

PENGIDENTIFIKASIAN PARAMETER FUNGSI ALIH SISTEM PADA PLANT SIMULASI ORDE TIGA DAN EMPAT DENGAN METODE ALGORITMA GENETIK

Jody Roostandy¹, Sumardi², Trias Andromeda²

Abstrak – Identifikasi adalah suatu proses pendekatan dengan cara pemodelan matematis pada suatu sistem yang tidak diketahui lewat pembelajaran terhadap data - data yang dikumpulkan dari eksperimen sebelumnya maupun data masukan keluaran dari sistem terkontrol yang telah ada sebelumnya. Data tersebut diperbandingkan dengan model acuan, yang kemudian persamaan beda (error) antara kedua sistem tersebut dijadikan fungsi objektif yang nantinya akan diminimalkan. Identifikasi dan estimasi dari suatu sistem dapat dilakukan dengan berbagai metode, dan salah satunya adalah dengan menggunakan algoritma genetik. Salah satu alasan dari penggunaan algoritma genetik adalah seperti yang diketahui bahwa grafik dari error tidak unimodal, dimana hanya satu puncak nilai yang optimal, dimana melainkan ada beberapa nilai puncak (multimodal). Sehingga, penyelesaian yang tidak memperhatikan hal ini akan menghasilkan nilai parameter yang hanya merupakan solusi optimal lokal dan akan membentuk sistem yang tidak stabil. Algoritma genetik yang merupakan algoritma optimasi multimodal dan diaplikasikan lewat perangkat lunak MATLAB akan dapat menyelesaikan permasalahan ini.

Kata kunci : pengidentifikasi sistem, error(MSE), algoritma genetik

I. PENDAHULUAN

1.1. Latar Belakang

Proses estimasi dan identifikasi, yang intinya adalah pengenalan suatu sistem atau *plant* yang akan dikendalikan, merupakan salah satu faktor penting yang menunjang keberhasilan dalam merencanakan suatu sistem kendali yang stabil, *robust* serta mampu beradaptasi dengan baik terhadap lingkungan.

Proses estimasi dan identifikasi dapat dilakukan lewat berbagai cara, antara lain secara *on-line*, yaitu dimana sistem terhubung langsung ke perangkat identifikasi dan hasil dari proses tersebut dapat langsung diaplikasikan atau dengan kata lain proses dilakukan secara *real time*. Ataupun dengan metode lain, yaitu secara *off-line*, dimana sistem yang akan diestimasi tidak dihubungkan langsung dengan perangkat identifikasi, tetapi cukup dikenali dengan menggunakan data masukan dan keluaran yang telah dikumpulkan.

Metode identifikasi dan estimasi yang umum digunakan antara lain metode non-parametrik, parametrik non rekursif, dan parametrik rekursif. Sebenarnya penggunaan metode konvensional ini sudah dapat dianggap mencukupi untuk menyelesaikan permasalahan identifikasi pada sebagian besar sistem yang ada. Namun pada beberapa sistem, ataupun untuk

aplikasi nyata, biasanya grafik dari *error* tidak *unimodal*, dimana hanya satu nilai atau satu puncak nilai yang optimal, melainkan ada beberapa nilai puncak (*multimodal*). Sehingga, penyelesaian yang tidak memperhatikan hal ini akan menghasilkan nilai parameter yang hanya merupakan solusi optimal lokal dan akan membentuk sistem yang tidak stabil.

1. Salah satu metode yang dapat menyelesaikan permasalahan ini adalah dengan menggunakan algoritma genetik. Algoritma optimasi ini mampu menyelesaikan masalah *multimodal*, sehingga nantinya akan didapatkan solusi optimal secara global.

1.2. Tujuan

Tujuan yang hendak dicapai dalam Tugas Akhir ini adalah mampu untuk menyelesaikan permasalahan *Pengidentifikasi secara off-line dari suatu sistem* berupa *plant* kendali simulasi berorde tiga dan empat dengan menggunakan metode algoritma genetik yang berbasis perangkat lunak MATLAB.

1.3. Pembatasan Masalah

Pembatasan masalah yang dilakukan dalam perancangan perangkat pengidentifikasi sistem berbasis algoritma genetik pada *plant* simulator ini adalah:

1. Identifikasi sistem dilakukan secara *off-line*.
2. Sistem yang digunakan merupakan *plant* simulasi orde tiga dan empat.
3. Proses komputasi akan menghasilkan parameter optimum pembentuk fungsi alih yang modelnya telah ditentukan sebelumnya.
4. Metode yang digunakan dalam perancangan hanya algoritma genetik dan tidak menggunakan metode lain sebagai pembanding.
5. Perancangan pengidentifikasi dengan Algoritma Genetik ini dilakukan dengan bantuan program MATLAB.
6. Algoritma genetik yang digunakan adalah model SGA (*Simple Genetic Algorithm*).
7. Analisa dilakukan dengan cara membandingkan antara nilai parameter *plant* nyata dengan nilai yang dihasilkan lewat proses komputasi algoritma genetik.

II. IDENTIFIKASI DAN ESTIMASI SISTEM

Dalam analisis dan perancangan sistem kendali dibutuhkan suatu pengetahuan tentang sistem atau *plant* yang

¹ Mahasiswa Teknik Elektro Undip

² Staf Pengajar Teknik Elektro Undip

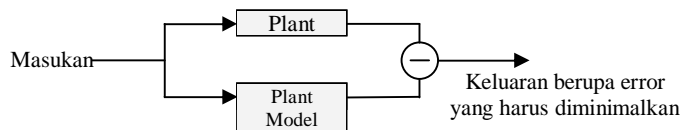
akan kendalikan. Bentuk dari pengetahuan tersebut bisa berupa model matematis dari *plant* tersebut.

Ada dua cara atau pendekatan untuk mendapatkan model matematik dari sistem yang akan dikendalikan. Caranya, antara lain :

1. Lewat penelaahan hukum fisis yang berhubungan dengan sistem yang dikendalikan.
2. Dengan mencari penyelesaian lewat metode eksperimental. Metode pencarian model matematik melalui penelaahan hukum fisis tidak akan memberi hasil identifikasi yang akurat, sebab ada parameter – parameter yang dibutuhkan dan hanya bisa didapatkan lewat proses eksperimental. Sehingga seharusnya, dibutuhkan suatu kombinasi dari dua pendekatan tersebut.

Identifikasi sistem sendiri didefinisikan^{[14] [21]} sebagai usaha merekonstruksi model parametrik *plant* lewat usaha eksperimental dengan menggunakan data yang berasal dari masukan dan keluaran. Dan selanjutnya diakhiri dengan proses estimasi yang akan memberikan parameter optimal yang lebih menjelaskan tentang sistem yang akan dikenali.

seperti yang terlihat pada gambar 1 maka beda antara keluaran *plant* nyata dan model disebut dengan *error*. Jika sebuah set parameter *plant* dipilih untuk meminimalkan *error* yang misalnya dalam bentuk MSE dan parameter ini telah optimal dalam mencapai tujuannya, maka parameter ini dapat dianggap sebagai parameter *plant* hasil estimasi terbaik.



Gambar 1. Proses Pembentukan Nilai Error

Persyaratan umum dari pengidentifikasian suatu sistem secara *on-line* adalah dibatasinya waktu pengambilan data cuplik (*sample*) dari *plant* yang sedang dipantau. Selain itu waktu proses pengidentifikasian sistem hingga mendapatkan hasil tidak boleh terlalu lama. Karena, dikhawatirkan telah terjadi perubahan pada *plant* yang disebabkan perubahan lingkungan terhadap waktu. Perubahan pada *plant* jelas mengakibatkan perubahan pada parameter – parameter karakteristik dari *plant* tersebut. Sehingga hasil yang diperoleh nantinya tidak valid atau sudah tidak sesuai.

