



**PENERAPAN ALGORITMA GENETIKA
PADA PEYELESAIAN SISTEM PERSAMAAN LINIER**

SKRIPSI

Oleh

**Zulfi Azhari
NIM 111810101052**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**



**PENERAPAN ALGORITMA GENETIKA
PADA SISTEM PERSAMAAN LINIER**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

oleh

Zulfi Azhari
NIM 111810101052

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**

PERSEMBAHAN

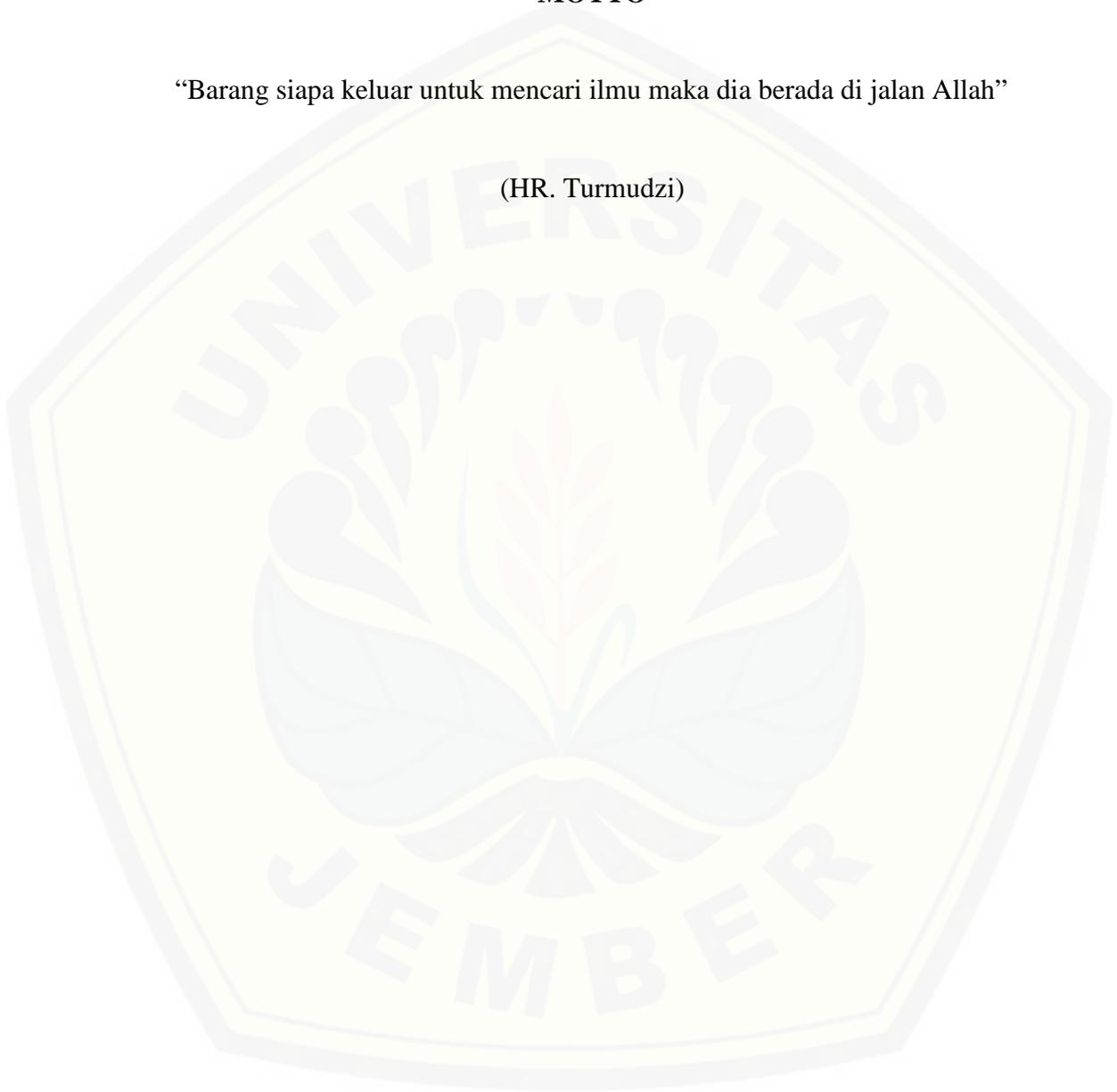
Skripsi ini saya persembahkan untuk:

1. Allah SWT yang telah memberikan kehidupan sempurna ini;
2. Bapak Abd. Haqi dan Ibunda Ririn, yang telah memberikan doa, cinta dan kasih sayangnya dalam perjalanan hidupku;
3. guru-guru sejak taman kanak-kanak sampai perguruan tinggi, yang telah memberikan ilmu dan membimbing dengan penuh kesabaran;
4. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

MOTTO

“Barang siapa keluar untuk mencari ilmu maka dia berada di jalan Allah”

(HR. Turmudzi)



<http://uzumet.blogspot.com/2014/12/motto-hidup-berdasarkan-al-quran-dan.html>

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Zulfi Azhari

NIM : 111810101052

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma Genetika Pada Sistem Persamaan Linier” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juni 2016

Yang menyatakan,

Zulfi Azhari

NIM 111810101052

SKRIPSI

**PENERAPAN ALGORITMA GENETIKA
PADA SISTEM PERSAMAAN LINIER**

oleh

Zulfi Azhari

NIM 111810101052

Pembimbing:

Dosen Pembimbing Utama : M. Ziaul Arif S. Si., M. Sc

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni S. Si., M. Kom

PENGESAHAN

Skripsi berjudul “Penerapan Algoritma Genetika Pada Sistem Persamaan Linier”
telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember.

Tim Penguji:

Ketua,

M. Ziaul Arif, S. Si., M. Sc
NIP. 1985011 120081 2 100 2

Penguji I,

Prof. Drs. Kusno, DEA.,Ph.D
NIP. 19610108 198602 1 001

Sekretaris,

Ahmad Kamsyakawuni, S. Si., Kom
NIP. 19721129 199802 100 1

Penguji II,

Drs. Rusli Hidayat, M. Sc
NIP. 19661012 199363 1 001

Mengesahkan

Dekan,

Drs. Sujito, Ph.D
NIP 19610108 198602 1 001

RINGKASAN

Penerapan Algoritma Genetika Pada Peyelesaian Sistem Persamaan Linier;
Zulfi Azhari; 111810101052; 2016; 41 Halaman; Jurusan Matematika Fakultas
Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Pencarian solusi Sistem Persamaan Linier (SPL) merupakan masalah yang sering muncul pada analisis numerik. Terdapat 2 metode yang biasanya digunakan yaitu metode eksak dan metode iterasi. Beberapa contoh dari metode eksak adalah Eliminasi Gauss, Eliminasi Gauss–Jordan, Faktorisasi LU dan lainnya. Metode iterasi yang paling populer adalah Gauss–Seidel, Jacobi, Newton, Conjugate Gradient dan Broyden .

Algoritma Genetika adalah sebuah metode untuk menyelesaikan masalah optimasi berdasarkan kecerdasan buatan (*Artificial Intelligence*). Algoritma Genetika menggunakan konsep evolusi dengan cara mengembangkan generasi-generasi dari populasi solusi dengan menggunakan nilai *fitness* untuk menunjukkan nilai solusi terbaik pada suatu masalah.

Penelitian ini bertujuan untuk mengetahui cara menyelesaikan permasalahan Sistem Persamaan Linier dengan Algoritma Genetika dan mengetahui perbandingan hasil solusi dari Sistem Persamaan Linier dengan algoritma genetika jika dibandingkan dengan metode eksak seperti Gauss–Jordan dan metode numerik iteratif seperti Gauss–Seidel. Sistem Persamaan Linier yang dipakai dalam penelitian ini diambil dari beberapa sumber referensi. Terdapat beberapa jenis Sistem Persamaan Linier antara lain Sistem Persamaan Linier yang berukuran n , m , dan *sparse*.

Penelitian ini dilakukan melalui beberapa langkah, yaitu Studi literatur, menentukan masalah, menentukan masukan dan luaran, membuat program, implementasi algoritma genetika, analisis hasil, dan mengambil kesimpulan. Langkah-langkah penerapan algoritma genetika pada Sistem Persamaan Linier adalah inisialisasi parameter, pembentukan populasi awal, menghitung nilai *fitness*,

menyeleksi solusi, melakukan proses reproduksi yaitu *crossover* dan mutasi, dimana pada proses *crossover* dan mutasi algoritma genetika berjalan paling optimal saat *crossover rate* bernilai 0,9 dan *mutation rate* bernilai 0,1. Selanjutnya proses yang terakhir yaitu mengevaluasi solusi, dan mengulangi tahapan-tahapan tersebut hingga kriteria pemberhentian terpenuhi.

Berdasarkan hasil penelitian yang dilakukan beberapa kali pengujian diperoleh hasil yang nilainya hampir sama dengan metode eksak, metode iteratif, dan hasil dari sumber referensi pada beberapa jenis SPL. Algoritma Genetika dapat memperoleh solusi yang dapat diterima, hal ini dapat dilihat dari nilai MSE solusi yang sangat kecil.

