



**PENERAPAN *HARMONY SEARCH ALGORITHM WITH
MODIFIED DIFFERENTIAL MUTATION OPERATOR (HSMD)*
PADA PENYELESAIAN SISTEM PERSAMAAN NON LINIER**

SKRIPSI

Oleh :
Siti Solehatin
NIM 141810101053

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018**



**PENERAPAN *HARMONY SEARCH ALGORITHM WITH
MODIFIED DIFFERENTIAL MUTATION OPERATOR (HSMD)*
PADA PENYELESAIAN SISTEM PERSAMAAN NON LINIER**

SKRIPSI

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

Oleh

**Siti Solehatin
NIM 141810101053**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018**

PERSEMBAHAN

Alhamdulillah dengan segala kerendahan hati, saya persembahkan skripsi ini sebagai bentuk tanggung jawab, bakti dan ungkapan terimakasih kepada :

1. Allah SWT yang telah memberikan rahmat, kasih sayang dan karunia-Nya kepada hamba-Nya untuk kemudahan serta kelancaran dalam pembuatan skripsi ini;
2. Kedua orang tua saya, almarhumah ibu Sutik dan bapak Abd. Hadi tercinta yang telah memberikan doa, kasih sayang, inspirasi, dukungan serta nasehat yang senantiasa mengiringi setiap langkah pembuatan skripsi ini;
3. Almarhumah nenek saya, nenek Sutiah yang sudah berbaik hati mendedikasikan seluruh hidupnya untuk menemani, merawat, mendoakan, serta memberikan kasih sayang yang tak terhingga kepada saya dan adik-adik saya layaknya seorang ibu;
4. Kedua adik saya, Siti Aisyah dan Ahmad Khusairi yang selalu memberikan dukungan dan senantiasa menjadi tempat berbagi suka duka dan kasih sayang;
5. Segenap keluarga besar yang telah memberikan semangat serta kasih sayangnya;
6. Seluruh guru dan dosen sejak taman kanak-kanak hingga perguruan tinggi yang telah memberikan ilmu dan bimbingannya;
7. Almamater Jurusan Matematika FMIPA Universitas Jember, SMA Negeri 1 Gending, SMP Negeri 1 Banyu Anyar, SD Negeri Klenang Lor 1 dan TK Kasih Ibu.

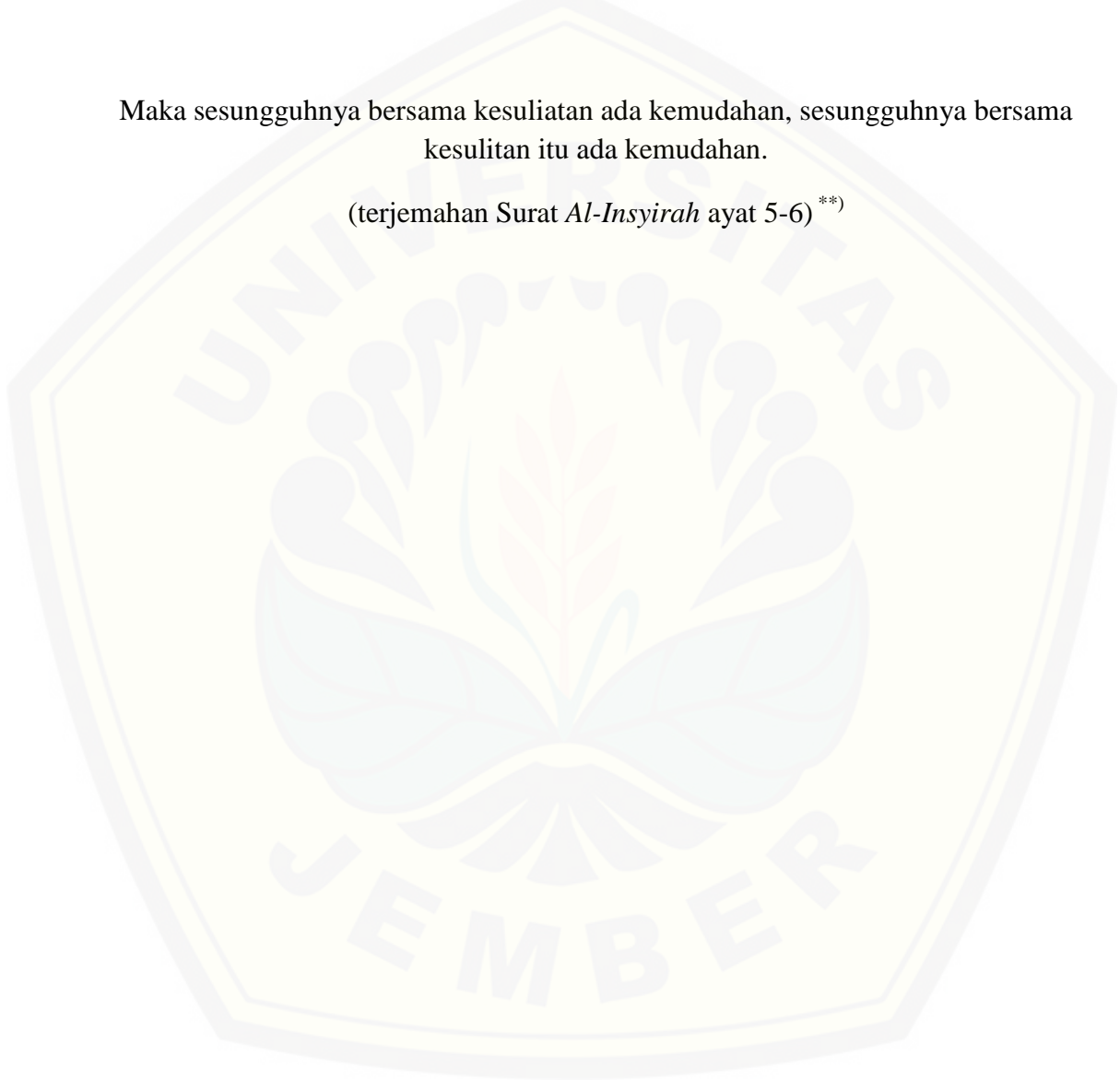
MOTO

Allah tidak membebani seseorang melainkan sesuai kesanggupannya.

(terjemahan Surat *Al-Baqarah* ayat 286) *)

Maka sesungguhnya bersama kesulitan ada kemudahan, sesungguhnya bersama kesulitan itu ada kemudahan.

(terjemahan Surat *Al-Insyirah* ayat 5-6) **)



*) Kementerian Agama RI. 2014. *Al-Qur'an dan Terjemahannya*. Bandung : CV Mikraj Khazanah Ilmu.

**) Motto Hidup dari Al-Quran
<https://nikenpuspitasari.wordpress.com/2012/03/11/motto-hidup-dari-al-quran/>

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Siti Solehatin

NIM : 141810101053

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) pada Penyelesaian Sistem Persamaan Non Linier” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun 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, 16 Juli 2018

Yang menyatakan,

Siti Solehatin
NIM 141810101053

SKRIPSI

**PENERAPAN *HARMONY SEARCH ALGORITHM WITH
MODIFIED DIFFERENTIAL MUTATION OPERATOR (HSMD)*
PADA PENYELESAIAN SISTEM PERSAMAAN NON LINIER**

Oleh

Siti Solehatin
NIM 141810101053

Pembimbing

Dosen Pembimbing Utama : Drs. Rusli Hidayat, M.Sc.

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

PENGESAHAN

Skripsi berjudul “Penerapan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) pada Penyelesaian Sistem Persamaan Non Linier” karya Siti Solehatin telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember

Tim Penguji:

Ketua,

Anggota I,

Drs. Rusli Hidayat, M.Sc.
NIP 196610121993031001

Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP 197211291998021001

Anggota II,

Anggota III,

Kosala Dwidja Purnomo, S.Si., M.Si.
NIP 196908281998021001

Kusbudiono, S.Si, M.Si
NIP 197704302005011001

Mengesahkan
Dekan,

Drs. Sujito, Ph.D
NIP 196102041987111001

RINGKASAN

Penerapan *Harmony Search Algorithm With Modified Differential Mutation Operator (HSMD)* pada Penyelesaian Sistem Persamaan Non Linier; Siti Solehatin, 141810101053; 2018: 60 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Sistem persamaan non-linier adalah himpunan dari beberapa persamaan non linier yang solusinya memenuhi semua persamaan tersebut. Pencarian solusi sistem persamaan non linier dapat dilakukan menggunakan metode analitik, namun dalam beberapa kasus penyelesaian analitik dari sistem persamaan non linier tidak mudah ditemukan sehingga dibutuhkan metode-metode lain untuk menyelesaikannya. Salah satu metode yang dapat digunakan untuk menyelesaikan sistem persamaan non linier yaitu dengan menggunakan algoritma metaheuristik. Salah satu contoh dari algoritma metaheuristik adalah *Harmony Search Algorithm With Modified Differential Mutation Operator (HSMD)*.

