PENERAPAN ALGORITMA *CAT SWARM OPTIMIZATION* (CSO) PADA PENYELESAIAN SISTEM PERSAMAAN NON-LINIER

SKRIPSI

Oleh
Nur Achmad Baihaki
111810101029

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

PENERAPAN ALGORITMA *CAT SWARM OPTIMIZATION* (CSO) 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
Nur Achmad Baihaki
111810101029

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Ibunda Hamida, Ayahanda Buanto, dan sekeluarga tercinta yang telah memberikan do'a, kasih sayang dan motivasi yang sangat berharga bagi putra tercintanya;
2. Bapak Yusron Moeryanto, Ibu Rini Indriyanti, dan sekeluarga yang telah memberikan dorongan dan motivasi dalam menyelesaikan skripsi ini;
3. Karina Aprilia Anugrah Sari, yang telah banyak membantu dan memberi semangat dalam penyelesaian skripsi ini;
4. Guru-guruku sejak taman kanak – kanak sampai dengan perguruan tinggi;
5. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, SMA Negeri 1 Probolinggo, SMP Negeri 10 Probolinggo, SD Negeri Pesisir 01 Sumberasih Probolinggo dan SD Negeri Rogotrunan 03 Lumajang.

MOTTO

Pekerjaan hebat tidak dilakukan dengan kekuatan, tapi dengan ketekunan dan kegigihan.
(Thomas Szasz)*)

Keberhasilan adalah kemampuan untuk melewati dan mengatasi dari satu kegagalan ke kegagalan berikutnya tanpa kehilangan semangat.
(Winston Chuchill)**)

---

*) Kata Bijak Motivasi dari Tokoh Dunia https://galeriabiee.wordpress.com/kata-kata-bijak/kata-bijak-motivasi-dari-tokoh-dunia/

**) 70 Kata-Kata Bijak dari Tokoh Terkenal di Dunia http://joko-motivasi.blogspot.co.id/2011/06/70-kata-kata-bijak-dari-tokoh-terkenal.html

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Nur Achmad Baihaki

NIM : 111810101029

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penerapan Algoritma *Cat Swarm Optimization* (CSO) 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, 17 Juni 2016

Yang menyatakan,

Nur Achmad Baihaki

NIM. 111810101029

SKRIPSI

PENERAPAN ALGORITMA *CAT SWARM OPTIMIZATION* (CSO) PADA PENYELESAIAN SISTEM PERSAMAAN NON-LINIER

Oleh

Nur Achmad Baihaki

NIM 111810101029

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 *Cat Swarm Optimization* (CSO) pada Penyelesaian Sistem Persamaan Non-Linier" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Dosen Pembimbing Utama,	Dosen Pembimbing Anggota,
M. Ziaul Arif, S.Si., M.Sc.	Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP. 198501112008121002	NIP. 197211291998021001
Penguji I,	Penguji II,
Kusbudiono, S.Si., M.Si.	Dian Anggraeni, S.Si., M.Si.
NIP. 197704302005011001	NIP. 198202162006042002

Mengesahkan

Dekan,

Dr. Sujito, Ph. D

NIP. 196102041987111001

RINGKASAN

Penerapan Algoritma *Cat Swarm Optimization* (CSO) pada Penyelesaian Sistem Persamaan Non-Linier, Nur Achmad Baihaki, 111810101029; 2016: 45 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

Sistem persamaan non-linier adalah himpunan dari beberapa persamaan non-linier. Umumnya, sistem persamaan non-linier sulit atau bahkan tidak dapat diselesaikan menggunakan metode analitik. Terdapat banyak cara untuk mencari solusi sistem persamaan non-linier. Solusi sistem persamaan non-linier adalah nilai-nilai variabel bebas yang mengakibatkan sistem persamaan bernilai nol. Salah satu metode yang dapat digunakan untuk menyelesaikan sistem persamaan non-linier yaitu metode optimasi *metaheuristic*. Salah satu contoh dari metode optimasi *metaheuristic* adalah algoritma *Cat Swarm Optimization*.

Penelitian ini dimulai dari menentukan beberapa sistem persamaan non-linier dari beberapa artikel yang akan diteliti berupa sistem persamaan non-linier dua variabel dan sistem persamaan non-linier tiga variabel. Kemudian menentukan nilai parameter yang digunakan yaitu SMP, SRD, CDC, dan MR serta nilai c, vmax, dan jumlah populasi kucing. Setelah itu, mencari solusi sistem persamaan non-linier menggunakan algoritma *Cat Swarm Optimization*. Kemudian menguji keakuratan dari algoritma *Cat Swarm Optimization* dengan membandingkan solusi sistem persamaan non-linier menggunakan algoritma *Cat Swarm Optimization* dengan solusi sistem persamaan non-linier dari beberapa artikel yang dikutip.

Hasil penyelesaian sistem persamaan non-linier menggunakan algoritma *Cat Swarm Optimization* menunjukkan bahwa solusi dari sistem persamaan non-linier yang dihasilkan oleh algoritma *Cat Swarm Optimization* mendekati solusi eksak. Jika dibandingkan dengan solusi sistem persamaan non-linier dari beberapa metode yang telah diteliti, penyelesaian sistem persamaan non-linier

menggunakan algoritma *Cat Swarm Optimization* (CSO) mendekati solusi yang dihitung secara langsung dengan matlab. Jadi, dapat dikatakan bahwa algoritma *Cat Swarm Optimization* memiliki akurasi yang baik untuk menyelesaikan sistem persamaan non-linier.

Penelitian ini juga dilakukan simulasi program guna mengetahui keakuratan program jika beberapa parameter diubah dan mengetahui pengaruh dari setiap parameter terhadap waktu atau iterasi program. Dari simulasi yang telah dilakukan, perbedaan nilai MR, SMP, dan *nPop* dapat mempengaruhi waktu yang diperlukan untuk mencari solusi sistem persamaan non-linier. Semakin besar nilai MR, maka semakin sedikit waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier. Sebaliknya, semakin kecil nilai MR, maka semakin banyak waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier. Sedangkan semakin besar nilai SMP dan *nPop*, maka semakin banyak waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier. Sebaliknya, semakin kecil nilai SMP dan *nPop*, maka semakin sedikit waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier.

PRAKATA

Puji syukur kehadirat Allah SWT. Atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Penerapan Algortima *Cat Swarm Optimization* (CSO) 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. M. Ziaul Arif, S.Si., M.Sc, selaku Dosen Pembimbing Utama, Ahmad Kamsyakawuni, S.Si., M. Kom, selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Drs. Moh. Hasan M.Sc., Ph.D. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
3. Ibunda Hamida dan Ayahanda Buanto yang telah memberikan do'a dan motivasi demi terselesaikannya skripsi ini;
4. Bapak Yusron Moeryanto dan Ibu Rini Indriyanti sekeluarga yang telah memberikan motivasi dan dorongan demi terselesaikannya skripsi ini;
5. Karina Aprilia Anugrah Sari yang telah memberi semangat dan dukungan;
6. Teman – teman angkatan 2011 (KRAMAT 11) atas dukungan, keceriaan, dan canda – tawa yang telah diberikan;
7. Teman – teman kontrakan Jawa VII (5men) atas dukungan dan nasehat yang telah diberikan;
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, 17 Juni 2016

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pada era modern sekarang ini, matematika adalah ilmu hitung yang sangat diperlukan oleh berbagai pihak. Salah satu kajian di dalam matematika adalah sistem persamaan. Terdapat dua jenis sistem persamaan yaitu sistem persamaan linier dan sistem persamaan non-linier. Sistem persamaan non-linier merupakan salah satu permasalahan yang cukup sulit untuk diselesaikan.

Dalam menyelesaikan suatu sistem persamaan, secara umum terdapat dua metode yang digunakan yaitu metode analitik atau eksak dan metode numerik. Pada umumnya, suatu sistem persamaan yang memiliki bentuk sederhana dapat diselesaikan dengan menggunakan metode analitik. Akan tetapi penggunaan metode analitik sangat sulit untuk mencari penyelesaian dari suatu sistem persamaan yang bersifat kompleks seperti sistem persamaan non-linier. Oleh karena itu, dalam menyelesaikan sistem persamaan non-linier, diperlukan suatu algoritma metode numerik untuk mencari solusi dari sistem persamaan tersebut.

