

**HIBRID ALGORITMA GENETIKA DAN
NEW CONJUGATE GRADIENT
UNTUK MASALAH OPTIMASI NON-LINIER**

SKRIPSI

Oleh:
KRISTY MERDHANIA
0510940031-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2009**

**HIBRID ALGORITMA GENETIKA DAN
NEW CONJUGATE GRADIENT
UNTUK MASALAH OPTIMASI NON-LINIER**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains dalam bidang matematika

Oleh:

KRISTY MERDHANIA

0510940031-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG**

2009

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI
HIBRID ALGORITMA GENETIKA DAN
NEW CONJUGATE GRADIENT
UNTUK MASALAH OPTIMASI NON-LINIER

Oleh :
KRISTY MERDHANIA
0510940031-94

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 10 Agustus 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam bidang matematika

Pembimbing I

Pembimbing II

Syaiful Anam, SSi., MT
NIP. 132 300 237

Drs. M. Muslikh, M.Si
NIP. 131 871 740

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, M.Sc
NIP. 132 126 049

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : KRISTY MERDHANIA
NIM : 0510940031
Jurusan : MATEMATIKA
Penulis Skripsi berjudul : HIBRID ALGORITMA GENETIKA
DAN NEW CONJUGATE
GRADIENT UNTUK MASALAH
OPTIMASI NON-LINIER

Dengan ini menyatakan bahwa :

1. Isi Skripsi yang saya buat adalah benar-benar karya sendiri, dan bukan plagiat karya orang lain. Karya-karya yang tercantum dalam Daftar Pustaka Skripsi ini, semata-mata digunakan sebagai acuan/referensi.
2. Apabila di kemudian hari diketahui bahwa isi Skripsi saya merupakan hasil plagiat, maka saya bersedia menanggung akibat hukum dari keadaan tersebut.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 3 Agustus 2009

Yang menyatakan,

(Kristy Merdhania)

NIM. 0510940031

UNIVERSITAS BRAWIJAYA



HIBRID ALGORITMA GENETIKA DAN NEW CONJUGATE GRADIENT UNTUK MASALAH OPTIMASI NON-LINIER

ABSTRAK

Algoritma genetika adalah algoritma optimasi global yang banyak diterapkan dalam berbagai aplikasi. Pada penerapannya, algoritma genetika sering mengalami kekonvergenan prematur. Hal ini disebabkan oleh kriteria terminasi yang telah terpenuhi, padahal populasi algoritma genetika belum optimal. Dalam skripsi ini digunakan paralel algoritma genetika yang diperoleh dengan mengkombinasikan beberapa peluang silang dan peluang mutasi. Hal ini bertujuan untuk mendapatkan solusi yang paling optimum dari beberapa algoritma genetika. Tetapi pada skripsi ini ditunjukkan bahwa pada paralel algoritma genetika masalah kekonvergenan prematur masih muncul. Untuk mengatasi kondisi tersebut, diperkenalkan algoritma optimasi lokal (metode *new conjugate gradient*). Metode ini akan digabungkan dengan paralel algoritma genetika dan disebut hibrid algoritma genetika. Pada hibrid algoritma genetika, solusi yang dihasilkan oleh paralel algoritma genetika digunakan sebagai titik awal metode *new conjugate gradient*. Efektifitas penggunaan metode *new conjugate gradient* akan ditunjukkan dengan membandingkan kesalahan mutlak dan waktu komputasi dari hibrid algoritma genetika dan paralel algoritma genetika untuk beberapa tes fungsi. Dengan waktu komputasi yang sebanding, hasil solusi numerik hibrid algoritma genetika diketahui jauh lebih baik daripada paralel algoritma genetika.

Kata kunci: optimasi nonlinier, paralel algoritma genetika, *new conjugate gradient*.

UNIVERSITAS BRAWIJAYA



HYBRID GENETIC ALGORITHM AND NEW CONJUGATE GRADIENT FOR NON-LINEAR OPTIMIZATION PROBLEMS

ABSTRACT

Genetic algorithm is a global optimization algorithm which has been applied in many applications. The implementation of genetic algorithm frequently produce the premature convergence. It is caused by the fact that the termination criteria is already satisfied, whereas the population of genetic algorithm is not optimal yet. Genetic algorithm parallel is therefore used in this final project which is obtained by combining some crossover and mutation probabilities. It is purposed to get the solution which is the most optimum from several genetic algorithms. But in this final project show that genetic algorithm parallel premature convergence still appear. To solve this condition, local optimization algorithm (new conjugate gradient method) is introduced. This method will be combined with genetic algorithm parallel which is called genetic algorithm hybrid. In this genetic algorithm hybrid, the solution of genetic algorithm parallel is used as starting point for new conjugate gradient method. The effectivity of new conjugate gradient method usage will be shown by comparing absolute error and computation time of genetic algorithm hybrid and genetic algorithm parallel for several function tests. With equal computation time, the results of numeric solution are known that genetic algorithm hybrid better than genetic algorithm parallel.

Key words: nonlinear optimization, parallel genetic algorithm, new conjugate gradient.

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat ALLAH SWT atas segala rahmat serta hidayahNya yang telah dilimpahkan kepada penulis sehingga dapat menyelesaikan penulisan Skripsi yang berjudul **"Hibrid Algoritma Genetika dan New Conjugate Gradient Untuk Masalah Optimasi Non-Linier"**. Shalawat, salam, dan terima kasih sebesar-besarnya juga penulis curahkan kepada Rasulullah MUHAMMAD SAW. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Matematika, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.

Penulis menyadari bahwa penulisan Skripsi ini tidak dapat terealisasikan tanpa bantuan baik yang bersifat moral maupun spiritual dari berbagai pihak. Untuk itu penulis menyampaikan terima kasih yang sedalam - dalamnya kepada :

1. Syaiful Anam, S.Si, M.T selaku pembimbing I atas segala motivasi, bimbingan serta pengarahan dan nasihat selama proses pengerjaan Skripsi ini.
2. Drs. M. Muslikh, MSi selaku Dosen Pembimbing II dan dosen penasihat akademik atas bimbingan, pengarahan dan nasihat selama proses pengerjaan Skripsi.
3. Drs. Noor Hidayat, MSi, Dr. Agus Suryanto, M.Sc, Drs. Hery Subagio, M.Kes selaku dosen penguji atas koreksi serta motivasinya sehingga Skripsi ini dapat terselesaikan.
4. Bapak dan Ibu Dosen Matematika yang telah memberikan bekal ilmu pengetahuan, serta para Staf Tata Usaha Jurusan Matematika atas segala bantuannya.
5. Kedua orang tua, Bapak Agoes Edie Poernomo, Ibu Dwi Astuti serta adik-adikku Maya Kristiningrun, Freddie Rachman Hakim dan Krisnina Maharani atas segala kasih sayang, do'a dan keteladanannya.
6. Matris Fortunatus, Kartikowati, Zuraidah Fitriah, Nia Astri, Lisa Tridiana, Farhanah, Indra Noordianto dan teman-teman Math '05 serta keluarga besar Raya Sumbersari 254 B/D Lia, Melsi, Evi atas dukungan dan kekeluargaannya.

