

SISTEM TUTORIAL CERDAS BERBASIS FUZZY GENETIK

**Mewati Ayub¹, The Houw Liong²,
Inggriani Liem³, Achmad Hinduan⁴**

¹Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Indonesia,
Bandung 40116

e-mail : mewatia@yahoo.com

²Departemen Fisika - FMIPA, Institut Teknologi Bandung

³Departemen Teknik Informatika - FTI, Institut Teknologi Bandung

⁴FPMIPA/PPs Universitas Pendidikan Indonesia

Intisari

Kinerja sistem kendali fuzzy bergantung pada fungsi keanggotaan dan aturan-aturan kendali fuzzy yang digunakan. Pengaturan parameter sistem untuk memperoleh hasil yang optimal sangat diperlukan. Dalam penelitian ini, algoritma genetik akan digunakan untuk mengatur (tuning) parameter fungsi keanggotaan variabel input dalam sistem kendali fuzzy. Tujuannya adalah untuk memperoleh parameter fungsi keanggotaan sistem fuzzy sehingga sistem kendali fuzzy dapat memberikan hasil yang optimal. Adapun sistem kendali fuzzy yang akan diatur kembali parameternya adalah sistem yang menjadi bagian dari sebuah sistem tutorial cerdas, yang telah dikembangkan dalam penelitian sebelumnya. Hasil penelitian menunjukkan algoritma genetik dapat melakukan tuning terhadap parameter sistem kendali fuzzy sehingga diperoleh hasil yang lebih optimal.

Kata kunci: algoritma genetik, sistem kendali fuzzy

Abstract

Fuzzy controlled system performance depends on its membership functions and fuzzy rules. Therefore, system parameter adjustment is required to obtain optimal results. In this research, genetic algorithm is used for tuning membership function parameters of input variable in a fuzzy controlled system that has been used in previous research as a part of an intelligent tutoring system. The goal of our research is to determine membership function parameters in order to obtain optimal results. Experiment shows that genetic algorithm is able to perform membership functions parameter tuning to give better performance.

Key words: genetic algorithm, fuzzy control system

Diterima : 19 April 2005

Disetujui untuk dipublikasikan : 18 Mei 2005

1. Pendahuluan

Pada [1,2] telah dipaparkan hasil penelitian mengenai penerapan kendali *fuzzy* dalam Sistem Tutorial Cerdas untuk membantu sistem dalam menentukan pemberian layanan pengajaran kepada siswa sehingga siswa dapat mencapai tingkat pengetahuan yang sesuai dengan kemampuannya.

Proses belajar dengan lebih memperhatikan kebutuhan dan kemampuan siswa didukung oleh model pembelajaran yang dikembangkan dalam *mastery learning* yang diusulkan oleh Carroll [3]. Jika sekelompok siswa yang kemampuannya terdistribusi secara normal untuk suatu pelajaran diberikan bahan pelajaran yang sama dengan cara pembelajaran yang sama dan dalam jangka waktu yang sama panjang, maka pencapaian yang diperoleh adalah terdistribusi secara normal juga. Sebaliknya, jika sekelompok siswa yang kemampuannya terdistribusi secara normal untuk suatu pelajaran diberikan bahan pelajaran yang sama namun dengan cara pembelajaran dan jangka waktu yang disesuaikan dengan karakteristik dan kebutuhan setiap siswa, maka sebagian besar siswa akan mencapai penguasaan pelajaran tersebut. Dengan demikian sebagian besar siswa dapat menguasai pelajaran yang diberikan jika disediakan waktu belajar yang cukup dan bantuan yang sesuai. Beberapa siswa akan membutuhkan lebih banyak usaha, waktu dan bantuan untuk mencapai penguasaan pelajaran.

Algoritma Genetik merupakan teknik optimisasi yang meniru adaptasi genetik dalam seleksi alam untuk menemukan solusi yang optimal dari suatu permasalahan. Kemampuan pencarian dan kemampuan untuk menggabungkan pengetahuan a priori telah memperluas penggunaan algoritma genetik dalam pengembangan berbagai cara untuk memperbaiki perancangan suatu sistem

fuzzy. Sistem yang menerapkan pendekatan perancangan tersebut disebut dengan Sistem *Fuzzy-Genetik* (SFG) [4,5,6].

