

## Optimalisasi Jaringan Neuro-Fuzi dengan Algoritma Genetik Multiresolusi untuk Identifikasi Sistem Nonlinier

Gunawan Ariyanto  
Teknik Elektro UMS

Nurgiyatna  
Teknik Elektro UMS

Jl. A. Yani Tromol Pos I Kartasura Surakarta

Jl. A. Yani Tromol Pos I Kartasura Surakarta

### Abstrak

Pengetahuan tentang perilaku atau karakteristik dari sistem sangat penting dalam pengendalian sehingga diperlukan pemodelan melalui proses identifikasi sistem berdasarkan pada pengukuran input dan output sistem. Untuk sistem yang bersifat nonlinier, pemodelan kualitatif semisal dengan logika fuzi mulai disukai dan banyak digunakan. Sistem logika fuzi memiliki kekurangan utama berupa kebutuhan akan seorang ahli dan ketidakmampuannya untuk belajar. Dengan menggabungkan prinsip kerja jaringan syaraf tiruan kedalam sistem logika fuzi maka akan terbentuk sistem baru yang bersifat adaptif yaitu jaringan neuro-fuzi yang dapat mengatasi kekurangan di atas. Supaya kinerja jaringan neuro-fuzi menjadi lebih baik maka digunakan algoritma genetik multiresolusi sebagai algoritma optimalisasi. Hasil simulasi memperlihatkan algoritma genetik multiresolusi mampu memperbaiki kinerja jaringan neuro-fuzi untuk mengidentifikasi sistem nonlinier yang ditandai dengan perbaikan nilai mse pembelajaran dari nilai 0,0422 menjadi 0,00337.

**Kata kunci :** sistem logika fuzi, jaringan syaraf tiruan, algoritma genetik, identifikasi sistem

### I. Pendahuluan

Istilah *Soft Computing* sebagai kerangka pikir yang baru di dalam sistem kecerdasan buatan pertama kali dikemukakan oleh Zadeh. Sistem kecerdasan buatan berupaya membuat mesin yang dapat meniru kecerdasan manusia dan sistem biologis. Di dalam *Soft Computing* dikembangkan paradigma dengan bentuk logika baru semisal logika fuzi, jaringan syaraf tiruan dan algoritma genetik yang semuanya memiliki kelebihan berupa kemampuan untuk menangani sesuatu yang bersifat tidak presisi dan tidak pasti.

Sistem logika fuzi adalah nama bagi suatu sistem yang memiliki kaitan langsung dengan konsep-konsep fuzi (semisal himpunan fuzi, variable fuzi, dll) dan logika fuzi [Li Xin Wang, 1994]. Sedangkan logika fuzi adalah logika yang dikembangkan berdasar pada himpunan fuzi yang mampu memanipulasi berbagai konsep yang samar (tidak presisi dan tidak pasti). Himpunan fuzi merupakan perluasan dari himpunan tegas (*crisp set*). Pada himpunan fuzi, nilai keanggotaan suatu elemen tidak hanya terbatas pada 0 dan 1 saja, tetapi dapat berupa nilai bilangan nyata antara 0 dan 1.

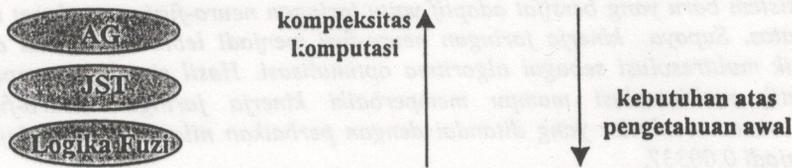
Sistem logika fuzi terdiri dari empat komponen utama yaitu unit fuzifikasi, basis pengetahuan, mesin inferensi fuzi, dan unit defuzifikasi [Yun Yang, 1994]. Kesulitan utama dalam perancangan sistem logika fuzi adalah ketika membuat basis pengetahuan karena harus dibuat berdasarkan pada pengalaman seorang ahli atau operator yang telah memahami sistem (*plant*) yang akan dimodelkan. Selain itu, sistem logika fuzi juga tidak memiliki kemampuan untuk belajar sehingga sulit untuk memperoleh unjuk kerja yang benar-benar optimal (terbaik).