2.1 Identifikasi Sistem Teknik Parametrik

Metode identifikasi sistem dengan teknik parametrik berbeda dengan non parametrik. Jika pada model parametrik hasil yang didapatkan adalah dalam bentuk parameter maka dengan model non-parametrik, yang diperoleh adalah kurva – kurva dinamik sistem seperti tanggapan step, *impulse*, tanggapan frekuensi dan sebagainya.

Kekurangan metode non-parametrik adalah :

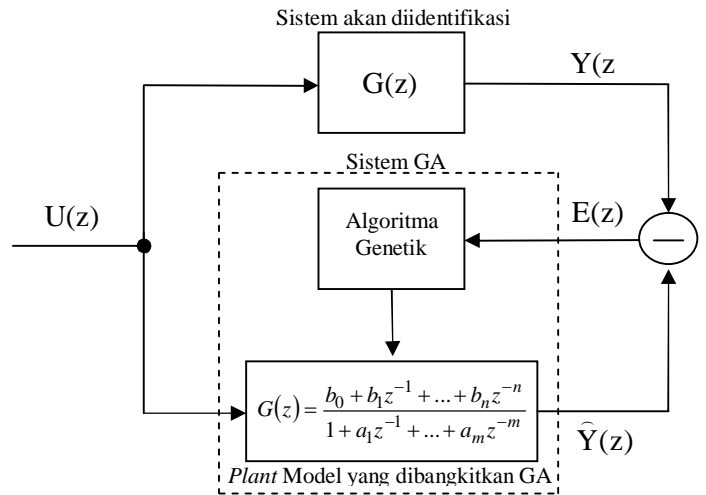
1. Parameter model sistem tidak bisa diperoleh secara langsung
2. Tidak mampu menganalisa dan mengidentifikasi sistem kompleks berorde lebih dari dua (tiga, empat, dan seterusnya).

3. Kurang mampu mengidentifikasi sistem berorde satu atau dua dengan faktor gangguan (*disturbance*) berintensitas tinggi dan hasil yang diperoleh tidak memuaskan.
4. Tidak dapat diandalkan untuk kebutuhan identifikasi secara *on-line*, yang misalnya aplikasi pada kontrol otomatis.

Karenanya untuk saat ini model parametrik lebih banyak digunakan dalam aplikasi riil.

2.2 Identifikasi Dengan Menggunakan Algoritma Genetik

Secara umum, suatu perangkat pengidentifikasi sistem model parametrik dengan menggunakan algoritma genetik yang bekerja baik secara *real time* atau *on-line* maupun secara *off-line* dapat dijelaskan berdasarkan gambar 2^[6].



Gambar 2. Proses Identifikasi Sistem Dengan Menggunakan GA

Sinyal masukan yang sama diumpungkan ke dalam *plant* nyata dan *plant* model parametrik sistem yang dibangkitkan oleh kromosom dari algoritma genetik. Sinyal masukan dapat berbentuk tegangan, sinyal masukan khusus yang sengaja diumpungkan, atau bentuk sinyal lainnya.

Keluaran dari kedua *plant*, baik nyata maupun model pada akhir proses komputasi identifikasi dengan algoritma genetik diharapkan memiliki keluaran yang sama.

Parameter fungsi alih yang ingin dicari pada proses identifikasi dan estimasi sistem ini diwakili dengan komponen variabel b_n dan a_m , dimana n dan m adalah orde dari fungsi alih yang dibangkitkan oleh algoritma genetik. Keluaran dari *plant* nyata $Y(z)$ dan model $\hat{Y}(z)$ akan saling mengurangi dan menghasilkan nilai beda (*error*). Dan kemudian dari nilai beda tersebut akan dibentuk persamaan beda lewat formula kesalahan kuadrat rata-rata (*mean square error*) yang prinsip dasarnya akan diterangkan secara lebih jelas pada subbab 2.5.

$$\varepsilon = \frac{\sum (\hat{Y}(z) - Y(z))^2}{N}$$

ε = persamaan beda (*error*)

$Y(z)$ = keluaran dari sistem yang akan diidentifikasi

$\hat{Y}(z)$ = keluaran dari model yang dibangkitkan oleh sistem algoritma genetik.

N = Jumlah persamaan beda.

Persamaan ini nantinya akan diumpamakan ke sistem algoritma genetik sebagai fungsi objektif yang harus diminimalkan.

kelebihan metode GA dibandingkan dengan metode identifikasi konvensional seperti RLS (*Recursive Least Square*) yaitu, GA dapat bekerja pada fungsi nonlinier, *non-differerensiable*, diskontinyu, sehingga bisa digunakan secara langsung untuk mengidentifikasi parameter dari sistem.

III. TEORI ALGORITMA GENETIK

Algoritma genetik pertama kali dikembangkan oleh John Holland dari Universitas Michigan pada tahun 1975 dengan paper “*adaptation in Natural and Artificial System*”. Dalam paper ini banyak dibicarakan tentang proses adaptasi natural dan pengembangan perangkat lunak yang berbasis pada mekanisme alam.

3.1 Tinjauan Umum

Algoritma genetik adalah suatu metode pencarian atau optimasi diskrit yang menggunakan prinsip dasar dari mekanisme evolusi alam. Algoritma genetik bekerja pada suatu populasi solusi permasalahan. Perkembangan populasi pada generasi berikutnya dihasilkan melalui mekanisme genetik yang dihasilkan dari perkawinan dua individu yang memiliki kualitas di atas rata – rata. Proses perkembangbiakan individu akan terus berlangsung dan akan memberikan hasil yaitu suatu populasi dimana setiap individu memiliki kualitas baik, proses evolusi akan berlangsung dalam beberapa generasi yang ditentukan. Setiap individu dalam suatu populasi memiliki kromosom yang mempunyai struktur gen yang sama. Setiap kromosom membawa informasi tentang parameter-parameter yang dibutuhkan dalam suatu proses pemecahan masalah.

Algoritma genetik memiliki perbedaan yang mendasar dengan metode pencarian solusi optimal berbasis model matematika kalkulus, perbedaan tersebut adalah sebagai berikut :

1. Mekanisme optimasi algoritma genetik bekerja berdasarkan kromosom, dimana setiap kromosom menyimpan informasi parameter – parameter tersebut.
2. Proses pencarian solusi optimal pada mekanisme algoritma genetik tidak dilakukan pada satu titik pencarian, tetapi pada sekumpulan titik pencarian.
3. Algoritma genetik tidak membutuhkan prosedur – prosedur matematis dalam mencari solusi optimal tetapi algoritma genetik menggunakan informasi langsung dari hasil transfer tiap – tiap parameternya ke suatu fungsi yang dapat mewakili tujuan dari proses optimasi yang sedang dilakukan.
4. Mekanisme genetik digunakan dalam pemrosesan kode parameter suatu permasalahan, melalui proses seleksi, rekombinasi dan mutasi untuk memperoleh solusi optimal.
5. Proses pencarian solusi optimal menggunakan metode algoritma genetik menggunakan titik acuan sembarang, untuk menghindari solusi optimal lokal.

6. Mekanisme pencarian terbimbing diberikan melalui penilaian terhadap kualitas kode atau kromosom yang dimiliki oleh setiap individu dalam suatu generasi.

3.2 Mekanisme Kerja Algoritma Genetik

Algoritma genetik memiliki proses melalui suatu tahapan siklus kerja sebagai berikut:

1. Membentuk suatu populasi string awal secara acak
2. Mengevaluasi masing – masing string
3. Memilih string dengan kualitas atau fitness terbaik
4. Memanipulasi string-string secara genetik untuk menghasilkan populasi string baru

3.3 Operator-operator Algoritma Genetik

Operator – operator genetik merupakan ‘tool’ yang sangat penting dalam proses algoritma genetik. Ada empat operator genetik yang digunakan pada perancangan tugas akhir ini, yang antara lain :

1. Seleksi dan Reproduksi

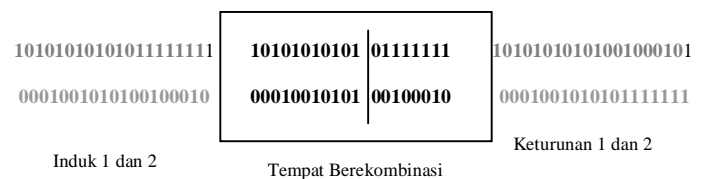
Proses Seleksi adalah suatu proses awal berupa pemilihan individu – individu terbaik dalam suatu populasi yang pantas untuk mengikuti proses genetik selanjutnya. Operator seleksi dapat dikatakan merupakan model dari siapa yang terkuat (*fittest*) dialah yang dapat bertahan (*survive*).