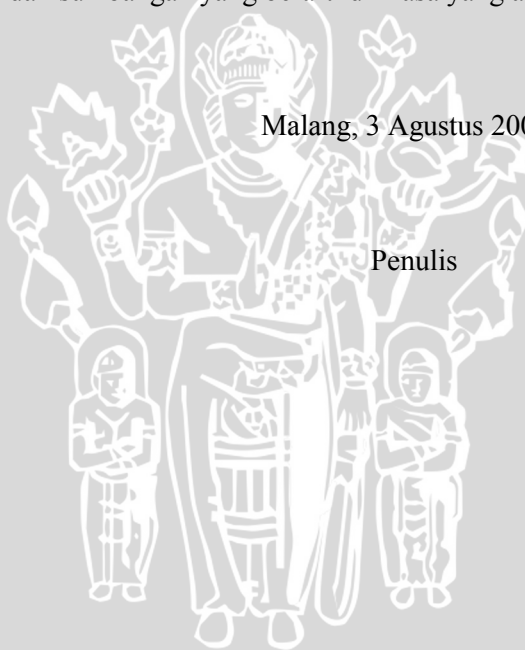
7. Mbak Itri, Mas Ve, Yeyen, Faiz, Yayan, Lanang dan Teman-teman paduan suara Cientifico Choir yang mendukung juga sebagai tempat menghilangkan pikiran penat.
8. Seluruh adik (anak didik) yang telah bertukar pikiran dan ilmu pengetahuan, khususnya Ova, Ovi, Sangkan dan Felich yang menginspirasi pengerjaan Skripsi ini.

Sebagai manusia yang memiliki keterbatasan dan dengan segala kerendahan hati, penulis menyadari bahwa dalam penulisan Skripsi ini masih terdapat banyak kekurangan sehingga belum dapat dikatakan sempurna. Untuk itu penulis sangat mengharapkan kritik dan saran dari berbagai pihak demi kesempurnaan Skripsi ini.

Akhir kata, penulis berharap semoga Skripsi ini dapat memberikan manfaat dan sumbangan yang berarti di masa yang akan datang.

Malang, 3 Agustus 2009

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
BAB II TINJAUAN PUSTAKA	3
2.1 Konsep Dasar Optimasi.....	3
2.2 Metode <i>Conjugate Gradient</i>	9
2.3 Metode <i>New Conjugate Gradient</i>	11
2.4 Algoritma Genetika.....	14
2.4.1 Pandangan Umum.....	14
2.4.2 Komponen-komponen Algoritma Genetika.....	15
2.4.2.1 Skema Pengkodean.....	15
2.4.2.2 Evaluasi.....	17
2.4.2.3 Seleksi.....	18
2.4.2.4 Pindah Silang (<i>Crossover</i>).....	19
2.4.2.5 Mutasi.....	19
2.4.2.6 <i>Elitisme</i>	20
2.4.2.7 Penggantian Populasi.....	20
2.4.3 Parameter Genetika.....	20
2.4.3.1 Ukuran Populasi.....	20
2.4.3.2 Jumlah Generasi.....	21
2.4.3.3 Peluang <i>Crossover</i> (Pc).....	21
2.4.3.4 Peluang Mutasi (Pm).....	21
2.4.4 Kekonvergenan Algoritma Genetika.....	21

2.5 Paralel Algoritma Genetika	22
2.6 Hibrid Algoritma Genetika dengan <i>New Conjugate Gradient</i>	22
BAB III PEMBAHASAN	25
3.1 Arah <i>Descent</i> dan Kondisi <i>Descent</i>	25
3.2 Studi Kasus	30
3.3.1 Penyelesaian Secara Analitik	30
3.3.2 Penyelesaian Numerik Menggunakan Hibrid Algoritma Genetika	34
3.3 Diagram Alir Algoritma Genetika	36
3.4 Diagram Alir <i>New Conjugate Gradient</i>	37
3.5 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Paralel Algoritma Genetika Menggunakan Beberapa Tes Fungsi	39
BAB IV PENUTUP	49
4.1 Kesimpulan	49
4.2 Saran	50
DAFTAR PUSTAKA	51
LAMPIRAN	53

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Minimum lokal dan minimum global..... 5
Gambar 2.2	(a) dan (b) Fungsi Unimodal pada I . (c) Fungsi yang tidak unimodal pada I 5
Gambar 2.3	Skema Hibridisasi Algoritma Genetika dan <i>New Conjugate Gradient</i> 23
Gambar 3.1	Fungsi $f = x_1 e^{(-x_1^2 - x_2^2)}$ 35
Gambar 3.2	Diagram Alir Algoritma Genetika 36
Gambar 3.3	Diagram Alir <i>New Conjugate Gradient</i> 38
Gambar 3.4	Fungsi De Jong 41
Gambar 3.5	<i>Contour</i> dan Titik-titik Solusi Tes Fungsi De Jong dengan Metode Hibrid Algoritma Genetika $f(x^*) = 2.4431 \times 10^{-13}$ dan $(x^*) = (-3.2548 \times 10^{-7}, -3.7198 \times 10^{-7})$ 41
Gambar 3.6	<i>Contour</i> dan Titik-titik Solusi Tes Fungsi De Jong dengan Metode Paralel Algoritma Genetika $f(x^*) = 4.3106$ dan $(x^*) = (-1.3672, -1.5625)$ 42
Gambar 3.7	Fungsi Rastrigin..... 45
Gambar 3.8	<i>Contour</i> dan Titik-titik Solusi Tes Fungsi Rastrigin dengan Metode Hibrid Algoritma Genetika $f(x^*) = 0$ dan $(x^*) = (-8.6388 \times 10^{-10}, -1.4125 \times 10^{-9})$ 46
Gambar 3.9	<i>Contour</i> dan Titik-titik Solusi Tes Fungsi Rastrigin dengan metode Paralel Algoritma Genetika $f(x^*) = 1.0572$ dan $(x^*) = (0.035156, 0.064453)$ 47
Gambar A.1	Fungsi De Jong 53
Gambar A.2	Fungsi Ackley 54
Gambar A.3	Fungsi Drop Wave 55
Gambar A.4	Fungsi Shekel 56
Gambar A.5	Fungsi Rastrigin 56
Gambar D.1	Proses Pindah Silang 71
Gambar D.2	Proses Mutasi 71

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

	Halaman
Tabel 2.1 Skema Pengkodean <i>real number encoding</i>	16
Tabel 2.2 Skema Pengkodean <i>discrete decimal encoding</i>	16
Tabel 2.3 Skema Pengkodean <i>binary encoding</i>	16
Tabel 3.1 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi De Jong	39
Tabel 3.2 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Ackley.	42
Tabel 3.3 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Drop Wave	43
Tabel 3.4 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Shekel..	43
Tabel 3.5 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Rastrigin.....	44
Tabel D.1 Kromosom dalam Satu Populasi	65
Tabel D.2 Nilai $\bar{x} = (x_1, x_2)$ dalam Satu Populasi	66
Tabel D.3 Nilai f Setiap Individu	66
Tabel D.4 Nilai f_b Setiap Individu	67
Tabel D.5 Nilai <i>Fitness</i> Setiap Individu	68
Tabel D.6 Nilai <i>Fitness</i> yang Telah Diurutkan	68
Tabel D.7 Nilai <i>Linear Fitness Ranking</i>	69
Tabel D.8 Nilai <i>Linear Fitness Ranking</i> yang Telah Diurutkan.....	69
Tabel D.9 Nilai Probabilitas q_k	70
Tabel D.10 Solusi Algoritma Genetika	72
Tabel D.11 Solusi Metode <i>New Conjugate Gradient</i>	79

UNIVERSITAS BRAWIJAYA



BAB I PENDAHULUAN

1.1 Latar Belakang

Algoritma genetika adalah suatu teknik optimasi global yang terkenal, diinspirasi oleh proses seleksi alam dan teori genetik populasi. Algoritma Genetika menyediakan suatu arsitektur umum untuk menyelesaikan masalah optimasi yang kompleks dan banyak digunakan pada masalah praktis dengan fokus pada pencarian parameter-parameter optimal.