PRAKATA

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga tugas akhir yang berjudul “Penerapan Algoritma Genetika Pada Sistem Persamaan Linier” dapat terselesaikan dengan baik. Tugas akhir ini disusun untuk memenuhi syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis menyampaikan terima kasih kepada:

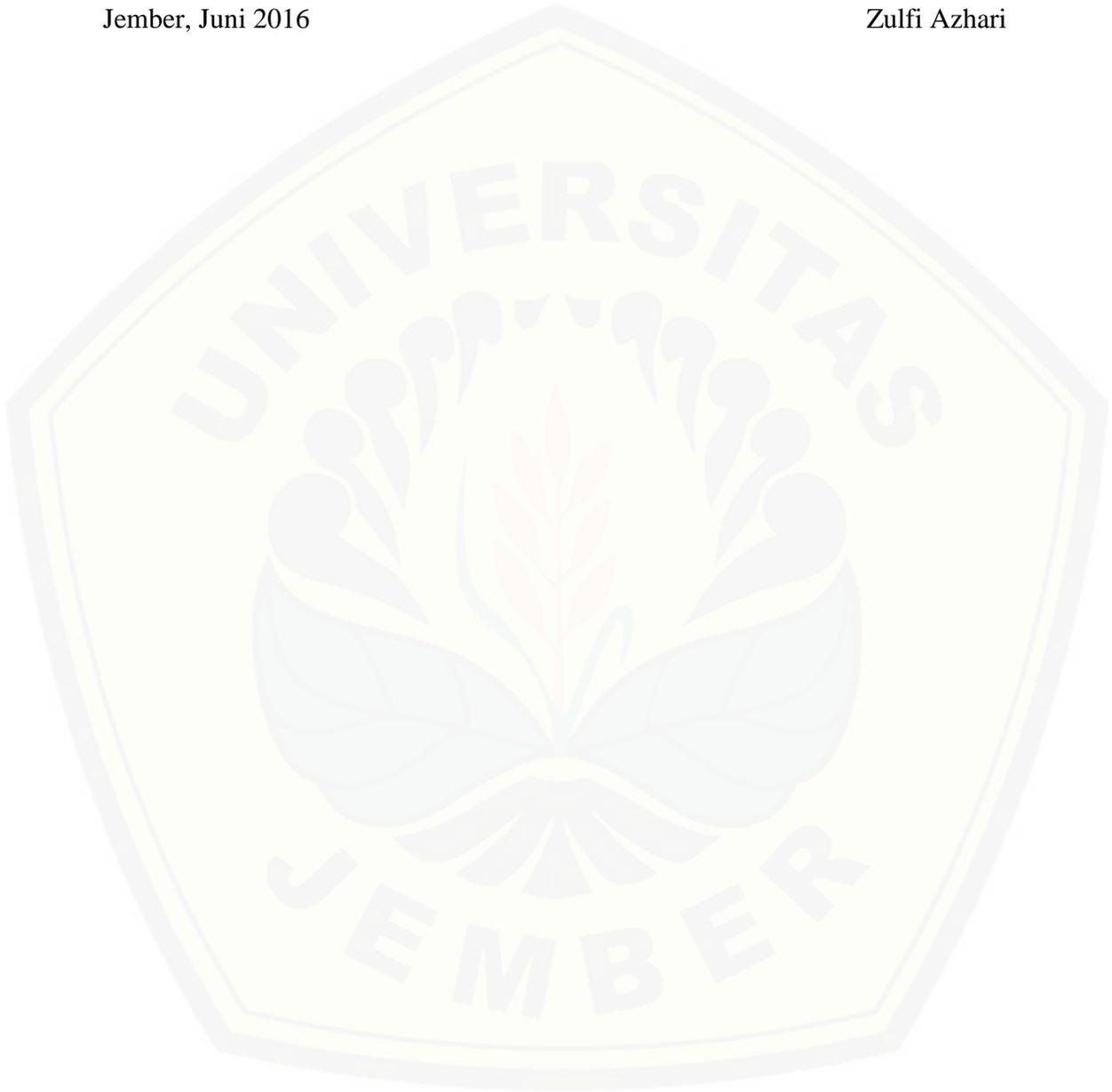
1. Drs. Drs. Sujito, Ph.D selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah mengesahkan penulisan skripsi ini;
2. M. Ziaul Arif, S. Si, M. Sc selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni S. Si, M. Kom selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Prof. Drs. Kusno, DEA., Ph.D dan Drs. Rusli Hidayat, M. Sc selaku Dosen Penguji yang telah memberikan kritik dan saran sehingga skripsi ini menjadi lebih baik lagi;
4. Seluruh dosen dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam;
5. Keluarga besar KRAMAT '11 yang selalu memberikan dukungan dalam hal apapun.
6. Semua pihak terkait yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam menyusun skripsi ini masih terdapat kekurangan baik isi maupun susunannya. Oleh karena itu, penulis mengharapkan saran dan kritik

demi penyempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat memberi manfaat dan sumbangan bagi pembaca.

Jember, Juni 2016

Zulfi Azhari



DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN	v
HALAMAN PEMBIMBING	vi
HALAMAN PENGESAHAN	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Persamaan dan Sistem Persamaan	4
2.1.1 Sistem Persamaan Linier (SPL)	4
2.1.1 Sistem Persamaan Non Linier (SPNL).....	6
2.2 Algoritma Genetika Pada Sistem Persamaan Linier	7
2.2.1 Nilai <i>Fitness</i>	8
2.2.2 Seleksi Induk	9
2.2.3 Pindah Silang (<i>Crossover</i>)	10

2.2.4 Mutasi	11
2.2.5 Kriteria pemberhentian.....	12
2.3 Metode Numerik	13
2.3.1 Gauss-Jordan	13
2.3.2 Gauss-Seidel.....	14
2.3 Mean Square Erroe (MSE).....	14
BAB 3. METODE PENELITIAN.....	14
BAB 4. HASIL DAN PEMBAHASAN.....	15
4.1 Hasil.....	15
4.2 Pembahasan.....	21
4.2.1 Program	21
4.2.2 Penyelesaian SPL dengan Algoritma Genetika.....	23
4.2.3 Pengujian Beberapa Parameter	29
4.2.4 Analisis Hasil	31
BAB 5. PENUTUP.....	38
5.1 Kesimpulan.....	38
5.2 Saran	38
DAFTAR PUSTAKA	40
LAMPIRAN.....	42

DAFTAR TABEL

	Halaman
2.1 Contoh Nilai <i>Fitness</i>	9
2.2 Contoh <i>Tournament Selection</i>	9
4.1 Persamaan yang Dipakai	19
4.2 Hasil Perhitungan	20
4.3 Populasi Awal	24
4.4 Nilai <i>Fitness</i>	25
4.5 <i>Tournament Selection</i>	26
4.6 Hasil Seleksi	26
4.7 Evaluasi Kromosom Hasil Reproduksi	28
4.8 Kromosom Baru	29
4.9 Uji <i>Pc</i> dan <i>Pm</i>	29
4.10 Rata-rata Waktu Komputasi	30

DAFTAR GAMBAR

	Halaman
2.1 Contoh <i>One-Point Crossover</i>	10
2.2 Contoh Proses Mutasi	11
2.3 <i>Flowchart</i> Algoritma Genetika.....	12
3.1 Skema Metode Penelitian.....	16
4.1 Tampilan Akhir Program	23
4.2 Hasil <i>Crossover</i>	27
4.3 Hasil Mutasi.....	27
4.4 Kurva Kekonvergenan Nilai <i>fitness</i> Sistem Persamaan Limier 1	32
4.5 Kurva Kekonvergenan Nilai <i>fitness</i> Sistem Persamaan Limier 2.....	33
4.6 Kurva Kekonvergenan Nilai <i>fitness</i> Sistem Persamaan Limier 3	34
4.7 Kurva Kekonvergenan Nilai <i>fitness</i> Sistem Persamaan Limier 4.....	34
4.8 Kurva Kekonvergenan Nilai <i>fitness</i> Sistem Persamaan Limier 5	35
4.9 Kurva Kekonvergenan Nilai <i>fitness</i> Sistem Persamaan Limier 6.....	36

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pencarian solusi Sistem Persamaan Linier (SPL) merupakan masalah yang sering muncul pada analisis numerik. Masalah ini telah dikembangkan sepanjang waktu oleh ahli matematika dan ahli komputer. Sebagai buktinya adalah banyak metode-metode numerik dikembangkan untuk menyelesaikannya. Terdapat 2 metode yang biasanya digunakan yaitu metode eksak dan metode iterasi. Metode eksak digunakan ketika ukuran sistem persamaannya kecil. Beberapa contoh dari metode ini adalah Eliminasi Gauss, Eliminasi Gauss–Jordan, Faktorisasi LU dan lainnya. Metode iterasi dipakai ketika ukuran sistem besar. Metode iterasi yang paling populer adalah Gauss–Seidel, Jacobi, Newton, Conjugate Gradient dan Broyden.

SPL sering muncul pada berbagai bidang ilmu pengetahuan sosial dan eksakta untuk menemukan solusi dari beberapa masalah. Ada beberapa algoritma numerik untuk menyelesaikan Sistem Persamaan Linier (SPL) yang berdasarkan beberapa prinsip teoritis dari sebuah matrik. Menemukan solusi dari suatu himpunan persamaan melalui proses Algoritma Genetika (AG) adalah suatu hal yang baru dalam pengembangan metode pada bidang riset dan teknologi yang sudah diteliti oleh Abiodun *et al* pada tahun 2011.