Kinerja sistem kendali *fuzzy* sangat bergantung pada keahlian dan pengalaman seseorang secara empirik dalam menentukan aturan-aturan dan fungsi keanggotaannya. Algoritma genetik dapat digunakan untuk melakukan *tuning* terhadap parameter sistem kendali *fuzzy* sehingga dapat diperoleh sistem kendali *fuzzy* yang lebih optimal. Algoritma genetik berperan di dalam pengubah posisi dan lebar fungsi keanggotaan, juga mampu mengoptimasikan aturan-aturan dalam suatu sistem kendali *fuzzy* [4,5,6].

Pada makalah ini, akan dibahas hasil penelitian mengenai sistem *fuzzy* genetik yang digunakan untuk membantu sistem tutorial cerdas di dalam memilih materi pelajaran yang sesuai dengan kemampuan siswa. Dalam hal ini, SFG digunakan untuk mengatur (*tuning*) parameter fungsi keanggotaan variabel input dalam sistem kendali *fuzzy* yang diterapkan dalam sistem tutorial cerdas. Tujuannya adalah untuk memperoleh parameter sistem *fuzzy* dalam menentukan layanan pembelajaran sehingga sistem kendali *fuzzy* dapat memberikan hasil pembelajaran yang lebih optimal. Untuk itu dilakukan penelitian terhadap parameter algoritma genetik yang dapat memberikan nilai fitness optimal. Adapun sistem kendali *fuzzy* yang akan diatur kembali parameternya adalah sistem yang dipakai dalam sistem tutorial cerdas OFCL (*Object oriented Fuzzy Control based Learning*) yang telah dikembangkan dalam penelitian sebelumnya [1,2].

2. Sistem Fuzzy Genetik

Algoritma Genetik adalah algoritma pencarian (*search algorithm*) yang menggunakan prinsip seleksi alam

dalam ilmu genetika untuk mengembangkan solusi terhadap suatu permasalahan [5]. Ide dasar algoritma genetik adalah mengelola suatu populasi individu yang merepresentasikan kandidat solusi untuk suatu permasalahan. Kumpulan kandidat solusi tersebut kemudian dikembangkan melalui suatu proses kompetisi. Kromosom merupakan representasi suatu individu dalam suatu populasi yang menggambarkan suatu solusi masalah. Setiap kromosom di dalam populasi mempunyai suatu nilai *fitness* yang menentukan kromosom mana yang akan dipilih untuk membentuk generasi baru dalam proses kompetisi. Nilai *fitness* adalah nilai yang menunjukkan derajat ketangguhan kromosom dalam beradaptasi terhadap masalah. Selama iterasi yang suksesif, yang disebut dengan generasi, kromosom di dalam populasi dievaluasi berdasarkan kemampuan adaptasinya. Berdasarkan hasil evaluasi ini, populasi kromosom yang baru dibentuk menggunakan mekanisme seleksi dan operator genetik. Dengan demikian populasi tersebut seolah-olah menjalani evolusi seperti yang terjadi pada seleksi alam. Sistem *Fuzzy-Genetik* (SFG) [4,5,6] adalah gabungan dari sistem kendali *fuzzy* dengan algoritma genetik untuk melakukan pembangkitan atau modifikasi basis pengetahuan atau basis data dari sistem kendali *fuzzy*. Basis pengetahuan yang dimaksud mencakup basis aturan dan basis data dari suatu sistem kendali *fuzzy*. Basis data meliputi fungsi penskalaan dan fungsi keanggotaan yang didefinisikan untuk sistem kendali *fuzzy*. Basis aturan adalah kumpulan aturan yang digunakan untuk inferensi dalam sistem kendali *fuzzy*. Pembangkitan atau modifikasi basis pengetahuan atau basis data dilakukan melalui proses *tuning* dan atau *learning*. Proses *tuning* merupakan proses untuk mengatur kembali parameter dari sistem kendali *fuzzy*, yaitu modifikasi fungsi

penskalaan dan atau fungsi keanggotaan. Proses *learning* adalah proses untuk memodifikasi basis aturan dari sistem kendali *fuzzy* [4,5,6].

Proses *tuning* secara genetik terhadap basis data sistem kendali *fuzzy* dapat dibedakan atas *approximate genetic tuning* dan *descriptive genetic tuning*. Pada *approximate genetic tuning* kromosom akan mengkodekan basis pengetahuan secara lengkap (baik premis maupun konklusi dari aturan). Sedangkan pada *descriptive genetic tuning* kromosom hanya mengkodekan definisi basis data (bagian premis aturan) berdasarkan partisi domain *fuzzy* dari variabel input.