Keunggulan algoritma genetika terlihat pada kemudahan penggunaannya dan mampu mendapatkan solusi secara cepat khususnya untuk masalah-masalah berdimensi tinggi. Salah satu kelemahan algoritma genetika adalah kondisi konvergen tidak menjamin bahwa solusi yang diperoleh merupakan solusi optimal (konvergen prematur). Oleh karena itu, untuk menutupi kelemahannya, algoritma genetika akan digabungkan dengan algoritma optimasi lokal. Penggabungan tersebut dimaksudkan untuk mendapatkan metode alternatif dalam menentukan optimasi global (Gudla dan Ganguli, 2005).

Dalam skripsi ini, metode optimasi lokal yang digunakan adalah metode *new conjugate gradient*. Penggunaan metode ini dipilih karena ukuran langkah yang dihasilkan lebih besar dibandingkan penggunaan metode *conjugate gradient*, sehingga proses penentuan titik optimal lebih cepat. *New conjugate gradient* adalah modifikasi metode *conjugate gradient*. Modifikasi yang dilakukan adalah menggabungkan parameter *conjugate gradient* (β_k) Polak-Ribiere-Polyak (1969) dengan Dai-Yuan (2001). Pada metode tersebut, proporsi nilai parameter yang digunakan adalah θ_k dengan $0 < \theta_k < 1$.

Penggabungan metode algoritma genetika dengan metode *new conjugate gradient* disebut hibrid algoritma genetika. Diharapkan kedua metode ini dapat saling menutupi kekurangan metode lainnya dan memperoleh solusi global seoptimal mungkin.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, pokok permasalahan yang dibahas dalam skripsi ini adalah sebagai berikut:

1. bagaimana pemilihan parameter θ_k untuk menentukan nilai β_k pada algoritma *new conjugate gradient* sebagai perhitungan *descent direction*?
2. bagaimana solusi numerik yang dihasilkan oleh hibrid algoritma genetika dengan solusi numerik yang dihasilkan oleh paralel algoritma genetika dilihat dari nilai kesalahan mutlak?
3. bagaimana waktu komputasi yang dihasilkan oleh hibrid algoritma genetika dibandingkan dengan waktu komputasi yang dihasilkan oleh paralel algoritma genetika?

1.3 Batasan Masalah

1. Fungsi yang dioptimalkan adalah fungsi *nonlinear* tanpa kendala dengan dua variabel.
2. Nilai optimum yang dicari adalah nilai minimum global.

1.4 Tujuan

Berdasarkan pada masalah di atas, tujuan dari penulisan skripsi ini adalah untuk:

1. menentukan pemilihan parameter θ_k untuk menentukan nilai β_k pada algoritma *new conjugate gradient* sebagai perhitungan *descent direction*,
2. membandingkan solusi numerik yang dihasilkan oleh hibrid algoritma genetika dengan solusi numerik yang dihasilkan oleh paralel algoritma genetika dilihat dari nilai kesalahan mutlak,
3. membandingkan waktu komputasi yang dihasilkan oleh hibrid algoritma genetika dengan waktu komputasi yang dihasilkan oleh paralel algoritma genetika.

BAB II TINJAUAN PUSTAKA

Dalam bab ini diberikan beberapa teori yang menunjang pembahasan hibrid algoritma genetika yaitu optimasi, *conjugate gradient*, *new conjugate gradient* dan algoritma genetika.

2.1 Konsep Dasar Optimasi

Optimasi merupakan masalah memaksimalkan atau meminimalkan suatu besaran tertentu yang disebut dengan fungsi tujuan. Fungsi tujuan bergantung pada sejumlah variabel. Variabel-variabel ini tidak saling berhubungan atau saling bergantung melalui satu atau lebih kendala (Luknanto, 2000).

Optimasi terbagi menjadi dua, optimasi *linear* dan *nonlinear*. Suatu permasalahan optimasi disebut *nonlinear* jika fungsi tujuan dan kendalanya berbentuk *nonlinear* pada salah satu atau keduanya (Luknanto, 2000).

Optimasi *nonlinear* itu sendiri dapat dibagi menjadi empat berdasarkan kriteria permasalahannya yaitu optimasi *nonlinear* satu variabel tanpa kendala, multi variabel tanpa kendala, satu variabel dengan kendala, dan multi variabel dengan kendala (Luknanto, 2000).

Penulisan ini difokuskan pada optimasi *nonlinear* multi variabel tanpa kendala untuk kasus meminimumkan suatu fungsi tujuan. Secara umum optimasi *nonlinear* multi variabel tanpa kendala untuk permasalahan minimum dinyatakan sebagai berikut:

$$\min f(x) \text{ dengan } x = (x_1, x_2, \dots, x_n) \in I \subset \mathbb{R}^n$$

di mana $f : I \rightarrow \mathbb{R}$ dan I kompak.

Sasaran penyelesaian optimasi secara umum untuk permasalahan minimum adalah memperoleh solusi seminimum mungkin (solusi global). Solusi dalam permasalahan minimum adalah mencari suatu nilai yang membuat fungsi tujuan menjadi minimum global. Agar nilai fungsi tujuan memenuhi minimum global, ada beberapa hal yang perlu diperhatikan yaitu:

Definisi 2.1 (Fungsi Kontinu)

Diberikan fungsi $f: I \rightarrow \mathfrak{R}$ dengan $I \subset \mathfrak{R}^n$ dan $x_0 \in I$. f dikatakan kontinu di titik x_0 jika memenuhi kondisi:

1. $f(x_0)$ terdefinisi pada I ,
2. $\lim_{x \rightarrow x_0} f(x)$ ada,
3. $\lim_{x \rightarrow x_0} f(x) = f(x_0)$.

Jika satu atau lebih dari kondisi di atas tidak terpenuhi, maka f dikatakan diskontinu di x_0 . f kontinu dalam himpunan terbuka $I \subset \mathfrak{R}^n$ jika f kontinu di setiap titik $x \in I$ (Grossman, 1995).

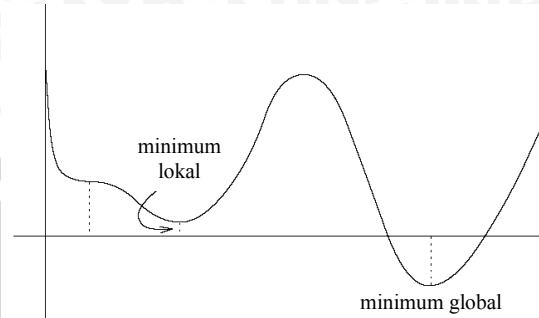
Definisi 2.2 (Minimum Lokal)

Diberikan $x^* \in I \subset \mathfrak{R}^n$. x^* dikatakan sebagai titik minimum lokal fungsi f pada I jika terdapat persekitaran $N(x^*)$ di dalam I sedemikian sehingga untuk semua $x \in N(x^*)$ berlaku $f(x^*) \leq f(x)$ (Leader, 2004).