Algoritma Genetika adalah sebuah metode untuk menyelesaikan masalah optimasi berdasarkan kecerdasan buatan (*Artificial Intelligence*). Konsep dari Algoritma Genetika diperkenalkan oleh John Holland pada 1975 dengan tujuan membuat komputer bisa melakukan seperti sifat dasar makhluk hidup. Ide dari Algoritma Genetika berasal dari teori Evolusi yang menyatakan bahwa hanya spesies yang mampu bertahan hidup yang bisa menghasilkan keturunan. John Holland pada jurnalnya yang berjudul *Adaptation in Natural and Artificial Systems* menyatakan

bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi dan operasi genetika atas kromosom. Algoritma Genetika menggunakan konsep evolusi dengan cara stokastik dalam mengembangkan generasi-generasi dari populasi solusi dengan menggunakan nilai *fitness* untuk menunjukkan nilai solusi terbaik pada suatu masalah.

Algoritma Genetika telah banyak diterapkan dalam berbagai masalah seperti penelitian yang dilakukan oleh Ariastuti (2015) dengan judul Penerapan Algoritma Genetika dan Algoritma *Harmony Search* pada Permasalahan *Knapsack* 0-1. Dalam penelitian tersebut disimpulkan bahwa algoritma genetika lebih cepat konvergen jika dibandingkan dengan algoritma *Harmony Search*. Selanjutnya, penelitian Hanggraeni (2014) yang berjudul Perbandingan Algoritma *Harmony Search* dan Genetika pada Penjadwalan *Jobshop* menyimpulkan bahwa Algoritma *Harmony Search* dan Genetika memiliki tingkat efisiensi yang sama. Dari beberapa referensi tersebut dapat disimpulkan bahwa Algoritma Genetika efisien dalam menyelesaikan beberapa macam permasalahan optimasi.

Terdapat beberapa keunggulan dari aplikasi Algoritma Genetika dalam proses penentuan solusi dari suatu masalah optimasi. Pertama, Algoritma Genetika tidak memerlukan persyaratan matematika dalam penyelesaian proses optimasi. Kedua, Algoritma Genetika dapat diaplikasikan pada beberapa jenis fungsi obyektif dengan beberapa fungsi pembatas berbentuk linier. Selanjutnya, Operasi evolusi dari Algoritma Genetika sangat efektif untuk mengobservasi posisi global secara acak dan mempunyai fleksibilitas untuk diimplementasikan secara efisien pada problematika tertentu.

Problematika dalam metode numerik misalnya pencarian solusi Sistem persamaan linier dapat dipandang sebagai kasus optimasi. Kasus optimasi tersebut merupakan masalah optimasi minimal yang solusi minimumnya merupakan solusi bagi SPL tersebut. Berdasarkan keunggulan algoritma genetika diatas, penulis tertarik

melakukan penelitian mengenai penerapan Algoritma Genetika untuk menyelesaikan suatu sistem persamaan linier.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka rumusan masalah pada penelitian ini adalah:

- a. Bagaimana cara menyelesaikan permasalahan Sistem Persamaan Linier dengan Algoritma Genetika?
- b. Bagaimana hasil solusi dari Sistem Persamaan Linier dengan algoritma genetika jika dibandingkan dengan metode eksak seperti Gauss–Jordan, metode numerik iteratif seperti Gauss-Seidel, dan dari sumber referensi ?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini antara lain:

- a. Mengetahui cara menyelesaikan permasalahan Sistem Persamaan Linier dengan Algoritma Genetika
- b. Mengetahui perbandingan hasil solusi dari Sistem Persamaan Linier dengan algoritma genetika jika dibandingkan dengan metode eksak seperti Gauss–Jordan, metode numerik iteratif seperti Gauss–Seidel, dan dari sumber referensi.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah dapat memberi pengetahuan tentang algoritma genetika sebagai salah satu cara untuk mencari solusi Sistem Persamaan Linier. Selain itu, penelitian ini juga dapat digunakan sebagai referensi dan wawasan baru bagi peneliti lainnya tentang solusi sistem persamaan linier.

BAB 2. TINJAUAN PUSTAKA

2.1 Persamaan dan Sistem Persamaan

Persamaan adalah kalimat terbuka yang menggunakan relasi “=”, sedangkan apabila menggunakan relasi “<,>,<=, atau >=” dinamakan pertidaksamaan. Persamaan linier merupakan sebuah persamaan yang mengandung variabel berpangkat satu, contohnya $2x + 3y - 4z = 3$. Sedangkan persamaan non linier adalah persamaan yang mengandung variabel berderajat tidak sama dengan satu atau mengandung nilai fungsi non linear, atau fungsi transenden, seperti contoh berikut :

- a. $\log 2x + \log 6y = 23$
- b. $\cos 23x + \tan y = 120$
- c. $2x^2 - 3y^3 = 10$

Jika terdapat dua atau lebih persamaan yang saling terkait, maka disebut dengan sistem persamaan. Sistem persamaan secara umum dibagi menjadi dua, yaitu SPL dan Sistem Persamaan Non-Linier (SPNL). Sistem Persamaan Linier (SPL) adalah suatu kumpulan dari 2 atau lebih persamaan linier yang saling berhubungan dan Sistem Persamaan Non Linier (SPNL) merupakan kumpulan dari 2 atau lebih persamaan non linier yang saling berhubungan (Markaban, 2004). Pada penelitian ini, peneliti menfokuskan permasalahan pada SPL.

2.1.1 Sistem Persamaan Linier (SPL)

Masalah SPL sering dijumpai dalam kehidupan nyata maupun dalam simulasi numerik. Masalah ini memunculkan banyak metode untuk menyelesaikannya seperti eliminasi Gauss, metode Newton-Raphson dan lain sebagainya.

SPL adalah kumpulan dari 2 atau lebih persamaan linier dengan himpunan penyelesaian yang tidak diketahui. Dalam menyelesaikan SPL, akan dicari suatu nilai yang akan memenuhi setiap persamaan di dalam sistem tersebut.

Bentuk umum dari SPL dengan m persamaan dan n variabel adalah sebagai berikut:

$$\begin{array}{cccccc}
 a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\
 a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\
 \vdots & & \vdots & & & & \vdots & & \vdots \\
 a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m
 \end{array} \tag{2.1}$$

x_1, \dots, x_n adalah variabel yang tidak diketahui, a_{11}, \dots, a_{mn} adalah koefisien pada sistem persamaan dan b_1, \dots, b_m adalah konstanta ruas kanan.

Bentuk diatas dapat ditulis dalam bentuk persamaan matrik $AX = b$ dengan A adalah matrix $m \times n$, X adalah kolom vektor yang berisi n buah masukan, dan b adalah kolom vektor dengan isi m konstanta ruas kanan.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Sebuah solusi dari sistem persamaan linier adalah penempatan nilai pada variabel x_1, \dots, x_n sehingga setiap persamaan terpenuhi. Himpunan dari semua solusi yang mungkin disebut himpunan solusi. SPL bisa memiliki satu dari tiga kemungkinan berikut :

- a. SPL memiliki solusi tak hingga banyaknya, yaitu terdapat lebih dari satu nilai X yang membuat $f(X) = 0$, contohnya SPL dalam bentuk matrik $n \times m$, dengan m persamaan dan n variable, $n > m$.

$$x_1 + x_2 + x_3 = 7$$

$$x_1 + 2x_2 + 3x_3 = 8$$

- b. SPL memiliki solusi yang unik. Hanya ada satu nilai X yang membuat $f(X) = 0$.

c. SPL tidak memiliki solusi, yaitu tidak ada nilai X yang membuat $f(X) = 0$.

(Abiodun *et al*, 2011)

Secara umum, jika matrik berukuran relatif kecil maka solusi eksak bisa ditemukan. Namun jika SPL itu besar, sulit ditentukan solusi eksaknya. Maka metode numerik bisa diterapkan dalam mencari solusi pendekatannya. Akan tetapi metode numerik memiliki kekurangan diantaranya, bergantung dengan syarat tertentu, ukuran matrik harus persegi dan lambat jika ukuran matrik relative besar, sehingga untuk menyelesaikan masalah tersebut peneliti menerapkan algoritma genetika.

2.1.2 Sistem Persamaan Non-Linier

Munif (dalam Mulyono 2007) menyatakan jika persamaan non-linier dinyatakan dalam sumbu kartesius berupa kurva (garis melengkung). Sistem persamaan non-linier memiliki beberapa karakteristik penyelesaian, diantaranya tidak memiliki akar real, satu akar, dan lebih dari satu akar. Contohnya:

$$\begin{aligned} 2x_1^2 - 3x_2^3 + 3x_3^4 &= -1 \\ x_1^2 + 4x_2^3 + 4x_3^4 &= 13 \\ 2x_1^2 - x_2^3 + x_3^4 &= 2 \end{aligned}$$

Penyelesaian sistem ini terdiri dari himpunan nilai – nilai X yang secara simultan memenuhi seluruh persamaan tersebut.

2.2 Algoritma Genetika Pada Sistem Persamaan Linier

Menurut Abiodun *et al* (2011) Algoritma Genetika adalah sebuah metode untuk menyelesaikan masalah berdasarkan kecerdasan buatan (*Artificial Intelligence*). Konsep dari GA diperkenalkan oleh John Holland pada 1975 dengan tujuan membuat komputer yang bisa melakukan seperti sifat dasar makhluk hidup.

Pada algoritma Genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Populasi awal dibangkitkan secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi.