Pada proses *learning* secara genetik terhadap basis aturan dapat dilakukan tiga macam pendekatan belajar, yaitu pendekatan Michigan, pendekatan Pittsburgh dan pendekatan *Iterative Rule Learning* (IRL). Kromosom pada pendekatan Michigan akan merepresentasikan suatu aturan sebagai suatu individu, sehingga populasi dinyatakan sebagai kumpulan aturan. Pendekatan ini digunakan dalam proses yang bersifat *on-line* dan dikenal sebagai sistem klasifikasi (*classifier system*). Pada pendekatan Pittsburgh, kromosom akan merepresentasikan seluruh basis aturan sebagai suatu individu, sehingga populasi dinyatakan sebagai kumpulan basis aturan. Dalam hal ini operasi genetik dilakukan untuk menghasilkan kombinasi aturan yang baru. Pendekatan ini digunakan dalam proses yang bersifat *off-line*. Pada pendekatan IRL, kromosom akan merepresentasikan suatu aturan sebagai suatu individu, sehingga populasi dinyatakan sebagai kumpulan aturan. Pembentukan basis aturan dilakukan dengan memilih hanya individu terbaik pada setiap iterasi dan membuang kumpulan individu sisanya sehingga menghasilkan solusi parsial.

Pendekatan ini digunakan dalam proses yang bersifat *off-line*.

Tujuan dari proses *tuning* aturan kendali dalam sistem kendali *fuzzy* adalah untuk membuat sistem inferensi *fuzzy* memberikan hasil yang optimal dengan cara mengatur kembali parameter fungsi keanggotaan yang dipakai dalam sistem. Untuk itu diperlukan sekumpulan data *training*, yaitu pasangan data input dan output yang dihasilkan dari sistem

kendali *fuzzy*. Dalam hal ini data input berhubungan dengan nilai variabel input dan data output dianggap sebagai nilai output yang diharapkan dapat dihasilkan oleh sistem. Proses *tuning* akan memodifikasi parameter himpunan *fuzzy* dari variabel input sehingga sistem dapat memberikan output yang mendekati data *training* yang diberikan. Pendekatan yang dipakai dalam penelitian ini adalah pendekatan Pittsburgh.

```

Program Algoritma Genetik
begin
    t = 0;
    Initalisasi P(t); {P(t):populasi kromosom pada generasi ke
t}
    Evaluasi P(t);
    while (not kondisi-terminasi) do
    begin
        t = t + 1;
        seleksi P(t) dari P(t-1);
        rekombinasi P(t);
        evaluasi P(t);
    end;
end;

```

Algoritma 1 : Struktur Algoritma Genetik

Kromosom yang merupakan suatu basis aturan dari sistem *fuzzy* direpresentasikan sebagai Cr. Setiap aturan dalam sistem akan direpresentasikan sebagai kromosom Cri. Dengan demikian suatu basis aturan dalam sistem *fuzzy* dengan m aturan akan direpresentasikan sebagai $C_r = C_{r1} C_{r2} \dots C_{rm}$.

Suatu aturan yang berbentuk Ri : IF x_1 is A_{i1} and and x_n is A_{in} THEN y is B_i dengan fungsi keanggotaan untuk pemetaan data input dan output menggunakan fungsi trapezoidal, akan dikode menjadi kromosom [4,5,6]: $C_{ri} = (a_{i1}, b_{i1}, c_{i1}, d_{i1} \dots, a_{in}, b_{in}, c_{in}, d_{in}, a_i, b_i, c_i, d_i)$

Dalam hal ini $(a_{ij}, b_{ij}, c_{ij}, d_{ij})$ adalah parameter fungsi trapezoidal untuk variabel input ke j pada aturan ke i. Dengan demikian basis aturan akan direpresentasikan sebagai vektor

kromosom yang dikodekan sebagai vektor bilangan real. Populasi dari R basis aturan dinyatakan sebagai : $C = (C_1 \dots C_R)$

Populasi awal dibuat dari basis aturan mula-mula yang diberikan oleh pakar. Kumpulan basis aturan awal ini merupakan kromosom yang dinyatakan sebagai C_1 . Untuk keperluan proses *tuning*, kita definisikan untuk setiap gene c_h dari C_1 , $h = 1 \dots H$, $H = (n+1) \times 4$, suatu interval $[c_{h1}, c_{hr}]$, yang merupakan interval dari pengaturan c_h tersebut, $c_h \in [c_{h1}, c_{hr}]$. Jika $(t \bmod 4) = 1$ maka c_t adalah nilai terkirri dari titik yang membentuk fungsi trapezoidal untuk himpunan *fuzzy*.