Definisi 2.3 (Minimum Global)

Diberikan $x^* \in I \subset \mathfrak{R}^n$, x^* disebut sebagai titik minimum global fungsi f jika untuk semua $x \in I$ berlaku $f(x^*) \leq f(x)$ (Bronshtein, dkk. 2007).

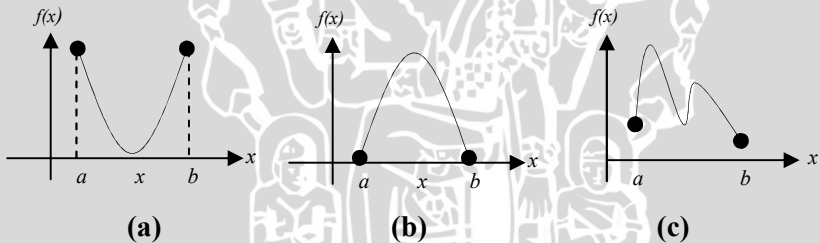
Untuk lebih jelasnya dapat dilihat pada Gambar 2.1 berikut ini:



Gambar 2.1 Minimum lokal dan minimum global

Definisi 2.4 (Fungsi Unimodal)

Sebuah fungsi riil $f(x)$ dikatakan unimodal [multimodal] pada $I=[a,b]$ jika ada titik $x \in I$ yang tunggal [tidak tunggal], $f(x)$ turun (naik) pada $[a,x]$ dan naik (turun) pada $[x,b]$ (Mathews dan Fink, 2004). Contoh fungsi unimodal dan multimodal dapat dilihat pada Gambar 2.2.



Gambar 2.2 (a) dan (b) Fungsi unimodal pada I (c) Fungsi multimodal pada I

Definisi 2.5 (Gradient)

Misalkan $f(x)$ adalah fungsi x , di mana $x=(x_1, x_2, K, x_n)$ suatu vektor dalam himpunan terbuka \mathfrak{R}^n , sedemikian sehingga $\frac{\partial f(x)}{\partial x_k}$ ada untuk $k=1,2,K,n$. Gradient $f(x)$, dinotasikan dengan $\nabla f(x)$,

adalah vektor kolom $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^T$ (Mathews dan Fink, 2004).

Definisi 2.6 (Matriks Hessian)

Misalkan $f(x)$ adalah fungsi x , dimana $x = (x_1, x_2, \dots, x_n)$ suatu vektor dalam himpunan terbuka \mathcal{R}^n , sedemikian sehingga $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ ada untuk $i, j = 1, 2, \dots, n$. Matriks Hessian $f(x)$, dinotasikan dengan $Hf(x)$ dengan ukuran matriks $n \times n$

$$Hf(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

(Mathews dan Fink, 2004).

Definisi 2.7 (Definit Positif dan Definit Semi Positif)

Bentuk kuadrat $x^T Ax$ disebut sebagai definit positif (definit semi positif) jika $x^T Ax > 0$ ($x^T Ax \geq 0$) untuk semua $x \neq 0$, sedangkan matriks simetri A disebut matriks definit positif (matriks definit semi positif) (Nocedal dan Wright, 1999).

Teorema 2.1

Matriks simetri A adalah definit positif (definit semi positif) jika dan hanya jika determinan setiap submatriks utama > 0 (determinan setiap submatriks utama ≥ 0). Submatriks utama A adalah submatriks yang terbentuk dari r baris pertama dan r kolom pertama matriks A untuk $r = 1, 2, \dots, n$ (Anton, 2000).

Contoh, matriks $A = \begin{bmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{bmatrix}$ definit positif karena untuk

semua determinan submatriks utama A adalah positif, yaitu:

$$\text{determinan submatriks utama 1 } (A_1) : |2| = 2, \quad |2| = 2, \quad |9| = 9,$$

$$\text{determinan submatriks utama 2 } (A_2) : \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} = 3, \quad \begin{vmatrix} 2 & 4 \\ 4 & 9 \end{vmatrix} = 2,$$

$$\begin{vmatrix} 2 & -3 \\ -3 & 9 \end{vmatrix} = 27,$$

$$\text{determinan submatriks utama 3 } (A_3) : \begin{vmatrix} 2 & -1 & -3 \\ -1 & 2 & 4 \\ -3 & 4 & 9 \end{vmatrix} = 1.$$

Definisi 2.8 (Fungsi Terdiferensial)

Diberikan fungsi $f : I \rightarrow \mathfrak{R}$ dengan $I \subset \mathfrak{R}^n$ dikatakan terdiferensial di $X_0 = (x_{01}, x_{02}, \dots, x_{0n}) \in I$ jika semua turunan parsial $\frac{\partial f}{\partial x_j}$ ada

(untuk semua $j = 1, 2, \dots, n$) dan jika

$$\lim_{X \rightarrow X_0} \frac{\|f(X) - f(X_0) - (Df)(X_0)(X - X_0)\|}{\|X - X_0\|} = 0$$

$$\text{di mana } (Df)(X_0) = \left(\frac{\partial f}{\partial x_1}(X_0) \quad \frac{\partial f}{\partial x_2}(X_0) \quad \dots \quad \frac{\partial f}{\partial x_n}(X_0) \right)$$

dan $(X - X_0)$ adalah vektor kolom (Chen, 1997).

Teorema 2.2 (Teorema Taylor)

Dimisalkan $f : I \rightarrow \mathfrak{R}$ mempunyai turunan parsial kedua yang kontinu dengan $I \subset \mathfrak{R}^n$ dan $X_0 \in I$, maka berlaku:

$$f(X) = f(X_0) + \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i}(X_0) \right) (x_i - x_{0i}) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(X_0) \right) (x_i - x_{0i})(x_j - x_{0j}) + R_2(X)$$

di mana $X_0 = (x_{01}, x_{02}, \dots, x_{0n})$ dan $X = (x_1, x_2, \dots, x_n)$.

$\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i}(X_0) \right)$ adalah turunan total yaitu $(Df)(X_0)$,

$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(X_0) \right) (x_i - x_{0i})(x_j - x_{0j})$ adalah matriks *Hessian*

f di X_0 dan $R_2(X)$ adalah suku sisa (Chen, 1997).

Definisi 2.9 (Titik Stasioner)

Titik $X_0 \in I$ disebut titik stasioner f dengan $I \subset \mathfrak{R}^n$, jika turunan total $(Df)(X_0) = O$ dan $O = (0, 0, \dots, 0)$. Dengan kata lain,

$X_0 \in I$ adalah titik stasioner fungsi f jika $\frac{\partial f}{\partial x_i}(X_0) = 0$ untuk setiap $i = 1, 2, \dots, n$ (Chen, 1997).

Teorema 2.3 (Minimum Fungsi)

Dimisalkan $f: I \rightarrow \mathfrak{R}$ mempunyai turunan parsial kedua yang kontinu dengan $I \subset \mathfrak{R}^n$ dan X_0 titik stasioner dari f . Jika matriks Hessian $Hf(X_0)(X - X_0)$ adalah definit positif (definit semi positif) maka f mempunyai minimum lokal di X_0 (Chen, 1997).