Jaringan syaraf tiruan (JST) merupakan sistem pemroses informasi yang dibentuk dari generalisasi model matematis jaringan syaraf biologis manusia. Kelebihan utama dari JST adalah kemampuan untuk belajar dan sifat adaptif yang dimiliki. Suatu JST dapat dicirikan dengan tiga sifat yang melekat padanya yaitu : pola hubungan antar neuron (arsitektur), metode penentuan bobot-bobot jaringan (algoritma pembelajaran), dan fungsi aktivasi. Pada dasarnya, fungsi JST adalah untuk melakukan pemetaan nonlinier vektor-vektor masukan ke dalam vektor-vektor keluaran yang dikodekan melalui bobot-bobot jaringan, fungsi aktivasi dan arsitektur jaringan. Kemampuan lainnya yang dimiliki oleh JST adalah kemampuan untuk melakukan generalisasi.

Algoritma genetik (AG) adalah algoritma pencarian dan optimalisasi stokastik yang bekerja secara paralel dan berdasarkan pada kaidah seleksi alam dan genetika alami. Perancangan AG berturut-turut dimulai dengan pengkodean individu/kandidat solusi, pembuatan fungsi ketangguhan, penentuan metode seleksi, dan penentuan metode rekombinasi (persilangan dan

mutasi). Pengkodean individu/kandidat solusi harus disesuaikan dengan jenis permasalahan dan fungsi ketangguhan harus dapat memberikan penilaian obyektif terhadap setiap individu. Metode seleksi adalah metode untuk mempertahankan individu yang baik (berdasarkan nilai ketangguhan) yang akan dipersiapkan pada proses generasi berikutnya. Proses rekombinasi yang bertujuan untuk menghasilkan generasi baru berupa persilangan antar beberapa pasang individu dalam satu populasi dan mutasi, yaitu peristiwa berubahnya satu atau beberapa kode pada individu yang terkena mutasi. Proses sekuensial yang dimulai dari seleksi (berdasar evaluasi nilai ketangguhan), rekombinasi, evaluasi (menentukan nilai ketangguhan) dan berulang lagi ke seleksi akan berjalan terus menerus sampai kondisi terminasi (semisal jumlah generasi yang dibatasi) yang ditentukan telah terpenuhi. Dengan semakin bertambahnya jumlah generasi maka individu-individu yang dihasilkan diharapkan dapat berevolusi menjadi semakin baik. Cara pencarian stokastis evolutif yang sistematis seperti inilah yang menjadi ciri bagi AG.

Ditinjau dari segi kompleksitas komputasi dan segi kebutuhan akan pengetahuan awal (*a-priory knowledges*) maka paradigma logika fuzi, JST, dan AG memiliki hirarki/tingkatan seperti terlihat pada gambar 1 [Russo,1998].

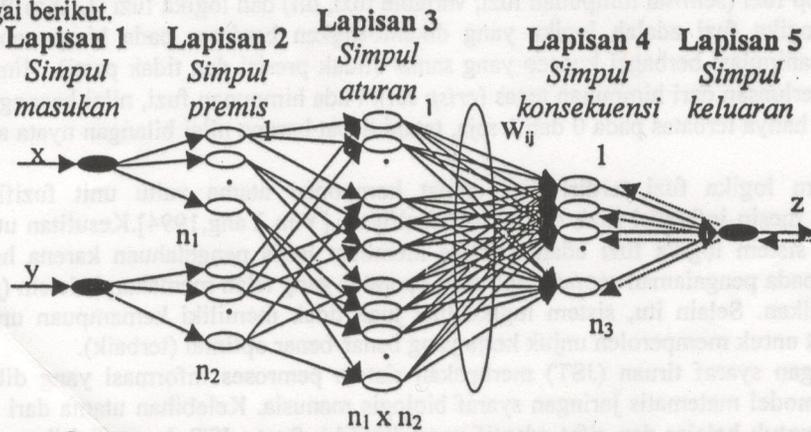


Gambar 1. Hirarki logika fuzi, JST dan AG [Russo,1998].

Makalah ini akan menerangkan pengembangan salah satu sistem hybrid fuzi-neural-genetik untuk tugas identifikasi sistem nonlinier. Penekanan makalah ini terletak pada pengembangan algoritma genetik agar memiliki kemampuan yang lebih handal sesuai dengan permasalahan yang dihadapi yaitu pengoptimalan pembelajaran jaringan neuro-fuzi.