Banyak algoritma yang telah digunakan untuk menyelesaikan sistem persamaan non-linier. Wahyudianto (2014) meneliti solusi dari hasil perbandingan Metode Regula Falsi dengan Metode Secant. Dari penelitian tersebut disimpulkan bahwa dalam menyelesaikan sistem persamaan non-linier, metode Secant lebih baik daripada metode Regula Falsi. Selanjutnya, pencarian solusi sistem persamaan non-linier dengan menggunakan metode Jaringan Syaraf Tiruan Hopfield telah dilakukan oleh Devi (2011) dan didapat kesimpulan bahwa metode Jaringan Syaraf Tiruan Hopfield selalu konvergen untuk sembarang nilai awal, tetapi waktu yang dibutuhkan Jaringan Syaraf Tiruan Hopfield untuk menghasilkan solusi dengan perangkat komputer tidak signifikan. Sedangkan metode yang paling sering digunakan untuk menyelesaikan sistem persamaan non-linier yaitu metode Newton-Raphson seperti pada penelitian Nasiha tahun 2008, hasil kajian dari Noor dan Waseem tahun 2009, penelitian Ozel tahun 2010,

dan penelitian tahun 2007 oleh Darvishi dan Barati. Selain itu, Darvishi dan Barati pada tahun 2007 juga meneliti tentang modifikasi dari metode Newton-Raphson, yaitu *A Fourth-Order Method From Quadrature Formula*. Metode Newton-Raphson tergolong cepat untuk menyelesaikan sistem persamaan non-linier. Akan tetapi, masih terdapat kekurangan pada metode Newton–Raphson, yaitu metode ini mengharuskan untuk menghitung turunan dari fungsi $f(x_n)$. Sedangkan tidak semua fungsi dapat dicari turunannya dengan mudah. Kekurangan yang lainnya adalah penetapan harga awal (x_n) yang sulit dan tidak selalu menemukan akar. Oleh karena itu, dibutuhkan suatu algoritma atau metode yang bisa mengatasi hal tersebut.

Beberapa metode numerik yang merupakan pengembangan metode Newton-Raphson telah diteliti oleh banyak matematikawan, sehingga didapatkan suatu metode yang mampu mengeliminasi kelemahan metode Newton-Raphson. Akan tetapi, penerapan metode-metode tersebut tidak lepas dari permasalahan konvergenitas metode dalam pengambilan nilai awal yang akurat. Untuk mengatasi hal tersebut, permasalahan pencarian solusi sistem persamaan non-linier dapat dipandang sebagai suatu permasalahan teknik optimasi yang membutuhkan suatu algortitma *computational intelligence*. *Computational intelligence* adalah riset penelitian yang digunakan dalam bidang teknik optimasi berdasarkan perhitungan cerdas dari suatu langkah yang terstruktur (Chu *et al.*, 2006). Contoh algoritma yang tergabung dalam *computational intelligence* adalah *Ant Colony Optimization* (ACO), *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), *Hybrid Artificial Glowworm Swarm Optimization* (HAGSO), *Cat Swarm Optimization* (CSO), dan *Simulated Annealing* (SA). Pada tahun 2012, Rosita meneliti penerapan salah satu dari *swarm intelligence* yaitu PSO untuk menyelesaikan sistem persamaan non-linier. Sedangkan, Yang *et al.* tahun 2010 menerapkan HAGSO pada topik permasalahan yang sama.

Secara khusus, *Cat Swarm Optimization* adalah algoritma baru yang diusulkan oleh Chu *et al*. (2006). *Cat Swarm Optimization* merupakan suatu algoritma yang berdasarkan tingkah laku kucing (Chu *et al.*, 2006). Algoritma *Cat Swarm Optimization* termasuk dalam kelompok *swarm intelligence*. Keuntungan

menggunakan salah satu dari kelompok *swarm intelligence* dalam menyelesaikan sistem persamaan non-linier adalah solusi yang dihasilkan mendekati solusi eksak. Oleh karena itu, pada penelitian ini, penulis tertarik untuk meneliti penerapan salah satu dari kelompok *swarm intelligence* yaitu *Cat Swarm Optimization* untuk menyelesaikan sistem persamaan non-linier. Dengan menggunakan algoritma *Cat Swarm Optimization*, diharapkan dapat mengatasi kekurangan-kekurangan yang ada pada metode Newton-Raphson serta metode-metode pengembangannya dan menjadi khazanah baru dalam menyelesaikan sistem persamaan non-linier. Selain itu, penulis juga ingin mengetahui konvergenitas dan keakuratan algoritma *Cat Swarm Optimization* dalam menyelesaikan sistem persamaan non-linier.

1.2 Rumusan Masalah

Berdasarkan pada latar belakang yang telah diuraikan diatas, maka permasalahan yang akan dibahas dalam penelitian ini yaitu:

a. bagaimana menyelesaikan suatu sistem persamaan non-linier menggunakan algoritma *Cat Swarm Optimization*?;
b. bagaimana hasil penyelesaian dan konvergenitas algoritma *Cat Swarm Optimization* jika dibandingkan dengan metode Newton-Raphson, *A Fourth-Order Method From Quadrature Formula* dan *Hybrid Artificial Glowworm Swarm Optimization*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

a. mengetahui cara menyelesaikan sistem persamaan non-linear dengan menggunakan algoritma *Cat Swarm Optimization*;
b. mengetahui hasil penyelesaian dengan menggunakan algoritma *Cat Swarm Optimization* jika dibandingkan dengan metode Newton-Raphson, *A Fourth-Order Method From Quadrature Formula* dan *Hybrid Artificial Glowworm Swarm Optimization*.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan wawasan baru cara menyelesaikan sistem persamaan non-linier yang dipandang sebagai suatu permasalahan teknik optimasi dengan menggunakan algoritma *Cat Swarm Optimization*. Penelitian ini sekaligus memberikan alternatif untuk menyelesaikan sistem persamaan non-linier disamping metode Newton-Raphson dan metode-metode pengembangannya.

BAB 2. TINJAUAN PUSTAKA

2.1 Persamaan Linier

Persamaan linier adalah suatu persamaan yang memuat paling sedikit satu variabel (peubah) yang mempunyai pangkat bulat positif dan pangkat tertinggi variabelnya satu. Bentuk umum dari persamaan linier adalah

$$ax + b = 0$$

dengan $a, b \in R$ dan $a \neq 0$. x disebut variabel, sedangkan a dan b disebut konstanta.

Ada beberapa jenis persamaan linier, diantaranya adalah persamaan linier dengan satu variabel dan persamaan linier lebih dari dua variabel. Persamaan linier dengan satu variabel adalah suatu persamaan yang memuat satu variabel. Persamaan linier dengan satu variabel mempunyai dua kemungkinan penyelesaian, yaitu persamaan linier satu variabel yang tidak mempunyai penyelesaian dan persamaan linier satu variabel yang mempunyai satu penyelesaian saja. Contoh dari persamaan linier satu variabel adalah $3x + 6 = 9$.

Sedangkan persamaan linier lebih dari dua variabel adalah suatu persamaan yang memiliki dua atau lebih variabel dimana variabel (peubahnya) berpangkat satu.. Himpunan semua penyelesaian dari suatu persamaan linier dua variabel disebut Himpunan Penyelesaian atau yang biasa disingkat HP. Contoh dari persamaan linier dengan dua variabel adalah $x + y = 5$.