Dalam skripsi ini, kajian lebih difokuskan pada permasalahan optimasi nonlinier tanpa kendala, yaitu optimasi yang berbentuk

$$\min f(x), x \in I \subset \mathfrak{R}^n \quad (2.1)$$

dimana $f: I \rightarrow \mathfrak{R}$ merupakan suatu fungsi yang dapat diturunkan dan kontinu. Kondisi yang diperlukan untuk memperoleh titik x^* menjadi optimal adalah

$$\nabla f(x^*) = 0.$$

Pada umumnya permasalahan tersebut akan dipecahkan menggunakan algoritma perulangan yang membentuk sebuah barisan titik-titik $x_0, x_1, \dots, x_k \in I$, dengan $f(x_k)$ konvergen untuk $k \rightarrow \infty$.

Untuk menyelesaikan persamaan (2.1) digunakan metode iteratif *conjugate gradient* dengan bentuk $x_{k+1} = x_k + \alpha_k d_k$ di mana $\alpha_k > 0$ adalah ukuran langkah dan d_k adalah arah penurunan (*descent direction*) (Boyd dan Lieven, 2004).

Definisi 2.9 (Konvergen Global)

$I \subset \mathbb{R}^n$, $f: I \rightarrow \mathbb{R}$, $x_s = f(x_s)$ Barisan x_n dikatakan konvergen global ke x_s jika setiap orbit sistem konvergen ke x_s .

Definisi 2.10 (Kesalahan Mutlak)

Hubungan antara nilai eksak y dengan nilai hampiran \tilde{y} adalah :

$$\text{NilaiEksak} = \text{NilaiHampiran} + \text{Kesalahan}$$

atau

$$y = \tilde{y} + E$$

di mana E adalah nilai kesalahan antara nilai eksak dengan nilai hampiran yaitu:

$$E = y - \tilde{y}$$

dan nilai kesalahan mutlak adalah

$$|E| = |y - \tilde{y}| \tag{2.2}$$

(McKinney, 2008).

2.2 Metode Conjugate Gradient

Metode *conjugate gradient* merupakan salah satu metode untuk meminimumkan suatu masalah optimasi tanpa kendala. Suatu fungsi

akan dikatakan menurun jika memenuhi $f(x_{k+1}) < f(x_k)$ dengan arah penurunan yang harus memenuhi $\nabla f(x_k)^T d_k < 0$.

Untuk menentukan hasil konvergensi metode *conjugate gradient*, biasanya membutuhkan ukuran langkah α_k yang harus memenuhi kondisi Wolfe mutlak:

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq -\rho \alpha_k g_k^T d_k \\ \left| g(x_k + \alpha_k d_k)^T d_k \right| &\leq \sigma \left| g_k^T d_k \right| \end{aligned}$$

di mana $g_k = \nabla f(x_k)$, d_k adalah arah *descent* dan biasanya $d_k = -g_k$, dan $0 < \rho < \sigma < 1$. Beberapa analisis konvergensi, tetap membutuhkan bahwa α_k yang diperhitungkan dengan garis pencarian pasti yaitu

$$\alpha_k = \arg \min_{\alpha > 0} f(x_k + \alpha_k d_k).$$

Tetapi, banyak metode numerik seperti BFGS, *gradient descent*, *conjugate gradient* untuk optimisasi tak terbatas yang dibuktikan konvergen di dalam kondisi Wolfe:

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\rho \alpha_k g_k^T d_k \quad (2.3)$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \quad (2.4)$$

Algoritma metode *conjugate gradient* digambarkan sebagai berikut:

- Langkah 0. disajikan $x_0 \in \mathcal{R}^n$, misalkan $d_0 = -g_0$ dan $k=0$, jika $g_0 = 0$, maka berhenti
- Langkah 1. hitung $\alpha_k > 0$ memenuhi kondisi wolfe (2.3) dan (2.4)
- Langkah 2. misalkan $x_{k+1} = x_k + \alpha_k d_k$. Jika $g_{k+1} = 0$, maka berhenti
- Langkah 3. hitung β_{k+1} dan hasilkan d_{k+1} dengan $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$
- Langkah 4. tetapkan $k = k + 1$ dan kembali ke langkah 1.

Rumus terkenal yang sering digunakan untuk β_k adalah

rumus Fletcher-Reeves $\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$, Polak-Ribiere-

Polyak $\beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}$ dan Hestene-Stiefel $\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$

(Tomizuka dan Yabe, 2004).

Dalam skripsi ini, akan digunakan metode *new conjugate gradient* sebagai penyempurna metode *conjugate gradient* dengan perbaikan parameter β_{k+1} .

2.3 Metode New Conjugate Gradient

Optimasi *nonlinear* multi variabel tanpa kendala untuk permasalahan minimum telah didefinisikan dalam persamaan (2.1) di mana $f: I \rightarrow \mathfrak{R}$ adalah fungsi yang terdiferensiabel secara kontinu, terbatas ke bawah. Untuk menyelesaikan masalah ini, dimulai dari suatu titik awal $x_0 \in I$. Metode *nonlinear conjugate gradient* menghasilkan barisan $\{x_k\}$ sebagai berikut:

$$x_{k+1} = x_k + \alpha_k d_k \quad (2.5)$$

di mana $\alpha_k > 0$ diperoleh dari *line search* dan arah d_k dihasilkan dari:

$$d_{k+1} = -g_{k+1} + \beta_k S_k, \quad d_0 = -g_0 \quad (2.6)$$

Pada persamaan (2.6), β_k dikenal sebagai parameter *conjugate gradient*. $S_k = x_{k+1} - x_k$, $g_k = \nabla f(x_k)$, $\|\cdot\|$ merupakan *norm euclidean* dan $y_k = g_{k+1} - g_k$.

Di dalam metode ini, parameter β_k menggunakan perpaduan antara metode Polak-Ribiere dan Polyak (PRP) dengan metode Dai dan Yuan (DY). Dua metode ini dipilih sebab PRP memiliki perhitungan yang baik pada satu sisi dan pada sisi lain DY memiliki pemusatan yang baik (Neculai, 2008).

Dalam prakteknya, metode PRP lebih baik mencapai solusi optimal daripada metode DY. Oleh karena itu, metode PRP lebih sering digunakan daripada metode DY.

Iterasi untuk mendapatkan $x_0, x_1, x_2, \dots, x_n$ pada algoritma *new conjugate gradient* dihitung menggunakan persamaan (2.5) di mana ukuran langkah $\alpha_k > 0$ didapatkan dengan memenuhi kondisi Wolfe pada pertidaksamaan (2.3) dan (2.4). Arah d_k dihasilkan dari persamaan (2.6) di mana

$$\beta_k^N = (1 - \theta_k) \beta_k^{PRP} + \theta_k \beta_k^{DY} = (1 - \theta_k) \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{g}_k^T \mathbf{g}_k} + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{S}_k}. \quad (2.7)$$

θ_k adalah parameter dalam selang $[0, 1]$ yang nantinya ditentukan. Jika $\theta_k = 0$ maka parameter *conjugate gradient* pada persamaan (2.7) menjadi $\beta_k^N = \beta_k^{PRP}$, dan jika $\theta_k = 1$ maka $\beta_k^N = \beta_k^{DY}$. Di sisi lain, jika $0 < \theta_k < 1$ maka β_k^N merupakan kombinasi β_k^{PRP} dan β_k^{DY} . Selanjutnya diimplementasikan dalam pencarian d_k yaitu:

$$d_{k+1} = -\mathbf{g}_{k+1} + (1 - \theta_k) \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \mathbf{S}_k + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{S}_k \quad (2.8)$$

dan diperkirakan bahwa d_k merupakan arah *descent* ($d_0 = -\mathbf{g}_0$).