Sistem persamaan linier (2.1) dapat diselesaikan dengan menggunakan algoritma Genetika jika dapat diubah dalam bentuk fungsi objektif, dan solusi dikatakan benar jika dapat memenuhi semua persamaan, yaitu:

$$\begin{aligned}f_1(x_1, x_2, \dots, x_n) - a_1 &= 0 \\f_2(x_1, x_2, \dots, x_n) - a_2 &= 0 \\&\vdots \\f_n(x_1, x_2, \dots, x_n) - a_n &= 0\end{aligned}$$

Atau nilai dari $|f_1(x_1, x_2, \dots, x_n) - a_1|, |f_2(x_1, x_2, \dots, x_n) - a_2|, \dots, |f_n(x_1, x_2, \dots, x_n) - a_n|$, nilai absolutnya semuanya harus minimum atau mendekati nol.

Langkah-langkah dari algoritma Genetika secara umum dalam Abiodun *et al* (2011) adalah sebagai berikut:

- Buatlah domain dari variabel masalah sebagai kromosom dengan panjang tertentu. Tentukan ukuran populasi kromosom dan kemungkinan *crossover* ;
- Definisikan fungsi *fitness*, yang digunakan untuk mengukur kualitas setiap kromosom di domain masalah;
- Munculkan generasi populasi kromosom awal acak dengan panjang tertentu;
- Hitung fungsi *fitness* untuk setiap kromosom;
- Pilih sepasang kromosom untuk dikawinkan dari populasi kromosom induk (*parent*) dipilih berdasarkan nilai *fitness* mereka;
- Terapkan operasi genetika (*crossover* dan mutasi) untuk membuat sepasang kromosom anak (*offspring*);

- g. Tempatkan *offspring* yang telah dibuat di populasi awal;
- h. Tukar populasi kromosom awal (sebelumnya) dengan yang baru;
- i. Ulangi langkah d sampai i sampai kriteria pemberhentian terpenuhi. Kriteria pemberhentian bisa dari mendapat nilai optimal yang diketahui atau suatu tingkat solusi yang dapat diterima atau generasi maksimal telah tercapai;

2.2.1 Nilai *fitness*

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performanya yang disebut nilai *fitness*. Di dalam evolusi alam, individu yang bernilai *fitness* yang tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati

Untuk mencari nilai *fitness* kromosom pada masalah sistem persamaan linier, digunakan nilai dari kromosom-kromosom untuk mengevaluasi setiap baris dari persamaan. Hasil dari evaluasi kromosom pada setiap persamaan yang dikurangi dari nilai ruas kanan (*right hand side* (RHS)) dari persamaan aslinya disebut dengan selisih atau error. Selisih–selisih tersebut selanjutnya dijumlahkan kemudian dikuadratkan dan disebut (jumlah selisih)².

Nilai RHS pada setiap persamaan dijumlahkan dan juga di kuadratkan yang disebut (jumlah RHS)². Sehingga nilai *fitness* diperoleh dari formula berikut:

$$fitness = ((jumlah\ RHS)^2 - (jumlah\ selisih)^2) / (jumlah\ RHS)^2 \quad (2.2)$$

Persamaan 2.2 menggambarkan bahwa jika nilai *fitness* semakin mendekati 1 maka semakin baik solusi yang didapat dan jika nilai *fitness*nya sama dengan 1 maka akan didapat solusi eksak untuk sistem persamaan tersebut (Abiodun *et al*, 2011).

2.2.2 Seleksi Induk

Seleksi merupakan proses yang bertanggung jawab atas pemilihan kromosom dalam proses reproduksi. Seleksi digunakan untuk mendapatkan calon induk yang baik. Semakin tinggi nilai *fitness* suatu kromosom semakin besar

kemungkinannya untuk terpilih. Pada umumnya metode seleksi yang dipakai adalah *roulette wheel* dan *tournament selection*. Pada penelitian ini akan dipakai *tournament selection* karena metode ini menjamin bertahannya kromosom terbaik untuk direproduksi.

Pada metode *tournament selection*, ditetapkan sejumlah turnamen untuk kromosom-kromosom yang dipilih secara acak dari suatu populasi. Kromosom kromosom yang memiliki nilai *fitness* terbaik dalam turnamen ini akan diseleksi menjadi induk. Parameter yang digunakan pada metode ini adalah ukuran *tournament* yang nilainya dapat dibagi dengan jumlah kromosom.

Tabel 2.1 Contoh Nilai *Fitness*

Kromosom	<i>Fitness</i>
Kromosom 1 (K_1)	0,8
Kromosom 2 (K_2)	0,9
Kromosom 3 (K_3)	0,3
Kromosom 4 (K_4)	0,4

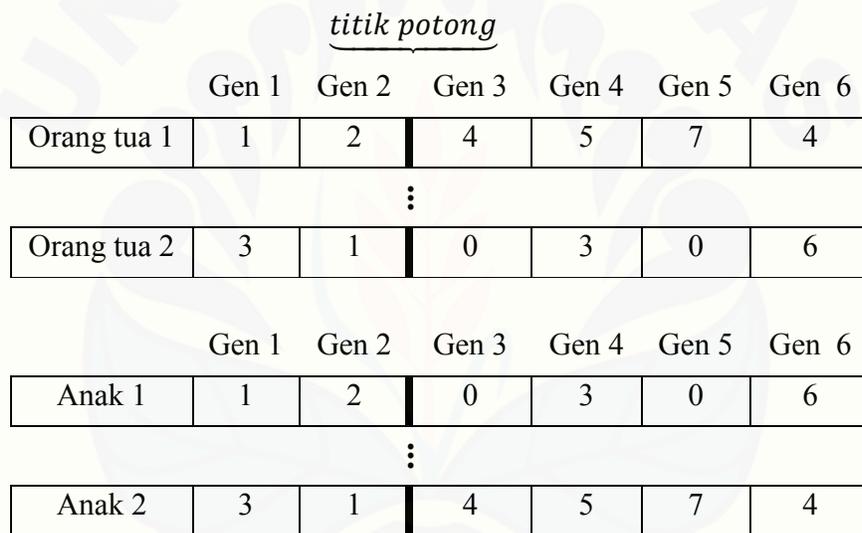
Tabel 2.2 Contoh *Tournament Selection*

	<i>Fitness</i> terpilih		Kromosom terpilih
<i>Tournament</i> 1	K_1	K_2	K_2
<i>Tournament</i> 2	K_2	K_4	K_2
<i>Tournament</i> 3	K_3	K_4	K_4
<i>Tournament</i> 4	K_1	K_3	K_1

2.2.3 Pindah Silang (*Crossover*)

Sebuah kromosom yang mengarah pada solusi yang baik dapat diperoleh dari proses memindah – silangkan dua buah kromosom. Pindah silang hanya bisa dilakukan dengan suatu probabilitas (P_c) tertentu, artinya pindah silang bisa

dilakukan hanya jika suatu bilangan random yang dibangkitkan kurang dari probabilitas yang ditentukan tersebut. Pada umumnya probabilitas tersebut ditetapkan mendekati 1. Pindah silang yang paling sederhana adalah pindah silang satu titik potong (*one - point crossover*). Suatu titik potong dipilih secara acak, kemudian bagian pertama dari orang tua 1 digabungkan dengan bagian kedua dari orang tua 2. Orang tua 1 dan orangtua 2 adalah kromosom yang memiliki bilangan random kurang P_c .



Gambar 2.2 Contoh *One-Point Crossover*

Crossover adalah operator algoritma Genetika yang utama karena beroperasi pada dua kromosom pada suatu waktu dan membentuk *offspring* dengan mengkombinasikan dua bentuk kromosom. Cara sederhana untuk memperoleh *crossover* adalah dengan memilih suatu titik yang ditempatkan secara acak dan kemudian membentuk *offspring* dengan cara mengkombinasikan segmen dari satu induk bagian kiri dari titik yang dipisahkan, dengan segmen dari induk yang lain bagian kanan dari titik yang dipisahkan. Metode ini akan berjalan normal dengan representasi bit string. Kemampuan dari algoritma Genetika bergantung pada operator

crossover yang digunakan. *Crossover rate* merupakan rasio antara jumlah *offspring* yang dihasilkan pada setiap generasi terhadap luas populasinya. Semakin tinggi *crossover rate* akan memungkinkan eksplorasi ruang solusi yang lebih luas dan mereduksi kemungkinan jatuh pada kondisi optimum yang salah. Namun dengan memberikan nilai *crossover rate* yang tinggi memberikan konsekuensi makin lamanya waktu perhitungan yang diperlukan oleh karena eksplorasi pada luas populasi yang ada. Amiolemhen (2004) menyatakan umumnya *crossover rate* yang nilainya antara 0,6 sampai 0,9 dapat bekerja dengan baik pada sebagian besar situasi.

2.2.4 Mutasi

Mutasi dapat dikatakan sebagai operasi pendukung yang menghasilkan perubahan secara acak dan seketika pada berbagai jenis kromosom. Cara mudah untuk mendapatkan mutasi dengan mengubah satu atau lebih *genes*. Pada algoritma Genetika, mutasi memainkan peran penting. Pertama, menggantikan *genes* yang hilang dari populasi selama proses seleksi, sehingga dapat diujikan pada suatu kondisi yang baru. Kedua, menyediakan *genes* yang tidak ditampilkan pada populasi awal.