2.2 Sistem Persamaan Linier

Sistem persamaan linier adalah salah satu model dari permasalahan matematika yang banyak dijumpai dalam berbagai ilmu, yaitu matematika, statistika, fisika, biologi, teknik, ilmu–ilmu sosial, dan bisnis. Sistem persamaan linier muncul secara langsung dari berbagai masalah di kehidupan nyata. Sistem persamaan linier terdiri atas sejumlah berhingga persamaan linier dalam jumlah berhingga variabel. Untuk menyelesaikan suatu sistem persamaan linier adalah

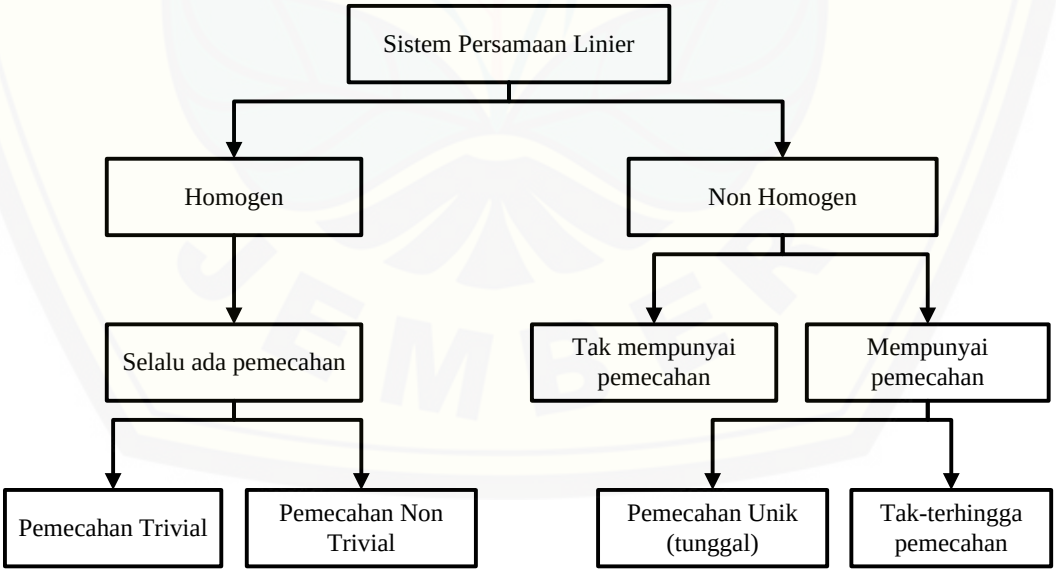
dengan mencari nilai–nilai variabel–variabel tersebut yang memenuhi semua persamaan linier yang diberikan (Sahid, 2005).

Sistem persamaan linier merupakan himpunan berhingga dari persamaan – persamaan linier dalam peubah $x_1, x_2, x_3, \dots, x_n$. Bentuk umum sistem persamaan linier dalam m persamaan dengan n bilangan yang tidak diketahui adalah sebagai berikut.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

Tidak semua sistem persamaan linier mempunyai pemecahan. Sistem persamaan yang tidak mempunyai pemecahan dikatakan tak konsisten (*inconsistent*). Jika ada setidak–tidaknya satu pemecahan, maka sistem persamaan tersebut dinamakan konsisten (*consistent*) (Kusumawati, 2009).

Menurut Kusumawati (2009), garis besar skema susunan sistem persamaan linier adalah sebagai berikut.



Gambar 2.1 Skema Susunan Sistem Persamaan Linier

Sistem persamaan linier dikatakan homogen jika semua suku konstan sama dengan nol. Bentuk umum sistem persamaan linier homogen dalam m persamaan dengan n bilangan yang tidak diketahui adalah sebagai berikut.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= 0 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= 0 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= 0 \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= 0 \end{aligned}$$

Sistem persamaan linier homogen adalah sistem yang konsisten, karena $x_1 = 0, x_2 = 0, \dots, x_n = 0$ selalu merupakan pemecahan. Pemecahan tersebut dinamakan pemecahan trivial (*trivial solution*). Jika ada pemecahan lain, maka pemecahan tersebut dinamakan pemecahan tak–trivial (*non trivial solution*).

Pada dasarnya, terdapat dua kelompok metode yang digunakan untuk menyelesaikan sistem persamaan linier, yaitu metode langsung dan metode tidak langsung atau metode iteratif. Metode langsung adalah metode yang digunakan untuk mencari penyelesaian suatu sistem persamaan linier dalam langkah berhingga. Metode ini disarankan untuk pemakaian secara umum dan dijamin keberhasilannya. Sedangkan metode tidak langsung atau metode iteratif adalah metode yang bermula dari suatu hampiran awal kemudian memperbaiki hampiran dalam jumlah tak berhingga namun dengan langkah konvergen. Metode ini digunakan untuk menyelesaikan sistem persamaan linier berukuran besar (Sahid, 2005).

2.3 Persamaan Non-Linier

Menurut Devi (2011), persamaan non-linier adalah persamaan yang melibatkan fungsi transenden dan fungsi non-transenden. Fungsi transenden tersebut meliputi:

a. Fungsi sinus, cosinus, eksponensial, logaritma, dan lainnya

$$\cos 2x + \tan^{-1} x - 2 = 0$$

b. Kecepatan ke atas sebuah roket dapat dihitung dengan persamaan :

$$v = u \ln \left| \frac{m_0}{m_0 - qt} \right| - gt$$

dengan v kecepatan ke atas, u kecepatan saat bahan bakar dikeluarkan, m_0 massa awal roket, q laju pemakaian bahan bakar, g percepatan gravitasi, t waktu

c. Suatu arus osilasi dalam rangkaian listrik

$$I = 10e^{-t}\sin(2\pi t)$$

dengan t waktu, dan I arus

Menurut Devi (2011), selain fungsi transenden, persamaan non-linier juga melibatkan fungsi non-transenden. Persamaan tersebut adalah persamaan polinomial. Bentuk umum dari persamaan polinomial adalah

$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 = 0$$

Contoh dari persamaan polinomial adalah:

a. Satu variabel (x)

$$x^2 + 4x - 3 = 0$$

b. Dua variabel (x dan y)

$$2x^2 - 5xy - y = 0$$

Untuk menyelesaikan persamaan non-linier dilakukan secara iteratif. Hingga saat ini sudah banyak metode yang dilakukan untuk menyelesaikan persamaan non-linier. Metode tersebut dapat dikelompokkan menjadi dua kelompok, yaitu Metode Tertutup dan Metode Terbuka (Munir, 2003).

Metode tertutup disebut juga metode Pengurung (*bracketing method*). Metode yang termasuk dalam golongan ini mencari akar didalam selang [a,b] sudah pasti berisi minimal sebuah akar penyelesaian. Karena itu, metode jenis ini selalu berhasil menemukan akar penyelesaian. Dalam metode tertutup iterasinya selalu konvergen ke akar penyelesaiannya sehingga metode tertutup juga dinamakan metode konvergen. Sedangkan metode terbuka adalah metode yang tidak memerlukan selang [a,b] yang mengandung akar penyelesaian. Metode terbuka memerlukan hampiran awal penyelesaian kemudian menggunakan prosedur iterasi untuk menghitung hampiran akar penyelesaian yang baru. Setiap iterasi, hampiran akar yang lama digunakan untuk menghitung hampiran akar yang baru. Metode terbuka tidak selalu berhasil menemukan akar penyelesaiannya (Munir, 2003).

2.4 Sistem Persamaan Non-Linier

Selain sistem persamaan linier, dalam permasalahan matematika juga terdapat sistem persamaan non-linier. Sistem persamaan non-linier merupakan sistem persamaan yang mempunyai bentuk persamaan yang kompleks dimana persamaan tersebut tidak dapat dipecahkan secara analitik. Apabila suatu persamaan tidak dapat diselesaikan dengan metode analitik, maka persamaan tersebut dapat diselesaikan menggunakan metode numerik (Munir, 2003).

Menurut Devi (2011), sistem persamaan non-linier adalah suatu sistem dengan n buah persamaan non-linier yang harus diselesaikan secara simultan dalam suatu sistem. Sistem persamaan non-linier merupakan kumpulan dari dua atau lebih dari persamaan – persamaan non-linier. Bentuk umum dari sistem persamaan non-linier adalah sebagai berikut.

$$f(x) = \begin{cases} f_1(x) = 0 \\ f_2(x) = 0 \\ \quad \vdots \\ f_n(x) = 0 \end{cases}$$

Salah satu contoh dari sistem persamaan non-linier adalah sebagai berikut (Nasiha, 2008).

$$\begin{aligned} x + \cos(xy) - z^2 - 1{,}1 &= 0 \\ x^2 - 10y - e^{yz} + 0{,}8 &= 0 \\ xz + y^2 - \mathrm{z} - 0{,}3 &= 0 \end{aligned} \tag{2.1}$$

Penyelesaian sistem persamaan non-linier terdiri dari himpunan nilai–nilai variabel atau peubah yang secara simultan memenuhi semua persamaan tersebut.

2.5 *Cat Swarm Optimization* (CSO)

Cat Swarm Optimization (CSO) adalah algoritma yang dibuat berdasarkan perilaku dari sekelompok kucing. CSO diterapkan untuk menyelesaikan masalah optimasi, langkah pertama adalah menentukan berapa banyak individu yang akan digunakan. Individu yang dimaksud adalah sekawanan kucing (Chu *et al.*, 2006).