Penelitian ini dimulai dengan menentukan beberapa sistem persamaan non linier dari beberapa referensi rujukan yang akan diteliti berupa sistem persamaan non linier dua variabel dan sistem persamaan non linier tiga variabel. Kemudian menentukan nilai parameter yang digunakan yaitu HMS , $HMCR$, N , L_i , U_i , NI , F_{awal} , F_{akhir} , a_1 dan a_2 . Setelah itu mencari perkiraan solusi dari sistem persamaan non linier dengan mencari nilai *fitness* (hasil penjumlahan $|f(x)|$) dari beberapa kemungkinan vektor solusi yang sama dengan 0. Apabila nilai *fitness* yang diteliti tidak ada yang sama dengan 0, maka dipilih nilai *fitness* yang paling minimum. Kemudian menguji keakuratan dari HSMD dengan cara membandingkan solusi sistem persamaan non linier menggunakan HSMD dengan solusi sistem persamaan non linier menggunakan metode Newton-Raphson dari beberapa referensi rujukan yang digunakan dalam penelitian ini.

Hasil penyelesaian sistem persamaan non linier menggunakan HSMD menghasilkan solusi yang mendekati solusi eksak. Nilai *fitness* yang dihasilkan menggunakan HSMD lebih mendekati 0 daripada nilai *fitness* yang dihasilkan

menggunakan Metode Newton-Raphson. Kecepatan konvergensi, iterasi dan waktu komputasi program HSMD dipengaruhi oleh beberapa faktor, yaitu perubahan nilai parameter HMS , $HMCR$, a_1 dan a_2 . Apabila nilai HMS yang digunakan kecil, maka nilai *fitness* yang dihasilkan besar. Selain itu semakin besar nilai HMS , semakin sedikit iterasi dan semakin besar waktu komputasi program yang dibutuhkan untuk menemukan solusi sistem persamaan non linier. Sedangkan apabila nilai $HMCR$, a_1 dan a_2 yang digunakan sama dengan 0, maka nilai *fitness* yang dihasilkan besar. Perubahan nilai $HMCR$ tidak berpengaruh terhadap jumlah iterasi dan waktu komputasi program. Untuk $a_1 > 0$, semakin besar nilai a_1 , jumlah iterasi yang dibutuhkan untuk mencapai kekonvergenan juga semakin besar. Perubahan nilai a_1 tidak berpengaruh terhadap waktu komputasi program. Sedangkan semakin besar nilai a_2 , jumlah iterasi dan waktu komputasi program yang dibutuhkan untuk menemukan solusi sistem persamaan non linier juga semakin besar.

PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) pada Penyelesaian Sistem Persamaan Non Linier”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Drs. Rusli Hidayat, 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;
2. Kosala Dwidja Purnomo, S.Si., M.Si., selaku Dosen Penguji I, dan Kusbudiono, S.Si., M.Si., selaku Dosen Penguji II yang telah memberikan kritik serta sarannya terhadap penulisan skripsi ini;
3. Drs. Rusli Hidayat, M.Sc., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. seluruh staf pengajar Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember yang telah memberikan ilmu serta bimbingannya sehingga penulis dapat menyelesaikan skripsi ini;
5. almarhumah ibu Sutik, bapak Abd. Hadi, almarhumah nenek Sutiah dan kedua adik saya, Siti Aisyah dan Ahmad Khusairi yang telah memberikan doa dan motivasi demi terselesaikannya skripsi ini;
6. sahabat-sahabat “IPA 2 SMANIG”, “Kontrakan 66” dan “The Gang” yang selalu memberikan keceriaan, dorongan motivasi dan nasehat;
7. teman-teman angkatan 2014 (EXTREME) atas keceriaan, canda tawa serta dukungan yang selalu diberikan selama pembelajaran dalam masa perkuliahan;
8. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Juli 2018

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Persamaan Non Linier.....	5
2.2 Sistem Persamaan Non Linier	5
2.3 Metode Numerik	6
2.4 Metode <i>Newton-Raphson</i>	7
2.5 Algoritma <i>Harmony Search (HS)</i>.....	8
2.5.1 Algoritma <i>Differential Harmony Search (DHS)</i>	11
2.5.2 Modifikasi <i>Harmony Search Algorithm With Modified</i> <i>Differential Mutation Operator (HSMD)</i>	12
BAB 3. METODE PENELITIAN.....	15

BAB 4. HASIL DAN PEMBAHASAN	18
4.1 Hasil	18
4.2 Pembahasan	21
4.2.1 Perhitungan Manual.....	21
4.2.2 Program.....	31
4.2.3 Simulasi Perubahan Nilai Parameter	38
BAB 5. PENUTUP	45
5.1 Kesimpulan	45
5.2 Saran	45
DAFTAR PUSTAKA	46
LAMPIRAN	49

DAFTAR GAMBAR

	Halaman
2.1 <i>Flowchart Harmony Search Algorithm With Modified Differential Operator</i> (HSMD)	13
3.1 Skema Metode Penelitian.....	17
4.1 Tampilan Program.....	31
4.2 Grafik Kekonvergenan Nilai <i>Fitness</i> SPNL No. 1 pada Tabel 4.1	33
4.3 Grafik Kekonvergenan Nilai <i>Fitness</i> SPNL No. 2 pada Tabel 4.1	34
4.4 Grafik Kekonvergenan Nilai <i>Fitness</i> SPNL No. 3 pada Tabel 4.1	35
4.5 Grafik Kekonvergenan Nilai <i>Fitness</i> SPNL No. 4 pada Tabel 4.1	36
4.6 Grafik Kekonvergenan Nilai <i>Fitness</i> SPNL No. 5 pada Tabel 4.1	37

DAFTAR TABEL

	Halaman
4.1 Perbandingan Hasil Penyelesaian Sistem Persamaan Non Linier dari HSMD dengan Perhitungan Langsung pada MATLAB dan Metode Newton-Raphson	19
4.2 Solusi dan <i>Fitness</i> Setiap Solusi	22
4.3 Vektor Solusi Baru dan <i>Fitness</i> Baru.....	28
4.4 Vektor Solusi Baru dan <i>Fitness</i> HM Baru	29
4.5 Simulasi Hasil dengan Nilai <i>HMS</i> yang Berbeda	39
4.6 Simulasi Hasil dengan Nilai <i>HMCR</i> yang Berbeda	40
4.7 Simulasi Hasil dengan Nilai a_1 yang Berbeda.....	42
4.8 Simulasi Hasil dengan Nilai a_2 yang Berbeda.....	43

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Salah satu permasalahan yang sering terjadi dalam bidang ilmiah adalah mencari solusi dari persamaan non linier dan sistem persamaan non linier. Persamaan non linier adalah semua persamaan selain persamaan linier yang derajat terkecil peubahnya satu. Pencarian solusi persamaan non linier dilakukan dengan menentukan nilai x yang memenuhi $f(x) = 0$, yaitu nilai $x = s$ sedemikian hingga $f(s) = 0$. Sedangkan sistem persamaan non linier adalah himpunan n persamaan non linier, dengan $n > 1$, yang penyelesaiannya harus memenuhi semua n persamaan tersebut..

Pencarian akar sistem persamaan non linier dapat dilakukan menggunakan metode analitik. Namun dalam beberapa kasus penyelesaian analitik dari sistem persamaan non linier tidak mudah ditemukan, kecuali pada kasus yang sederhana. Oleh sebab itu digunakan alternatif lain untuk mencari solusi dari sistem persamaan non linier, yaitu menggunakan pendekatan metode numerik.

Proses perhitungan secara numerik dilakukan dengan menentukan batas-batas akar dan pencarian pendekatan nilai akar melalui proses perhitungan secara berulang hingga tercapai kekonvergenan. Terdapat beberapa metode numerik yang telah digunakan untuk menyelesaikan sistem persamaan non linier. Utami (2013) melakukan penelitian tentang perbandingan solusi sistem persamaan non linier menggunakan Metode Newton-Raphson dan Metode Jacobian. Hasil yang diperoleh yaitu solusi yang diperoleh dari Metode Newton-Raphson lebih baik daripada Metode Jacobian, dimana iterasi untuk menghasilkan solusi yang konvergen pada Metode Newton-Raphson yaitu 5 iterasi, sedangkan jumlah iterasi pada Metode Jacobian adalah 58.