Kromosom awal	2	3	0	1	1	2	3	4
Hasil mutasi	2	4	0	1	3	2	3	4

Gambar 2.3 Contoh Proses Mutasi

Mutation rate menyatakan presentase dari total jumlah *genes* dalam populasi. *Mutation rate* ini melakukan kontrol dimana *genes* baru dalam populasi dapat diuji seleksi. Jika *rate* terlalu kecil akan banyak *genes* yang sebenarnya bermanfaat tetapi tidak pernah diuji seleksi. Namun jika *rate* terlalu tinggi akan mengakibatkan *offspring* mulai kehilangan kemiripan dengan induknya dan Algoritma Genetika akan kehilangan kemampuan untuk melihat urutan langkah

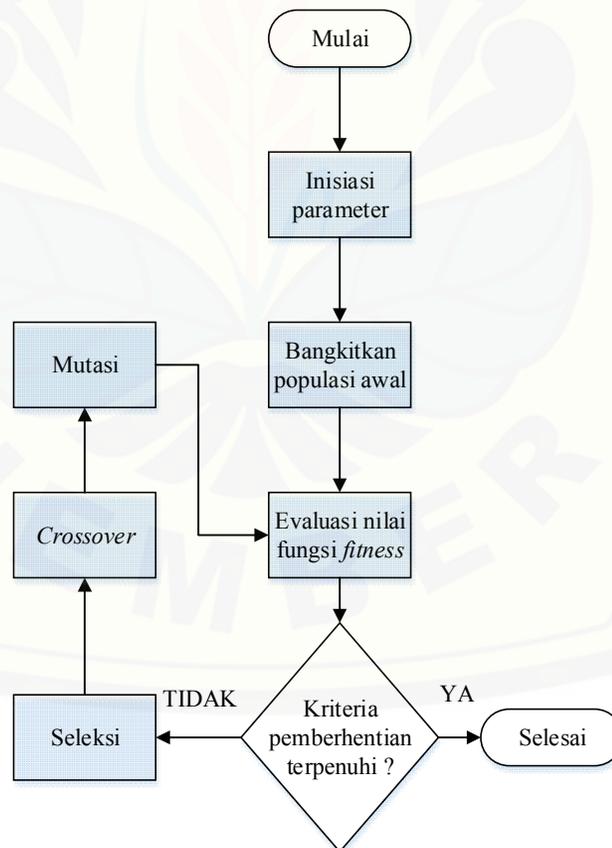
observasinya. Pada umumnya *mutation rate* nilainya ditetapkan antara 0,01 sampai 0,1 (Ali, 2009).

Setelah proses mutasi, terbentuklah suatu populasi kromosom baru yang selanjutnya akan diperiksa nilai *fitness* masing-masing kromosom. Kromosom terbaik akan dijadikan dasar pembentukan kromosom untuk iterasi berikutnya.

2.2.5 Kriteria pemberhentian

Algoritma Genetika akan berhenti jika jumlah iterasi (generasi) yang ditetapkan sudah terpenuhi.

Langkah-langkah pada algoritma Genetika dapat diilustrasikan pada *flowchart* dibawah ini:



Gambar 2.4 *Flowchart* Algoritma Genetika

2.3 Metode Numerik

Metode Numerik merupakan salah satu cabang atau bidang ilmu matematika. Metode numerik adalah teknik untuk menyelesaikan permasalahan-permasalahan yang diformulasikan secara matematis dengan cara operasi hitung aritmatika (Triatmodjo, 1992).

Menurut Santoso (2011) pada umumnya metode numerik tidak mengutamakan diperolehnya nilai yang eksak (tepat), tetapi mengusahakan perumusan metode yang menghasilkan nilai pendekatan yang berbeda dari nilai yang eksak sebesar suatu nilai yang dapat diterima berdasarkan pertimbangan praktis, tetapi cukup dapat memberikan penjelasan pada persoalan yang dihadapi. Banyak metode dalam metode numerik yang dapat digunakan untuk menyelesaikan suatu persamaan matematika. Setiap metode memiliki prosedur yang berbeda dalam menentukan nilai pendekatannya. Metode ini memberikan keefisienan dan keefektifan di dalam menyelesaikan persoalan-persoalan matematis dikarenakan berkembangnya perangkat keras dan lunak komputer akhir-akhir ini.

Beberapa metode yang umumnya dipakai dalam menyelesaikan sistem persamaan linier antara lain metode Eliminasi Gauss, metode Thomas, metode Gauss-Jordan, dan metode Gauss-Seidel.

Beberapa kelemahan dari metode numerik iteratif antara lain lambat dalam menyelesaikan sistem persamaan linier besar, memerlukan persyaratan-persyaratan khusus untuk metode-metode tertentu, dan hanya dapat menyelesaikan matrik $n \times n$, jika ukuran matrik $m \times n$ maka metode tersebut gagal mencapai kekonvergenan.

2.3.1 Gauss-Jordan

Menurut Sahid (2005) pada eliminasi Gauss Jordan dibuat nol elemen-elemen dibawah maupun di atas diagonal suatu matrik. Hasilnya adalah matriks tereduksi yang berupa matrik diagonal satuan (semua elemen pada diagonal utama bernilai 1, elemen elemen lainnya bernilai 0). Dengan demikian, solusi SPL yang bersangkutan dapat dibaca pada kolom terakhir matriks *augmented*, tanpa perlu

melakukan perhitungan lagi. Metode ini memerlukan lebih banyak operasi dari pada eliminasi Gauss, selama proses reduksi matriks. Akan tetapi setelah itu tidak lagi memerlukan operasi hitung untuk mendapatkan penyelesaian SPL. Dengan demikian metode eliminasi Gauss-Jordan lebih efisien untuk menyelesaikan sebuah SPL. Dalam Matlab metode Gauss Jordan dapat digunakan dengan perintah $rref(A)$, dimana A adalah sebuah SPL berbentuk matrik dan b adalah vektor RHS dari SPL

2.3.2 Gauss-Seidel

Metode iterasi Gauss-Seidel digunakan untuk menyelesaikan SPL berukuran besar. Gauss-Seidel konvergen untuk setiap SPL yang memiliki matriks koefisien bersifat dominan secara diagonal. Suatu SPL yang tidak bersifat dominan secara diagonal, mungkin dapat disusun ulang menjadi sedemikian sehingga iterasi gauss seidel akan konvergen. Dalam metode Gauss-Seidel nilai error dapat diperkecil karena dapat melakukan iterasi sampai solusinya seteliti mungkin.

Pada Sahid (2005) rumus iterasi untuk hampiran ke- k pada metode Gauss-Seidel adalah sebagai berikut:

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k-1)} \right)$$

Untuk $i = 1, 2, \dots, n$ dan $k = 1, 2, 3, \dots$

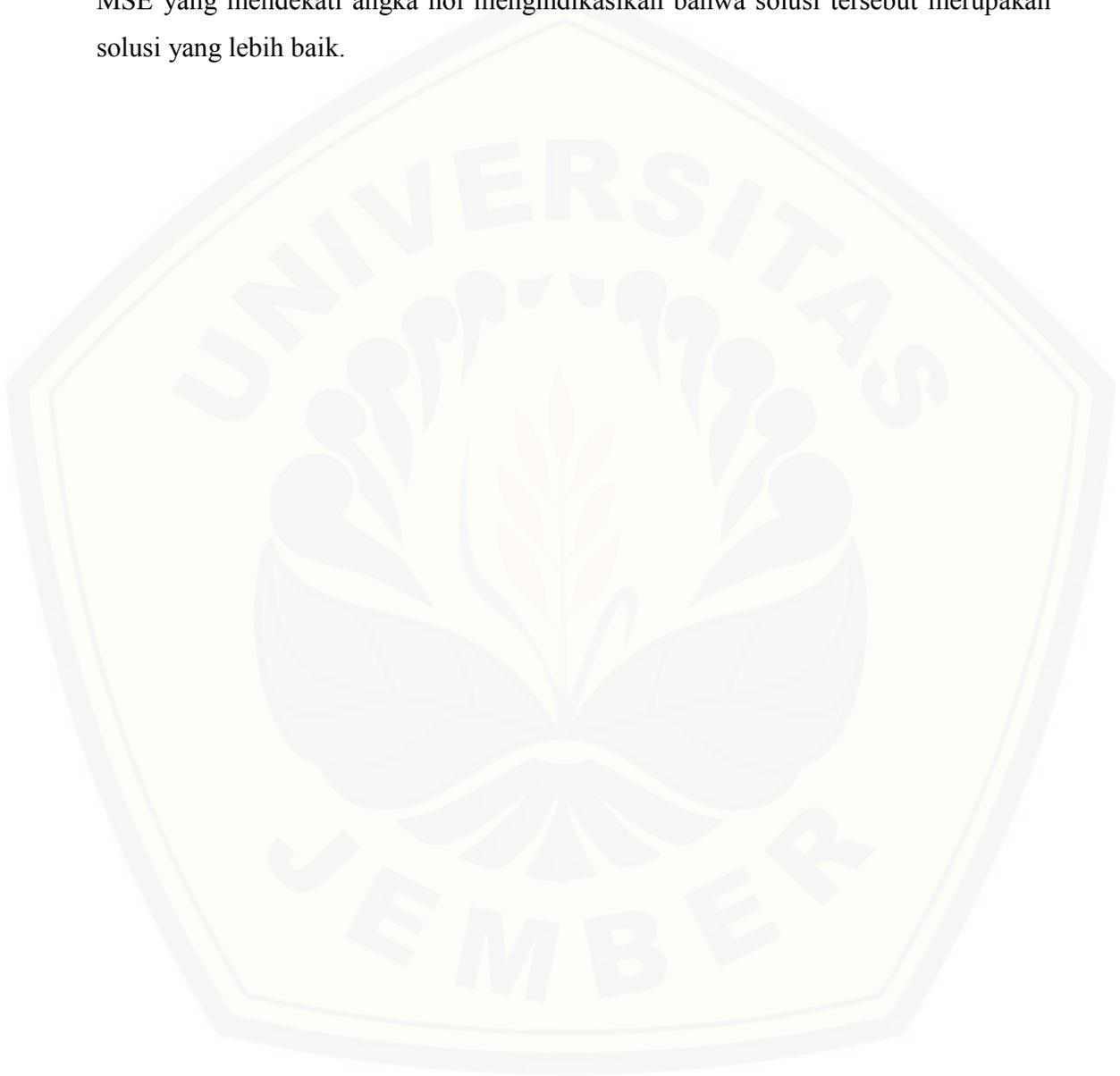
2.4 Mean Square Error (MSE)