CSO terdiri dari dua sub-model, yaitu *seeking mode* dan *tracing mode*. Untuk menggabungkan dua mode tersebut ke dalam satu algoritma, maka ditetapkan *Mixture Ratio* (MR) dari gabungan *seeking mode* dengan *tracing mode*.

Jika MR sama dengan 0, semua kucing akan dimasukkan ke dalam proses *seeking mode*. Jika MR sama dengan 1, semua kucing akan dimasukkan ke dalam proses *tracing mode*. Pada umumnya kucing memiliki waktu istirahat lebih banyak, maka nilai MR sangat kecil sehingga banyak kucing yang masuk dalam proses *seeking mode* (Lin *et al.*, 2015).

2.5.1 *Seeking Mode*

Submodel ini digunakan untuk memodelkan situasi kucing yang sedang istirahat, melihat sekeliling, dan mencari posisi berikutnya untuk bergerak. Dalam *seeking mode* ini, didefinisikan empat faktor penting, yaitu : *seeking memory pool* (SMP), *seeking range of the selected dimension* (SRD), *counts of dimension to change* (CDC), dan *self-position considering* (SPC) (Chu *et al.*, 2006).

SMP digunakan untuk mendefinisikan ukuran memori pencarian pada setiap kucing, yang menunjukkan titik yang dicari oleh kucing tersebut. Kucing akan memilih sebuah titik dari gabungan memori berdasarkan aturan yang akan dijelaskan kemudian (Chu *et al.*, 2006).

SRD menunjukkan rentang perpindahan dari dimensi terpilih. Dalam *seeking mode* ini, jika dimensi diputuskan berpindah, selisih nilai baru dan lama tidak boleh melebihi suatu rentang yang didefinisikan oleh SRD (Chu *et al.*, 2006).

CDC memperlihatkan banyaknya dimensi yang akan berubah. Faktor–faktor tersebut memainkan peran yang sangat penting dalam *seeking mode* ini (Chu *et al.*, 2006).

SPC adalah variabel *Boolean* (bernilai benar atau salah), yang menetapkan suatu titik yang pernah menjadi posisi kucing, akan dijadikan salah satu kandidat posisi untuk bergerak. Tidak masalah nilai SPC benar atau salah, nilai SMP tidak akan terpengaruh. Langkah *seeking mode* dapat dideskripsikan dalam 5 langkah berikut (Chu *et al.*, 2006) :

Langkah 1 : Bangkitkan j tiruan dari posisi saat ini dari kucing k, dimana j = SMP. Jika nilai SPC benar, maka j = (SMP – 1), lalu pertahankan posisi saat ini sebagai salah satu kandidat.

Langkah 2 : Tambahkan atau kurangkan dengan SRD dalam bentuk persen secara acak dari nilai sekarang dan ganti yang lama berdasarkan persamaan (2.2).

$$x_{j,d} \text{ new} = (1 + (rand).(SRD)).(x_{j,d}), \quad (2.2)$$

dimana $rand = [-1,1]$, $d = 1,2,\dots,M$

Langkah 3 : Hitung nilai fungsi *fitness* (FS) dari semua titik kandidat. Berikut adalah hitungan matematis dari nilai fungsi *fitness*.

$$FS = \begin{cases} minimize\ abs\left(f_1(x_1, x_2, \dots, x_n)\right) + \\ minimize\ abs\left(f_2(x_1, x_2, \dots, x_n)\right) + \\ \vdots \\ minimize\ abs\left(f_n(x_1, x_2, \dots, x_n)\right) \end{cases} \quad (2.3)$$

Langkah 4 : Jika semua FS tidak benar-benar sama, hitung probabilitas terpilih dari setiap kandidat titik dari persamaan (2.4).

$$P_i = \frac{|FS_i - FS_b|}{FS_{\max} - FS_{\min}}, \text{ dimana } 0 < i < j \quad (2.4)$$

Jika tidak atur semua probabilitas terpilih dari tiap titik kandidat sama dengan 1.

Langkah 5 : Pilih titik secara acak atau melalui *Roulette Wheel* (RW) untuk bergerak dari kandidat titik, dan gantilah posisi kucing.

Jika tujuan fungsi *fitness* adalah untuk menemukan solusi minimal, $FS_b = FS_{max}$, jika tidak $FS_b = FS_{min}$.

2.5.2 *Tracing Mode*

Tracing mode adalah submodel untuk memodelkan sebuah kasus dari kucing dalam menemukan targetnya.

Sekali kucing masuk ke dalam *tracing mode*, kucing tersebut bergerak menurut kecepatannya untuk setiap dimensi. Perilaku kucing dalam *tracing mode* dapat dideskripsikan dalam 3 langkah berikut (Chu *et al.*, 2006):

Langkah 1 : Perbarui kecepatan kucing untuk setiap dimensi ($v_{k,d}$ new) berdasarkan persamaan (2.5).

$$v_{k,d} \text{ new} = v_{k,d} + (r).(c).\left(x_{best,d} - x_{k,d}\right) \quad (2.5)$$

dimana $d = 1,2,\dots,M$

$x_{best\ ,d}$ adalah posisi kucing, yang mempunyai nilai *fitness* terbaik ; $x_{k,d}$ adalah posisi kucing ke-k. c konstan dan r adalah nilai acak dalam interval [0,1].

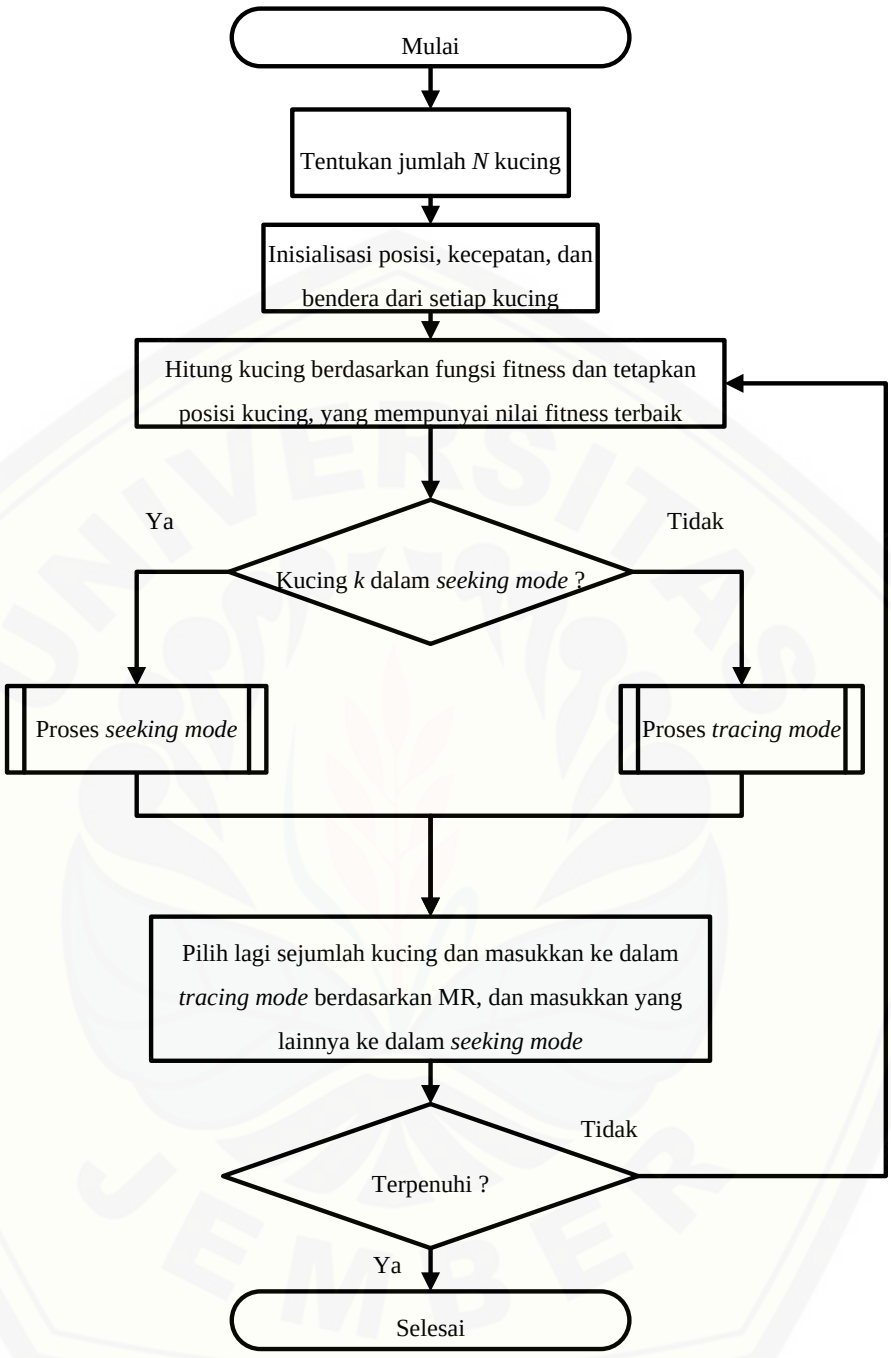
Langkah 2 : Periksa apakah kecepatannya berada dalam range kecepatan maksimum. Apabila kecepatan baru melebihi batas rentang, tetapkan nilai sama dengan batas.