Algoritma metode *new conjugate gradient* digambarkan sebagai berikut:

Langkah 1

- menetapkan nilai awal $x_0 \in \mathfrak{R}^n$ dan parameter $0 < \rho < \sigma < 1$,
- menghitung nilai $f(x_0)$,
- menghitung nilai $\mathbf{g}_0(x_0) = \nabla f(x_0)$,
- menetapkan $d_0 = -\mathbf{g}_0$, dan
- menetapkan dugaan awal nilai α_0 yaitu $\alpha_0 = \frac{1}{\|\mathbf{g}_0\|}$.

Langkah 2

- untuk $k = 0$ sampai dengan maksimum iterasi, melakukan **Langkah 3** sampai **Langkah 6**,
- jika $\|\mathbf{g}_k\|_\infty \leq 10^{-6}$ maka iterasi dihentikan.

Langkah 3

- menghitung nilai $\alpha_k > 0$ yang memenuhi kondisi pencarian arah wolfe pada persamaan (2.3) dan (2.4),
- mengganti nilai $x_{k+1} = x_k + \alpha_k d_k$,
- menghitung nilai $g_{k+1}(x_{k+1}) = \nabla f(x_{k+1})$,
- menentukan nilai $S_k = x_{k+1} - x_k$,
- menentukan nilai $y_k = g_{k+1} - g_k$.

Langkah 4

- menghitung parameter θ_k :
 - ♦ jika $(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2 = 0$ maka, $\theta_k = 0$ dan
 - ♦ jika $(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2 \neq 0$ maka,
$$\theta_k = \frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2}$$

Langkah 5

- menghitung parameter conjugate gradient β_k^N :
 - ♦ jika $\theta_k \leq 0$ maka
$$\beta_k^N = \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{g_k^T y_k},$$
 - ♦ jika $0 < \theta_k < 1$ maka
$$\beta_k^N = (1 - \theta_k) \beta_k^{PRP} + \theta_k \beta_k^{DY}$$
$$\beta_k^N = (1 - \theta_k) \frac{g_{k+1}^T y_k}{g_k^T y_k} + \theta_k \frac{g_{k+1}^T g_k}{y_k^T S_k}, \text{ dan}$$
 - ♦ jika $\theta_k \geq 1$ maka
$$\beta_k^N = \beta_k^{DY} = \frac{g_{k+1}^T g_k}{y_k^T S_k}.$$

Langkah 6

- menghitung arah d_{k+1} ,
- menentukan nilai $d = -g_{k+1} + \beta_k^N S_k$,
- menghitung arah d_{k+1} :
 - ♦ jika $|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2$ maka
$$d_{k+1} = -g_{k+1},$$
 - ♦ jika $|g_{k+1}^T g_k| < 0.2 \|g_{k+1}\|^2$ maka
$$d_{k+1} = d,$$
- menghitung nilai $\alpha_{k+1} = \alpha_k \frac{\|d_k\|}{\|d_{k+1}\|}$,
- menetapkan nilai $k = k + 1$, dan
- kembali ke **Langkah 2.** jika salah satu kondisi telah terpenuhi, yaitu iterasi telah sampai maksimum iterasi atau $\|g_k\|_\infty \leq 10^{-6}$ maka proses dihentikan

(Neculai, 2008).

2.4 Algoritma Genetika

2.4.1 Pandangan Umum

Terlepas benar tidaknya teori evolusi Darwin, algoritma genetika menggunakan konsep evolusi Darwin dan diaplikasikan untuk memecahkan masalah optimasi.

John Holland dalam Suyanto (2005) menjelaskan bahwa proses evolusi dapat dipakai untuk memecahkan berbagai masalah optimasi dengan suatu teknik paralel yang kini dikenal sebagai algoritma genetika.

Variabel permasalahan algoritma genetika dikodekan menjadi struktur string dengan panjang berhingga. Setelah skema pengkodean ditentukan, algoritma genetika diinisialisasi untuk sebuah populasi dengan N kromosom. Masing-masing kromosom

akan didekodekan menjadi individu dengan nilai *fitness* tertentu. Sebuah populasi baru dihasilkan dengan menggunakan mekanisme seleksi alamiah, yaitu memilih individu-individu secara proporsional terhadap nilai *fitness*-nya dan genetika alamiah, yakni pindah silang dan mutasi (Suyanto, 2005).

Secara garis besar, algoritma genetika dapat dijelaskan sebagai berikut:

1. [**mulai**] membangun populasi secara acak sebanyak n kromosom (sesuai dengan masalahnya),
2. [**fitness**] evaluasi nilai *fitness* setiap kromosom x pada populasi,
3. [**populasi baru**] membuat populasi baru dengan mengulang langkah-langkah berikut sampai populasi baru lengkap:
 - a. [**seleksi**] pilih dua kromosom induk dari populasi berdasarkan *fitness*-nya (semakin besar *fitness*-nya semakin besar kemungkinannya untuk terpilih),
 - b. [**perkawinan silang**] sesuai dengan besarnya kemungkinan perkawinan silang, induk terpilih disilangkan untuk membentuk anak. Jika tidak ada perkawinan silang, maka anak merupakan salinan induknya,
 - c. [**mutasi**] sesuai dengan besarnya kemungkinan mutasi, anak dimutasi pada *loci* (posisi pada kromosom),
 - d. [**penerimaan**] tempatkan anak baru pada 'populasi baru',
4. [**ganti**] gunakan populasi yang terbentuk pada generasi selanjutnya untuk proses algoritma selanjutnya,
5. [**tes**] jika kondisi akhir terpenuhi, berhenti, dan hasilnya adalah solusi terbaik dari populasi saat itu, dan
6. [**ulangi**] ke nomer 2

(Leonardo, 2003).

2.4.2 Komponen-komponen Algoritma Genetika

Berikut ini tujuh komponen algoritma genetika yang banyak digunakan dalam berbagai metode:

2.4.2.1 Skema Pengkodean

Terdapat tiga skema yang paling umum digunakan dalam pengkodean, yaitu:

1. *real number encoding*

Pada skema ini, nilai gen berada dalam interval $[0, R]$, di mana R adalah bilangan real positif dan biasanya $R=1$. Lebih jelasnya pada Tabel 2.1.

Tabel 2.1 Skema Pengkodean *real number encoding*

x_1	x_2	x_3
0,2390	1,0000	0,0131
g_1	g_2	g_3

2. *discrete decimal encoding*

Pada skema ini, setiap gen bernilai salah satu bilangan bulat dalam interval, yaitu pada contoh Tabel 2.2.

Tabel 2.2 Skema Pengkodean *discrete decimal encoding*

x_1			x_2			x_3		
2	3	9	9	9	9	0	1	3
g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9

3. *binary encoding*

Pada skema ini, setiap gen hanya bernilai 0 atau 1. Lebih jelasnya dapat dilihat pada Tabel 2.3.