Peneliti menggunakan metode *Mean Square Error* (MSE) untuk menentukan nilai error dari solusi yang diperoleh. Nilai MSE diperoleh dengan membagi *Sum Square Error* (SSE) dengan jumlah data atau dalam bentuk matematis dapat ditulis sebagai berikut:

$$MSE = \frac{\sum (y - \hat{y})^2}{n}$$

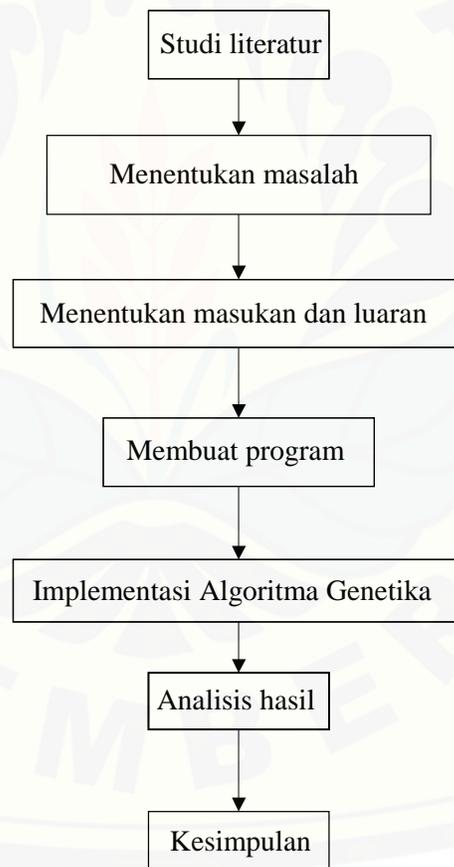
dimana y merupakan variabel respon dalam penelitian ini berupa nilai RHS SPL yang seharusnya, \hat{y} merupakan prediktor dalam penelitian ini berupa nilai RHS dari solusi

yang diperoleh dan n adalah banyaknya persamaan pada SPL. Model dengan nilai MSE yang mendekati angka nol mengindikasikan bahwa solusi tersebut merupakan solusi yang lebih baik.



BAB 3. METODE PENELITIAN

Pada penelitian ini, penulis meneliti tentang penggunaan algoritma Genetika pada pencarian solusi suatu sistem persamaan linier untuk beberapa kasus sistem persamaan linier. Langkah-langkah yang akan dilakukan dalam penelitian ini, secara skematik dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema Metode Penelitian

Penjelasan skema pada Gambar 3.1 untuk memperoleh hasil yang diinginkan sebagai berikut:

a. Studi literatur

Studi literatur yang dilakukan pada penelitian ini adalah mempelajari mengenai Algoritma Genetika dan sistem persamaan linier.

b. Menentukan masalah

Masalah yang akan diteliti pada skripsi ini yaitu enam buah sistem persamaan linier $A = b$ dimana A adalah suatu bentuk matrik koefisien. Matrik yang diteliti yaitu:

1. Matrik berordo 2×2
2. Matrik berordo 3×3 yang memiliki banyak penyelesaian
3. Matrik berordo 3×3 yang dominal diagonal
4. Matrik berordo 4×4 yang tidak dominal diagonal
5. Matrik berordo 4×4 yang dominal diagonal
6. Matrik tidak persegi berordo 4×6

c. Menentukan masukan dan keluaran

Peneliti akan menentukan masukan berupa suatu sistem persamaan yang akan dicari penyelesaiannya, kriteria pemberhentian dan parameter- parameter yang dipakai seperti probabilitas *crossover*, probabilitas mutasi dan jumlah populasi. Sedangkan keluaran berupa kromosom atau hasil penyelesaian yang memiliki nilai *fitness* terbaik.

d. Membuat program

Peneliti akan menulis *script* ke dalam program Matlab R2009a. Selanjutnya, program akan diuji guna menemukan kesalahan yang ada.

e. Implementasi algoritma Genetika

Peneliti akan menyelesaikan suatu sistem persamaan linier dengan program Algoritma Genetika yang telah dibuat pada beberapa bentuk sistem persamaan linier, sehingga didapat suatu solusi penyelesaian yang dianggap terbaik.

f. Analisis hasil

Peneliti akan melakukan analisis terhadap hasil keluaran program. Algoritma Genetika dianggap baik jika hasil yang diperoleh mendekati atau sama dengan hasil sebenarnya. Dalam hal ini peneliti juga menghitung dengan metode eksak seperti Gauss–Jordan dan metode numerik iteratif seperti Gauss–Seidel sebagai tolak ukur keberhasilan algoritma Genetika.

g. Kesimpulan

Peneliti akan menarik kesimpulan berdasarkan analisis hasil yang diperoleh dengan menjawab rumusan masalah yang telah disebutkan.

BAB 5. PENUTUP

5.1 Kesimpulan

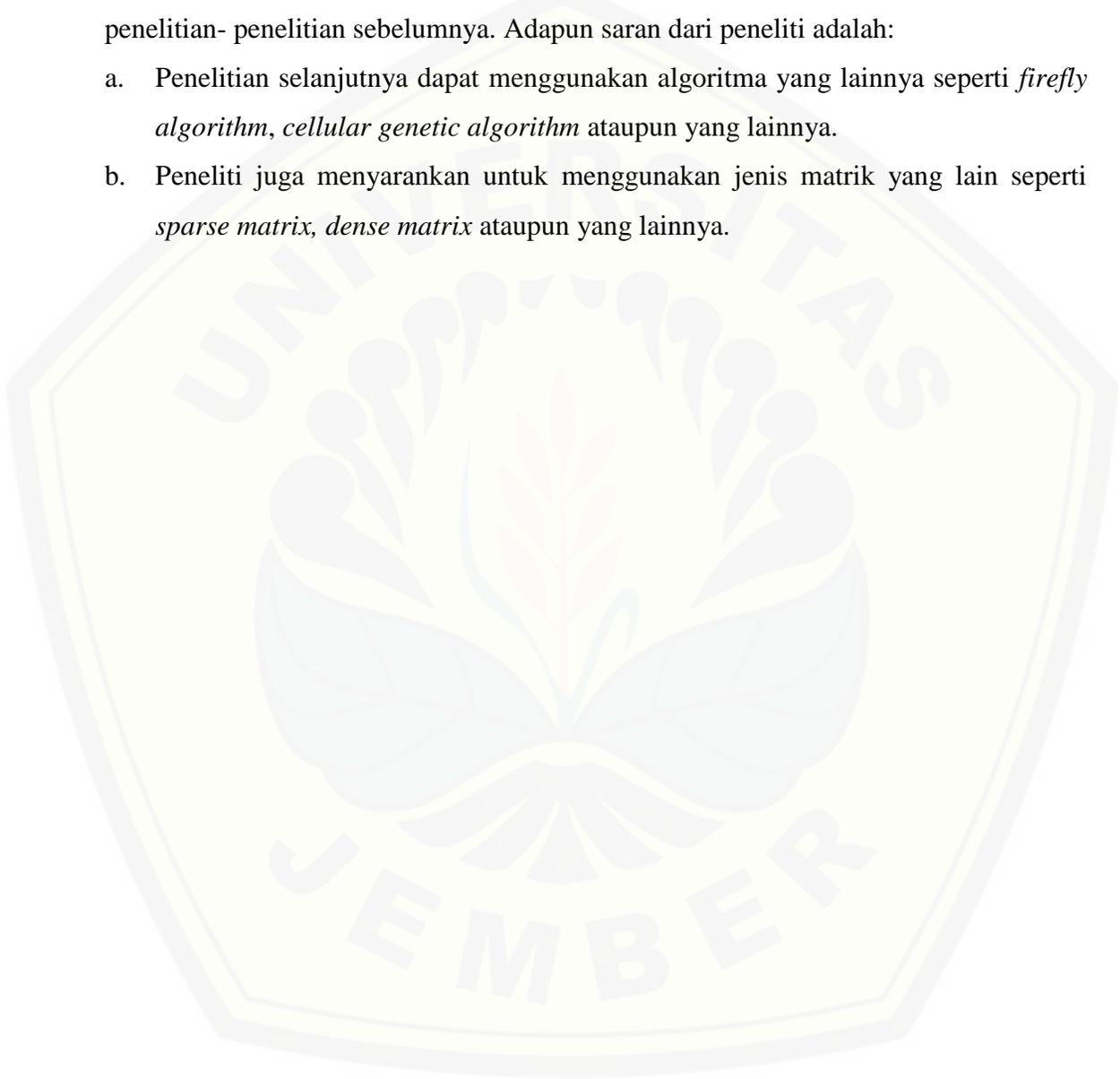
Dari hasil dan pembahasan algoritma genetika pada sistem persamaan linier dengan menggunakan beberapa sistem persamaan dari beberapa sumber referensi dapat disimpulkan sebagai berikut:

- a. Penerapan algoritma Genetika pada sistem persamaan linier adalah dengan cara melakukan pengolahan solusi menggunakan algoritma Genetika dengan tahapan inisialisasi parameter, pembentukan populasi awal, menghitung nilai *fitness*, menyeleksi solusi, melakukan proses reproduksi yaitu *crossover* dan mutasi, dimana pada proses *crossover* dan mutasi algoritma Genetika berjalan paling optimal saat *crossover rate* bernilai 0,9 dan *mutation rate* bernilai 0,1. Selanjutnya proses yang terakhir yaitu mengevaluasi solusi, dan mengulangi tahapan-tahapan tersebut hingga kriteria pemberhentian terpenuhi.
- b. Penelitian ini telah melakukan beberapa kali percobaan pada sistem persamaan linier. Metode eliminasi Gauss-Jordan menghasilkan solusi eksak pada sistem persamaan linier yang berbentuk persegi serta menghasilkan solusi parametrik pada sistem persamaan linier yang memiliki persamaan berkelipatan dan memiliki jumlah variabel tidak sama dengan jumlah persamaan. Metode Gauss-Seidel menghasilkan solusi pendekatan pada sistem persamaan linier yang persegi dan dominal diagonal dan gagal memperoleh hasil pada sistem persamaan linier yang tidak persegi dan tidak dominal diagonal. Algoritma Genetika yang dipakai pada penelitian ini menemukan dapat solusi pendekatan pada beberapa sistem persamaan linier yang memiliki karakteristik matrik seperti diatas dengan nilai MSE yang mendekati angka nol dan nilai *fitness* yang mendekati 1.

5.2 Saran

Setiap hasil penelitian selalu memberikan saran untuk dapat mengembangkan penelitian- penelitian sebelumnya. Adapun saran dari peneliti adalah:

- a. Penelitian selanjutnya dapat menggunakan algoritma yang lainnya seperti *firefly algorithm*, *cellular genetic algorithm* ataupun yang lainnya.
- b. Peneliti juga menyarankan untuk menggunakan jenis matrik yang lain seperti *sparse matrix*, *dense matrix* ataupun yang lainnya.



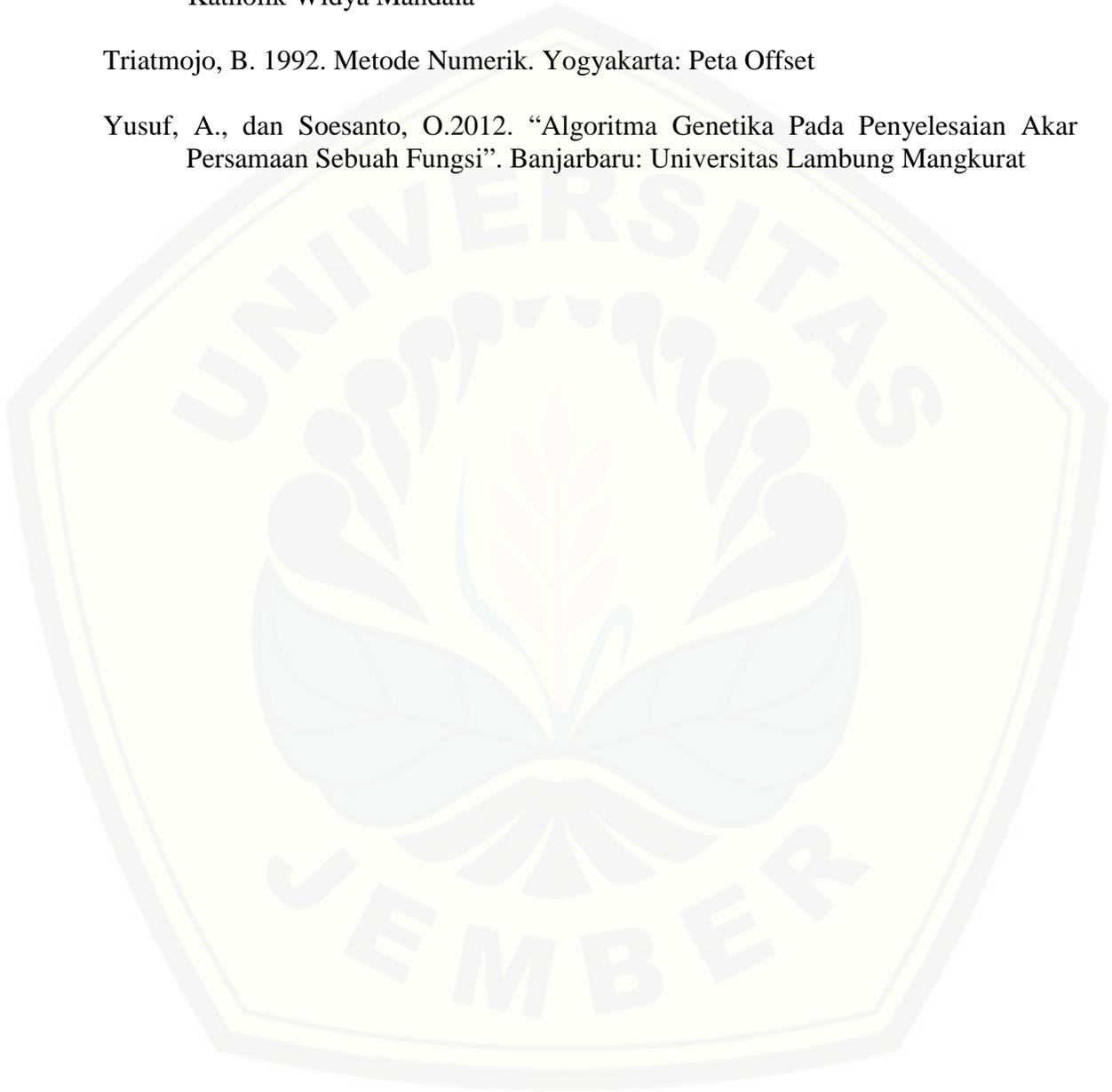
DAFTAR PUSTAKA

- Abiodun, I. M., Olawale, L. N., & Adebowale, A. P. 2011. The Effectiveness of Genetic Algorithm in Solving Simultaneous Equations. *International Journal of Computer Applications* (0975 –8887) vol. 14 no.8
- Ali, A.D., Emary, I. M. M. E., Karem, M. M. A. E. 2009. Application of Genetic Algorithm in Solving Linear Equation Systems. *MASAUM Journal of Basic and Applied Science* vol.1 no.2
- Anton, H & Rorres, C. 2004. Aljabar Linier Elementer Versi Aplikasi Edisi kedelapan jilid 1. Jakarta: Erlangga
- Ariastuti, P. R. 2015. “Penerapan Algoritma Genetika dan Algoritma *Harmony Search* pada Permasalahan *Knapsack* 0-1”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Hanggraeni, S. 2014. “Perbandingan Algoritma *Harmony Search* Dan Genetika Pada Penjadwalan *Jobshop*”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Holland J. H. 1975. *Adaptation in Natural and Artificial Systems*. USA: University of Michigan Press.
- Maftiu, L. O. 2015. *A New Approach for Solving Equations Systems Inspired from Brainstorming*. Romania: West University of Timisoara.
- Markaban, 2004. Logika Matematika. Yogyakarta : PPPG Matematika.
- Mulyono. 2007. “Penyelesaian Persamaan Non – Linear Metode Biseksi dan Metode Regula Falsi Menggunakan Cara Komputasi”. Tidak diterbitkan. Purwokerto: FKIP Universitas Muhammadiyah Purwokerto.
- Sahid. 2005. Pengantar Komputasi Numerik Dengan Matlab. Yogyakarta: Andi
- Sanjoyo. 2006. Aplikasi algoritma genetika. <https://sanjoyo55.files.wordpress.com/2008/11/non-linier-gen-algol.pdf>
[1 November 2015, pukul 09.00]

Santoso, F.G.I. 2011. “Analisis Perbandingan Metode Numerik Dalam Menyelesaikan Persamaan – Persamaan Serentak”. Madiun: Universitas Katholik Widya Mandala

Triatmojo, B. 1992. Metode Numerik. Yogyakarta: Peta Offset

Yusuf, A., dan Soesanto, O.2012. “Algoritma Genetika Pada Penyelesaian Akar Persamaan Sebuah Fungsi”. Banjarbaru: Universitas Lambung Mangkurat



LAMPIRAN

1. Nilai uji parameter P_c dan P_m

a. $P_c = 1$ dan $P_m = 0,1$

Percobaan ke	<i>Fitness</i>	MSE
1	0.999999999812561	1.408009024275262e-006
2	0.99999999994278721	2.1312118010394099e-006
3	0.9999999998664402	3.9404884931591266e-007
4	0.9999999998472877	8.6806788090556478e-007
5	0.9999999998138445	6.6069299847610768e-007
6	0.9999999996914468	1.0550486195049582e-006
7	0.9999999996391709	2.2642886237680645e-006
8	0.9999999997212086	1.331106708313045e-006
9	0.9999999999147349	3.1320571724242688e-007
10	0.9999999998891353	6.3485492226833624e-007
Rata-rata	0.99999999976237	1.106053514510909e-006

b. $P_c = 0,9$ dan $P_m = 0,1$

Percobaan ke	<i>Fitness</i>	MSE
1	0.9999999998191424	5.3152899098207735e-007
2	0.999999999458713	1.7603857124454741e-006
3	0.9999999996610445	1.9707179276855863e-006
4	0.9999999997025635	9.3440691766078057e-007
5	0.999999999857021	4.7827973007644969e-007
6	0.9999999998025879	8.074774119813565e-007
7	0.9999999996967592	9.02844605888391e-007