Langkah 3 : Perbarui posisi kucing menurut persamaan (2.6).

$$x_{k,d} \text{ new} = x_{k,d} + v_{k,d} \tag{2.6}$$

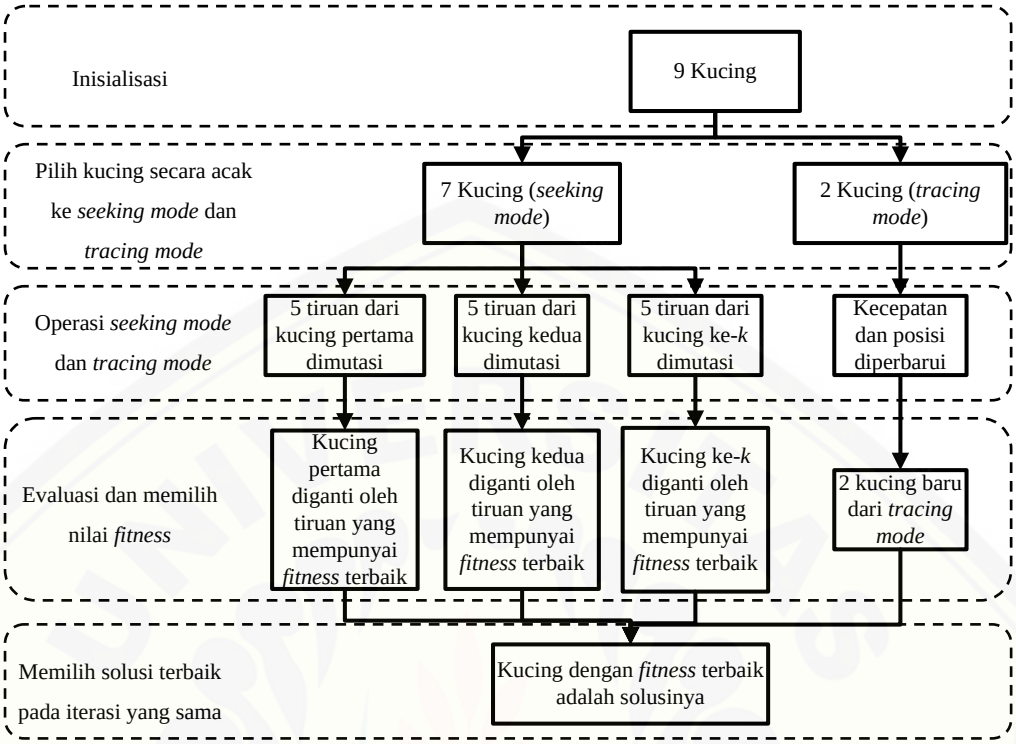
Langkah–langkah CSO adalah sebagai berikut (Chu *et al.*, 2006).

a. Tentukan jumlah *N* kucing dalam proses.
b. Sebarkan secara acak kucing tersebut ke dalam ruang solusi *M*-dimensi dan tetapkan nilai secara acak yang terdapat di dalam barisan kecepatan maksimum, untuk menjadi kecepatan setiap kucing. Lalu pilih sejumlah kucing secara acak dan masukkan ke dalam *tracing mode* sesuai dengan MR, dan yang lainnya masukkan ke dalam *seeking mode*.
c. Hitung nilai *fitness* dari setiap kucing dengan memasukkan nilai posisi kucing ke dalam fungsi *fitness* dan simpan kucing terbaik ke dalam memori. Sebagai catatan bahwa yang perlu disimpan adalah posisi kucing terbaik (x_{best}) karena mewakili solusi terbaik.
d. Pindahkan kucing–kucing tersebut berdasarkan benderanya, jika kucing ke-*k* berada dalam *seeking mode*, terapkan kucing tersebut ke dalam langkah *seeking mode*, jika sebaliknya terapkan dalam langkah *tracing mode*.
e. Pilih lagi sejumlah kucing dan masukkan ke dalam *tracing mode* berdasarkan MR, kemudian masukkan kucing lainnya ke dalam *seeking mode*.
f. Perhatikan kriteria pemberhentiannya. Jika telah terpenuhi, maka hentikan program. Jika sebaliknya maka ulangi langkah (c) sampai dengan langkah (e).



Gambar 2.2. Flowchart CSO (Tsai *et al.*, 2008)

Menurut Hadi dan Sabah (2015), skema dalam pengerjaan *Cat Swarm Optimization* (CSO) adalah sebagai berikut.



Gambar 2.3 Skema Pengerjaan *Cat Swarm Optimization* (CSO)

2.6 Metode Numerik

Dalam permasalahan matematika terdapat dua metode untuk menyelesaikannya. Metode tersebut adalah metode analitik dan metode numerik. Metode numerik merupakan beberapa cara yang digunakan untuk memecahkan masalah matematis agar permasalahan tersebut dapat dipecahkan dengan operasi perhitungan. Meskipun terdapat beberapa metode numerik, tetapi ada satu kesamaan ciri yaitu semua metode numerik merupakan perhitungan yang menjemukan. Meskipun demikian, peranan metode numerik dalam memecahkan permasalahan rekayasa telah meningkat secara dramatis akibat perkembangan komputer digital yang cepat (Chapra dan Canale, 1988).

Nugroho (dalam Wahyudianto, 2014) menyatakan bahwa penyelesaian secara numerik dari suatu persamaan hanya memberikan hampiran yang mendekati solusi eksak dari penyelesaian analitis. Dalam penyelesaian numerik tersebut terdapat beberapa kesalahan terhadap nilai eksak. Kesalahan-kesalahan tersebut adalah:

a. kesalahan bawaan, terjadi karena kekeliruan dalam menyalin data, salah membaca skala, atau karena kurangnya pengertian mengenai hukum–hukum fisik dari data yang diukur.
b. kesalahan pembulatan, terjadi karena tidak diperhitungkan beberapa angka terakhir dari suatu bilangan.
c. kesalahan pemotongan, terjadi karena tidak dilakukannya hitungan sesuai dengan prosedur matematika yang benar.

Metode numerik biasanya digunakan untuk menyelesaikan permasalahan–permasalahan yang diformulasikan secara matematis dengan cara operasi hitungan (aritmatika). Beberapa permasalahan tersebut dapat digambarkan dalam bentuk persamaan matematika. Jika persamaan tersebut memiliki bentuk yang sederhana, maka penyelesaiannya dapat menggunakan metode analitik, tetapi jika bentuk persamaannya sulit diselesaikan dengan metode analitik, maka untuk menyelesaikannya dapat menggunakan metode numerik (Nasiha, 2008).

Teradapat beberapa metode numerik yang digunakan untuk menyelesaikan sistem persamaan non-linier. Metode numerik yang paling sering digunakan adalah Metode Newton–Raphson. Metode Newton–Raphson adalah salah satu metode yang paling populer untuk menghitung hampiran akar–akar dari sistem persamaan non-linier. Sebagai contoh $f(x)$ suatu fungsi diferensiabel pada $[a, b]$. Maka $f(x)$ memiliki kemiringan tertentu dan garis singgung tunggal pada setiap titik di dalam (a, b). Garis singgung di titik $(x_0, f(x_0))$ merupakan pendekatan grafik $f(x)$ di dekat titik $(x_0, f(x_0))$. Jadi, pembuat nilai nol garis singgung tersebut merupakan hampiran pembuat nol $f(x)$ (Sahid, 2005).

