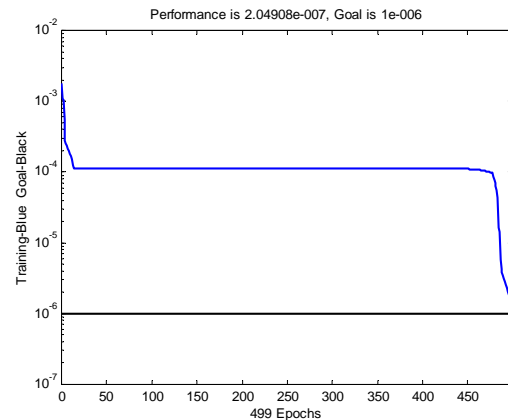
Tabel 10 Tabel SSE untuk NN_2 iterasi pertama.

	SSE
Semua variabel digunakan	3,2297
Variabel x_1 dieliminasi	3,1144
Variabel x_3 dieliminasi	3,3422
Variabel x_4 dieliminasi	5,8729

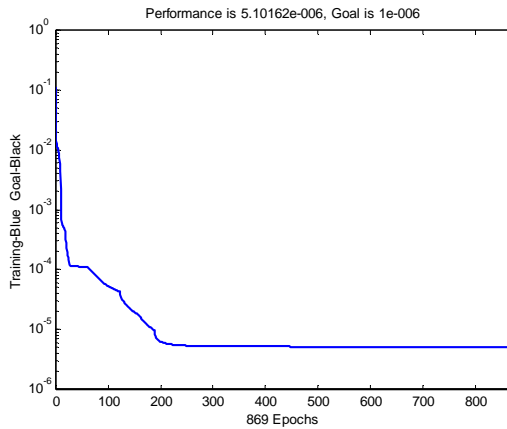
Karena nilai SSE dengan mengeliminasi variabel x_1 lebih kecil daripada SSE dengan menggunakan semua variabel, maka variabel x_1 dapat dihilangkan.



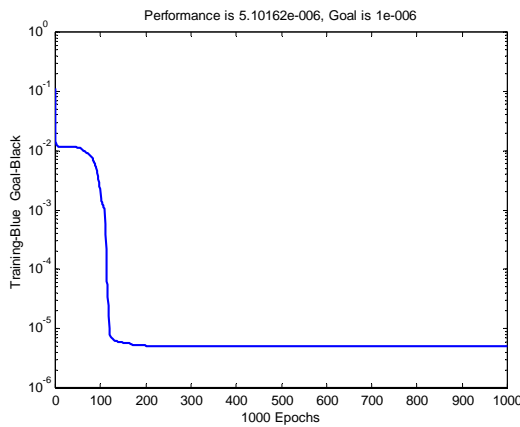
Gambar 13 Hasil pembelajaran dengan semua variabel pada NN_2 .



Gambar 14 Hasil pembelajaran dengan eliminasi x_1 pada NN_2 .



Gambar 15 Hasil pembelajaran dengan eliminasi x_3 pada NN_2 .



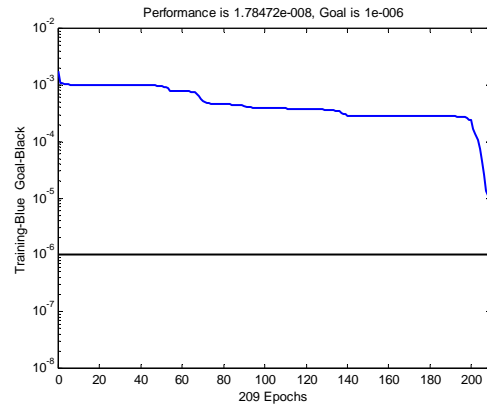
Gambar 16 Hasil pembelajaran dengan eliminasi x_4 pada NN_2 .

Kemudian, dengan setelah x_1 dihilangkan, NN_2 dilatih kembali dengan mengeliminasi x_3 atau x_4 . Nilai SSE dapat dilihat pada **Tabel 11**. Proses pembelajaran dilakukan dengan mengeliminasi variabel x_3 (**Gambar 17**), dan mengeliminasi variabel x_4 (**Gambar 18**).

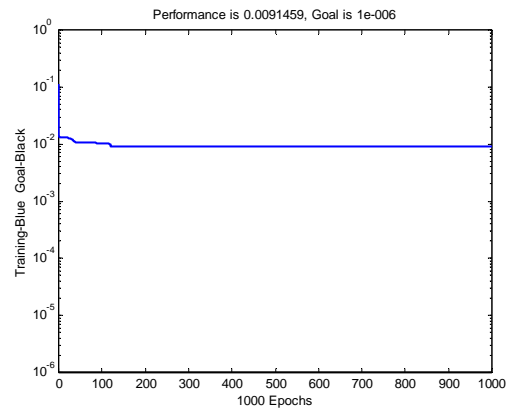
Tabel 11 Tabel SSE untuk NN_2 iterasi kedua.

	SSE
Variabel x_3 dieliminasi	4,9913
Variabel x_4 dieliminasi	4,0307

Karena nilai SSE dengan mengeliminasi variabel x_3 maupun variabel x_4 tidak ada yang lebih kecil daripada 3,2297 (menggunakan semua variabel, lihat **Tabel 11**), maka tidak ada variabel input yang dapat dihilangkan.



Gambar 17 Hasil pembelajaran dengan eliminasi x_3 pada NN_2 .



Gambar 18 Hasil pembelajaran dengan eliminasi x_4 pada NN_2 .