II. Model Jaringan Neuro-Fuzi

Dalam makalah ini digunakan model jaringan neuro-fuzi seperti model yang digunakan oleh Lin [Lin C.T.,1991]. Jaringan ini menggunakan proses inferensi fuzi mamdani. Struktur jaringan dengan 2 masukan dan 1 keluaran digambarkan pada gambar 2. Keterangan tiap lapis adalah sebagai berikut.



Gambar 2. Jaringan *neuro-fuzi* dengan inferensi fuzi mamdani.

Notasi yang menjelaskan fungsi tiap simpul pada semua lapisan adalah sebagai berikut :

- $O_{k,i}$  : keluaran simpul ke-i pada lapisan k
  - $W_{k,i,j}$  : bobot hubungan dari simpul ke-i lapisan (k-1) ke simpul j pada lapisan k
  - $netf_{k,i}$  : melambangkan jumlah total masukan simpul ke-i pada lapisan k.
  - $m_{k,i}, \sigma_{k,i}$  : pusat dan varian fungsi gussian pada simpul ke-i lapisan k
- Proses yang terjadi pada setiap lapisan dapat dijelaskan sebagai berikut :

**Lapisan 1.** Setiap simpul pada lapisan ini melewati secara langsung sinyal lapisan ke lapisan berikutnya.

$$O_{1,1} = I_1 \quad \text{dan} \quad O_{1,2} = I_2$$

Semua bobot sambungan pada lapisan ini bernilai 1 ( $W_{1,i} = 1$ ).

**Lapisan 2.** Semua simpul pada lapisan ini berperan sebagai fungsi keanggotaan. Dalam penelitian digunakan fungsi keanggotaan gaussian, sehingga keluaran setiap simpul pada lapisan ini adalah sebagai berikut.

$$net_{2,i} = \begin{cases} O_{1,1} & \text{untuk } i = 1, 2, \dots, n_1 \\ O_{1,2} & \text{untuk } i = 1, 2, \dots, n_2 \end{cases}$$

$$O_{2,i} = \exp \frac{-(net_{2,i} - m_{2,i})^2}{(\sigma_{2,i})^2} \quad \text{dengan } i = 1, 2, \dots, n_1 + n_2$$

**Lapisan 3.** Setiap simpul pada lapisan ini berfungsi sebagai simpul aturan. Sambungan ke simpul ini melakukan proses evaluasi bagian premis aturan fuzzy. Apabila digunakan sistem inferensi minimum untuk menentukan kuat penyulutan setiap aturan, maka keluaran simpul pada lapisan ini adalah.

$$net_{3,i} = \min(O_{2,j}, O_{2,h}) \quad \text{dengan } i = n_1(j-1) + (k - n_2); \quad j = 1, 2, \dots, n_1; \quad h = n_1 + 1, \dots, n_1 + n_2$$

$$O_{3,i} = net_{3,i} \quad \text{dengan } i = 1, 2, \dots, n_1 * n_2$$

Bobot sambungan pada lapisan ini adalah 1 ( $W_{3,i} = 1$ ).

**Lapisan 4.** Setiap simpul pada lapisan ini mempunyai dua ragam operasi, yaitu ragam transmisi bawah-atas dan ragam transmisi atas-bawah. Pada transmisi bawah-atas, sambungan ke lapisan 4 melakukan operasi OR untuk menggabungkan kuat penyulutan dari aturan-aturan yang mempunyai konsekuensi yang sama.

$$net_{4,i} = \sum_{j=1}^{n_1 * n_2} (W_{4,ij} O_{3,i}) \quad \text{dengan } i = 1, 2, \dots, n_3$$

$$O_{4,1} = net_{4,1}$$

Bobot sambungan pada lapisan ini menyatakan hubungan aturan ke-j dengan peubah keluaran-i. Bobot pada lapisan ini hanya mempunyai dua nilai, yaitu 0 atau 1. Pada transmisi atas-bawah, sambungan dari lapisan 5 ke lapisan 4 ini mempunyai fungsi yang sama dengan sambungan pada lapisan 2.