2.7 MATLAB

MATLAB adalah bahasa pemrograman level tinggi yang dikhususkan untuk komputasi teknis. Semakin tinggi level bahasa semakin mudah cara menggunakannya. Bahasa dalam MATLAB mengintegrasikan kemampuan komputasi, visualisasi, dan pemrograman dalam suatu lingkungan yang tunggal dan mudah untuk digunakan. Selain itu, MATLAB juga menggunakan konsep array/matrik sebagai standar variabel elemennya. Dalam lingkungan pendidikan

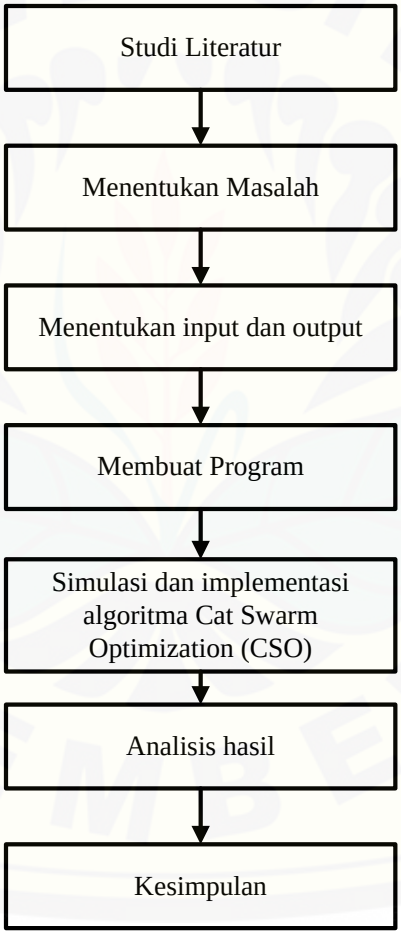
ilmiah, MATLAB menjadi alat pemrograman standar bidang Matematika. Sedangkan dalam lingkungan industri, MATLAB dapat menjadi pilihan paling produktif untuk riset, pengembangan dan analisa.

Dengan adanya MATLAB maka dapat memberikan jawaban sekaligus tantangan. MATLAB juga menyediakan beberapa pilihan untuk dipelajari. Mempelajari metode visualisasi saja, pemrograman saja atau kedua–keduanya. Selain itu, MATLAB juga memberikan keuntungan bagi *programmer–developer program* yaitu untuk menjadi program pembanding yang sangat handal, hal tersebut dapat dilakukan karena kekayaannya akan fungsi matematika, fisika, statistik dan visualisasi (Away, 2010).

BAB 3. METODE PENELITIAN

Dalam penelitian ini, peneliti akan menerapkan *Cat Swarm Optimization* (CSO) pada beberapa contoh sistem persamaan non-linier dari sumber referensi yang dirujuk.

Langkah – langkah yang akan dilakukan dalam penelitian ini, dapat digambarkan dengan diagram alir pada Gambar 3.1.



Gambar 3.1 Skema Metode Penelitian

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

a. Studil iteratur

Studi literatur yang dilakukan pada penelitian ini adalah mempelajari mengenai algoritma *Cat Swarm Optimization* dan sistem persamaan non-linier.

b. Menentukan masalah

Masalah yang akan diteliti pada skripsi ini yaitu suatu sistem persamaan non-linier.

c. Menentukan masukan dan keluaran

Peneliti akan menentukan masukan berupa suatu sistem persamaan yang akan dicari penyelesaiannya, kriteria pemberhentian dan parameter yang dipakai yaitu SMP, SRD, MR, dan CDC, serta nilai *c*, vmax, dan jumlah populasi. Sedangkan keluaran berupa nilai *fitness* terbaik, solusi, dan grafik kekonvergenan.

d. Membuat program

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

e. Simulasi dan implementasi algoritma *Cat Swarm Optimization* (CSO)

Peneliti akan menyelesaikan suatu sistem persamaan non-linier dengan program algoritma *Cat Swarm Optimization* (CSO) yang telah dibuat sehingga diperoleh suatu solusi penyelesaian yang dianggap terbaik.

f. Analisis hasil

Peneliti akan melakukan analisis terhadap hasil keluaran program. Algoritma *Cat Swarm Optimization* (CSO) dianggap baik jika hasil yang diperoleh mendekati atau sama dengan hasil sebenarnya. Dalam hal ini peneliti akan membandingkan hasil penelitian dengan menggunakan beberapa metode dari referensi yang dirujuk sebagai tolak ukur keberhasilan algoritma *Cat Swarm Optimization* (CSO).

g. Kesimpulan

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

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada bab 4, didapatkan kesimpulan sebagai berikut:

a. Algoritma *Cat Swarm Optimization* dapat digunakan untuk menyelesaikan sistem persamaan non-linier dalam beberapa contoh kasus.
b. Algoritma *Cat Swarm Optimization* memiliki akurasi yang baik dalam pencarian solusi sistem persamaan non-linier karena solusi yang dihasilkan mendekati solusi eksak. Hasil yang diperoleh algoritma *Cat Swarm Optimization* dalam pencarian solusi selalu konvergen. Namun, kelemahan dari algoritma *Cat Swarm Optimization* adalah proses perhitungan dalam pencarian solusi sistem persamaan non-linier memerlukan waktu yang relatif lama. Ada tiga parameter yang mempengaruhi waktu proses pencarian solusi, yaitu MR, SMP, dan *nPop*. Semakin besar nilai MR, maka semakin sedikit waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier. Sebaliknya, semakin kecil nilai MR, maka semakin banyak waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier. Sedangkan semakin besar nilai SMP dan *nPop*, maka semakin banyak waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier. Sebaliknya, semakin kecil nilai SMP dan *nPop*, maka semakin sedikit waktu yang diperlukan untuk menemukan solusi sistem persamaan non-linier.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, disarankan kepada peneliti selanjutnya untuk menerapkan algoritma *Cat Swarm Optimization* yang lain untuk menyelesaikan sistem persamaan non-linier, misalnya *Binary Cat Swarm Optimization*, *Parallel Cat Swarm Optimization*, *Average-Inertia Weighted Cat Swarm Optimization*, dan lainnya.

DAFTAR PUSTAKA

Away, G. A. 2010. *The Shortcut of MATLAB Programming*. Bandung: Informatika.

Chapra, S. C. & Canale, R. P. 1988. *Metode Numerik*. Jakarta: Erlangga.

Chu, S. C., Tsai, P. W., & Pan, J. S. 2006. "Cat Swarm Optimization". *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence LNAI 4099*.

Chu, S. C., Tsai, P. W., & Pan, J. S. 2006. "Computational Intelligence Based on the Behavior of Cats". *International Journal of innovative Computing, Information and Control.*

Darvishi, M. T. & Barati, A. 2007. "A Fourth-Order Method from Quadrature Formulae to Solve Systems of Nonlinear Equations". *Applied Mathematics and Computation*: 257-261.

Darvishi, M. T. & Barati, A. 2007. "A Third-Order Newton-Type Method to Solve Systems of Nonlinear Equations". *Applied Mathematics and Computation*: 630-635.

Devi, F. M. 2011. *Penyelesaian Sistem Persamaan Nonlinier dengan Metode Jaringan Syaraf Tiruan Hopfield*. Jakarta: Universitas Islam Negeri Syarif Hidayatullah.

Grailoo, M., Bakhshi, A., & Grailoo, M. 2011. "Solving Systems of Nonlinear Equations With an Improved Particle Swarm Optimization". *IEEE*.

Hadi, I. & Sabah, M. 2015. "Improvement Cat Swarm Optimization for Efficient Motion Estimation". *International Journal of Hybrid Information Technology*. ISSN 1738-9968. **8 (1)**: 279-294.

Kusumawati, R. 2009. *Aljabar Linier & Matriks*. Malang: UIN-Malang Press.

Lin, K. C., Huang, Y. H., Hung, J. C., & Lin, Y. T. 2015. "Feature Selection and Parameter Optimization of Support Vector Machines Based on Modified Cat Swarm Optimization". *International Journal of Distributed Sensor Networks*.

Nasiha, K. 2008. *Penyelesaian Sistem Persamaan Tak Linier dengan Metode Newton-Raphson*. Malang: Universitas Islam Negeri Malang.

Noor, M. A. & Waseem, M. 2009. "Some Iterative Methods for Solving a System of Nonlinear Equations". *Computers and Mathematics with Applications*.