Selanjutnya populasi awal untuk $(C_2 \dots C_R)$ diisi dengan basis aturan yang dibangkitkan secara acak dari interval

himpunan *fuzzy* dari variabel input seperti terlihat pada Gambar 1. Dengan demikian interval yang didefinisikan dalam sebagai berikut [6] :

$$\begin{aligned}
 c_i &\in [c_{il}, c_{ir}] = \left[c_i - \frac{c_{i+1} - c_i}{2}, c_i + \frac{c_{i+1} - c_i}{2} \right] \\
 c_{i+1} &\in [c_{i+1l}, c_{i+1r}] = \left[c_{i+1} - \frac{c_{i+2} - c_{i+1}}{2}, c_{i+1} + \frac{c_{i+2} - c_{i+1}}{2} \right] \\
 c_{i+2} &\in [c_{i+2l}, c_{i+2r}] = \left[c_{i+2} - \frac{c_{i+3} - c_{i+2}}{2}, c_{i+2} + \frac{c_{i+3} - c_{i+2}}{2} \right] \\
 c_{i+3} &\in [c_{i+3l}, c_{i+3r}] = \left[c_{i+3} - \frac{c_{i+4} - c_{i+3}}{2}, c_{i+3} + \frac{c_{i+4} - c_{i+3}}{2} \right]
 \end{aligned}
 \tag{1}$$

Jika terdapat sekumpulan data input-output *training* yang berukuran K, yaitu $\{(x_i, y_i) = (x_{i1}, \dots, x_{in}, y_i), i = 1, \dots, K\}$ maka kesalahan inferensi dari suatu sistem berbasis *fuzzy* S, dengan kumpulan basis aturan BR dan kumpulan data input-output TRD dihitung sebagai jumlah dari kesalahan kuadrat, yaitu [6]:

$$E(S, BR, TRD) = \frac{1}{2K} \sum_{i=1}^K (y_i - S(x_i))^2
 \tag{2}$$

Fungsi *fitness* digunakan untuk mengukur ketangguhan setiap kromosom di dalam suatu populasi. Fungsi *fitness* dari algoritma genetik untuk kromosom C yang merepresentasikan kumpulan basis aturan BR, dan data *training* TRD didefinisikan sebagai berikut :

$$F(C_j) = E(S, BR, TRD) = \frac{1}{2K} \sum_{i=1}^K (y_i - S(x_i))^2
 \tag{3}$$

Evaluasi nilai *fitness* dilakukan dengan tujuan untuk meminimumkan fungsi *fitness*.

Operator genetik yang dapat digunakan adalah *crossover* dan mutasi. Operator *crossover* yang digunakan adalah *max-min arithmetical crossover*. Untuk

operator mutasi digunakan mutasi *non uniform*.

Operator *max-min arithmetical crossover* bekerja sebagai berikut :

Jika

$$C_v^t = (c_1, \dots, c_k, \dots, c_H)$$

$$C_w^t = (c_1', \dots, c_k', \dots, c_H')$$

dikawin silang, maka turunan yang dihasilkan adalah [4,6]:

$$\begin{aligned}
 C_1^{t+1} &= aC_w^t + (1-a)C_v^t \\
 C_2^{t+1} &= aC_v^t + (1-a)C_w^t \\
 C_3^{t+1}, c_{3k}^{t+1} &= \min(c_k, c_k') \\
 C_4^{t+1}, c_{4k}^{t+1} &= \max(c_k, c_k')
 \end{aligned}
 \tag{4}$$

dengan a adalah konstanta atau variabel yang nilainya bergantung pada usia populasi. Turunan yang dipilih adalah dua turunan dengan nilai *fitness* terbaik.

Operasi mutasi *non uniform* dilakukan sebagai berikut :

Jika

$$C_v^t = (c_1, \dots, c_k, \dots, c_H)$$

adalah kromosom suatu aturan dan elemen c_k dipilih untuk dimutasi (domain c_k adalah $[c_{kl}, c_{kr}]$), maka hasilnya adalah vektor [4,6] :