Tabel 2.3 Skema Pengkodean *binary encoding*

x_1			x_2			x_3		
0	1	0	1	1	1	0	0	0
g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9

Berdasarkan Tabel 2.1, terdapat tiga variabel x_1, x_2, x_3 yang dikodekan ke dalam sebuah kromosom yang terdiri dari 3 gen (untuk *real number encoding*). Sedangkan pada *discrete decimal encoding* maupun *binary encoding*, ketiga variabel dikodekan ke dalam kromosom yang terdiri dari 9 gen (masing-masing variabel dikodekan ke dalam 3 gen) (Suyanto, 2005).

Dekode kromosom adalah mengubah kromosom dari *real-number encoding*, *discrete decimal encoding* dan *binary encoding* menjadi bilangan riil kemudian disubstitusikan sebagai nilai dari variabel $x_{i,j}$ untuk dievaluasi ke dalam fungsi yang dicari nilai

optimalnya. Dekode kromosom untuk masing-masing *encoding*, adalah sebagai berikut:

1. *real number encoding*

$$x_{i,j} = r_b + (r_a - r_b)g_k$$

2. *discrete decimal encoding*

$$x_{i,j} = r_b + (r_a - r_b)(g_1 \times 10^{-1} + \dots + g_k \times 10^{-k} + \dots + g_N \times 10^{-N})$$

3. *binary encoding*

$$x_{i,j} = r_b + (r_a - r_b)(g_1 \times 2^{-1} + \dots + g_k \times 2^{-k} + \dots + g_N \times 2^{-N}) \quad (2.9)$$

di mana r_a = batas bawah interval

r_b = batas atas interval

g_i = nilai gen pada gen ke- i

i = indeks individu (individu ke-1,2,3,..., n)

j = indeks variabel

k = indeks gen (gen ke-1,2,..., j ,..., N)

n = jumlah individu dalam satu populasi (ukuran populasi)

N = jumlah gen yang mewakili satu variabel

(Suyanto, 2005).

2.4.2.2 Evaluasi

Hal penting dalam algoritma genetika adalah evaluasi individu/kromosom untuk menghasilkan keturunan berikutnya. Suatu individu dievaluasi berdasarkan nilai fungsi *fitness* tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Terdapat dua macam *fitness* yang digunakan, yaitu :

1. pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi f maka fungsi *fitness* yang digunakan adalah:

$$fitness = f,$$

2. sedangkan, jika solusi yang dicari adalah meminimumkan sebuah fungsi f , maka fungsi fitness yang digunakan adalah

$$fitness = \frac{1}{(f + a)}$$

a adalah bilangan yang sangat kecil untuk menghindari pembagian dengan 0 dan nilai f harus positif.

Linear fitness ranking adalah mekanisme penskalaan nilai-nilai fitness. Metode yang digunakan adalah:

$$LFR(kromosom_ke(urut_ke(j))) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(j)-1}{n-1} \right) \quad (2.10)$$

- di mana
- $j = n, n-1, \dots, 3, 2, 1$
 - n = jumlah individu dalam populasi (ukuran populasi)
 - $R(j)$ = ranking individu ke j
 - $R(j) = 1$, jika j individu bernilai fitness tertinggi
 - $R(j) = n$, jika j individu bernilai fitness terendah
- (Suyanto, 2005).

2.4.2.3 Seleksi

Suatu metode seleksi umum yang digunakan untuk seleksi adalah *Roulette Wheel Selection* (seleksi roda roulette). Langkah-langkah metode ini adalah:

1. bangkitkan bilangan acak riil (r) pada selang $[0, 1]$,
2. jika $r < q_1$, maka pilih individu pertama. Selain itu pilih individu ke- k ($2 \leq k \leq n$) di mana $q_{k-1} < r < q_k$.

Langkah-langkah untuk mendapatkan nilai q_i (probabilitas kumulatif) adalah sebagai berikut:

1. hitung nilai *Linear Fitness Ranking* masing-masing individu
- $$LFR(k), \quad (2.11)$$
2. jumlahkan nilai fitness semua individu dalam populasi

$$Tot.LFR = \sum_{k=1}^n LFR(k), \quad (2.12)$$

3. hitung probabilitas terpilih setiap individu

$$p_k = \frac{LFR(k)}{Tot.LFR}, \quad (2.13)$$

4. hitung probabilitas kumulatif

$$q_k = \sum_{j=1}^k p_k, \quad (2.14)$$

di mana $LFR(k)$ = *Linear Fitness Ranking* individu ke k

k = 1,2,3,...,n

n = jumlah individu dalam populasi (ukuran populasi)

(Gen dan Cheng, 1997).

2.4.2.4 Pindah Silang (*Crossover*)

Pindah silang adalah proses menyilangkan gen-gen pada dua individu induk untuk mendapatkan dua individu baru. Metode yang digunakan adalah pindah silang satu titik potong (*one-point crossover*). Suatu titik potong dipilih secara acak, kemudian bagian pertama dari induk 1 digabungkan dengan bagian kedua dari induk 2, sebaliknya bagian kedua dari induk 1 digabungkan dengan bagian pertama dari induk 2.

Pemilihan individu induk hanya dapat dilakukan dengan suatu probabilitas tertentu $P_{crossover}$ sebagai berikut:

1. bangkitkan bilangan acak riil (r) pada selang $[0,1]$,
2. jika $r < P_{crossover}$, pilih individu untuk dilakukan *crossover*.

2.4.2.5 Mutasi

Proses ini mengubah gen-gen pada setiap individu untuk mendapatkan individu baru. Pada *binary encoding*, dilakukan dengan cara sebagai berikut:

1. bangkitkan bilangan acak riil (r) pada selang $[0,1]$,
2. jika $r < P_{mutasi}$, maka 0 diubah 1 dan 1 diubah 0.

Pada umumnya P_{mutasi} dideklarasikan sebagai $\frac{1}{N}$, di mana N adalah jumlah gen dalam satu individu.

2.4.2.6 *Elitisme*

Proses pembuatan satu atau beberapa salinan individu ini dilakukan untuk menjaga agar individu dengan nilai *fitness* tertinggi tidak hilang selama evolusi (Suyanto, 2005).

2.4.2.7 **Penggantian Populasi**

Ukuran populasi menyatakan banyaknya individu dalam satu generasi. Setelah dilakukan proses seleksi, pindah silang dan mutasi, kemudian dilakukan penggantian populasi (*generational replacement*). Artinya semua individu generasi sebelumnya digantikan sekaligus oleh individu hasil seleksi, pindah silang dan mutasi (Suyanto, 2005).

i.2.4.3 **Parameter Genetika**

Terdapat beberapa parameter algoritma genetika yang perlu diberi nilai, antara lain:

2.4.3.1 **Ukuran Populasi**

Ukuran populasi atau *pop_size*, merupakan parameter yang berfungsi untuk menentukan jumlah individu/kromosom yang berada pada populasi (untuk setiap generasi). Semakin banyak dan beragamnya individu akan memberikan peluang lebih besar untuk menemukan individu yang mendekati sempurna. Jika individu yang ada terlalu sedikit, algoritma genetika hanya memiliki sedikit kemungkinan untuk melakukan *crossover* dan hanya sebagian kecil *search space* yang akan di-*explore*. Namun jika terdapat terlalu banyak individu, maka *performance* algoritma genetika juga akan menjadi lambat. Penelitian menunjukkan bahwa pada batasan tertentu, meningkatkan ukuran populasi akan menjadi sangat tidak berguna, karena pemecahan masalah tidak akan berjalan lebih cepat (Obitko, 1998).