Ozel, M. 2010. "A New Decomposition Method for Solving System of Nonlinear Equations". *Mathematical and Computational Applications*. **15 (1)**: 89-95.

Munir, R. 2003. *Metode Numerik*. Jakarta: Erlangga.

Rosita, A. 2012. "Implementasi Algoritma *Particle Swarm* untuk Menyelesaikan Sistem Persamaan Nonlinier". *Jurnal Teknik ITS*. ISSN 2301-9271. **1**: 211-215.

Sahid. 2005. *Pengantar Komputasi Numerik dengan MATLAB*. Yogyakarta: ANDI.

Singh, S. 2013. "A System of Nonlinear Equations with Singular Jacobian". *International Journal of Innovative Research in Science, Engineering and Technology*. ISSN 2319-8753. **2**

Tsai, P. W., Pan, J. S., Chen, S. M., Liao, B. Y., & Hao, S. P. 2008. "Parallel Cat Swarm Optimization". *Procceedings of the Seventh International Conference on Machine Learning and Cybernetics*.

Wahyudianto, E. 2014. "Perbandingan Metode Regula Falsi dan Secant dalam Menyelesaikan Persamaan Non-Linear." Tidak Diterbitkan. Skripsi. Jember: Program Sarjana Universitas Jember.

Yang, Y., Zhou, Y., & Gong, Q. 2010. "Hybrid Artificial Glowworm Swarm Optimization Algorithm for Solving System of Nonlinear Equations". *Journal of Computational Information Systems*.

LAMPIRAN

a. Sistem Persamaan Non-Linier Tiga Variabel

```
% clc;
clear all;
format long
disp('Penerapan Algoritma Cat Swarm Optimization pada Sistem
Persamaan Non-Linier');
MaxIt=1000;
nPop=10;
Mr=0.4;
SMP=15;
CDC=1;
SRD=0.4;
c=0.04;
vmax=2;
fmin_all=zeros(1,MaxIt);
xmin_all=zeros(1,MaxIt);
ymin_all=zeros(1,MaxIt);
zmin_all=zeros(1,MaxIt);

f=inline('(x)+cos(x*y*pi/180)-(z^2)-(1.1)','x','y','z');
g=inline('(x^2)-(10*y)-exp(y*z)+(0.8)','x','y','z');
h=inline('(x*z)+(y^2)-(z)-(0.3)','x','y','z');

x=rand(nPop,1);
y=rand(nPop,1);
z=rand(nPop,1);

vx=rand(nPop,1)*10^(-2);
vy=rand(nPop,1)*10^(-2);
vz=rand(nPop,1)*10^(-2);
disp('-------------------------------------------------------------
-------------------------------------------------------------------
-------------------------------------------------------------')
fprintf('Iterasi          x                    y
z                        f1                          f2
f3                      f(tot)\n')
fprintf('-------------------------------------------------------------
-------------------------------------------------------------------
-------------------------------------------------------------\n')
tic
for k=1:MaxIt

    if k>1
       x=xmin_all(k-1)+randn(nPop,1)*fmin_all(k-1);
       y=ymin_all(k-1)+randn(nPop,1)*fmin_all(k-1);
       z=zmin_all(k-1)+randn(nPop,1)*fmin_all(k-1);

    end

    ff=zeros(nPop,1);
    fg=zeros(nPop,1);
    fh=zeros(nPop,1);
    ftot=zeros(nPop,1);
```

```
for i=1:nPop
   ff(i,1)=abs(f(x(i),y(i),z(i)));
   fg(i,1)=abs(g(x(i),y(i),z(i)));
   fh(i,1)=abs(h(x(i),y(i),z(i)));
   ftot(i,1)=ff(i)+fg(i)+fh(i);
end
data=[x y z vx vy vz ff fg fh ftot];

%=======================
%seeking mode
nt=round(Mr*nPop);
ns=nPop-nt;

pos=randperm(nPop);
data_tc=[];
data_sk=[];
for i=1:nt
    data_tc(i,:)=data(pos(i),:);
end
for i=1:ns
    data_sk(i,:)=data(pos(nt+i),:);
end

SPC=round(rand(1));
if SPC==1
x1=repmat(data_sk(:,1),1,SMP-1);
y1=repmat(data_sk(:,2),1,SMP-1);
z1=repmat(data_sk(:,3),1,SMP-1);

xbaru=(1+randn(ns,SMP-1)*SRD).*x1;
ybaru=(1+randn(ns,SMP-1)*SRD).*y1;
zbaru=(1+randn(ns,SMP-1)*SRD).*z1;
else
x1=repmat(data_sk(:,1),1,SMP);
y1=repmat(data_sk(:,2),1,SMP);
z1=repmat(data_sk(:,3),1,SMP);

xbaru=(1+randn(ns,SMP)*SRD).*x1;
ybaru=(1+randn(ns,SMP)*SRD).*y1;
zbaru=(1+randn(ns,SMP)*SRD).*z1;

end
ff_baru=zeros(ns,SMP-1);
fg_baru=zeros(ns,SMP-1);
fh_baru=zeros(ns,SMP-1);

for i=1:ns
    for j=1:SMP-1
        ff_baru(i,j)=abs(f(xbaru(i,j),ybaru(i,j),zbaru(i,j)));
        fg_baru(i,j)=abs(g(xbaru(i,j),ybaru(i,j),zbaru(i,j)));
        fh_baru(i,j)=abs(h(xbaru(i,j),ybaru(i,j),zbaru(i,j)));
    end
end
        ftot_baru=ff_baru+fg_baru+fh_baru;
P=zeros(ns,SMP-1);
```

```
for i=1:ns

    for j=1:SMP-1
        P(i,j)=abs((ftot_baru(i,j)-
max(ftot_baru(i,:))))/(max(ftot_baru(i,:))-min(ftot_baru(i,:)));
    end

end
for i=1:ns
    indx=find(P(i,:)==1);
    data_sk_baru(i,:)=[xbaru(i,indx(1)) ybaru(i,indx(1))
zbaru(i,indx(1)) data_sk(i,4:6) ff_baru(i,indx(1))
fg_baru(i,indx(1)) fh_baru(i,indx(1)) ftot_baru(i,indx(1))];
end

%=========================
%tracing mode
for i=1:nt
    [a1 a]=min(ftot);
    vx1=data_tc(i,4)+rand(1)*c*(data(a,1)-data_tc(i,1));
    vy1=data_tc(i,5)+rand(1)*c*(data(a,2)-data_tc(i,2));
    vz1=data_tc(i,6)+rand(1)*c*(data(a,3)-data_tc(i,3));
    if vx1>vmax
        vx1=vmax;
    end
    if vy1>vmax
        vy1=vmax;
    end
    if vz1>vmax
        vz1=vmax;
    end
    x0=data_tc(i,1:3)+[vx1 vy1 vz1];
    ftot2=sum([abs(f(x0(1),x0(2),x0(3)))
abs(g(x0(1),x0(2),x0(3)))  abs(h(x0(1),x0(2),x0(3)))]);
   data_tc_baru(i,:)=[data_tc(i,1:3)+[vx1 vy1 vz1] vx1 vy1 vz1
abs(f(x0(1),x0(2),x0(3)))  abs(g(x0(1),x0(2),x0(3)))
abs(h(x0(1),x0(2),x0(3))) ftot2];
end
data_baru=[data_sk_baru; data_tc_baru];

if k==1
    [fmin_all0 indx]=min(data(:,end));
    fmin_all(k)=fmin_all0;
    xmin_all(k)=data(indx(1),1); ymin_all(k)=data(indx(1),2);
zmin_all(k)=data(indx(1),3); f1(k)=data(indx(1),7);
f2(k)=data(indx(1),8); f3(k)=data(indx(1),9);
else
   [fmin0 indx]=min(data(:,end));
   if fmin0<=fmin_all(k-1)
       fmin_all(k)=fmin0;
    xmin_all(k)=data(indx(1),1); ymin_all(k)=data(indx(1),2);
zmin_all(k)=data(indx(1),3); f1(k)=data(indx(1),7);
f2(k)=data(indx(1),8); f3(k)=data(indx(1),9);
   else
       fmin_all(k)=fmin_all(k-1);
```

```
    xmin_all(k)=xmin_all(k-1); ymin_all(k)=ymin_all(k-1);
zmin_all(k)=zmin_all(k-1); f1(k)=f1(k-1); f2(k)=f2(k-1);
f3(k)=f3(k-1);

   end
end
data=data_baru;
x=data(:,1);
y=data(:,2);
z=data(:,3);
vx=data(:,4);
vy=data(:,5);
vz=data(:,6);
%=========================================
if fmin_all(k)<=0
    break
end

if k>1
plot([k-1 k],fmin_all(k-1:k),'Color','b','LineWidth',3);
title('Kurva Kekonvergenan Algoritma Cat Swarm Optimization');
hold on
pause(0.000000000001)
xlim([-30 MaxIt])
ylim([-0.3 fmin_all(1)+0.3])
end
%==================================
end
if fmin_all(k)>0

k=k+1;
   [fmin0 indx]=min(data(:,end));
   if fmin0<=fmin_all(k-1)
       fmin_all(k)=fmin0;
    xmin_all(k)=data(indx(1),1); ymin_all(k)=data(indx(1),2);
zmin_all(k)=data(indx(1),3); f1(k)=data(indx(1),7);
f2(k)=data(indx(1),8); f3(k)=data(indx(1),9);
   else
       fmin_all(k)=fmin_all(k-1);
    xmin_all(k)=xmin_all(k-1); ymin_all(k)=ymin_all(k-1);
zmin_all(k)=zmin_all(k-1); f1(k)=f1(k-1); f2(k)=f2(k-1);
f3(k)=f3(k-1);

   end
end
fprintf('%3.0f    %10.20f    %10.20f   %10.20f   %10.20f   %10.20f
%10.20f   %10.20f\n',[k-
1;xmin_all(MaxIt);ymin_all(MaxIt);zmin_all(MaxIt);f1(MaxIt);f2(Max
It);f3(MaxIt);fmin_all(MaxIt);])
toc
plot(fmin_all); title('Kurva Kekonvergenan Algoritma Cat Swarm
Optimization');
```