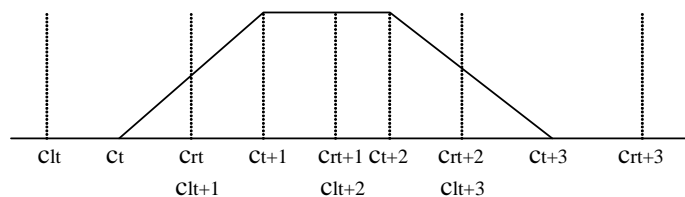
$$\begin{aligned}
 C_v^{t+1} &= (c_1, \dots, c_k', \dots, c_H) \\
 c_k' &= \begin{cases} c_k + \Delta(t, c_{kr} - c_k)[\text{rumus} - 1] \\ c_k - \Delta(t, c_k - c_{kl})[\text{rumus} - 2] \end{cases}
 \end{aligned}
 \tag{5}$$

dengan $k \in 1, \dots, H$. Jika digit acak = 0 digunakan rumus pertama, jika digit acak = 1 digunakan rumus kedua. Fungsi $\Delta(t,y)$ mengembalikan nilai dalam interval $[0,y]$ sehingga probabilitas dari $\Delta(t,y)$ mendekati 0 meningkat seiring dengan meningkatnya t.

3. Aplikasi Sistem Fuzzy Genetik dalam Sistem Tutorial Cerdas

OFCL yang telah dikembangkan dalam penelitian sebelumnya [1,2] memakai konsep Sistem Tutorial Cerdas (STC) yaitu suatu sistem pengajaran berbasis komputer yang berusaha meniru peran dari seorang tutor (guru) manusia [7]. Untuk mewujudkan STC diperlukan

representasi dari domain pengetahuan (*Domain Knowledge*), strategi pengajaran (*Pedagogy Module*) dan status belajar siswa (*Student Model*). Melalui interaksi antara ketiga modul tersebut, STC mengevaluasi tingkat pengetahuan siswa, membuat keputusan mengenai layanan pembelajaran yang akan diberikan kepada siswa sesuai dengan kemajuannya dalam proses belajar.



Gambar 1 : Interval Fungsi Keanggotaan untuk Pembangkitan Basis Aturan Secara Acak

Tujuan dari OFCL adalah untuk menyediakan pengalaman belajar bagi setiap siswa yang mendekati standar pengajaran yang akan diperoleh siswa dalam pengajaran seorang guru seorang siswa dengan guru yang cukup berpengalaman. Guru yang cukup berpengalaman akan dapat memberikan layanan pembelajaran yang berbeda untuk kelompok siswa dengan kemampuan yang berbeda.

Dalam hal ini SFG akan membantu OFCL untuk dapat memilih materi pelajaran yang sesuai dengan kebutuhan dan kemampuan siswa. Hal ini dilakukan oleh SFG dengan mengatur lebar dan posisi fungsi keanggotaan untuk variabel yang berkaitan. Untuk itu sistem fuzzy genetik memerlukan data historis dari hasil pembelajaran sebelumnya.

SFG diterapkan pada modul pedagogi. Setelah siswa berinteraksi dengan OFCL selama beberapa saat, apabila kemajuan siswa kurang berarti, SFG akan mempelajari data sebelumnya untuk mengatur kembali parameter sistem fuzzy

sehingga proses pembelajaran siswa dapat dilayani dengan lebih optimal.

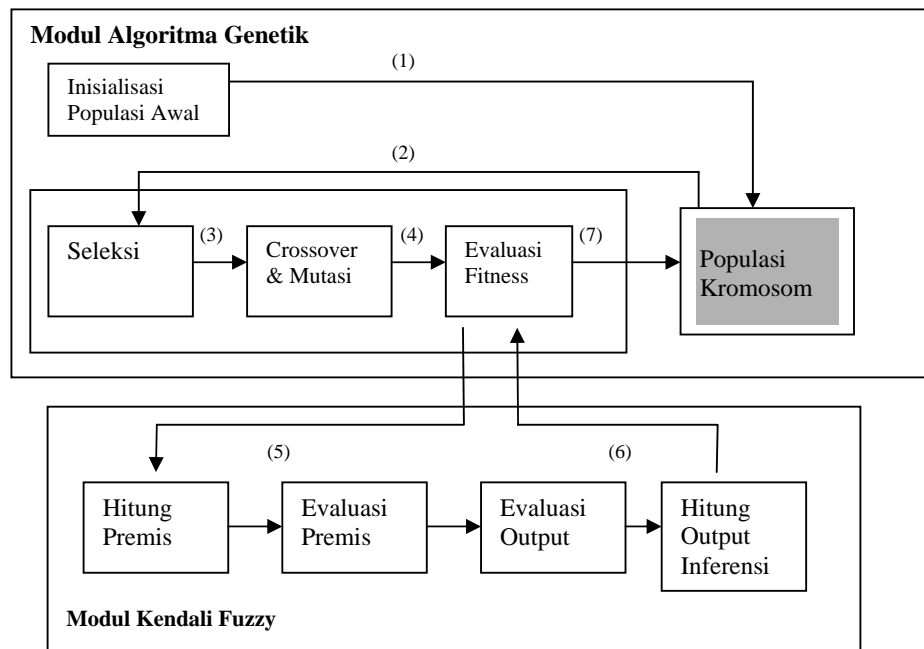
Aplikasi algoritma genetik untuk melakukan *tuning* sistem fuzzy diimplementasikan sebagai suatu modul tambahan dalam OFCL. Hubungan antara modul algoritma genetik dan modul inferensi fuzzy dapat ditunjukkan pada Gambar 2.