Metode Newton-Raphson merupakan metode yang sering digunakan untuk menyelesaikan sistem persamaan non linier karena tingkat kecepatan perhitungannya yang cukup baik untuk menghasilkan solusi yang mendekati atau sama dengan solusi eksak. Namun metode ini memiliki beberapa kelemahan. Nasiha (2008) menyatakan bahwa perkiraan solusi sangat mempengaruhi hasil

dan tingkat konvergensinya. Apabila pemilihan nilai perkiraan solusi awal tidak tepat maka algoritma gagal. Sedangkan penentuan perkiraan solusi awal yang baik pada sebagian besar sistem persamaan non linier tidak dapat ditentukan dengan mudah. Kelemahan komputasi ini mendorong para ilmuwan dunia untuk mengandalkan algoritma metaheuristik.

Belakangan ini, pengembangan dari beberapa jenis algoritma metaheuristik sudah terlihat, misalnya pada *Evolutionary Algorithm* (EAs) dan algoritma *Swarm Intelligence* (SI). Meskipun selama ini algoritma metaheuristik banyak digunakan pada permasalahan optimasi dan penjadwalan, tidak menutup kemungkinan juga bisa diterapkan pada penyelesaian persamaan non linier. Anwar (2016) menyatakan bahwa Metode *Zero Crossing* dan Algoritma *Virus Evolutionary Genetic Algorithm* (VEGA) mampu menyelesaikan persamaan non linier baik akar tunggal maupun ganda dengan hasil yang mendekati atau sama dengan solusi eksak. Penelitian lainnya oleh Baihaki (2016) adalah tentang penerapan algoritma *Cat Swarm Optimization* (CSO) pada penyelesaian sistem persamaan non linier. Hasil yang diperoleh yaitu algoritma CSO mampu menyelesaikan sistem persamaan nonlinier dengan hasil yang mendekati solusi eksak. Selanjutnya Prastowo (2016) menyatakan bahwa *Cockroach Swarm Optimization Algorithm* (CSOA) mampu menyelesaikan sistem persamaan non linier dengan hasil yang mendekati solusi eksak dan lebih baik daripada hasil yang diperoleh menggunakan Metode Newton-Raphson dan algoritma CSO.

Algoritma metaheuristik lain yang efektif dalam hal optimasi adalah algoritma *Harmony Search* (HS) yang terinspirasi dari proses improvisasi pemutaran musik (Geem, 2001). Beberapa penelitian sebelumnya tentang algoritma HS memberikan hasil yang baik. Geem (2002) mengaplikasikan algoritma HS pada optimasi desain jaringan pipa. Penelitian lainnya oleh Geem (2005) adalah tentang pengaplikasian algoritma HS pada *vehicle routing problem*.

Mahdavi (2007) dalam jurnalnya menyatakan bahwa algoritma HS juga memiliki kelemahan, yaitu mengalami prematur, tingkat konvergensi yang lambat serta kinerja dari algoritma HS akan memburuk seiring bertambahnya ruang

pencarian solusi. Oleh sebab itu dibutuhkan suatu teknik atau metode yang bisa mengatasi masalah tersebut.

Chakraborty (2012) mengusulkan sebuah algoritma pengembangan dari algoritma HS yaitu *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD). Langkah dari algoritma ini pada dasarnya sama dengan langkah kerja algoritma HS, hanya saja strategi *pitch adjustment* pada algoritma HS diganti dengan modifikasi operator mutasi yang diambil dari algoritma *Differential Evolution* (DE) untuk meningkatkan daya eksploratif dari algoritma gabungan ini. Hasil penelitian memberikan hasil yang baik ketika algoritma HS digabung dengan modifikasi operator mutasi yang ada pada algoritma DE.

Berdasarkan uraian di atas, menjadi pertimbangan bagi penulis untuk melakukan penelitian tentang penerapan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) pada penyelesaian sistem persamaan non linier.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang akan dibahas dalam penelitian ini adalah :

- a. Bagaimana menyelesaikan sistem persamaan non linier menggunakan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) ?
- b. Bagaimana hasil penerapan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) jika dibandingkan dengan Metode Newton-Raphson ?

1.3 Batasan Masalah

Pada penelitian ini menggunakan batasan masalah sebagai berikut :

- a. persamaan non linier yang digunakan hanya persamaan non linier berupa fungsi polinomial orde 4, transenden maupun gabungannya.

- b. fungsi transenden yang digunakan berupa fungsi trigonometri, eksponensial, dan logaritma.
- c. fungsi gabungan yang digunakan berupa gabungan polinomial dan logaritma, gabungan polinomial dan eksponensial, gabungan polinomial dan trigonometri, serta gabungan polinomial, eksponensial dan trigonometri.

1.4 Tujuan

Tujuan dari penelitian ini adalah :

- a. mengetahui cara menyelesaikan sistem persamaan non linier dengan menggunakan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD);
- b. mengetahui hasil penyelesaian sistem persamaan non linier menggunakan *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) jika dibandingkan dengan Metode Newton-Raphson.

1.5 Manfaat

Manfaat dari penelitian ini adalah mampu memberikan informasi kepada penulis dan pembaca mengenai penerapan metode optimasi metaheuristik yaitu *Harmony Search Algorithm With Modified Differential Mutation Operator* (HSMD) untuk menyelesaikan permasalahan sistem persamaan non linier.

BAB 2. TINJAUAN PUSTAKA

2.1 Persamaan Non Linier

Persamaan non linier adalah semua persamaan yang bukan persamaan linier dengan peubah berderajat terkecil sama dengan satu. Pencarian solusi persamaan non linier dilakukan dengan menentukan nilai x yang memenuhi $f(x) = 0$, yaitu nilai $x = s$ sedemikian hingga $f(s) = 0$

Penentuan akar-akar persamaan non linier dapat diselesaikan menggunakan metode analitik, misalnya untuk mencari akar fungsi kuadrat dapat ditentukan dengan menggunakan rumus abc. Namun dalam beberapa kasus terdapat beberapa fungsi selain fungsi kuadrat yang akar-akarnya tidak dapat ditentukan dengan mudah secara analitik. Oleh sebab itu digunakan alternatif lain untuk mencari akar persamaan, yaitu dengan menggunakan pendekatan metode numerik (Chapra, 1989).

Menurut Triatmodjo (1996), terdapat dua tahapan pada poses perhitungan akar secara numerik, yaitu penentuan batas-batas akar dan pencarian pendekatan nilai akar. Beberapa metode penyelesaian persamaan non linier dengan metode numerik yaitu metode bagi dua, metode newton-raphson dan metode secant.

2.2 Sistem Persamaan Non Linier

Menurut Kosasih (2006), sistem persamaan non linier adalah himpunan n persamaan non linier, dengan $n > 1$, yang penyelesaiannya harus memenuhi semua n persamaan tersebut.

Bentuk umum sistem persamaan non linier dapat dituliskan sebagai berikut :

$$f(x) = \begin{cases} f_1(x_1, x_2, x_3, \dots, x_n) = 0 \\ f_2(x_1, x_2, x_3, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, x_3, \dots, x_n) = 0 \end{cases} \quad (2.1)$$

Penyelesaian sistem persamaan (2.1) terdiri dari himpunan nilai-nilai variabel yang memenuhi semua persamaan tersebut.

Terdapat beberapa metode yang dapat digunakan untuk menyelesaikan sistem persamaan non linier, diantaranya yaitu Newton-Raphson, Regulasi falsi, CSO, *Particle Swarm Optimization* (PSO), dan *Glowworm Swarm Optimization* (GSO). Contoh sistem persamaan non linier yaitu :

$$\begin{aligned}f_1(x_1, x_2) &= 3x_1 + x_1^3 + x_2 + 1 = 0 \\f_2(x_1, x_2) &= x_1 + 2x_2 + e^{x_2} - 2 = 0\end{aligned}$$

Akar-akar dari sistem persamaan non linier diatas terdiri dari himpunan nilai-nilai variabel yang memenuhi semua persamaan tersebut (Azmi, 2017).

2.3 Metode Numerik

Triatmodjo (1996) menyatakan bahwa metode numerik adalah teknik untuk menyelesaikan permasalahan-permasalahan yang diformulasikan secara matematika sedemikian sehingga dapat diselesaikan dengan operasi aritmatika (penjumlahan, pengurangan, perkalian, dan pembagian). Penyelesaian suatu sistem persamaan yang besar, non linier dan sangat kompleks yang tidak mungkin diselesaikan secara analitis dapat diselesaikan menggunakan metode ini.