2. Rekombinasi (Crossover)

Rekombinasi adalah proses pertukaran struktur kromosom antara dua induk yang terpilih dalam proses seleksi dengan tujuan untuk menciptakan keberagaman individu – individu baru yang tetap mewarisi sifat – sifat terbaik dari induk – induknya. Dalam algoritma genetik dilakukan atau tidaknya proses rekombinasi ditentukan oleh nilai probabilitas terjadinya peristiwa rekombinasi (P_c) yang ditentukan oleh pengguna. Jenis rekombinasi yang digunakan pada tugas akhir ini adalah rekombinasi dengan jenis sisi tunggal (*single site crossover*).

Single Site Crossover (SSC)

Rekombinasi jenis ini memiliki ciri ada sebuah titik yang menjadi tempat terjadinya perpindahan struktur kromosom antar kromosom induk. Titik tempat terjadi awal dari pertukaran struktur kromosom induk ditentukan secara sembarang dalam rentang yang tidak melebihi dari panjang kromosom induk. Jenis rekombinasi seperti ini banyak digunakan karena efektifitas dalam penganekaragaman struktur kromosom individu lebih baik dan materi genetik dari kromosom induk tidak cepat hilang. Contoh mekanisme rekombinasi satu titik adalah sebagai berikut



Gambar 3. Proses Rekombinasi Satu Titik

3. Mutasi

Mutasi adalah operator genetik yang merubah satu atau lebih gen – gen dalam sebuah kromosom dari bentuk aslinya dan menghasilkan sebuah gen baru. Dengan gen baru yang dihasilkan, algoritma genetik dapat menghasilkan solusi yang lebih baik. Mutasi adalah bagian penting dalam penelusuran genetika karena dapat membantu menjaga populasi dari kemacetan pada saat optimasi lokal.

4. Reinsersi

Reinsersi dapat didefinisikan sebagai proses menjaga ukuran dari populasi asli, baik pada saat populasi kekurangan individu maupun saat kelebihan individu. Selain itu dapat juga diartikan sebagai skema yang harus digunakan untuk menentukan individu mana saja yang pantas untuk dipertahankan dalam populasi yang baru. Seperti yang telah dijelaskan di atas, setelah populasi baru telah diproduksi lewat proses seleksi dan rekombinasi, maka nilai fitness dari individu yang baru dapat ditentukan. Jika populasi individu baru yang dihasilkan lewat rekombinasi lebih sedikit dibanding dengan ukuran populasi sebenarnya, maka beda jumlah populasi tersebut disebut sebagai celah generasi (*generation gap*). Jika satu individu yang lebih atau paling fit pada generasi sebelumnya, disisipkan pada pembentukan populasi pada generasi selanjutnya, maka hal ini disebut juga strategi elitist. Pada Strategi elitist ini, anggota dengan nilai fitness yang rendah digantikan dengan individu yang lebih baik. Sedangkan strategi penggantian terbaik adalah dengan cara mengganti anggota populasi yang paling lama atau tua.

IV. PERANCANGAN PERANGKAT LUNAK

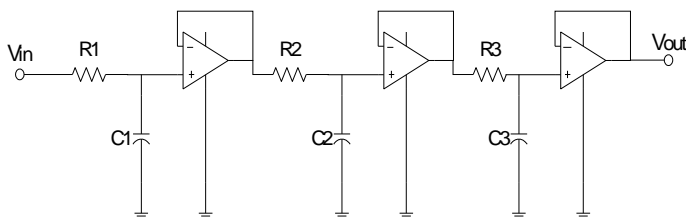
Pembentukan dan perancangan plant simulasi

Ketiga *plant* yang digunakan dalam perancangan ini merupakan *plant* simulasi yang dikembangkan berdasarkan rangkaian elektronik dengan fungsi alih berorde satu dan dua, yang terdiri dari komponen pasif resistor dan kapasitor serta komponen aktifnya yaitu penguat operasional (Op-amp).

Karakteristik ketiga *plant* tersebut dapat dijelaskan sebagai berikut :

1. Orde 3 (tipe 1).^[17]

Plant orde 3 jenis pertama dibentuk berdasarkan penggabungan tiga *plant* berorde 1 seperti gambar 4.



Gambar 4. Plant Simulasi Orde 3 Jenis Pertama

Dengan $R_1 = R_2 = R_3 = 1 \text{ K}\Omega$ dan $C_1 = C_2 = C_3 = 100\mu\text{F}$

Dimana fungsi alih rangkaian keseluruhan adalah:

$$G_p(s) = \frac{1}{sR_1C_1 + 1} \times \frac{1}{sR_2C_2 + 1} \times \frac{1}{sR_3C_3 + 1} \quad (3.1)$$

maka dengan memasukkan nilai dari variabel yang ada akan didapatkan fungsi alihnya dalam bentuk laplace

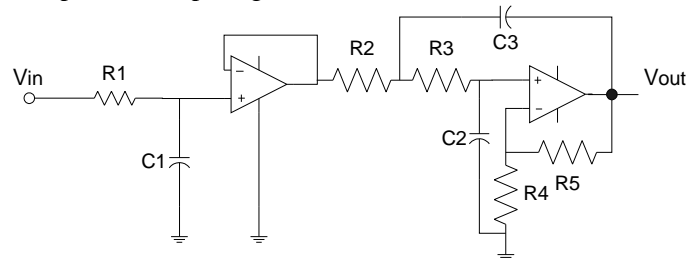
$$G_p(s) = \frac{10^3}{s^3 + 30s^2 + 300s + 1}$$

Dalam perancangan ini fungsi alih dari *plant* nyata yang akan digunakan harus berada dalam *domain z*. Sehingga ditransformasikan dalam bentuk *z*

$$G_p(z) = \frac{2,2964z^2 + 0,3325z}{z^3 - 0,9967z^2}$$

2. Orde 3 (tipe 2).^[17]

Plant orde 3 jenis kedua dibentuk berdasarkan penggabungan antara satu *plant* berorde 1 dan satu *plant* berorde 2. Bentuk rangkaian dari *plant* simulasi ini diperlihatkan pada gambar 5.



Gambar 5. Plant Simulasi Orde 3 Jenis Kedua

Dengan $R_1 = R_2 = R_3 = 10 \text{ K}\Omega$; $R_4 = R_5 = 1 \text{ K}\Omega$

$C_1 = C_2 = C_3 = 100\mu\text{F}$

Dimana fungsi alih rangkaian keseluruhan adalah :

$$G_p(s) = \frac{1}{sR_1C_1 + 1} \times \frac{k}{s^2(R_2C_2R_3C_3) + s(R_2C_2 + R_2C_3 + R_3C_2 - R_2C_3k) + 1} \quad (3.2)$$

dengan

$$k = 1 + \frac{R_5}{R_4}$$

maka dengan memasukkan nilai dari variabel yang ada akan didapatkan fungsi alihnya dalam bentuk laplace

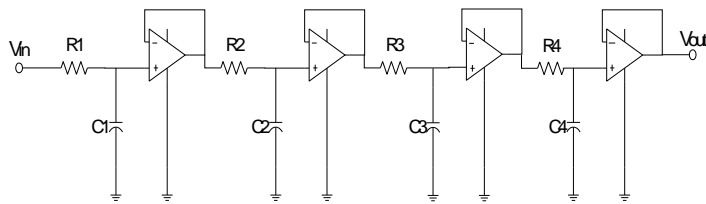
$$G_p(s) = \frac{2}{(s+1)(s^2 + s + 1)} = \frac{2}{s^3 + 2s^2 + 2s + 1}$$

Dalam perancangan ini fungsi alih dari *plant* nyata yang akan digunakan harus berada dalam *domain z*. Sehingga ditransformasikan dalam bentuk *z*

$$G_p(z) = \frac{0,1972z^2 + 0,4460z + 0,0726}{z^3 - 1,1538z^2 + 0,6507z - 0,1353}$$

3. Orde 4.^[17]

Plant orde 4 ini didapatkan dengan cara menggabungkan empat *plant* berorde 1. Bentuk rangkaian dari *plant* simulasi ini diperlihatkan pada gambar 6.