Modul algoritma genetik terdiri dari modul inialisasi populasi awal, modul seleksi, modul *crossover* dan mutasi, dan modul evaluasi *fitness*. Modul inialisasi populasi awal akan membangkitkan populasi kromosom awal secara acak dari aturan sistem fuzzy yang mula-mula dipakai berdasarkan interval nilai yang ditetapkan sesuai rumus (1) yang telah dijelaskan pada bagian 2. Selanjutnya modul seleksi akan memilih dua kromosom dari populasi tersebut. Kromosom yang terpilih akan direproduksi melalui modul *crossover* dan mutasi sehingga dihasilkan kromosom baru dengan rumus (4) dan (5). Modul evaluasi *fitness* akan menghitung nilai *fitness* dan mengevaluasi nilai *fitness* dari kromosom

baru dengan rumus (3). Kromosom dengan nilai *fitness* terbaik akan dipilih untuk membentuk populasi kromosom yang baru.

Pada saat perhitungan nilai *fitness*, modul algoritma genetik akan berhubungan dengan modul kendali *fuzzy*. Kromosom yang akan dihitung nilai *fitness*nya akan diencode menjadi aturan *fuzzy* yang siap

untuk digunakan dalam perhitungan nilai output dari proses inferensi. Modul kendali *fuzzy* akan memberikan nilai output dari sekelompok data hasil belajar berdasarkan aturan *fuzzy* yang diperoleh dari kromosom. Nilai output tersebut akan digunakan oleh modul evaluasi *fitness* untuk menghitung nilai *fitness* dari suatu kromosom.



Gambar 2 : Hubungan Modul Algoritma Genetik dengan Modul Kendali *Fuzzy*

Representasi kromosom menggunakan representasi non biner, yaitu menggunakan bilangan real. Kromosom yang merepresentasikan aturan *fuzzy* dalam algoritma genetik disimpan dalam suatu struktur tabel bilangan real. Sekumpulan aturan disimpan sebagai list dari kromosom. Populasi kumpulan aturan akan disimpan sebagai list dari alamat kumpulan aturan.

4. Eksperimen

Tujuan percobaan dalam penelitian ini adalah untuk melakukan *tuning* / modifikasi terhadap parameter fungsi keanggotaan dari variabel input dalam

suatu sistem *fuzzy* dalam OFCL dengan algoritma genetik. Adapun tujuan yang lebih spesifik adalah untuk meminimumkan fungsi *fitness* seperti pada rumus (3). Penelitian ini masih berupa simulasi, belum melibatkan interaksi langsung dengan siswa. Percobaan yang dilakukan menggunakan sistem *fuzzy* yang sudah diujicobakan pada penelitian terdahulu dalam perangkat lunak OFCL. Sistem *fuzzy* yang dimaksud telah diuraikan dalam [1,2]. Adapun variabel input yang dimaksud adalah nilai test dan tingkat kesulitan soal test. Nilai test memakai

fungsi trapezoidal, sedangkan tingkat kesulitan test memakai fungsi triangular.

Beberapa penelitian berbasis Sistem *Fuzzy* Genetik [4,6] menggunakan probabilitas *crossover* 0.6 dan probabilitas mutasi 0.2 sebagai parameter algoritma genetik. Sumber lain [8] menyebutkan probabilitas *crossover* 0.6 sebagai parameter yang dapat digunakan untuk memberikan hasil yang cukup optimal, sedangkan probabilitas mutasi diusahakan kecil. Untuk mengetahui besaran probabilitas *crossover* dan probabilitas mutasi yang dapat memberikan hasil yang optimal dilakukan sejumlah percobaan terhadap data hasil belajar dari eksperimen OFCL sebelumnya [2]. Data hasil belajar tersebut digunakan sebagai data *training* dalam algoritma genetik, terdiri dari data

hasil belajar dari eksperimen dalam kuliah Struktur Data di program studi sarjana Teknik Informatika STTI dan data hasil belajar dari eksperimen dalam kuliah Struktur Data di program studi sarjana Teknik Informatika ITB.