Hasil perhitungan akar yang didapat melalui metode numerik bukan berupa nilai eksak, namun berupa aproksimasi atau hampiran yang mendekati solusi eksak sehingga akan dimungkinkan muncul kesalahan. Tiga kesalahan yang pasti terjadi adalah kesalahan pembulatan, kesalahan pemotongan, dan kesalahan pemenggalan (Juliana, 2011).

Terdapat beberapa metode numerik yang digunakan untuk menyelesaikan sistem persamaan non linier, diantaranya yaitu metode Newton-Raphson, Metode Titik Tetap, Metode Jacobian, Metode Broyden dan sebagainya. Metode Newton-Raphson merupakan metode yang sering digunakan untuk menyelesaikan sistem persamaan non linier karena tingkat kecepatannya yang cukup baik untuk menghasilkan solusi yang mendekati atau sama dengan solusi eksak (Kosasih, 2006).

2.4 Metode Newton-Raphson

Metode Newton-Raphson merupakan salah satu contoh metode numerik yang sering digunakan untuk menemukan akar persamaan maupun sistem persamaan karena tingkat konvergensinya yang cepat terutama bila nilai hampiran awal akar yang digunakan cukup dekat dengan akar yang sebenarnya.

Perhatikan sistem persamaan non linier di bawah ini :

$$\begin{aligned} F_1 &= f_1(x_1, x_2, x_3, \dots, x_n) = 0 \\ F_2 &= f_2(x_1, x_2, x_3, \dots, x_n) = 0 \\ &\vdots \\ F_n &= f_n(x_1, x_2, x_3, \dots, x_n) = 0 \end{aligned} \quad (2.2)$$

penyelesaian sistem persamaan non linier tersebut terdiri dari himpunan nilai-nilai x yang memenuhi semua persamaan tersebut dan secara simultan memberikan semua persamaan tersebut nilai yang sama dengan nol. Sistem persamaan non linier tersebut dapat diselesaikan menggunakan perluasan metode Newton-Raphson melalui ekspansi deret Taylor pada masing-masing persamaan. Dengan ekspansi deret Taylor pertama (Nasiha, 2008) :

$$f(x_{i+1}) = f(x_i) + (x_{i+1} - x_i)f'(x_i)$$

Sehingga sistem persamaan (2.2) menjadi :

$$\begin{bmatrix} (F_1)_{i+1} - (F_1)_i \\ (F_2)_{i+1} - (F_2)_i \\ \vdots \\ (F_n)_{i+1} - (F_n)_i \end{bmatrix} = \begin{bmatrix} \frac{\partial(F_1)_i}{\partial x_1} & \frac{\partial(F_1)_i}{\partial x_2} & \dots & \frac{\partial(F_1)_i}{\partial x_n} \\ \frac{\partial(F_2)_i}{\partial x_1} & \frac{\partial(F_2)_i}{\partial x_2} & \dots & \frac{\partial(F_2)_i}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial(F_n)_i}{\partial x_1} & \frac{\partial(F_n)_i}{\partial x_2} & \dots & \frac{\partial(F_n)_i}{\partial x_n} \end{bmatrix} \begin{bmatrix} (x_1)_{i+1} - (x_1)_i \\ (x_2)_{i+1} - (x_2)_i \\ \vdots \\ (x_n)_{i+1} - (x_n)_i \end{bmatrix} \quad (2.3)$$

atau

$$\Delta F \approx J(x_1, x_2, x_3, \dots, x_n) \cdot \Delta X$$

karena yang akan dicari adalah nilai $(x_1, x_2, x_3, \dots, x_n)_{i+1}$ ketika $(F_1)_{i+1}, (F_2)_{i+1}, \dots, (F_n)_{i+1} = 0$ maka persamaan (2.3) dapat ditulis sebagai berikut :

$$X_{i+1} = X_i - J(X)_i^{-1} \cdot F(X_i)$$

2.5 Algoritma *Harmony Search*

Algoritma *harmony search* (HS) adalah salah satu algoritma metaheuristik yang diusulkan oleh Geem pada tahun 2001. Algoritma tersebut terinspirasi dari proses pertunjukan musik, dimana musisi melakukan proses pencarian harmoni yang lebih baik untuk mendapatkan keadaan terbaik berdasarkan perkiraan estetika.

Berdasarkan analogi tersebut, HS melakukan proses optimasi untuk mendapatkan keadaan terbaik dengan mengevaluasi fungsi objektif. Fungsi objektif pada HS dihitung menggunakan himpunan nilai-nilai pada setiap variabel keputusan. Perbaikan nilai fungsi objektif pada HS terus ditingkatkan disetiap iterasi seperti halnya perbaikan kualitas suara estetika yang diperbaiki dengan latihan demi latihan.

Berdasarkan konsep di atas, algoritma *harmony search* terdiri dari lima tahapan, yaitu (suyanto, 2010) :

a. Inisialisasi Masalah dan Parameter Algoritma

Langkah pertama yang harus dilakukan yaitu inisialisasi masalah dan parameter algoritma. Permasalahan optimasi dapat ditunjukkan seperti persamaan (2.4) berikut :

$$\text{Minimasi (atau maksimasi) } f(x) \quad (2.4)$$

dengan batasan $x_i \in X_i, i = 1, 2, \dots, N; L_i \leq X_i \leq U_i$

dimana

$f(x_i)$ = fungsi objektif

x_i = variabel keputusan ke- i

X_i = himpunan variabel keputusan ke- i

N = jumlah variabel keputusan

L_i = batas bawah variabel keputusan ke- i

U_i = batas atas variabel keputusan ke- i

Setelah menentukan permasalahan optimasi, inialisasi parameter-parameter dari algoritma HS. Parameter-parameter tersebut yaitu (Aulia dkk, 2012):

- 1) *Harmony Memory Size* (HMS) adalah banyaknya vektor solusi yang bisa disimpan di dalam harmony memory;
- 2) *Harmony Memory Considering Rate* (HMCR) adalah probabilitas harmony memory untuk digunakan kembali sebagai hasil dari vektor solusi yang bernilai $0 \leq \text{HMCR} \leq 1$, umumnya digunakan berkisar antara 0,7 sampai 0,95;
- 3) *Pitch Adjusting Rate* (PAR) adalah parameter yang berperan dalam menentukan jumlah nilai yang harus diubah, disesuaikan, atau ditukar dengan nilai yang lain. Nilainya $0 \leq \text{PAR} \leq 1$, umumnya digunakan berkisar antara 0,1 sampai 0,5;
- 4) *Bandwidth* adalah jumlah perubahan maksimal dalam penyesuaian, biasanya yang sering digunakan adalah 0,001 sampai 0,01;
- 5) *Number of Impovisation* (NI) adalah banyaknya iterasi untuk melakukan improvisasi.

b. Inialisasi *Harmony Memory*

Pada tahap kedua, dibangkitkan matriks *harmony memory* (HM) secara acak yang berisi vektor-vektor solusi sebanyak HMS. Pembangkitan vektor-vektor solusi pada HM dilakukan dengan menggunakan persamaan (2.5) seperti berikut:

$$x_i^j = L_i + \text{rand}(0,1) \cdot (U_i - L_i) \quad (2.5)$$

dimana $i = 1,2,3, \dots, N$, $j = 1,2,3, \dots, \text{HMS}$ dan $\text{rand}(0,1)$ adalah bilangan acak antara 0 dan 1 (Chakraborty, 2009).

Berikut adalah representasi dari matriks HM :

$$\begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_{N-1}^2 & x_N^2 \\ \dots & \dots & \dots & \dots & \dots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \dots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix}$$

c. Improvisasi Harmoni Baru

Pada tahap ini, dilakukan pembangkitan *harmony memory* baru sehingga membentuk vektor solusi baru $x' = (x'_1, x'_2, x'_3, x'_4, \dots, x'_N)$. Pembangkitan tersebut dilakukan berdasarkan tiga aturan, yaitu *harmony memory consideration*, *pitch adjustment* dan pemilihan acak.

1) *Harmony memory consideration*

Nilai dari variabel keputusan x'_1 untuk vektor yang baru dipilih dari salah satu nilai yang sudah ada di HM saat ini yaitu dari himpunan $\{x_1^1, x_1^2, \dots, x_1^{HMS}\}$ dengan probabilitas *HMCR*. Nilai dari variabel keputusan yang lain $x'_2, x'_3, x'_4, \dots, x'_N$ ditentukan dengan cara yang sama.