Gambar 6. Plant Simulasi Orde 4

Dengan $R_1 = 10K\Omega$; $R_2 = R_3 = R_4 = 3,3 K\Omega$

$C_1 = C_2 = C_3 = C_4 = 100\mu F$

Dimana fungsi alih rangkaian keseluruhan adalah :

$$Gp(s) = \frac{1}{sR_1C_1 + 1} \times \frac{1}{sR_2C_2 + 1} \times \frac{1}{sR_3C_3 + 1} \times \frac{1}{sR_4C_4 + 1} \quad (3.4)$$

maka dengan memasukan nilai dari variabel yang ada akan didapatkan fungsi alihnya dalam bentuk laplace

$$Gp(s) = \frac{27}{(s+1)(s+3)^3} = \frac{27}{s^4 + 9s^3 + 27s^2 + 54s + 27}$$

Dalam perancangan ini fungsi alih dari *plant* nyata yang akan digunakan harus berada dalam *domain z*. Sehingga ditransformasikan dalam bentuk z

$$Gp(z) = \frac{0,2050z^3 + 0,4222z^2 + 0,0924z + 0,0012}{z^4 - 0,1318z^3 - 0,1064z^2 - 0,0412z + 0,0001}$$

Model algoritma genetik yang digunakan pada perancangan tugas akhir ini adalah SGA yang merupakan kependekan dari *Simple Genetic Algorithms*. Dimana langkah – langkah perancangan dari algoritma genetik model SGA ini dapat dilihat pada diagram alir perancangan gambar 7.

Penggunaan algoritma genetik model SGA pada perangkat lunak pengidentifikasi ini didasarkan pada beberapa alasan yang menjadi kelebihananya, yaitu :

1. Tidak memerlukan sistem komputer yang besar untuk melakukan proses komputasinya.
2. Sistematika operator genetik yang digunakan cukup baik
3. kestabilan kualitas individu dalam suatu generasi dapat terjaga dengan baik lewat metode elitist.
4. tidak ada penumpukan (*overlapping*) pada populasi string, reproduksi, *crossover* dan mutasi.
5. adaptasi *pay-off* (fungsi obyektif) yang sederhana.

Pembentukan Kromosom

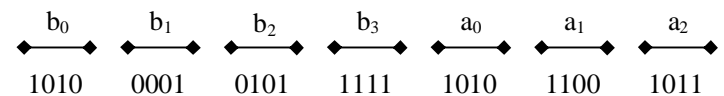
Kromosom dalam ruang lingkup algoritma genetik adalah suatu kumpulan dimana gen – gen berformat biner yang merupakan pembawa sifat berada.

Pada perancangan tugas akhir ini, kromosom akan terdiri dari gen – gen dalam format *binary gray* yang mewakili parameter pembentuk fungsi alih dari sistem yang akan diidentifikasi.

Jika dimisalkan, persamaan fungsi alih dari sistem yang akan diidentifikasi merupakan fungsi alih dalam z yang berorde tiga seperti persamaan (3.5).

$$G(z) = \frac{b_0 z^3 + b_1 z^2 + b_2 z + b_3}{z^3 - a_0 z^2 + a_1 z + a_2}$$

Maka bentuk kromosom berkepresisian empat bit per satuan gen yang dapat mewakili satu individu diperlihatkan pada gambar 8.



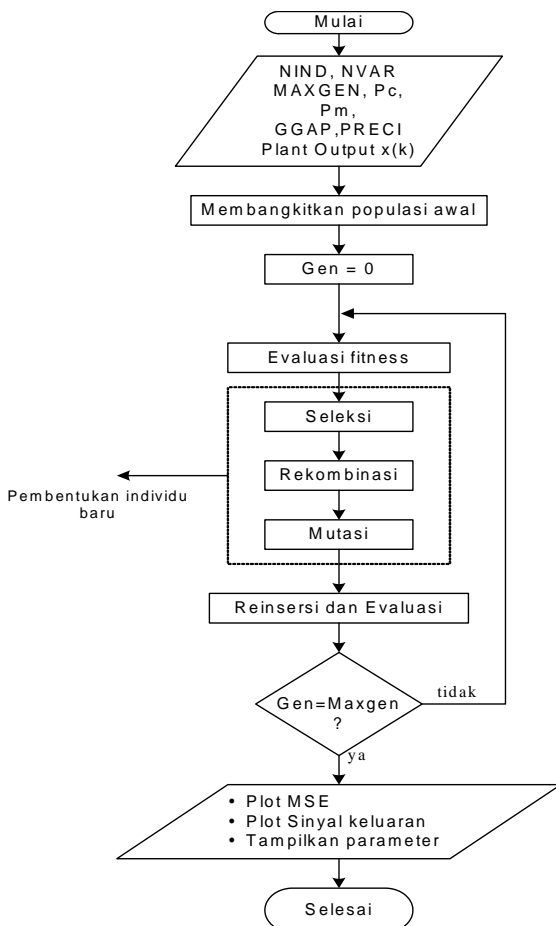
Gambar 8. Model Kromosom dari Parameter Fungsi Alih

Pembentukan Fungsi Obyektif dan Indeks Kebaikan (Fitness Value)

Fungsi obyektif mempunyai fungsi yang esensial dalam proses komputasi algoritma genetik. Sebab fungsi obyektif digunakan untuk menghasilkan harga indeks kebaikan dari setiap individu, dan indeks kebaikan (*fitness value*) berfungsi sebagai pengevaluasi kualitas dari individu dari suatu populasi. Dengan kata lain, untuk menentukan apakah suatu individu layak untuk dipertahankan atau pantas untuk dihilangkan dan hal ini biasa disebut dengan *proses seleksi*. Selain itu fungsi obyektif dibutuhkan untuk melihat, apakah tujuan dari proses komputasi ini telah tercapai.

Pada perancangan tugas akhir ini dipilih metode MSE (*Mean Square Error*) sebagai pembentuk fungsi obyektif. Alasan penggunaannya adalah karena kemampuannya dalam

Perancangan perangkat lunak sistem GA



Gambar 7. Diagram Alir Perancangan

mencari titik minimum, sehingga salah satu penentu keberhasilan dari identifikasi yang antara lain membuat *error* hingga mendekati nol memiliki probabilitas besar untuk dapat tercapai.

Proses pembentukan fungsi objektif dapat dijabarkan sebagai berikut :

1. Menentukan bentuk atau jenis serta nilai dari sinyal masukan $u(k)$.
2. Mencari bentuk serta nilai sinyal keluaran yang berasal dari *plant* nyata, $x(k)$ berdasarkan sinyal masukan.
3. Membangkitkan kromosom yang berisi gen – gen parameter fungsi alih, dicari nilai sebenarnya dari phenotip dan membentuk sinyal keluaran *plant* identifikasi $y(k)$ berdasarkan sinyal masukan.
4. Memasukan nilai $x(k)$ dan $y(k)$ kedalam fungsi MSE. Harga *error* yang dihasilkan keluaran fungsi MSE merupakan nilai fungsi objektif yang akan digunakan.

$$MSE = \frac{\sum_{n=1}^k (y(n) - x(n))^2}{k}$$

Data Pelengkap Proses Genetik

a. Data genetik

Data genetik adalah data yang berisikan parameter – parameter genetik yang dibutuhkan dalam menjaga kelangsungan proses komputasi algoritma genetik. Data – data tersebut antara lain,

1. Jumlah Generasi (MAXGEN)

Parameter MAXGEN digunakan untuk menentukan besar generasi maksimum yang akan dicapai dalam proses komputasi algoritma genetik. Generasi maksimum yang digunakan pada perancangan ini adalah 1000.