Untuk setiap kelompok data di atas, dilakukan 5 (lima) percobaan dengan probabilitas *crossover* 0.6 dan probabilitas mutasi berubah-ubah. Semua percobaan dilakukan dengan populasi berukuran 50 untuk 100 generasi. Hasil algoritma genetik untuk setiap percobaan dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Hasil Percobaan-1 sampai dengan Percobaan-5 untuk data hasil eksperimen di STTI dapat dilihat pada Gambar 3.

Nilai *Fitness* Awal : 2.202934942E-01

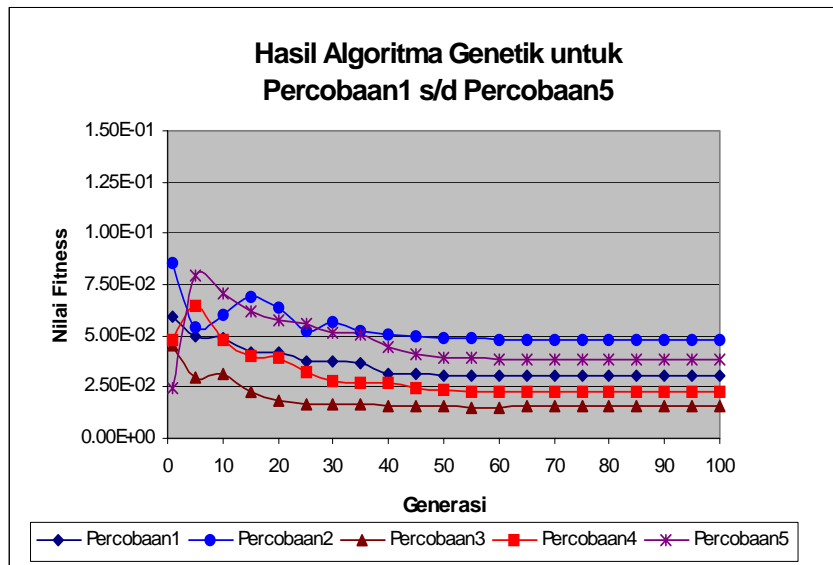
Nomor Percobaan	Probabilitas <i>crossover</i>	Probabilitas mutasi	Nilai <i>Fitness</i> Akhir
1	0.6	0.05	3.043155585E-02
2	0.6	0.1	4.814475279E-02
3	0.6	0.2	1.553226743E-02
4	0.6	0.3	2.256779159E-02
5	0.6	0.4	3.844027020E-02

Tabel 4.1 : Hasil Algoritma Genetik untuk Percobaan-1 sampai dengan Percobaan-5 terhadap Data Hasil Eksperimen di STTI.

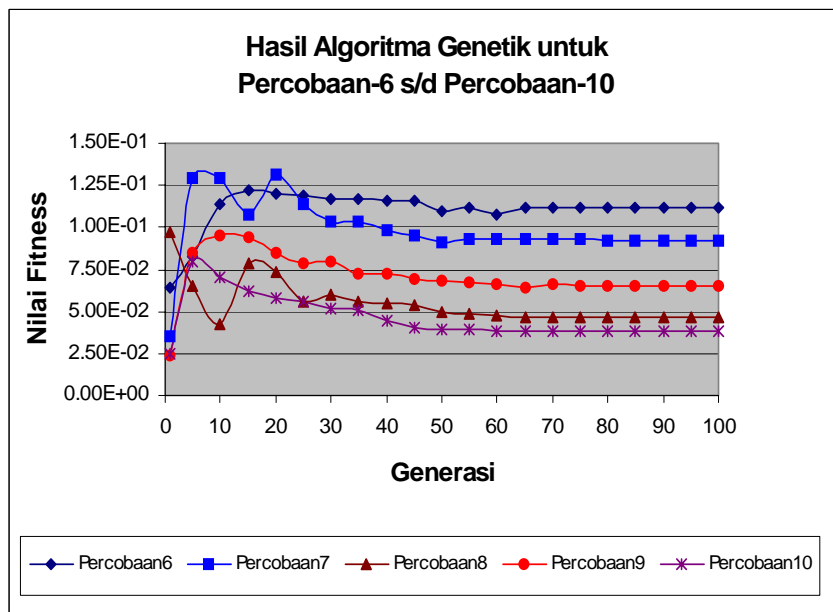
Nilai *Fitness* Awal : 3.2715228317E-01

Nomor Percobaan	Probabilitas <i>crossover</i>	Probabilitas mutasi	Nilai <i>Fitness</i> Akhir
6	0.6	0.05	1.114466611E-01
7	0.6	0.1	9.254981624E-02
8	0.6	0.2	4.618540617E-02
9	0.6	0.3	6.522564006E-02
10	0.6	0.4	3.844027020E-02

Tabel 4.2 : Hasil Algoritma Genetik untuk Percobaan-6 sampai dengan Percobaan-10 terhadap Data Hasil Eksperimen di ITB.