**Lapisan 5.** Lapisan ini juga mempunyai dua ragam operasi. Ragam atas-bawah untuk memasukan data pelatihan ke dalam jaringan dengan keluaran simpul adalah  $O_i^5 = y_i$ , dengan  $y_i$  adalah data pelatihan. Pada ragam bawah-atas simpul melakukan transmisi untuk menghasilkan sinyal keluaran hasil inferensi. Simpul ini dapat dikatakan sebagai defuzzifier dengan metode pusat luas (*center of area*). Keluaran simpul pada lapisan ini dapat dinyatakan sebagai berikut :

$$O_{5,1} = \frac{net_{5,1}}{\sum_{i=1}^{n_3} \sigma_{4,i} O_{4,i}} = \frac{\sum_{i=1}^{n_3} m_{4,i} \sigma_{4,i} O_{4,i}}{\sum_{i=1}^{n_3} \sigma_{4,i} O_{4,i}}$$

### III. Optimalisasi Jaringan dengan Algoritma Genetik Multiresolusi

Ada tiga tahapan dalam proses pembelajaran jaringan neuro-fuzi ini yaitu berupa inialisasi nilai pusat dan varian fungsi keanggotaan, pembangkitan aturan, dan yang terakhir optimisasi fungsi keanggotaan. Masing-masing tahapan dijelaskan sebagai berikut :

**A. Tahap pertama : inialisasi nilai pusat dan varian fungsi keanggotaan**

Untuk menginisialisasi nilai pusat dan varian digunakan metode penskalaan linier. Nilai pusat dicari dengan cara mencari terlebih dahulu nilai maksimum dan minimum pasangan data pelatihan kemudian dibagi secara proporsional sebanyak jumlah fungsi keanggotaan yang

diinginkan sedangkan nilai varian dicari dengan membagi jarak antar dua nilai pusat dengan suatu bilangan yang lebih besar 1 (dalam penelitian ini digunakan nilai 1,8).

**B. Tahap kedua : membangkitkan aturan fuzzy**

Model jaringan *neuro-fuzzy* di atas membutuhkan basis aturan dalam proses inferensinya. Aturan diekstrak tidak dari informasi seorang ahli tetapi dari pasangan data masukan-keluaran dengan algoritma derajat kesesuaian maksimum (*maximum matching factor algorithm*). Algoritma ini dikemukakan pertama kali oleh Faraq [Faraq,1998]. Algoritma derajat kesesuaian maksimum (ADKM) terdiri dari beberapa langkah sebagai berikut :

**Langkah 1.** Untuk setiap simpul aturan pada lapisan 3 dibuat  $n_3$  buah derajat kesesuaian (*matching factor*) sehingga diperoleh  $n_1 \times n_2 \times n_3$  derajat kesesuaian. Derajat kesesuaian dilambangkan dengan :

$M_{ij}$  ; dengan  $i =$  indeks simpul aturan ( $i = 1, 2, \dots, n_1 \times n_2$ )  
 $j =$  indeks simpul keluaran ( $j = 1, 2, \dots, n_3$ )

**Langkah 2.**  $M_{ij}$  dapat dihitung berdasar dengan *pseudo-code* berikut :

untuk  $i=1, 2, \dots, n_1 \times n_2$

    untuk  $j=1, 2, \dots, n_3$

        untuk  $k=1, 2, \dots, N$  (jumlah data pelatihan)

$$M_{ij} = \begin{cases} M_{ij} + O_{3,ki} & \text{jika } O_{4,kj} \text{ adalah elemen maksimum dalam } O_{4,k} \\ M_{ij} & \text{jika } O_{4,kj} \text{ lainnya} \end{cases}$$

        selesai

    selesai

selesai

**Langkah 3.** Setelah menghitung semua  $M_{ij}$  pada langkah 2, konsekuen aturan dapat ditentukan dengan *pseudo-code* berikut ini :

untuk  $i = 1, 2, \dots, n_1 \times n_2$

-Cari derajat kesesuaian maksimum  $M_{max}$  dari himpunan  $M_i$  ( $[M_{ij} ; j = 1, 2, \dots, n_3]$ ).

-Temukan indeks simpul keluaran ( $j_{max}$ ) dari  $M_{max}$ .