2. Jumlah Individu Pada Populasi (NIND)

Pemilihan NIND yang kecil dapat mengakibatkan kecilnya kemungkinan terjadi proses rekombinasi, sedangkan nilai yang besar berpengaruh pada kecepatan proses komputasi. Pada perancangan ini digunakan nilai NIND dengan besar 40, 70 dan 100.

3. Jumlah Variabel Dalam Kromosom (NVAR)

Penentuan jumlah variabel atau parameter mempengaruhi bentuk kromosom serta tujuan akhir dari proses komputasi algoritma genetik itu sendiri. Nilai NVAR yang digunakan adalah sebesar 6 sampai dengan 9, tergantung kepada model fungsi alih yang akan digunakan

4. Kepresisian (PRECI)

Nilai PRECI menandakan jumlah bit yang merepresentasikan nilai riil dari gen atau parameter fungsi alih. Makin besar nilai PRECI makin baik serta presisi pula hasil yang akan didapat, namun berakibat pada makin beratnya proses komputasi. Dalam perancangan ini, nilai PRECI yang digunakan adalah sebesar 40 bit per satuan gen atau parameter.

5. Celah Generasi (GGAP)

Nilai GGAP yang digunakan dapat mempengaruhi proses pencapaian hasil, karena parameter ini dibutuhkan untuk menjaga kelancaran proses reinserti. Dalam perancangan ini

nilai GGAP yang dipakai adalah sebesar 0,8. Maksudnya dari angka di atas adalah, jika dimisalkan setiap populasi berisi 40 individu maka pada akhir dari proses genetik pada tiap generasi, 80% posisi pada populasi baru akan diisi oleh individu baru hasil proses genetik, sedangkan sisanya akan diisi oleh individu terbaik dari populasi sebelumnya.

6. Probabilitas Rekombinasi (Pc)

Nilai Pc yang dapat digunakan adalah sangat bervariasi dan bergantung pada bentuk populasi yang digunakan. Biasanya, makin besar populasi makin besar pula nilai Pc yang digunakan. Pada perancangan tugas akhir ini digunakan nilai Pc yang berkisar antara 0,6 hingga 1. Atau tepatnya diambil tiga nilai yaitu, 0,75; 0,85; dan 1.

7. Probabilitas Mutasi (Pm)

Harga Pm yang akan digunakan dalam perancangan ini diambil dalam kisaran 0,005 hingga 0,01 atau tepatnya adalah 0,007; 0,008; 0,009; dan 0,01. Dari variasi nilai probabilitas mutasi tersebut, akan dianalisa pengaruhnya terhadap kinerja algoritma genetik dalam menyelesaikan tugasnya. (3.6)

b. Data masukan

Data input yang digunakan dalam perancangan ini berbentuk sinyal PRBS (*Pseudo – Random Binary Sequences*) yang dibangkitkan lewat fungsi yang telah disediakan oleh MATLAB. Metode pembangkitannya dapat dijelaskan sebagai berikut.

Pembangkitan sinyal PRBS dilakukan dengan perintah

$$u = idinput(K, 'prbs')$$

dimana,

1. u adalah sinyal keluaran $u(k)$ yang akan didapatkan.
2. K adalah panjang sinyal yang akan dibangkitkan. Dan pada perancangan ini nilai K yang digunakan adalah sebesar 500.
3. Perintah *idinput* merupakan fungsi dari MATLAB yang digunakan untuk membangkitkan sinyal yang dibutuhkan dalam proses identifikasi.
4. *Prbs* adalah jenis sinyal yang akan dibangkitkan lewat perintah *idinput*.

c. Data keluaran

Data keluaran yang diinginkan pada akhir proses komputasi algoritma genetik ini adalah data dalam bentuk grafik dan data dalam bentuk angka atau nilai. Data dalam bentuk grafik berisikan informasi mengenai dua hal, yang antara lain :

1. Grafik mengenai proses *error* menuju nilai terbaiknya yaitu nol. Sumbu y dari grafik ini menunjukkan nilai *error* paling minimal yang terdapat dalam satu populasi disuatu generasi. Sedangkan sumbu x -nya berisi informasi mengenai angka generasi yang telah dicapai.
2. Grafik yang memberi informasi tentang perbandingan bentuk antara sinyal keluaran yang berasal dari *plant* nyata dengan *plant* hasil estimasi algoritma genetik. Sedangkan data keluaran yang didapatkan dalam bentuk angka akan berisi informasi mengenai nilai parameter hasil

estimasi dan nilai *error* atau MSE yang diperoleh setelah proses komputasi berakhir.

V. ANALISA HASIL KOMPUTASI

4.1 Skenario Pengujian

Skenario pengujian terbagi menjadi tiga, yang antara lain :

a. Pengujian berdasarkan model fungsi alih

Ada tiga pengujian terhadap model fungsi alih yang terdiri dari sembilan skenario yang dilakukan yaitu :

1. Pengujian terhadap *plant* orde 3 tipe 1

a. Skenario 1.

Pada skenario ini model *improper* akan diterapkan pada fungsi alih berorde 3 tipe 1, sehingga fungsi alih dari *plant* model akan berbentuk sebagai berikut.

$$G(z) = \frac{b_0 z^3 + b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}$$

b. Skenario 2

Fungsi alih model *proper* diterapkan pada *plant* orde 3 tipe 1. Fungsi alihnya akan berbentuk,

$$G(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^3 + a_1 z^2 + a_2 z + a_3}$$

c. Skenario 3

Penerapan model konstanta pada *plant* orde 3 tipe 1, fungsi alihnya akan berbentuk

$$G(z) = \frac{b_0}{z^3 + a_1 z^2 + a_2 z + a_3}$$

2. Pengujian terhadap *plant* orde 3 tipe 2

Tiga pengujian berdasarkan model fungsi alih yang dilakukan yaitu :

a. Skenario 4

Pada skenario pengujian 4, model *improper* akan diterapkan untuk *plant* orde 3 tipe 2.

b. Skenario 5

Pada skenario ini model *proper* diterapkan untuk *plant* orde 3 tipe 2.

c. Skenario 6

Penerapan model konstanta pada *plant* orde 3 tipe 2.

3. Pengujian terhadap *plant* orde 4

Tiga buah skenario pengujian berdasarkan model fungsi alih yang dilakukan yaitu :

a. Skenario 7

Pada skenario pengujian 7, model *improper* diberikan pada fungsi alih orde 4.

b. Skenario 8

Model *proper* diterapkan pada fungsi alih berorde 4.

c. Skenario 9

Penerapan model konstanta untuk *plant* berorde.