b. Sistem Persamaan Non-Linier Dua Variabel

```
% clc;
clear all;
format long
disp('Penerapan Algoritma Cat Swarm Optimization pada Sistem
Persamaan Non-Linier');
MaxIt=1000;
nPop=10;
Mr=0.4;
SMP=15;
CDC=1;
SRD=0.4;
c=0.04;
vmax=2;
fmin_all=zeros(1,MaxIt);
xmin_all=zeros(1,MaxIt);
ymin_all=zeros(1,MaxIt);

f=inline('(x^2)-y-1','x','y');
g=inline('((x-2)^2)+((y-0.5)^2)-(1)','x','y');
x=rand(nPop,1);
y=rand(nPop,1);
vx=rand(nPop,1)*10^(-2);
vy=rand(nPop,1)*10^(-2);
disp('-----------------------------------------------------------------
-----------------------------------------------------------------
------------')
fprintf('Iterasi           x                    y
f1                        f2                        f(tot)\n')
fprintf('-----------------------------------------------------------------
-----------------------------------------------------------------
----------\n')
tic
for k=1:MaxIt

    if k>1
       x=xmin_all(k-1)+randn(nPop,1)*fmin_all(k-1);
       y=ymin_all(k-1)+randn(nPop,1)*fmin_all(k-1);
    end
    ff=zeros(nPop,1);
    fg=zeros(nPop,1);
    ftot=zeros(nPop,1);
for i=1:nPop
   ff(i,1)=abs(f(x(i),y(i)));
   fg(i,1)=abs(g(x(i),y(i)));
   ftot(i,1)=ff(i)+fg(i);
end
data=[x y vx vy ff fg ftot];

%======================
%seeking mode
nt=round(Mr*nPop);
ns=nPop-nt;
pos=randperm(nPop);
data_tc=[];
```

```
data_sk=[];
for i=1:nt
    data_tc(i,:)=data(pos(i),:);
end
for i=1:ns
    data_sk(i,:)=data(pos(nt+i),:);
end

SPC=round(rand(1));
if SPC==1
x1=repmat(data_sk(:,1),1,SMP-1);
y1=repmat(data_sk(:,2),1,SMP-1);

xbaru=(1+randn(ns,SMP-1)*SRD).*x1;
ybaru=(1+randn(ns,SMP-1)*SRD).*y1;
else
x1=repmat(data_sk(:,1),1,SMP);
y1=repmat(data_sk(:,2),1,SMP);

xbaru=(1+randn(ns,SMP)*SRD).*x1;
ybaru=(1+randn(ns,SMP)*SRD).*y1;

end
ff_baru=zeros(ns,SMP-1);
fg_baru=zeros(ns,SMP-1);

for i=1:ns
    for j=1:SMP-1
        ff_baru(i,j)=abs(f(xbaru(i,j),ybaru(i,j)));
        fg_baru(i,j)=abs(g(xbaru(i,j),ybaru(i,j)));
    end
end
        ftot_baru=ff_baru+fg_baru;
P=zeros(ns,SMP-1);
for i=1:ns

    for j=1:SMP-1
        P(i,j)=abs((ftot_baru(i,j)-
max(ftot_baru(i,:))))/(max(ftot_baru(i,:))-min(ftot_baru(i,:)));
    end

end
for i=1:ns
    indx=find(P(i,:)==1);
    data_sk_baru(i,:)=[xbaru(i,indx(1)) ybaru(i,indx(1))
data_sk(i,3:4) ff_baru(i,indx(1)) fg_baru(i,indx(1))
ftot_baru(i,indx(1))];
end

%=========================
%tracing mode
for i=1:nt
    [a1 a]=min(ftot);
    vx1=data_tc(i,3)+rand(1)*c*(data(a,1)-data_tc(i,1));
    vy1=data_tc(i,4)+rand(1)*c*(data(a,2)-data_tc(i,2));
```

```
    if vx1>vmax
        vx1=vmax;
    end
    if vy1>vmax
        vy1=vmax;
    end
    x0=data_tc(i,1:2)+[vx1 vy1];
    ftot2=sum([abs(f(x0(1),x0(2)))  abs(g(x0(1),x0(2)))]);
   data_tc_baru(i,:)=[data_tc(i,1:2)+[vx1 vy1] vx1 vy1
abs(f(x0(1),x0(2)))  abs(g(x0(1),x0(2)))  ftot2];
end
data_baru=[data_sk_baru; data_tc_baru];

if k==1
    [fmin_all(k) indx]=min(data(:,end));
    xmin_all(k)=data(indx(1),1); ymin_all(k)=data(indx(1),2);
f1(k)=data(indx(1),5); f2(k)=data(indx(1),6);
else
   [fmin0 indx]=min(data(:,end));
   if fmin0<=fmin_all(k-1)
       fmin_all(k)=fmin0;
    xmin_all(k)=data(indx(1),1); ymin_all(k)=data(indx(1),2);
f1(k)=data(indx(1),5); f2(k)=data(indx(1),6);
   else
       fmin_all(k)=fmin_all(k-1);
    xmin_all(k)=xmin_all(k-1); ymin_all(k)=ymin_all(k-1);
f1(k)=f1(k-1); f2(k)=f2(k-1);

   end
end
data=data_baru;
x=data(:,1);
y=data(:,2);
vx=data(:,3);
vy=data(:,4);
%=========================================
if fmin_all(k)<=0
    break
end

if k>1
plot([k-1 k],fmin_all(k-1:k),'Color','b','LineWidth',3);
title('Kurva Kekonvergenan Algoritma Cat Swarm Optimization');
hold on
pause(0.000000000001)
xlim([-30 MaxIt])
ylim([-0.3 fmin_all(1)+0.3])
end
%==================================
end
if fmin_all(k)>0

k=k+1;
   [fmin0 indx]=min(data(:,end));
   if fmin0<=fmin_all(k-1)
       fmin_all(k)=fmin0;
```

```
    xmin_all(k)=data(indx(1),1); ymin_all(k)=data(indx(1),2);
f1(k)=data(indx(1),5); f2(k)=data(indx(1),6);
   else
       fmin_all(k)=fmin_all(k-1);
    xmin_all(k)=xmin_all(k-1); ymin_all(k)=ymin_all(k-1);
f1(k)=f1(k-1); f2(k)=f2(k-1);

   end
end
fprintf('%3.0f    %10.20f    %10.20f   %10.20f   %10.20f
%10.20f\n',[k-
1;xmin_all(MaxIt);ymin_all(MaxIt);f1(MaxIt);f2(MaxIt);fmin_all(Max
It);])
toc
plot(fmin_all); title('Kurva Kekonvergenan Algoritma Cat Swarm
Optimization');
```