-Hapus semua sambungan dari simpul aturan ke- $i$  kecuali sambungan yang menuju ke simpul keluaran yang berindeks  $j_{max}$ .

selesai

**C. Tahap ketiga : optimisasi dengan algoritma genetik multiresolusi**

Gagasan AGM pertamakali dikemukakan oleh Faraq [Faraq,1998]. AGM bekerja seperti algoritma genetik konvensional tetapi dengan penambahan kemampuan mengembang-kempiskan ruang pencarian (*search space*). AGM menggunakan kromosom/deret berupa integer-desimal untuk mengkodekan perbaikan parameter-parameter jaringan *neuro-fuzzy*. Setiap gen (allele) memiliki nilai integer antara 1-9 ([ 1 2 ... 8 9]). Jumlah allele ditentukan dari jumlah total parameter fungsi keanggotaan peubah masukan dan keluaran. Jadi, akan digunakan  $n_1 + n_2 + n_3 = n_4$  fungsi keanggotaan. Setiap fungsi keanggotaan (semisal dipilih gaussian) dihitung jumlah parameternya dan dikalikan  $n_4$  untuk memperoleh jumlah allele. Karena parameter pada fungsi gaussian ada 2 yaitu pusat ( $m$ ) dan varian ( $\sigma$ ), maka tiap deret terdiri dari  $2 \times n_4$  allele. Parameter AGM adalah jumlah populasi ( $pop$ ), peluang persilangan dan mutasi ( $P_p$  dan  $P_m$ ), offset pusat dan varian ( $\delta_m$  dan  $\delta_\sigma$ ), faktor penyusutan offset ( $T_{m\sigma}$ ), dan parameter resolusi ( $r$ ). Prosedur atau cara kerja selengkapnya dari AGM adalah sebagai berikut :

1. Memberikan nilai awal pusat ( $m_o$ ), dan varian ( $\sigma_o$ ) jaringan *neuro-fuzzy* ke dalam AGM yang dilambangkan dengan  $m_{io}$  dan  $\sigma_{io}$  ( $i = 1, 2, \dots, n_4$ ) dan menginisialisasi nilai parameter terbaik B ( $m_{iB}$  dan  $\sigma_{iB}$ ) dengan nilai parameter awal ini.
2. Nilai pusat dan lebar yang baru dihitung dari nilai allele, yaitu sebagai berikut:

$$m_i = m_{io} + (s_i - 5) \delta_m$$

$$\sigma_i = \sigma_{io} + (s_{(i+n_4)} - 5) \delta_\sigma$$

dengan  $m_i$  dan  $\sigma_i$  adalah nilai center dan varian yang baru,  $S_i$  adalah nilai allele ke- $i$  pada string,  $\delta_m$  dan  $\delta_\sigma$  adalah nilai offset dari pusat dan varian.

3. Berdasarkan nilai pusat dan varian yang baru, uji sekumpulan individu dalam populasi untuk menentukan nilai ketangguhan bagi masing-masing individu. Fungsi ketangguhan yang digunakan berupa nilai mse (*mean square error*). Cari nilai mse terbaik dalam populasi dan bandingkan dengan nilai terbaik sebelumnya yang tersimpan pada B ( $m_{iB}$  dan  $\sigma_{iB}$ ). Bila lebih baik maka perbaharui nilai B dan perbaharui struktur model jaringan *neuro-fuzzy* dengan parameter terbaik B ( $m_{i0} = m_{iB}$  dan  $\sigma_{i0} = \sigma_{iB}$ ) yang telah diperoleh (proses optimisasi masih berjalan).
4. Setiap r generasi nilai offset  $\delta_m$  dan  $\delta_\sigma$  dikoreksi berdasarkan fungsi penyusutan berikut ini :  

$$\delta_m = \delta_m * T_{m\sigma}; \text{ dengan } 0 < T_{m\sigma} < 1$$

$$\delta_\sigma = \delta_\sigma * T_{m\sigma}; \text{ dengan } 0 < T_{m\sigma} < 1$$
 Untuk mencegah terjebak pada nilai minimum lokal maka setiap n\*r generasi (dipilih n=10) nilai offset  $\delta_m$  dan  $\delta_\sigma$  dikembalikan ke nilai awal.
5. Lakukan proses rekombinasi dan ulangi lagi ke nomor 2 sampai kondisi terminasi terpenuhi. AGM akan berhenti bila jumlah generasi maksimum telah tercapai.