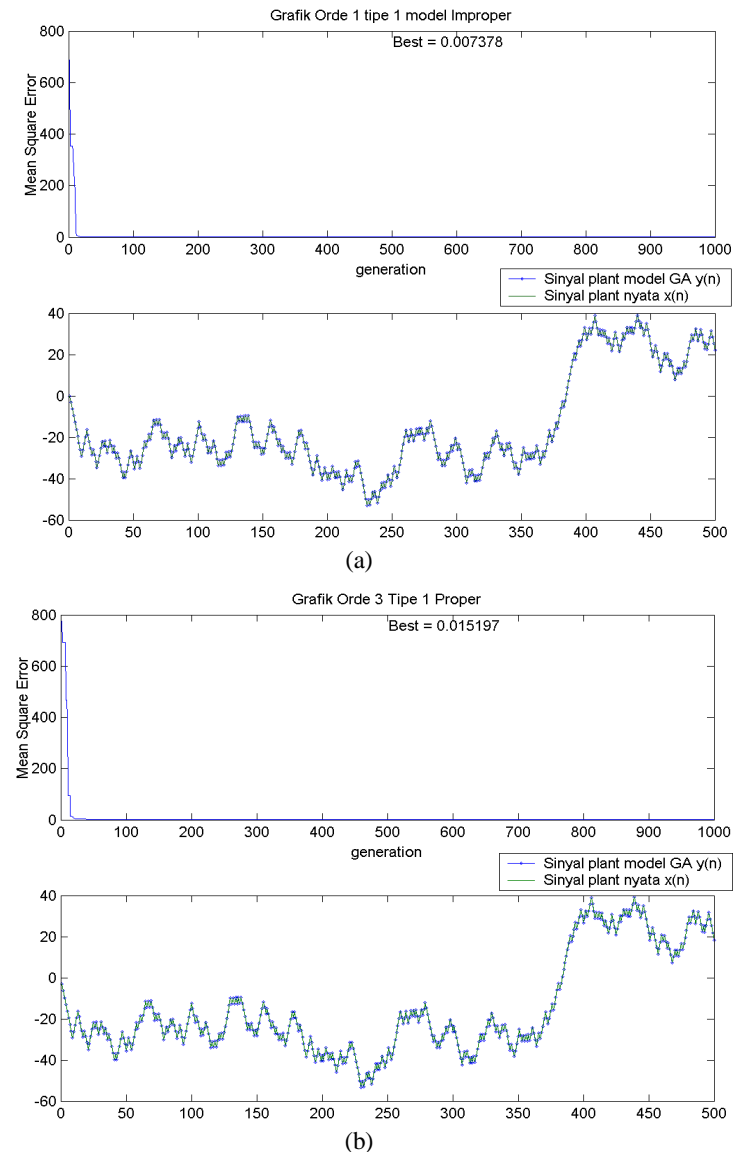
Dengan hasil pengujian pada *plant* orde 3 tipe 1 sebagai berikut.

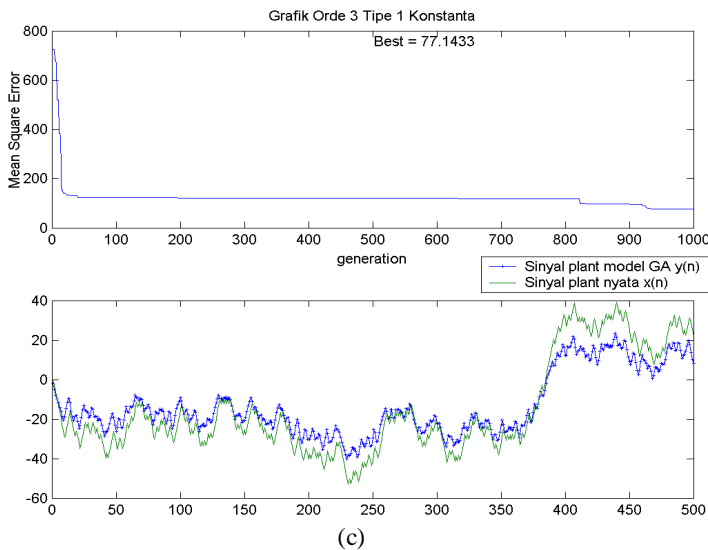
Tabel 1
Parameter Hasil Komputasi Skenario 1, 2, dan 3

Parameter	Plant	Improper	Proper	Konstanta
b_0	0	0.0468	3.0584	2,0000
b_1	2.9964	2.9524	2.0000	-
b_2	0.3325	1.3906	1.1311	-
b_3	0	-0.2213	-	-
a_1	-0.9967	-0.6160	-0.4228	-1,5312
a_2	0	-0.5079	-0.2856	0,9684
a_3	0	0.1280	-0.3629	0,9684
<i>MSE</i>		0.0074	0.0152	77,1433

Dari tabel 1 dapat diambil suatu kesimpulan bahwa kinerja paling baik yang teramati adalah pada saat fungsi alih model *improper* digunakan untuk merepresentasikan *plant* model.

Gambar 9 di bawah dapat digunakan untuk membandingkan hasil komputasi antara penggunaan model *improper*, ataupun model *proper* dan model konstanta yang *underparameterized*.





Gambar 9. hasil komputasi pada fungsi alih orde 3 tipe 1, (a)model improper, (b) proper, (c) konstanta.

b. Pengujian berdasarkan variasi nilai P_c dan P_m

Tujuannya adalah mengetahui tingkat kinerja perangkat lunak pengidentifikasi parameter sistem yang telah dibuat jika diberikan suatu variasi pada nilai Probabilitas rekombinasi dan mutasi.

1. Variasi probabilitas rekombinasi (P_c)

Probabilitas mutasi ditentukan sebesar 0,01, sedangkan P_c divariasikan ke dalam tiga nilai yang telah ditentukan.

a. Skenario 10

Fungsi alih *improper* orde 4 dengan $P_c = 0,45$ dan $P_m = 0,01$.

b. Skenario 11

Fungsi alih *improper* orde 4 dengan $P_c = 0,85$ dan $P_m = 0,01$.

c. Skenario 12

Fungsi alih *improper* orde 4 dengan $P_c = 1$ dan $P_m = 0,01$.

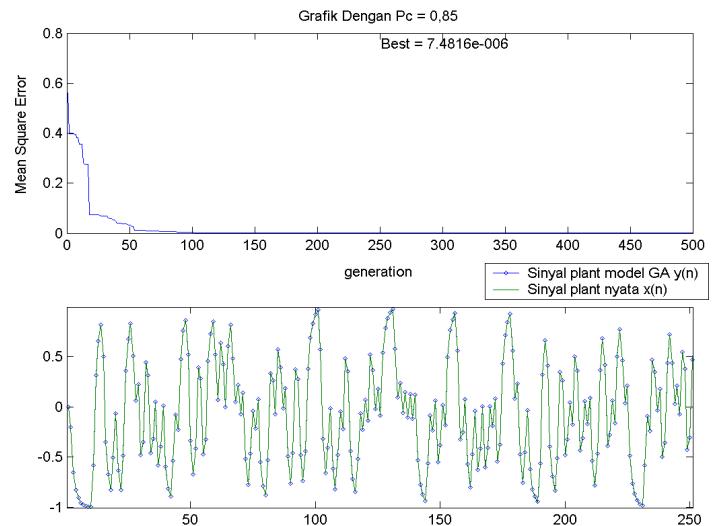
Dengan hasil pengujian sebagai berikut

Tabel 2
Parameter Hasil Komputasi Skenario 10, 11, dan 12

Parameter	Plant	$P_c = 0,45$	$P_c = 0,85$	$P_c = 1$
B_0	0	-0.0124	0,000274	$7,3142 \cdot 10^{-6}$
b_1	0.2050	0.2344	0.2049	0.2050
b_2	0.4222	0.1743	0.3374	0.4529
b_3	0.0924	-0.1230	-0.0327	0.2075
b_4	0.0012	0.0557	0.0526	0.1105
a_1	-0.1318	-1.1125	-0.5453	0.0177
a_2	-0.1064	0.6927	0.1885	-0.1499
a_3	-0.0412	-0.3521	-0.0850	-0.0850
a_4	0.0001	0.1169	0.0056	-0.0177
MSE		0.002715	$7,4816 \cdot 10^{-6}$	$7,6829 \cdot 10^{-6}$

Dengan hasil terbaik diberikan oleh skenario pengujian 11 yang menggunakan nilai P_c sebesar 0,85, sehingga dapat ditarik kesimpulan bahwa kinerja terbaik didapatkan pada probabilitas rekombinasi diatas 0,5 atau tepatnya berada pada

rentang 0,6 hingga 1. Sedangkan grafik hasil komputasinya diberikan pada gambar 10 dibawah



Gambar 10. hasil komputasi pada skenario dengan variasi $P_c = 0,85$.