Pembangkitan variabel keputusan baru pada tahap ini dapat ditunjukkan seperti pada persamaan (2.6) berikut (Chakraborty, 2012) :

$$x'_i \leftarrow \begin{cases} x_i \in \{x_i^1, x_i^2, x_i^3, x_i^4, \dots, x_i^{HMS}\} & \text{dengan probabilitas HMCR} \\ x_i \in X_i & \text{dengan probabilitas } (1 - \text{HMCR}) \end{cases} \quad (2.6)$$

2) *Pitch adjustment*

Tahap ini merupakan tahap penyesuaian variabel keputusan baru x_i yang dihasilkan pada tahap *harmony memory consideration*. Penyesuaian variabel pada tahap ini seperti pada persamaan (2.7) berikut :

$$x'_i \leftarrow \begin{cases} x'_i \pm \text{rand}(0,1) \cdot bw & \text{dengan probabilitas PAR} \\ x'_i & \text{dengan probabilitas } (1 - \text{PAR}) \end{cases} \quad (2.7)$$

d. Pembaruan *Harmony Memory*

Pada tahap ini, jika vektor solusi baru $x' = (x'_1, x'_2, x'_3, x'_4, \dots, x'_N)$ memiliki nilai *fitness* yang lebih baik dibandingkan vektor solusi dengan nilai *fitness* terburuk di dalam HM, maka vektor solusi baru dengan nilai *fitness* yang lebih baik tersebut dimasukkan ke dalam HM dan vektor solusi dengan nilai *fitness* terburuk dikeluarkan dari HM. Apabila nilai *fitness* vektor solusi tidak lebih baik maka tidak terjadi perubahan pada HM.

e. Cek Kriteria Berhenti

Kriteria pemberhentian yang dipakai yaitu jumlah iterasi maksimum dan nilai *fitness* terbaik. Nilai *fitness* adalah mutlak dari nilai fungsi ($|f(x)|$). Nilai *fitness* terbaik adalah sama dengan 0. Apabila kriteria pemberhentian sudah tercapai, maka proses pengerjaan dihentikan. Namun apabila kriteria pemberhentian belum tercapai maka akan kembali pada langkah c dan d.

2.5.1 Algoritma *Differential Harmony Search* (DHS)

Penelitian tentang algoritma metaheuristik HS menunjukkan bahwa algoritma tersebut mengalami prematur dan konvergensinya lambat. Selain itu kinerja algoritma HS memburuk seiring bertambahnya ruang pencarian dimensional. Sebuah pengembangan algoritma HS yaitu *Differential Harmony Search* (DHS) diajukan oleh Chakraborty untuk mengatasi masalah tersebut. Algoritma ini adalah gabungan antara algoritma HS dengan skema mutasi pada algoritma *Differential Evolution* (DE) (Chakraborty, 2009). Pada algoritma DHS, strategi *pitch adjustment* dari algoritma HS klasik diganti dengan skema mutasi pada algoritma DE untuk meningkatkan daya eksploratif dari algoritma. Hasil percobaan membuktikan keefektifan dari metode hibrid. Pada dasarnya, algoritma DHS mengikuti langkah a-e dari algoritma HS, dengan operasi *pitch adjustment* diganti dengan proses yang direpresentasikan dalam persamaan (2.8) berikut :

$$x'_i = x'_i + F(x_{r1} - x_{r2}) \quad (2.8)$$

dimana $r1 \neq r2$ adalah dua indeks acak dari vektor yang dipilih dari *harmony memory* dan F adalah faktor real dan konstan $\in [0,1]$ yang mengendalikan $(x_{r1} - x_{r2})$ (Chakraborty, 2009).

2.5.2 Harmony Search Algorithm With Modified Differential Mutation Operator (HSMD)

Harmony Search Algorithm With Modified Differential Mutation Operator (HSMD) merupakan modifikasi dari algoritma DHS. Skema mutasi DE yang digunakan pada algoritma DHS dimodifikasi menjadi skema mutasi yang direpresentasikan dalam persamaan (2.9) berikut:

$$x'_i = x'_i + F(a_1 \cdot \text{rand.}(x_{BEST} - x'_i) + a_2 \cdot \text{rand.}(x_{r1} - x_{r2})) \quad (2.9)$$

dengan nilai F diperoleh dari :

$$F = F - \frac{F_{awal} - F_{akhir}}{NI} \quad (2.10)$$

dimana $r1 \neq r2$ adalah dua indeks acak dari vektor yang dipilih dari *harmony memory*, a_1, a_2 adalah dua parameter skala yang digunakan untuk mencapai optimum global dan F merupakan faktor skala yang mengendalikan jarak antar nilai akar yang dicari. Langkah kerja dari HSMD hampir sama dengan langkah kerja algoritma DHS, perbedaannya terletak pada persamaan skema mutasi yang digunakan dan banyaknya vektor solusi baru yang dibangkitkan pada langkah improvisasi harmoni baru. Pada HSMD, vektor solusi baru yang dibangkitkan tiap iterasi adalah sebanyak m , dimana $m = 1, 2, 3, \dots, HMS$. Sedangkan pada algoritma DHS vektor solusi baru yang dibangkitkan hanya 1 tiap iterasi. Langkah ini menjamin bahwa untuk permasalahan kendala permasamaan yang kompleks, vektor dengan cepat memasuki zona solusi sedemikian hingga vektor mempunyai kesempatan untuk mengeksplorasi zona solusi dengan lebih efektif. Biasanya nilai yang digunakan untuk a_1 dan a_2 berturut-turut adalah 6 dan 0,3 (Chakraborty, 2012).

Langkah-langkah pada HSMD dapat diilustrasikan pada *flowchart* di bawah ini :



Gambar 2.1 *Flowchart Harmony Search Algorithm With Differential Mutation Operator (HSMD)*

BAB 3. METODE PENELITIAN

Penulis akan menerapkan *Harmony Search Algorithm With Differential Mutation Operator* (HSMD) untuk menyelesaikan sistem persamaan non linier pada penelitian ini. Langkah-langkah yang akan dilakukan pada penelitian ini adalah sebagai berikut :

a. Studi literatur

Studi literatur dilakukan dengan mengumpulkan dan mempelajari berbagai teori yang menunjang penelitian ini. Teori yang dipelajari meliputi algoritma *Harmony Search* (HS), *Differential Evolution* (DE), *Differential Harmony Search* (DHS), *Harmony Search Algorithm With Differential Mutation Operator* (HSMD), penyelesaian sistem persamaan non linier dan teori penunjang lainnya. Studi literatur dilakukan sebagai pedoman penelitian baik pengaplikasian dan analisis yang akan dilakukan. Selain itu juga bertujuan agar lebih memahami teori-teori yang digunakan dalam penelitian.

b. Menentukan Masalah

Masalah yang akan dikaji dalam penelitian ini adalah penyelesaian sistem persamaan non linier yang dirujuk dari beberapa referensi berupa fungsi polinomial orde 4, transenden maupun gabungannya dengan ketentuan sebagai berikut :

- 1) fungsi transenden yang digunakan berupa fungsi trigonometri, eksponensial, dan logaritma.
- 2) fungsi gabungan yang digunakan berupa gabungan polinomial dan logaritma, gabungan polinomial dan eksponensial, gabungan polinomial dan trigonometri, serta gabungan polinomial, eksponensial dan trigonometri.

c. Implementasi HSMD

Permasalahan yang menjadi objek penelitian ini akan diolah menggunakan HSMD. Adapun langkah-langkah yang harus dilakukan adalah sebagai berikut :

- 1) Pengaturan parameter-parameter dan inisialisasi masalah. Tentukan nilai parameter HMS , $HMCR$, N , L_i , U_i , NI , F_{awal} , F_{akhir} , a_1 dan a_2 .

- 2) Membangkitkan matriks *harmony memory* (HM) dengan persamaan (2.5).
- 3) Membangkitkan vektor solusi baru $x' = (x'_1, x'_2, x'_3, x'_4, \dots, x'_N)$ sebanyak m menggunakan aturan *harmony memory consideration* dengan persamaan (2.6) kemudian dilanjutkan dengan mutasi setiap vektor solusi baru menggunakan persamaan (2.9).
- 4) Mengevaluasi nilai *fitness* dari setiap vektor solusi baru. Jika nilai *fitness* vektor solusi baru $x' = (x'_1, x'_2, x'_3, x'_4, \dots, x'_N)$ lebih baik dibandingkan nilai *fitness* vektor solusi di dalam HM, maka vektor solusi baru dengan nilai *fitness* yang lebih baik tersebut dimasukkan ke dalam HM dan vektor solusi di dalam HM dengan nilai *fitness* yang lebih buruk dikeluarkan dari HM, sehingga nilai HM diperbarui. Jika nilai *fitness* vektor solusi baru tidak lebih baik maka tidak terjadi perubahan.
- 5) Pengecekan kriteria pemberhentian. Apakah mencapai iterasi maksimum atau nilai *fitness* terbaik? Jika ya maka keluar output. Sebaliknya, apabila tidak maka kembali ke langkah 3 .

d. Simulasi Program *Harmony Search Algorithm With Differential Mutation Operator* (HSMD)

Simulasi program dilakukan menggunakan *software* MATLAB R2015b.