**IV. Simulasi**

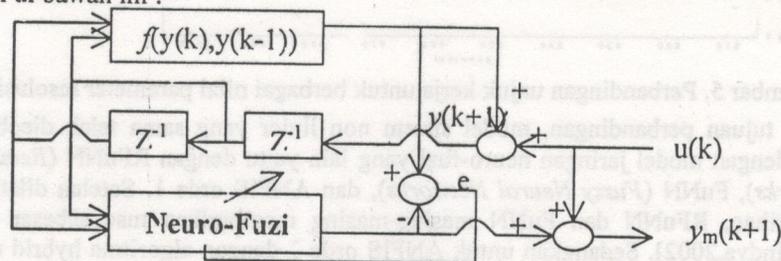
Untuk menguji kinerja AGM maka dibuat simulasi identifikasi sistem nonlinier dengan menggunakan perangkat lunak MATLAB versi 5.3. Sistem nonlinier yang akan diidentifikasi berupa sistem orde dua dengan persamaan matematis sebagai berikut :

$$y(k+1) = f(y(k), y(k-1)) + u(k) \tag{4.1}$$

dengan

$$f(y(k), y(k-1)) = \frac{y(k)(y(k-1) + 2)(y(k) + 2,5)}{8,5 + (y(k))^2 + (y(k-1))^2} \tag{4.2}$$

Bagian yang akan diidentifikasi adalah fungsi *f* yang merupakan persamaan matematis nonlinier. Proses identifikasi dilakukan dengan metode seri-paralel seperti terlihat dalam diagram blok proses identifikasi di bawah ini :



Gambar 3. Diagram blok proses identifikasi

Data pembelajaran berupa sinyal acak sebanyak 500 pasang data pelatihan dengan rentang nilai [-2 2] sedangkan variabel masukan yang dipilih adalah  $y(k)$  dan  $y(k-1)$ .

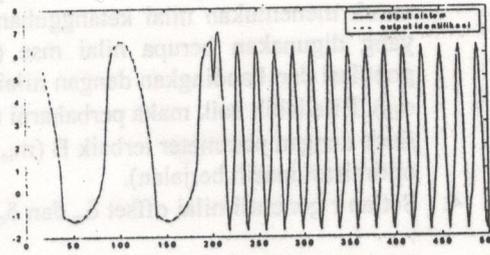
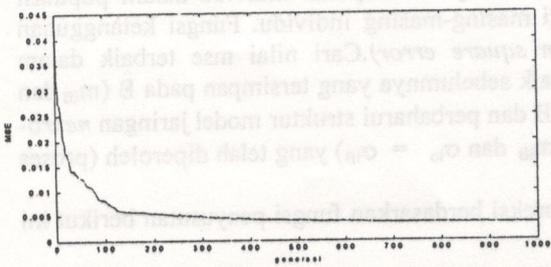
Tabel 1. Parameter jaringan neuro-fuzi dan AGM

Parameter	Nilai
Jumlah fungsi keanggotaan input	[7 7]
Jumlah fungsi keanggotaan output	7
Pop	20
Generasi	1000
$P_r$	0,6

Parameter	Nilai
$P_m$	0,1
$\delta_m$	0,01
$\delta_\sigma$	0,005
$T_m$	0,9
$T_r$	50

Dengan memberikan nilai parameter jaringan dan parameter AGM seperti dalam tabel 1, diperoleh hasil pembelajaran sebagai berikut : nilai mse (*mean square error*) pembelajaran setelah tahap kedua sebesar 0,0422 dan nilai mse setelah optimisasi dengan AGM (tahap ketiga) menjadi 0.00337. Setelah selesai tahap pembelajaran, kemudian dilanjutkan dengan menguji model. Pengujian dilakukan dengan memasukan sinyal uji  $u(k) = 1,2 \cos(2\pi k/100)$  untuk  $k \leq 200$ , dan  $u(k) = 1,5 \sin(2\pi k/20)$  untuk  $200 < k \leq 500$ . Nilai mse hasil pengujian adalah 0.0081. Hasil ini

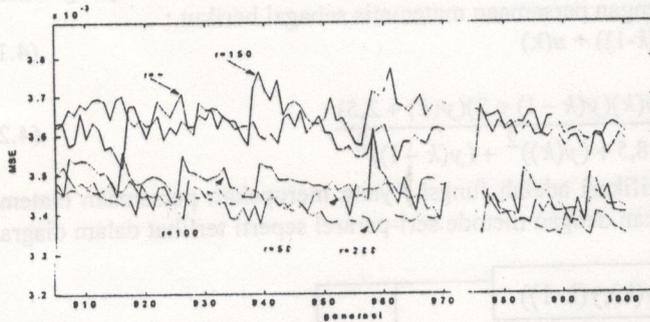
menunjukkan bahwa AGM mampu bekerja dengan baik dalam mengoptimalkan jaringan neuro-fuzi untuk identifikasi sistem nonlinier.