2. Variasi probabilitas mutasi (P_m)

Probabilitas rekombinasi ditentukan sebesar 0,75, dan P_m divariasikan ke dalam tiga nilai yang telah ditentukan.

a. Skenario 13

Fungsi alih model *improper* orde 4 dengan $P_m = 0,004$ dan $P_c = 0,75$.

b. Skenario 14

Fungsi alih model *improper* orde 4 dengan $P_m = 0,008$ dan $P_c = 0,75$.

c. Skenario 15

Fungsi alih model *improper* orde 4 dengan $P_m = 0,015$ dan $P_c = 0,75$.

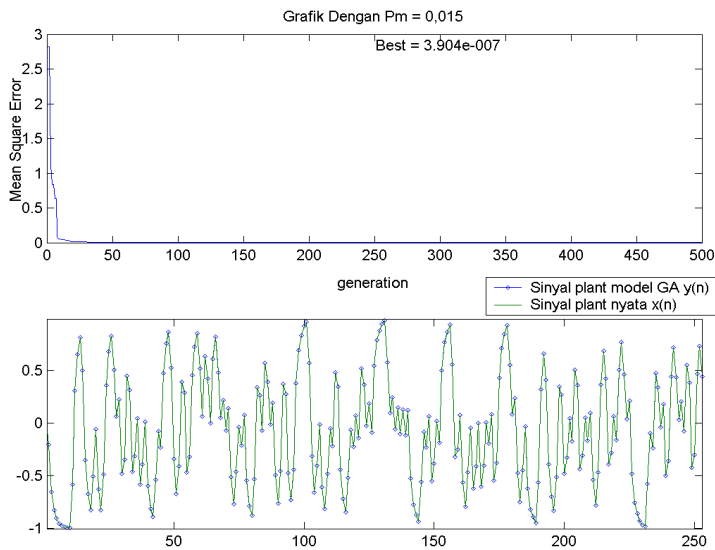
Dengan hasil pengujian sebagai berikut.

Tabel 3
Parameter Hasil Komputasi Skenario 13, 14, dan 15

Parameter	Plant	$P_m = 0,004$	$P_m = 0,008$	$P_m = 0,015$
B_0	0	$6,0678 \cdot 10^{-5}$	$1,6170 \cdot 10^{-6}$	$2,8548 \cdot 10^{-6}$
B_1	0.2050	0.2051	0.2050	0.2050
B_2	0.4222	0.3617	0.4597	0.4833
b_3	0.0924	-0.0718	0.1862	0.2299
b_4	0.0012	-0.0975	0.0483	0.0497
a_1	-0.1318	-0.4277	0.0511	0.1664
a_2	-0.1064	-0.2569	-0.0495	-0.0892
a_3	-0.0412	0.0569	-0.0911	-0.0949
a_4	0.0001	0.0246	-0.0108	-0.0141
MSE		$4,4001 \cdot 10^{-6}$	$7,8524 \cdot 10^{-7}$	$3,9040 \cdot 10^{-7}$

Dengan hasil terbaik diberikan oleh skenario pengujian 15 yang menggunakan nilai P_m sebesar 0,015 dan dapat dilihat hanya berbeda sedikit dengan pengujian dengan nilai $P_m = 0,008$. Hal ini sesuai dengan hasil percobaan yang menjadi rujukan yaitu kinerja terbaik akan didapatkan pada nilai P_m antara 0,5 % hingga $\pm 1\%$

Sedangkan grafik hasil komputasinya diberikan pada gambar 11 dibawah



Gambar 11. Hasil komputasi pada skenario dengan variasi $P_m = 0,015$.

3. Pengujian berdasarkan variasi jumlah individu per populasi

Tujuan skenario pengujian ini adalah mengetahui dan nantinya dapat menyimpulkan pengaruh pemberian variasi jumlah individu per populasi pada kinerja perangkat lunak ini dalam mengidentifikasi parameter dari sistem atau *plant* yang diberikan.

a. Skenario 16

Dimana populasi yang digunakan dalam pengujian ini adalah populasi berukuran kecil yang berisi individu sekitar 30 individu.

b. Skenario 17

Dimana populasi yang digunakan dalam pengujian ini adalah populasi yang berisi individu dengan jumlah 60 individu.

c. Skenario 18

Dimana populasi yang digunakan adalah populasi berukuran besar yang berisi individu sebanyak 100 individu.

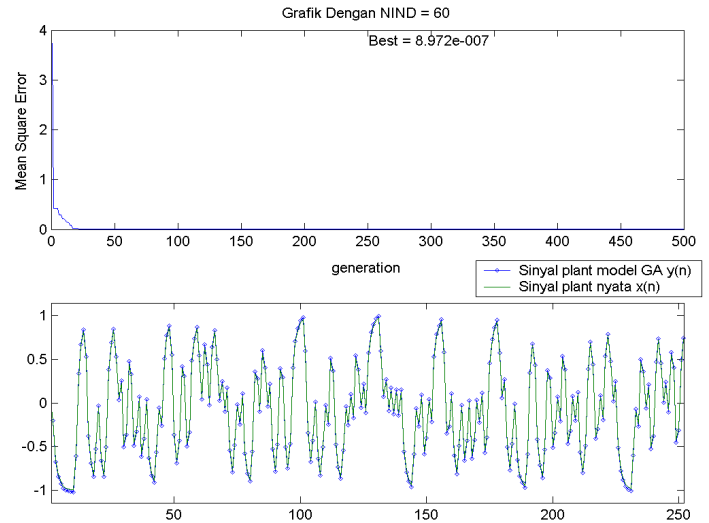
Berikut di bawah ini adalah tabel yang menjelaskan secara lengkap skenario pengujian beserta nilai – nilai parameter yang digunakan

Tabel 2

Parameter Hasil Komputasi Skenario 16, 17, dan 18

Parameter	Plant	NIND = 30	NIND = 60	NIND = 100
b_0	0	$-7.9904 \cdot 10^{-5}$	$3.0384 \cdot 10^{-6}$	$1.6213 \cdot 10^{-4}$
b_1	0.2050	0.2051	0.2050	0.2049
b_2	0.4222	0.4849	0.4881	0.3801
b_3	0.0924	0.1252	0.2454	0.1210
b_4	0.0012	-0.1416	0.0622	0.1996
a_1	-0.1318	0.1728	0.1895	-0.3380
a_2	-0.1064	-0.6112	-0.0644	0.4885
a_3	-0.0412	0.1042	-0.1078	-0.2147
a_4	0.0001	0.0050	-0.0161	-0.0262
MSE		$2.9517 \cdot 10^{-5}$	$8.9720 \cdot 10^{-7}$	$3.2827 \cdot 10^{-5}$

Hasil komputasi terbaik diberikan oleh skenario pengujian 17 yang menggunakan nilai NIND sebesar 60 individu per populasi. Hal ini sesuai dengan rujukan yang menyatakan bahwa kinerja terbaik didapatkan pada jumlah individu perpopulasi sebesar 50 hingga 100 individu. Sedangkan grafik hasil komputasinya diberikan pada gambar 12 dibawah



Gambar 12. Hasil komputasi pada skenario dengan variasi NIND = 60

VI. PENUTUP

Kesimpulan

Setelah melakukan serangkaian ujicoba serta pengujian terhadap perangkat lunak pengidentifikasi parameter sistem dengan metode algoritma genetik dalam mengenali *plant* simulasi berorde tiga dan empat, didapatkan kesimpulan sebagai berikut.