Langkah-langkah yang diperlukan adalah sebagai berikut :

- 1) Input
Peneliti akan memberikan input berupa sistem persamaan non linier (dalam bentuk fungsi) yang akan dicari penyelesaiannya, parameter-parameter yang digunakan dan kriteria pemberhentian. Parameter yang digunakan adalah HMS , $HMCR$, N , L_i , U_i , NI , F_{awal} , F_{akhir} , a_1 dan a_2 .
- 2) Proses
Proses pada simulasi ini seperti pada langkah c.
- 3) Output
Output yang dihasilkan dalam penelitian ini yaitu solusi sistem persamaan non linier. Pemaparan output berupa nilai *fitness* terbaik dan grafik kekonvergenan.

e. Analisis Hasil

Peneliti akan menganalisis hasil yang diperoleh menggunakan program. HSMD akan dianggap baik jika output mendekati atau sama dengan solusi eksak (nilai *fitness* sama dengan nol). Apabila tidak ditemukan solusi eksak, maka dilakukan perbandingan terhadap hasil yang diperoleh pada referensi rujukan di langkah b. Nilai *fitness* diperoleh dari hasil persamaan berikut :

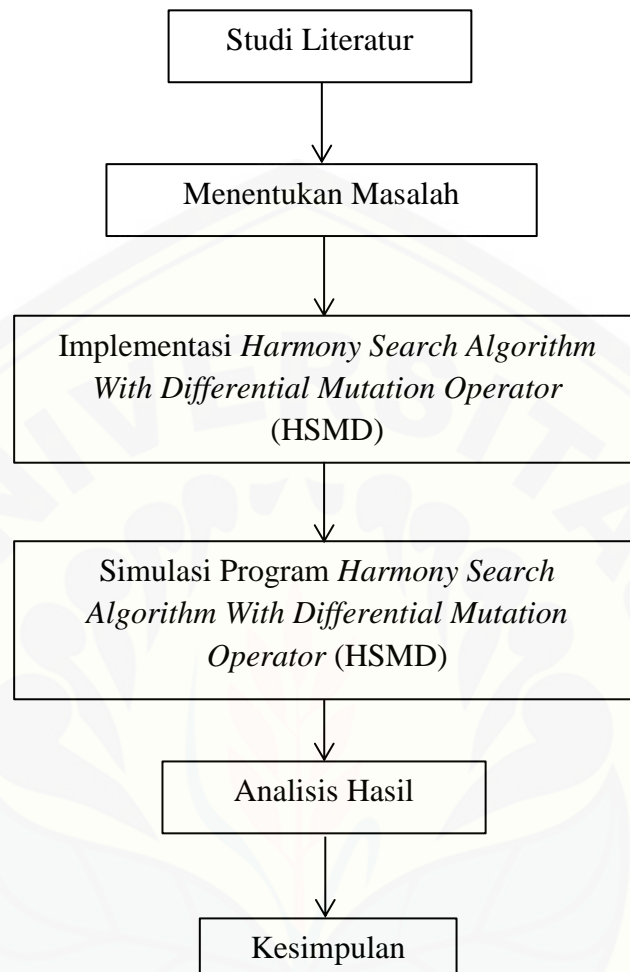
$$f(x) = |f_1(x_1, x_2, \dots, x_n)| + |f_2(x_1, x_2, \dots, x_n)| + \dots + |f_n(x_1, x_2, \dots, x_n)|$$

Perbandingan dilakukan dengan ketentuan suatu algoritma dikatakan lebih baik dari algoritma lainnya apabila nilai *fitness* yang dihasilkan lebih kecil dan mendekati nol (Azmi, 2018). Hal ini dilakukan guna menentukan akurasi hasil yang diperoleh oleh HSMD. Kriteria keberhasilan berupa tingkat akurasi hasil yang diperoleh. Selain itu peneliti akan menganalisis pengaruh parameter *HMS*, *HMCR*, a_1 dan a_2 terhadap solusi, iterasi maupun waktu komputasi program.

f. Kesimpulan

Pengambilan kesimpulan dilakukan dengan memberikan jawaban dari tujuan penelitian ini. Saran-saran juga diberikan untuk perbaikan bagi penelitian selanjutnya.

Untuk lebih jelasnya dapat dilihat pada skema metode penelitian berikut :



Gambar 3.1 Skema Metode Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dipaparkan pada bab sebelumnya, diperoleh kesimpulan sebagai berikut :

- a. HSMD dapat digunakan untuk menyelesaikan 5 contoh sistem persamaan non linier khususnya sistem persamaan non linier berupa fungsi polinomial orde 4, gabungan polinomial dan logaritma, gabungan polinomial dan eksponensial, gabungan polinomial dan trigonometri, serta gabungan polinomial, eksponensial dan trigonometri. Penyelesaian 5 sistem persamaan non linier tersebut menggunakan HSMD menghasilkan nilai fungsi yang mendekati solusi eksak. Namun untuk mendapatkan solusi yang mendekati solusi eksak, terdapat beberapa hal yang harus diperhatikan yaitu nilai HMS tidak boleh kecil dan nilai $HMCR$, a_1 dan a_2 yang digunakan harus lebih besar dari 0.
- b. HSMD memiliki akurasi dan efisiensi yang lebih baik jika dibandingkan dengan Metode Newton-Raphson dalam menemukan solusi sistem persamaan non linier karena nilai *fitness* yang dihasilkan lebih kecil atau mendekati 0 daripada nilai *fitness* yang dihasilkan dengan menggunakan Metode Newton-Raphson pada referensi rujukan. Selain itu HSMD tidak bergantung pada nilai awal dan tidak perlu mencari turunan dari fungsi penyusun sistem persamaan non linier seperti pada Metode Newton-Raphson.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, disarankan kepada peneliti selanjutnya untuk menerapkan HSMD pada kasus numerik yang lain guna menambah wawasan baru ketika suatu permasalahan numerik diselesaikan dari sudut pandang masalah optimasi, misalnya penyelesaian persamaan differensial, kasus integrasi, penyelesaian persamaan non linier akar ganda dan lainnya.

DAFTAR PUSTAKA

Anwar, Z. 2016. Penerapan Gabungan Metode Zero Crossing dan Virus Evolutionary Genetic Algorithm (VEGA) Pada Peyeleaian Persamaan Non-linier. *Skripsi*. Jember : Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Azmi, A.U. 2018. Perbandingan Algoritma Particle Swarm Optimization (PSO) dan Algoritma Glowworm Swarm Optimization (GSO) Dalam Penyelesaian Sistem Persamaan Non Linier. *Skripsi*. Jember : Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Aulia, I., E.B. Nababan, dan M.A. Muchtar. 2012. Penerapan harmony search algorithm dalam permasalahan penjadwalan flow shop. *Jurnal Dunia Teknologi Informasi*. 1(1) : 1-7.

Baihaki, N.A. 2016. Penerapan Algoritma Cat Swarm Optimiation (CSO) Pada Penyelesaian Sistem Persamaan Non-Linier. *Skripsi*. Jember : Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Chakraborty, P., G.G. Roy, B.K. Panigrahi, R.C. Bansal, dan A. Mahapatra. 2012. Dynamic economic dispatch using harmony search algorithm with modified differential mutation operator. *Electr Eng*. 94(2012) : 197-205.

Chakraborty, P., G.G. Roy, S. Das, D. Jain, dan A. Abraham. 2009. An improved harmony search algorithm with differential mutation operator. *Fundamental informaticae*. 95(2009) : 1-26.

Chapra, S.C. dan R.P. Canale. 1989. *Metode Numerik untuk Teknik*. Jakarta : Erlangga.

Chen, Z., X. Qiu, dan B. Chen. 2017. The iterative methods with higher order convergence for solving a system of nonlinear equations. *Journal of Nonlinear Sciences and Applications*. 10(2017) : 3834-3842.