Gambar 4. (a) mse pembelajaran dengan AGM

(b) pengujian

Pengaruh parameter resolusi ( $r$ ) terhadap kinerja AGM dapat diamati dengan cara mencoba berbagai nilai parameter. Grafik mse pembelajaran untuk nilai  $r=50, r=100, r=150, r=200$  dan  $r=$  dapat dilihat dalam gambar 5. AGM dengan nilai  $r=$  berarti tidak ada proses perubahan resolusi. Dari gambar tersebut terlihat bahwa untuk  $r=50, r=100,$  dan  $r=20$  memiliki mse yang lebih baik dibanding  $r=$ , tetapi sebaliknya untuk  $r=150$  memiliki mse yang lebih besar. Hal ini menandakan bahwa algoritma genetik dengan perubahan resolusi (mengembang-kempiskan ruang pencarian) berpeluang besar memiliki kinerja yang lebih baik bila dibanding dengan AG tanpa perubahan resolusi. Sedangkan untuk kasus  $r=150$ , hal tersebut dapat terjadi dikarenakan AG bekerja secara stokastik.



Gambar 5. Perbandingan unjuk kerja untuk berbagai nilai parameter resolusi

Untuk tujuan perbandingan, model sistem non linier yang sama telah dicobakan untuk diidentifikasi dengan model jaringan neuro-fuzi yang lain yaitu dengan RFuNN (*Recurrent Fuzzy Neural Networks*), FuNN (*Fuzzy Neural Networks*), dan ANFIS orde 1. Setelah dilatih sebanyak 300 kali pelatihan, RFuNN dan FuNN masing-masing memberikan mse sebesar 0,0346 dan 0.06376 [Trenindya,2002]. Sedangkan untuk ANFIS orde 1 dengan algoritma hybrid memberikan mse sebesar  $8,9 \times 10^{-5}$ . Perbandingan kinerja AGM dibandingkan dengan model jaringan neuro-fuzi yang lain selengkapnya dapat dilihat dalam tabel 4.

Tabel 2. Perbandingan AGM dengan algoritma lain

Jaringan Neuro-fuzi	Epoch	Generasi	Mse
ANFIS backpropagation	300	-	1,7658
ANFIS hybrid	300	-	$8,9 \times 10^{-5}$
RFuNN	300	-	0,0346
FuNN	300	-	0,0637
AGM	-	300	0,00475

### V. Kesimpulan

Berdasarkan simulasi dapat disimpulkan bahwa AGM dapat bekerja dengan baik untuk permasalahan optimisasi dengan jumlah variabel yang besar. Hal ini ditandai dengan perbaikan nilai mse pembelajaran yang signifikan pada jaringan neuro-fuzi ketika AGM diterapkan dari 0.0422 menjadi 0.00337. Penerapan ruang pencarian secara dinamis (multiresolusi) pada algoritma genetik terbukti dapat meningkatkan peluang memperoleh solusi yang lebih baik.

DAFTAR PUSTAKA

- Farag W, Quintana H, "A Genetik-Based Neuro-fuzzy Approach for Modeling and Control of Dynamical Systems," *IEEE Trans. Neural Networks.*, vol. 9, pp. 756-766, September 1998
- Gunawan, T., *Identifikasi Sistem Nonlinier Menggunakan Recurrent Fuzzy Neural Network*, Skripsi S-1 Teknik Elektro UGM, Yogyakarta, 2002
- Lin, C.T., "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, pp. 1320-1336, Dec. 1991.
- Russo, M., "FuGeNeSys - A Fuzzy genetik neural Sistem for Fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 373-387, August 1998.
- Wang, L. X., *Adaptive Fuzzy Systems and Control*, Prentice-Hall, New Jersey, 1994.
- Yan, J., *Using Fuzzy Logic*, Prentice-Hall, New York, 1994.