1. Algoritma genetik dapat digunakan dalam aplikasi pengidentifikasian parameter fungsi alih dari suatu sistem atau obyek berupa *plant* yang akan dikendalikan.
2. Berdasarkan hasil pengujian skenario, dapat diambil suatu resume akhir mengenai sifat atau perilaku komputasi dari pengidentifikasi GA ini, yang antara lain:
 - a. Model fungsi alih yang memberikan hasil identifikasi terbaik adalah model improper, dimana orde numerator sama dengan denumeratornya.
 - b. Kinerja algoritma genetik terbaik didapatkan pada probabilitas rekombinasi diatas 0,5 atau tepatnya berada pada rentang 0,6 hingga 1.
 - c. Sedangkan untuk probabilitas mutasi, kinerja terbaik didapatkan pada nilai P_m antara 0,5 % hingga $\pm 1\%$ dan makin besar nilainya, makin baik.
 - d. Kinerja terbaik didapatkan pada jumlah individu perpopulasi sebesar 50 hingga 100 individu. Namun hasil terbaik pada pengujian ini didapatkan pada angka sebesar 60 individu perpopulasi.
3. Kemampuan pengidentifikasian sistem dengan metode algoritma genetik dalam mengenali parameter fungsi alih

suatu *plant* dapat dibilang cukup baik, ini terbukti dari nilai parameter hasil identifikasi yang mendekati harga parameter sebenarnya.

4. Algoritma genetik, sesuai dengan fungsi objektifnya ditujukan untuk meminimasi harga *error* (MSE), dan bukan secara langsung mencari nilai parameter fungsi alih. Sehingga walaupun *error* yang didapat sudah sangat kecil atau mendekati nol, nilai parameter yang dihasilkan tetap saja dapat berbeda dengan nilai acuan. Terlebih lagi jika rentang pencarian yang digunakan sedemikian lebar.
5. Harga – harga parameter genetik seperti generasi maksimum, jumlah individu perpopulasi, Pc, Pm, dan lainnya yang cocok dengan suatu aplikasi GA bisa jadi memiliki nilai yang sangat berbeda dengan aplikasi lainnya. Sehingga harus diadakan pengujian untuk mendapatkan komposisi nilai – nilai parameter genetik yang sesuai.
6. Pemberian model fungsi alih serta harga parameter genetik yang sesuai sangat mempengaruhi kinerja dari perangkat lunak pengidentifikasi parameter sistem ini.

6.2 Saran

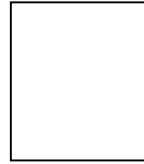
Kesimpulan lain yang dapat dijadikan saran dalam usaha mengembangkan perangkat lunak pengidentifikasi sistem ini sehingga menjadi lebih aplikatif, antara lain:

1. Untuk mendapatkan kinerja perangkat lunak serta parameter hasil identifikasi yang lebih baik, maka dapat digunakan operator genetik tingkat lanjut lain seperti dominasi, pemisahan dan algoritma genetik dinamis multiresolusi, yang memang dapat digunakan dalam aplikasi yang membutuhkan rentang pencarian lebar dan ukuran populasi besar.
2. Sebenarnya, perangkat lunak ini sudah bisa diterapkan pada keadaan nyata, namun dengan tambahan persyaratan, yaitu pengambilan data masukan –keluaran, sebaiknya dilakukan secara *off-line* dan harus dilakukan ujicoba (*trial and error*) secara intensif terhadap pengaruh batasan (*constrain*) nilai dari parameter yang akan dicari hingga didapatkan hasil yang baik, berupa nilai MSE yang minimal.
3. Sesuai dengan hasil pengujian yang telah dilakukan, maka, sebaiknya pada praktek pengidentifikasian sistem, menggunakan fungsi alih dengan model *improper*.
4. Karena hubungan antara parameter – parameter genetik dalam memberikan kinerja yang baik bersifat kompleks, maka harus dilakukan serangkaian ujicoba secara terus menerus untuk mendapatkan komposisi terbaik dari parameter genetik.

DAFTAR PUSTAKA

1. Andersson, M: *Matlab Tool for Rapid Process Identification and PID Design*, Master Thesis, Department of Automatic Control, Lund Institute of Technology, September 2000.
2. Bonadr, Renaldi : *Perbandingan Kinerja Algoritma LMS dan Algoritma Genetik Untuk Filter Adaptif Penghilang Noise*, Tugas Akhir, Universitas Diponegoro, Semarang, 2002.
3. Carlos Manuel Mira da Fonseca : *Multiobjective Genetic Algorithms with Application to Control Engineering Problems*, Ph.D Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield 1995.
4. Chipperfield, A., Fleming, P., Pohlheim, H., Fonseca, C : *Genetic Algorithm Toolbox for Use with MATLAB*, Version 1.2 User's Guide, Department of Automatic Control and Systems Engineering, University of Sheffield.
5. Davis, Lawrence E.D : *Hand Book of Genetic Algorithm*, Van Nostrand Reinhold, New York, 1991.
6. Downing, C.J., Byrne, B., Coveney, K., Marnane, W.P : *Controller Optimization dan System Identification using Genetic Algorithms*, Department of Electronics and Computer Science, University of Southampton.
7. DSG Pollock : *Handbook Time Series Analysis, Signal Processing and Dynamics*, Academic Press 1999.
8. Flockton, S.J., and White, M.S : *Pole-zero System Identification Using Genetic Algorithms*, In S. Forrest, Editor, Proceedings of Fifth International Conference on Genetic Algorithms, University of Illinois at Urbana-Champaign, pp: 531-535, Morgan Kaufman, 1993.
9. Goldberg, David E : *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.
10. Hadi Saadat : *Computational Aids in Control Systems Using MATLAB*, McGraw-Hill, Inc 1993.
11. Hariyono A. Tjokronegoro : *Identifikasi Parameter Sistem*, Jurusan Teknik Fisika , Institut Teknologi Bandung, 1996.
12. Katsuhiko Ogata : *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ.
13. Kumpati S. Narendra and Anwadha M. Annaswamy : *Stable Adaptive System*, Prentice-Hall, 1989.
14. Lennart Ljung : *System Identification : Theory for the User*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

15. Man, K.F., Tang, K.S., and Kwong, S : *Genetic Algorithms: Concepts and Applications*, IEEE Transactions on Industrial Electronics, Vol. 43, No.5, October 1996, pp: 519-534.
16. Man, K.F., Tang, K.S., Kwong, S., and Halang, W.A : *Genetic Algorithms for Control and Signal Processing Advances in Industrial Control*, Springer, London, 1997.
17. MathWorks : *The Student Edition of MATLAB High-Performance Numeric Computation and Visualization Software*, Version 4 User's Guide, Prentice-Hall, Englewood Cliffs, NJ, 1995.
18. Nailul Marom Kurniawan : *Penalaran Parameter Pengendali PID Dengan Logika Fuzzy*, Tugas Akhir, Universitas Diponegoro, Semarang 2000.
19. Reeves C.R, Dai P, and Burnham K.J : *A Hybrid Genetic Algorithm for System Identification*, Control Theory and Applications Centre, Coventry University, UK.
20. Richard H Middleton and Graham C Goodwin : *Digital Control and Estimation : a Unified Approach*, Prentice-Hall, International Edition, 1990.
21. Torsten Soderstrom, and Petre Stoica : *System Identification*, Prentice-Hall international, 1989.
22. Wright, Alden H : *Genetic Algorithms for Real Parameter Optimization*, Department of Computer Science, University of Montana, Missoula , Montana.



Jody Roostandy, lahir di Jakarta, pada 8 September 1978. Sejak tahun 1996 penulis berkesempatan menjadi mahasiswa S1 Jurusan Teknik Elektro Universitas Diponegoro Semarang, dan mengambil konsentrasi di Bidang Kontrol.

Semarang, Mei 2003

Mengetahui/Mengesahkan
Pembimbing I :

Mengetahui/Mengesahkan
Pembimbing II :

Sumardi, ST. MT.
NIP. 132 125 670

Trias Andromeda, ST. MT.
NIP. 132 231 134