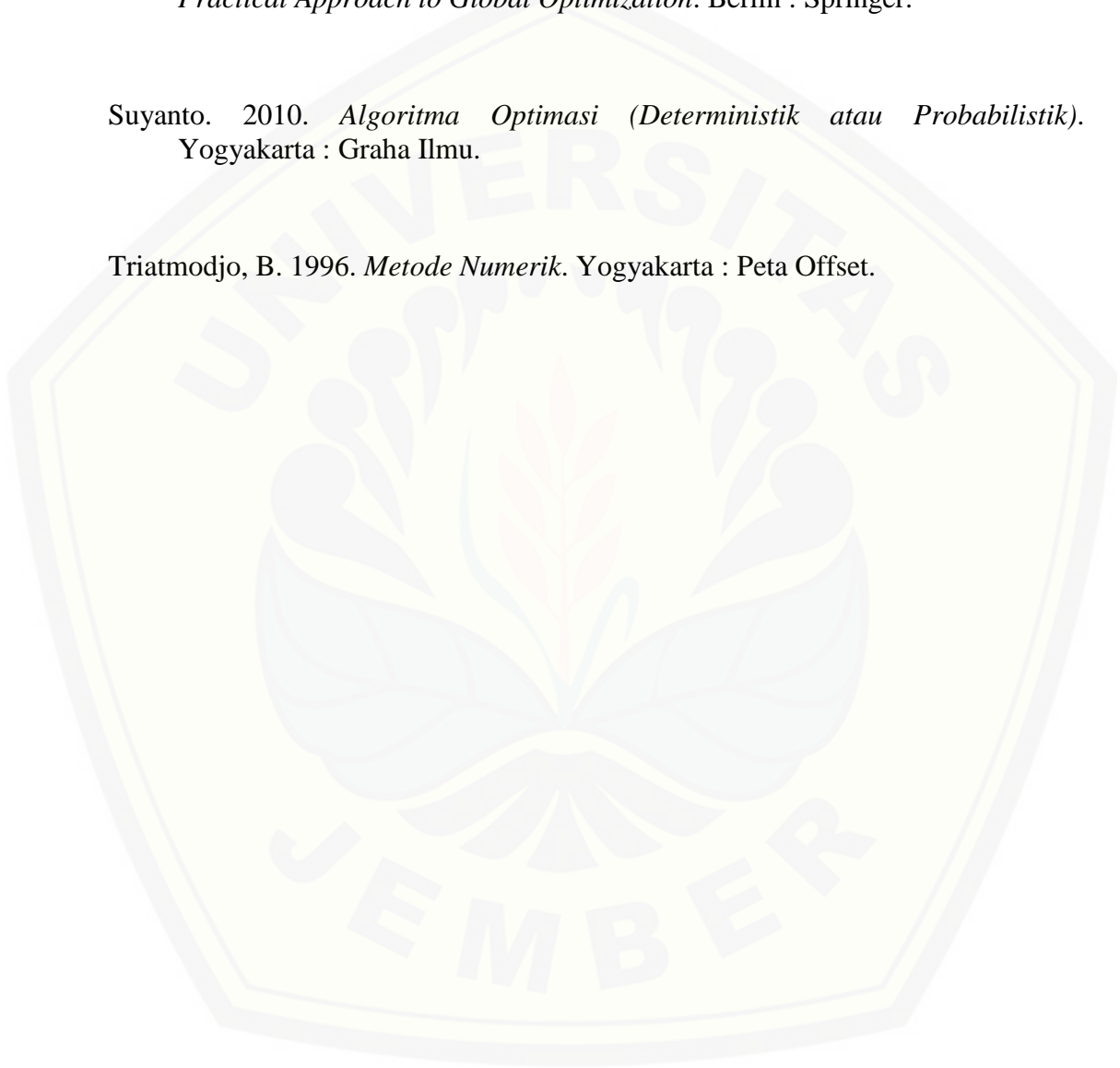
- Geem, Z.W., J.H. Kim, dan G.V. Loganathan. 2001. A new heuristic optimization algorithm : harmony search. *Simulation*. 76(2) : 60-68.
- Geem, Z.W., J.H. Kim, dan G.V. Loganathan. 2002. Harmony search optimization : application to pipe network design. *Int. J. Model. Simul.* 22(2) : 125-133.
- Geem, Z.W., K.S. Lee, dan Y. Park. 2005. Application of harmony search to vehicle routing. *American Journal of applied Science*. 2(12) : 1552-1557.
- Juliana, J.R., J. Siswanto, E. Asmawati, dan A. Herlambang. 2011. *Metode Numeric dengan Scilab*. Malang : Bayumedia Publishing.
- Kosasih, P.B. 2006. *Komputasi Numerik Teori dan Aplikasi*. Yogyakarta : C.V Andi Offset.
- Mahdavi, M., M. Fesanghary, dan E. Damangir. 2007. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*. 188(2) : 1567-1579.
- Nasiha, K. 2008. Penyelesaian Sistem Persamaan Tak Linier Dengan Metode Newton-Raphson. *Skripsi*. Malang : Universitas Islam Negeri Malang.
- Noor, M.A., dan M. Waseem. 2009. Some iterative methods for solving a system of nonlinear equations. *Computer and Mathematics with Applications*. 57(1) : 101-106.
- Prastowo, F.K. 2016. Penerapan Cockroach Swarm Optimization Algorithm (CSOA) Pada Penyelesaian Sistem Persamaan Nonlinier. *Skripsi*. Jember : Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Singh, S. 2013. A system of nonlinear equations with singular jacobian. *International Journal of Innovative Research in Science, Engineering and Technology*. 2(7) : 2650-2653.

Storn, R.M., K.V. Price, dan J. Lampinen. 2005. *Differential Evolution-A Practical Approach to Global Optimization*. Berlin : Springer.

Suyanto. 2010. *Algoritma Optimasi (Deterministik atau Probabilistik)*. Yogyakarta : Graha Ilmu.

Triatmodjo, B. 1996. *Metode Numerik*. Yogyakarta : Peta Offset.



LAMPIRAN

A. Skrip Program HSMD

```
% Update handles structure
guidata(hObject, handles);
movegui(gcf, 'center');
set(handles.edit1, 'string', '', 'style', 'edit');
set(handles.edit2, 'string', '', 'style', 'edit');
set(handles.edit3, 'string', '', 'style', 'text');
set(handles.edit5, 'string', '');
set(handles.edit6, 'string', '');
set(handles.edit7, 'string', '');
set(handles.edit8, 'string', '');
set(handles.edit9, 'string', '');
set(handles.edit10, 'string', '');
set(handles.edit11, 'string', '');
set(handles.edit12, 'string', '');
set(handles.edit13, 'string', '');
cla(handles.axes1, 'reset');
axes(handles.axes1);
set(handles.axes1, 'XLim', [0 1000], 'YLim', [-0.2 1]);
grid on
xlabel('Iterasi'); ylabel('Fitness');
set(handles.listbox1, 'string', sprintf('%5s %20s %40s
%42s', 'iter', 'x', 'y', 'Fitness'), 'value', 1, 'userdata', []);
set(handles.text17, 'string', '0 detik');
set(handles.text19, 'string', '0');
% UIWAIT makes HSMD wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

Seleksi Banyak Variabel

```
set(handles.edit1, 'string', '');
set(handles.edit2, 'string', '');
set(handles.edit3, 'string', '');
numvar=get(handles.popupmenu1, 'value')+1;
if numvar==2
    set(handles.edit1, 'style', 'edit');
    set(handles.edit2, 'style', 'edit');
    set(handles.edit3, 'style', 'text');
    set(handles.listbox1, 'string', sprintf('%5s %20s %40s
%42s', 'iter', 'x', 'y', 'Fitness'), 'value', 1);
elseif numvar==3
    set(handles.edit1, 'style', 'edit');
    set(handles.edit2, 'style', 'edit');
    set(handles.edit3, 'style', 'edit');
    set(handles.listbox1, 'string', sprintf('%5s %20s %40s %40s
%42s', 'iter', 'x', 'y', 'z', 'Fitness'), 'value', 1);
end
cla(handles.axes1, 'reset');
axes(handles.axes1);
```

```

set(handles.axes1,'XLim',[0 1000],'YLim',[-0.2 1]);
grid on
xlabel('Iterasi');ylabel('Fitness');
set(handles.listbox1,'userdata',[]);
set(handles.text17,'string','0 detik');
set(handles.text19,'string','0');

```

Tombol Proses

```

clc;
set(handles.listbox1,'userdata',[]);
set(handles.text17,'string','0 detik');
set(handles.text19,'string','0');
pause(0.1);
format long;
tic;
numvar=get(handles.popupmenu1,'value')+1;

if numvar==2
    f1=inline(get(handles.edit1,'string'),'x','y'); %input fungsi
    f2=inline(get(handles.edit2,'string'),'x','y'); %input fungsi
elseif numvar==3
    f1=inline(get(handles.edit1,'string'),'x','y','z'); %input
fungsi
    f2=inline(get(handles.edit2,'string'),'x','y','z'); %input
fungsi
    f3=inline(get(handles.edit3,'string'),'x','y','z'); %input
fungsi
end