8	0.999999999898922	3.7152385140040187e-007
9	0.9999999997310063	9.7293052201468671e-007
10	0.9999999997745648	8.9888294748287343e-007
Rata-rata	0.99999999974023	9.628978617618080e-007

c. $P_c = 0,8$ dan $P_m = 0,1$

Percobaan ke	<i>Fitness</i>	MSE
1	0.9999999998899913	3.8297727670510244e-007
2	0.9999999996617484	1.0272578525236077e-006
3	0.9999999995574973	2.5511538237714396e-006
4	0.9999999997223876	1.615076821453892e-006
5	0.99999999988529287	3.4498436935419414e-006
6	0.9999999995727629	2.459688349886156e-006
7	0.9999999999159106	2.5891943080353552e-007
8	0.9999999997312872	1.2938825245123918e-006
9	0.9999999999084022	4.4044123932838984e-007
10	0.999999999777861	1.2309304361797656e-006
Rata-rata	0.99999999965908	1.471017144870622e-006

d. $P_c = 1$ dan $P_m = 0,05$

Percobaan ke	<i>Fitness</i>	MSE
1	0.9999999999368316	2.5343356958014376e-007
2	0.9999999994136646	1.8929440292910105e-006
3	0.9999999998549216	9.5727247250749832e-007
4	0.9999999999713618	1.164908554633737e-007
5	0.9999999999861655	5.1792566604682309e-008
6	0.9999999999509714	2.3311518333455791e-007
7	0.9999999997184441	9.5584351271456324e-007

8	0.999999999734478	1.5006000246391219e-006
9	0.9999999992118582	2.3483520632213976e-006
10	0.9999999996648214	1.2839446309898605e-006
Rata-rata	0.99999999974435	9.793788908346211e-007

e. $P_c = 0,9$ dan $P_m = 0,05$

Percobaan ke	<i>Fitness</i>	MSE
1	0.999999999209099	2.9894545473443383e-007
2	0.9999999992748234	4.5651141975451109e-006
3	0.999999999271905	4.7436699336213282e-007
4	0.9999999996538957	1.3404736902117157e-006
5	0.999999999707556	1.2033212718878811e-007
6	0.9999999997624123	1.4338088497205201e-006
7	0.9999999997910449	9.4392825496339958e-007
8	0.9999999998637534	5.0848302831727613e-007
9	0.9999999998432709	5.7636529642465298e-007
10	0.9999999996498601	1.5045605691634559e-006
Rata-rata	0.99999999976579	1.176637846163149e-006

f. $P_c = 0,8$ dan $P_m = 0,05$

Percobaan ke	<i>Fitness</i>	MSE
1	0.9999999996965772	9.2181716473006608e-007
2	0.9999999998863576	5.9006894154931904e-007
3	0.9999999991083077	4.8930978478762845e-006
4	0.9999999996942779	1.0058872888625735e-006
5	0.9999999998431788	5.0770716040812891e-007
6	0.9999999998339995	1.219004314177588e-006
7	0.9999999993274868	2.2544954968858473e-006

8	0.9999999992646305	3.5454700643862276e-006
9	0.9999999999435651	1.8472553337553013e-007
10	0.9999999998800582	4.0285315582993526e-007
Rata-rata	0.99999999964785	1.552512696808150e-006

2. Data simulasi SPL yang dipakai untuk uji waktu komputasi

a. Ukuran data SPL dengan matriks pada SPL berordo 5x5

$$\begin{array}{r}
 1 \ 2 \ 3 \ 0 \ 1 \ = \ 19 \\
 1 \ 1 \ 1 \ 0 \ 0 \ = \ 6 \\
 3 \ 2 \ 1 \ 1 \ 1 \ = \ 19 \\
 2 \ 5 \ 8 \ 4 \ 6 \ = \ 82 \\
 2 \ 4 \ 7 \ 1 \ 2 \ = \ 45
 \end{array}$$

b. Ukuran data SPL dengan matriks pada SPL berordo 10x10

$$\begin{array}{r}
 1 \ 2 \ 3 \ 0 \ 1 \ 2 \ 6 \ 5 \ 4 \ 9 \ = \ 239 \\
 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 2 \ 4 \ 2 \ 3 \ = \ 106 \\
 3 \ 2 \ 1 \ 1 \ 1 \ 2 \ 3 \ 6 \ 9 \ 4 \ = \ 221 \\
 2 \ 5 \ 8 \ 4 \ 6 \ 3 \ 2 \ 1 \ 8 \ 7 \ = \ 264 \\
 2 \ 4 \ 7 \ 1 \ 2 \ 5 \ 4 \ 6 \ 3 \ 3 \ = \ 208 \\
 3 \ 1 \ 2 \ 2 \ 7 \ 8 \ 4 \ 6 \ 2 \ 1 \ = \ 206
 \end{array}$$

c. Ukuran data SPL dengan matriks pada SPL berordo 15x15

$$\begin{array}{r}
 1 \ 2 \ 3 \ 0 \ 1 \ 2 \ 6 \ 5 \ 4 \ 9 \ 3 \ 2 \ 3 \ 6 \ 4 \ = \ 479 \\
 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 2 \ 4 \ 2 \ 3 \ 8 \ 4 \ 2 \ 5 \ 2 \ = \ 368 \\
 3 \ 2 \ 1 \ 1 \ 1 \ 2 \ 3 \ 6 \ 9 \ 4 \ 4 \ 5 \ 1 \ 1 \ 4 \ = \ 412 \\
 2 \ 5 \ 8 \ 4 \ 6 \ 3 \ 2 \ 1 \ 8 \ 7 \ 4 \ 5 \ 2 \ 3 \ 9 \ = \ 571 \\
 2 \ 4 \ 7 \ 1 \ 2 \ 5 \ 4 \ 6 \ 3 \ 3 \ 7 \ 4 \ 3 \ 2 \ 8 \ = \ 520 \\
 3 \ 1 \ 2 \ 2 \ 7 \ 8 \ 4 \ 6 \ 2 \ 1 \ 4 \ 6 \ 2 \ 5 \ 4 \ = \ 478 \\
 5 \ 1 \ 8 \ 2 \ 1 \ 0 \ 0 \ 1 \ 4 \ 2 \ 2 \ 3 \ 1 \ 3 \ 5 \ = \ 296 \\
 6 \ 1 \ 3 \ 2 \ 9 \ 2 \ 4 \ 2 \ 2 \ 2 \ 4 \ 1 \ 6 \ 2 \ 6 \ = \ 416 \\
 9 \ 1 \ 1 \ 5 \ 5 \ 1 \ 4 \ 4 \ 5 \ 1 \ 5 \ 2 \ 0 \ 5 \ 23 \ = \ 674 \\
 8 \ 5 \ 1 \ 1 \ 2 \ 0 \ 2 \ 1 \ 3 \ 2 \ 1 \ 5 \ 9 \ 4 \ 21 \ = \ 663 \\
 0 \ 1 \ 1 \ 4 \ 7 \ 1 \ 4 \ 3 \ 4 \ 3 \ 1 \ 7 \ 4 \ 1 \ 1 \ = \ 356 \\
 2 \ 2 \ 9 \ 3 \ 8 \ 6 \ 2 \ 5 \ 9 \ 5 \ 1 \ 7 \ 33 \ 2 \ 1 \ = \ 873 \\
 5 \ 4 \ 8 \ 1 \ 15 \ 8 \ 0 \ 1 \ 7 \ 1 \ 1 \ 9 \ 2 \ 0 \ 3 \ = \ 435
 \end{array}$$

5	8	4	4	1	5	1	3	4	4	4	2	2	3	0	2	3	2	1	2
9	1	2	2	0	7	2	1	2	6	2	2	3	2	0	1	1	1	4	1
1	15	8	0	1	2	0	2	1	3	9	1	3	1	5	2	5	4	1	3
6	2	4	0	4	7	1	4	3	4	2	4	2	4	1	1	3	5	1	1
4	0	2	0	9	8	6	2	5	9	1	8	1	89	2	6	2	6	1	1
2	3	5	3	2	15	8	0	1	7	23	5	5	4	2	0	5	23	1	2
4	6	4	2	1	2	4	0	4	5	3	4	4	4	2	4	2	2	3	0

3	2	9	9	5	=	1401
1	3	8	9	5	=	1441
1	5	4	1	5	=	1186
2	4	5	3	5	=	1246
6	4	6	0	4	=	1116
1	2	23	1	4	=	2394
2	4	21	0	4	=	1459
2	9	1	0	7	=	1022
2	8	1	0	7	=	1617
21	4	3	2	8	=	2614
4	5	0	9	9	=	1235
52	6	1	9	6	=	3819
1	23	0	0	3	=	2848
3	21	0	1	21	=	1701
1	1	0	3	4	=	2452
11	1	2	2	5	=	1096
1	3	21	5	5	=	4245
1	0	4	3	0	=	918
1	2	52	2	5	=	1939
1	1	1	5	2	=	663
1	2	3	4	1	=	889
4	1	1	1	4	=	806
1	3	11	2	2	=	2407
1	1	1	0	3	=	1433
1	1	1	0	2	=	693