Gambar 3 : Hasil Algoritma Genetik untuk Percobaan1 s/d Percobaan5



Gambar 4 : Hasil Algoritma Genetik untuk Percobaan6 s/d Percobaan10

Dari Gambar 3 terlihat bahwa hasil algoritma genetik konvergen untuk semua percobaan, namun hasil yang

mempunyai nilai *fitness* minimum diperoleh dari percobaan-3 (probabilitas mutasi 0.2).

Hasil Percobaan-6 sampai dengan Percobaan-10 untuk data hasil eksperimen di ITB dapat dilihat pada Gambar 4. Dari Gambar 4, terlihat bahwa hasil algoritma genetik konvergen untuk semua percobaan, namun hasil yang mempunyai nilai *fitness* minimum diperoleh dari percobaan-10 yaitu percobaan dengan probabilitas mutasi 0.4.

Pada semua percobaan, ada kecenderungan konvergensi nilai *fitness* mulai terlihat setelah iterasi atau generasi lebih besar dari 50. Algoritma genetik juga menghasilkan komposisi fungsi keanggotaan yang dapat memberikan nilai *fitness* yang lebih minimum. Hal ini tampak dari nilai *fitness* akhir yang diperoleh dari hasil percobaan yang lebih kecil dari nilai *fitness* awal.

5. Kesimpulan

Kesimpulan yang dapat ditarik dari hasil penelitian ini adalah :
Algoritma genetik dapat digunakan untuk mengatur parameter sistem berbasis *fuzzy* dalam sistem tutorial cerdas sehingga sistem dapat memberikan layanan pembelajaran dengan optimal terhadap populasi siswa dengan kemampuan yang berbeda.

Dari hasil percobaan algoritma genetik untuk melakukan *tuning* terhadap posisi dan lebar fungsi keanggotaan sistem *fuzzy*, diperoleh hasil yang konvergen dengan nilai *fitness* yang minimum jika parameter probabilitas *crossover* yang dipakai adalah 0.6 dan parameter probabilitas mutasi yang berkisar antara 0.2 sampai 0.4. Pada semua percobaan,

ada kecenderungan konvergensi nilai *fitness* mulai terlihat setelah iterasi atau generasi lebih besar dari 50.

6. Daftar Pustaka

- [1] Ayub, M., Liem I., “*Fuzzy Control Application in Computer Assisted Learning*”, Proceeding CECI-SITIA, C22-C26, Juni 2003.
- [2] Ayub, M., The Houw Liong, Hinduan A., Liem I., “*Implementasi Kendali Fuzzy dalam Sistem Tutorial Cerdas*”, Proceeding of ASPAC on ASET, CP201-CP219, Oktober 2004.
- [3] Block, H.J., “*Mastery Learning Theory and Practice*”, Holt, Rinehart and Winston, Inc., 1971.
- [4] Cordon, O., Herrera F., Villar P., “*Generating the Knowledge Base of a Fuzzy Rule-Based System by the Genetic Learning of the Data Base*”, <http://decsai.ugr.es/~herrera/Pub-GFS.html#GFS-D>, 28 Mei 2004.
- [5] Herrera, F., Magdalena, L., “*Genetic Fuzzy Systems : A Tutorial*”, http://mat.upm.es/~llyayos/papers/GF_S.ps, 29 Oktober 2004.
- [6] Herrera, F., Lozano, M., Verdegay, J.L., “*Tuning Fuzzy Logic Controllers by Genetic Algorithms*”, <http://decsai.ugr.es/~herrera/Pub-GFS.html#GFS-D>, 25 April 2004.
- [7] Zakaria, A., Siraj, F., “*Intelligent Tutoring System for Web Based Education*”, <http://www.aisig.uum.edu.my/azizi/TN3063/its2.pdf>, 31 Juli 2003.
- [8] Goldberg, D.E., “*Genetic Algorithms in Search, Optimization and Machine Learning*”, Addison Wesley, 1989.