%parameter
HMS=str2num(get(handles.edit5,'string'));
HMCR=str2num(get(handles.edit6,'string'));
a1=str2num(get(handles.edit7,'string'));
a2=str2num(get(handles.edit8,'string'));
Fawal=str2num(get(handles.edit9,'string'));
Fakhir=str2num(get(handles.edit10,'string'));
Numiter=str2num(get(handles.edit13,'string'));

Lb=str2num(get(handles.edit11,'string'));
Ub=str2num(get(handles.edit12,'string'));

F=Fawal;

%Inisialisasi Harmony Memory
HM=Lb+rand(HMS,numvar)*(Ub-Lb);
for i=1:HMS
    if numvar==2
        Fitness(i)=abs(f1(HM(i,1),HM(i,2)))+...
            abs(f2(HM(i,1),HM(i,2)));
    elseif numvar==3
        Fitness(i)=abs(f1(HM(i,1),HM(i,2),HM(i,3)))+...
            abs(f2(HM(i,1),HM(i,2),HM(i,3)))+...
            abs(f3(HM(i,1),HM(i,2),HM(i,3)));
    end
end

```

```

        end
    end
    best=find(Fitness==min(Fitness));
    solusi=HM(best(1),:);
    konvergensi(1)=min(Fitness);
    ikon=0;
    if numvar==2
        prnt={sprintf('%5s %20s %40s %42s','iter','x','y','Fitness')
            sprintf('%5d %20.16f %24.16f
%28.16e',0,solusi(1),solusi(2),konvergensi(1))};
    elseif numvar==3
        prnt={sprintf('%5s %20s %40s %40s
%42s','iter','x','y','z','Fitness')
            sprintf('%5d %20.16f %24.16f %24.16f
%28.16e',0,solusi(1),solusi(2),solusi(3),konvergensi(1))};
    end
    set(handles.listbox1,'string',char(prnt));
    for iter=1:Numiter
        for m=1:HMS
            %HM Consideration
            for i=1:numvar
                r1=rand;
                if r1<=HMCR
                    xbaru(1,i)=HM(ceil(rand*HMS),i);
                else
                    xbaru(1,i)=HM(ceil(rand*HMS*numvar));
                end
            end
            end
            %Mutasi Operator
            [Fitnessurut, index]=sort(Fitness);
            xbest=HM(index(1),:);
            r1=ceil(rand*HMS);
            r2=ceil(rand*HMS);
            while r2==r1
                r2=ceil(rand*HMS);
            end
            xbaru=xbaru+F*(a1*rand*(xbest-xbaru)+a2*rand*(HM(r1,:)-
HM(r2,:)));
            if numvar==2
                Fitbaru=abs(f1(xbaru(1),xbaru(2)))+...
                    abs(f2(xbaru(1),xbaru(2)));
            elseif numvar==3
                Fitbaru=abs(f1(xbaru(1),xbaru(2),xbaru(3)))+...
                    abs(f2(xbaru(1),xbaru(2),xbaru(3)))+...
                    abs(f3(xbaru(1),xbaru(2),xbaru(3)));
            end
            end
            %Pergantian solusi
            if Fitbaru<Fitness(m)
                HM(m,:)=xbaru;
                Fitness(m)=Fitbaru;
            end
        end
    end
    %update F

```

```

F=F-(Fawal-Fakhir)/Numiter;
%plot
konvergensi(iter+1)=min(Fitness);
if konvergensi(iter+1)~=konvergensi(iter)
    ikon=iter;
end
axes(handles.axes1);
plot(0:iter,konvergensi,'b','LineWidth',2);

line(ikon,konvergensi(ikon+1),'Marker','s','Markersize',8,'Markerfacecolor','r');
grid on
xlabel('Iterasi');ylabel('Fitness');
set(handles.axes1,'Ylim',[-konvergensi(1)/8 konvergensi(1)]);
%print solusi
best=find(Fitness==min(Fitness));
solusi=HM(best(1),:);
if numvar==2
    prnt={get(handles.listbox1,'string');
        sprintf('%5d %20.16f %24.16f
%28.16e',iter,solusi(1),solusi(2),konvergensi(iter+1))};
elseif numvar==3
    prnt={get(handles.listbox1,'string');
        sprintf('%5d %20.16f %24.16f %24.16f
%28.16e',iter,solusi(1),solusi(2),solusi(3),konvergensi(iter+1))};

end

set(handles.listbox1,'string',char(prnt),'value',iter+2,'userdata',
,{ikon,konvergensi});
pause(0.0001);
if min(Fitness)==0
    break;
end
end
set(handles.text17,'string',[num2str(toc) ' detik']);
set(handles.text19,'string',num2str(ikon));

```

Tombol Reset

```

set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit5,'string','');
set(handles.edit6,'string','');
set(handles.edit7,'string','');
set(handles.edit8,'string','');
set(handles.edit9,'string','');
set(handles.edit10,'string','');
set(handles.edit11,'string','');
set(handles.edit12,'string','');
set(handles.edit13,'string','');
cla(handles.axes1,'reset');
axes(handles.axes1);
set(handles.axes1,'XLim',[0 1000],'YLim',[-0.2 1]);

```



```

grid on
xlabel('Iterasi');ylabel('Fitness');
numvar=get(handles.popupmenu1,'value')+1;
if numvar==2
    set(handles.listbox1,'string',sprintf('%5s %20s %40s
%42s','iter','x','y','Fitness'),'value',1);
elseif numvar==3
    set(handles.listbox1,'string',sprintf('%5s %20s %40s %40s
%42s','iter','x','y','z','Fitness'),'value',1);
end
set(handles.listbox1,'userdata',[]);
set(handles.text17,'string','0 detik');
set(handles.text19,'string','0');

```

Tombol Simpan

```

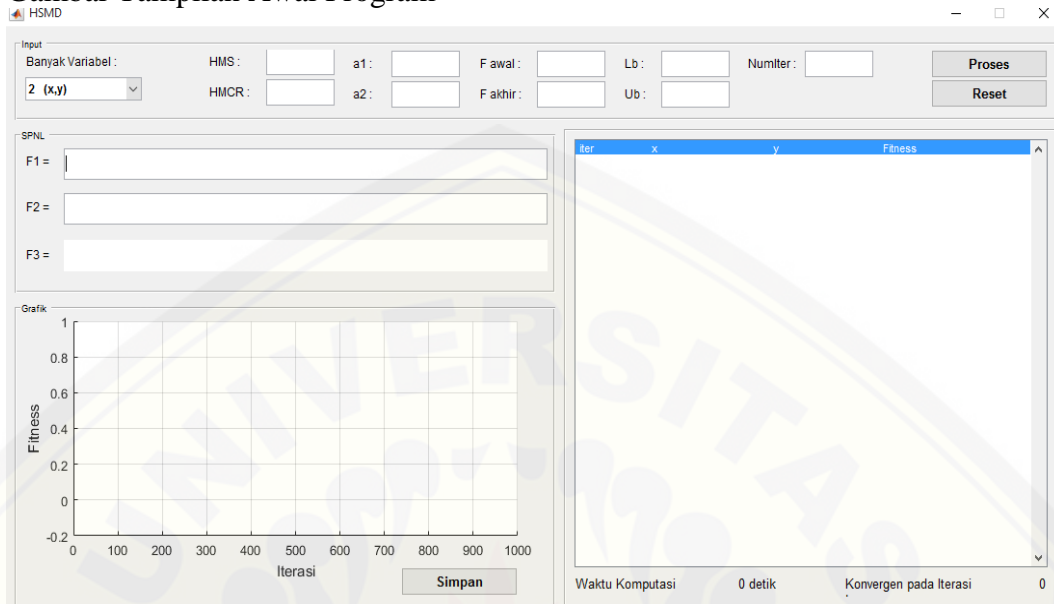
data=get(handles.listbox1,'userdata');
if ~isempty(data)
    [FileName,FilePath] = uiputfile('*.jpg','Save Plot As');
    if FileName~=0
        ikon=data{1};
        konvergensi=data{2};
        figure
        plot(0:length(konvergensi)-
1,konvergensi,'b','LineWidth',2);

        line(ikon,konvergensi(ikon+1),'Marker','s','Markersize',8,'Markerf
acecolor','r');
        grid on
        xlabel('Iterasi');ylabel('Fitness');
        ylim([-konvergensi(1)/8 konvergensi(1)]);
        saveas(gcf,fullfile(FilePath,FileName));
        close(gcf);
    end
end
end

```

B. Tampilan Program

Gambar Tampilan Awal Program



Gambar Hasil *Running* Program