Sehingga format untuk aturan pertama R^2 adalah:

$$R_2: \text{IF } x = (x_1, x_3, x_4) \text{ THEN } y_2 = NN_2(x_3, x_4).$$

3.7 Pengujian

Terakhir, nilai y_1^* dapat diperoleh dengan menggunakan **persamaan 8**, dan hasilnya dapat dilihat pada **Tabel 12**.

Tabel 12 Hasil pelatihan akhir.

Data ke-	Input jaringan			Target output (y)	Output jaringan (y^*)
	x_1	x_3	x_4		
1	0,52	0,31	0,43	0,5144	0,5996
2	0,45	0,57	0,48	0,5588	0,6263
3	0,60	0,51	0,40	0,4894	0,6214
4	0,81	0,42	0,52	0,6219	0,6219
5	0,25	0,64	0,46	0,5217	0,7292
6	0,91	0,73	0,62	0,7269	0,7269
7	0,63	0,50	0,71	0,8011	0,8011
8	0,82	0,24	0,74	0,8429	0,8429
9	0,90	0,35	0,90	1,0068	1,0000
10	0,65	0,42	0,63	0,7225	0,7225
11	0,79	0,91	0,79	0,8901	0,8901

Data ke-	Input jaringan			Target output (y)	Output jaringan (y*)
	x ₁	x ₃	x ₄		
12	0,81	0,75	0,85	0,9514	0,9514
13	0,23	0,82	0,52	0,5795	0,9966
14	0,17	0,27	0,44	0,4936	0,6014
15	0,52	0,61	0,68	0,7637	0,7637
16	0,35	0,54	0,46	0,5310	0,6178
17	0,12	0,44	0,67	0,7164	0,7678
18	0,43	0,56	0,58	0,6573	0,6566
18	0,28	0,27	0,87	0,9351	1,0000
20	0,45	0,66	0,58	0,6587	0,6704
21	0,14	0,44	0,33	0,3795	0,6137
22	0,32	0,66	0,88	0,9480	0,9998
23	0,17	0,54	0,55	0,6030	0,6356
24	0,32	0,43	0,75	0,8184	0,8352
25	0,41	0,65	0,56	0,6356	0,6609
26	0,32	0,44	0,65	0,7184	0,7449
27	0,17	0,23	0,33	0,3839	0,5935
28	0,51	0,48	0,65	0,7332	0,7408
29	0,65	0,45	0,89	0,9824	1,0000
30	0,34	0,44	0,46	0,5303	0,6043

Jumlah kuadrat error (SSE) antara target output dengan output jaringan adalah 0,3825; dan koefisien korelasi sebesar 0,8461. Berdasarkan hasil tersebut dapat disimpulkan bahwa sistem inferensi fuzzy yang terbentuk berdasarkan pembelajaran jaringan syaraf memiliki validitas yang cukup tinggi.

4. SIMPULAN

Dari hasil penelitian, dapat disimpulkan bahwa:

1. jaringan syaraf dengan algoritma pembelajaran *backpropagation* dapat digunakan untuk membangun sistem inferensi fuzzy berdasarkan data-data yang diberikan;
2. tidak semua variabel yang diberikan berpengaruh dalam pembentukan sistem inferensi fuzzy;.
3. hasil penelitian menghasilkan 2 aturan fuzzy sebagai berikut:
 $R_1: \text{IF } x = (x_1, x_3, x_4) \text{ THEN } y_1 = NN_1(x_3, x_4).$
 $R_2: \text{IF } x = (x_1, x_3, x_4) \text{ THEN } y_2 = NN_2(x_3, x_4).$
4. SSE antara target output dengan output jaringan adalah 0,3825; dan koefisien korelasi sebesar 0,8461; yang berarti bahwa sistem inferensi fuzzy yang terbentuk berdasarkan pembelajaran jaringan syaraf memiliki validitas yang cukup tinggi.

PUSTAKA

- [1] Cox, Earl. 1994. *The Fuzzy Systems Handbook (A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems)*. Massachusetts: Academic Press, Inc
- [2] Fausett, Laurence. 1994. *Fundamentals of Neural Networks (Architectures, Algorithms, and Applications)*. New Jersey: Prentice-Hall.
- [3] Jang, J, S, R; Sun, C.T; dan Mizutani, E. 1997. *Neuro-Fuzzy and Soft Computing*. London: Prentice-Hall.

- [4] Kusumadewi, Sri; dan Hartati, Sri. 2006. *Neuro-Fuzzy: Integrasi Jaringan Syaraf & Sistem Fuzzy*. Graha Ilmu, Yogyakarta
- [5] Kusumadewi, Sri. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB dan ExcelLink*. Graha Ilmu, Yogyakarta
- [6] Lin, Chin-Teng; dan Lee, George. 1996. *Neural Fuzzy Systems*. London: Prentice-Hall.
- [7] Pal, Nikhil, P.; Pal, Kuhu; Keller, James, M.; dan Bezdek, James, C. 2005. *A Possibilistic Fuzzy C-Means Clustering Algorithm*. IEEE Transactions on Fuzzy Systems, Vol. 13 No. 4.
- [8] Ross, Timothy J. 2005. *Fuzzy Logic with Engineering Applications*. Edisi ke-2. John Wiley & Sons Inc. Inggris.
- [9] Yan, Jun; Ryan, Michael; dan Power, James. 1994. *Using Fuzzy Logic Towards Intelligent Systems*. Prentice Hall. New York.