2.4.3.2 Jumlah Generasi

Jumlah generasi mempunyai andil yang besar dalam menemukan individu yang lebih baik, karena semakin besar generasi maka individu yang dihasilkan akan semakin baik dan sempurna. Tetapi tidak berarti semakin besar jumlah generasi maka individu yang dihasilkan selalu lebih baik karena ada suatu saat nilai *fitness* semua individu menjadi sama.

2.4.3.3 Peluang *Crossover* (P_c)

Parameter ini berfungsi untuk menentukan seberapa sering proses *crossover* dilakukan. Semakin besar probabilitas *crossover* akan memungkinkan pencapaian alternatif solusi yang lebih bervariasi dan mengurangi kemungkinan menghasilkan nilai optimum yang tidak dikehendaki. Tetapi bila nilai ini terlalu tinggi akan mengakibatkan pemborosan waktu untuk melakukan perhitungan di daerah solusi yang tidak menjanjikan hasil yang optimal. Nilai probabilitas *crossover* yang sering dipakai berkisar antara 0.65 sampai 0.95 (Obitko, 1998).

2.4.3.4 Peluang Mutasi (P_m)

Parameter ini menunjukkan seberapa sering sebuah individu akan dimutasi. Jika tidak terdapat mutasi, maka semua anak akan berasal dari proses *crossover* tanpa adanya perubahan. Jika terdapat proses mutasi, maka sebagian individu tersebut akan berubah. Nilai probabilitas mutasi yang terlalu rendah dapat mengakibatkan gen-gen yang berpotensi tidak dicoba. Sebaliknya jika terlalu tinggi, dapat menghilangkan sifat kemiripan dengan *parentnya*, sehingga proses pencarian salah dan menjauh dari daerah solusi yang diinginkan. Nilai probabilitas mutasi yang sering dipakai berkisar antara 0.01 sampai 0.1 (Obitko, 1998).

2.4.4 Kekonvergenan Algoritma Genetika

Algoritma genetika tidak menjamin dapat memperoleh solusi optimal. Algoritma genetika berhenti pada suatu kondisi jika memenuhi kriteria yang telah ditentukan sebelumnya. Dalam hal ini, proses berhenti setelah iterasi berulang sebanyak n kali. Konvergen

merupakan kondisi yang dicapai pada saat populasi kehilangan keanekaragaman, artinya sebagian besar individu dalam populasi mempunyai karakteristik yang sama.

Algoritma genetika dapat mencapai kondisi konvergen meskipun tidak dapat dipastikan bahwa solusi yang diperoleh merupakan solusi optimal. Kesalahan yang sering terjadi adalah populasi memperoleh solusi yang belum optimal, sementara kriteria terminasi sudah dipenuhi (Setiawan, 2003).

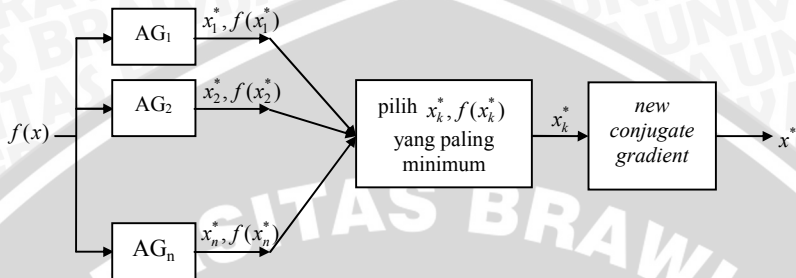
2.5 Paralel Algoritma Genetika

Dalam skripsi ini, model algoritma genetika yang digunakan adalah model paralel. Paralel algoritma genetika adalah kombinasi dari beberapa algoritma genetika berdasarkan banyaknya parameter peluang silang dan peluang mutasi. Misalkan terdapat tiga peluang silang yaitu $\{0.95, 0.85, 0.75\}$ dan dua peluang mutasi yaitu $\{0.01, 0.02\}$. Banyaknya paralel yang dibangkitkan sebanyak enam algoritma genetika dengan pasangan peluang silang dan peluang mutasi yaitu

$\{(0.95, 0.01); (0.95, 0.02); (0.85, 0.01); (0.85, 0.02); (0.75, 0.01); (0.75, 0.02)\}$.

2.6 Hibrid Algoritma Genetika dengan *New Conjugate Gradient*

Dalam banyak penelitian mengungkapkan bahwa algoritma genetika memiliki kelemahan dalam pencarian solusi yang belum optimal. Hal ini dikarenakan masalah kekonvergenan algoritma genetika yang prematur. Dalam skripsi ini, paralel algoritma genetika digabungkan dengan metode *new conjugate gradient* yang selanjutnya disebut hibrid algoritma genetika untuk mendapatkan solusi optimal seperti pada gambar berikut:



Gambar 2.3 Skema Hibridisasi Algoritma Genetika dan *New Conjugate Gradient*.



UNIVERSITAS BRAWIJAYA



BAB III PEMBAHASAN

Algoritma genetika adalah suatu teknik optimasi global yang terkenal, diinspirasi oleh proses seleksi alam dan teori genetik populasi. Algoritma genetika menyediakan suatu arsitektur umum untuk menyelesaikan masalah optimasi yang kompleks dan banyak digunakan pada masalah praktis dengan fokus pada pencarian parameter-parameter optimal.

Akan tetapi algoritma genetika tidak menjamin bahwa solusi yang diberikan adalah solusi optimal. Oleh karena itu, dalam skripsi ini algoritma genetika digabungkan dengan algoritma optimasi lokal *new conjugate gradient* yang akan melanjutkan pencarian solusi dari hasil yang diperoleh algoritma genetika. Dalam bab ini akan dibahas bagaimana gabungan antara algoritma genetika dengan algoritma *new conjugate gradient* pada metode hibrid algoritma genetika.

3.1 Arah *Descent* dan Kondisi *Descent*

Definisi 3.1 (Arah *Descent* dan kondisi *Descent*)

Arah *descent* dalam optimasi adalah vektor $d \in \mathbb{R}^n$ untuk mendekati nilai minimum $x^* \in \mathbb{R}^n$ fungsi objektif $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

Kondisi *descent* adalah kondisi $\langle d, \nabla f(x_k) \rangle < 0$ yang harus dipenuhi arah *descent* $d \in \mathbb{R}^n$ (Verena, 2009).

Teorema 3.1 (Arah *Descent*)

Diasumsikan bahwa α_k pada persamaan (2.5) ditentukan dengan *Wolfe line search* persamaan (2.3) dan (2.4).

$$\text{Jika } 0 < \theta_k < 1, \text{ dan } \left\| \frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \right\| \|\mathbf{g}_{k+1}\|^2 \geq \frac{(\mathbf{g}_{k+1}^T \mathbf{y}_k)(\mathbf{g}_{k+1}^T \mathbf{S}_k)}{\|\mathbf{g}_k\|^2} \quad (3.1)$$

maka arah d_{k+1} pada persamaan (2.8) adalah arah *descent*.

Bukti:

Persamaan (2.8) dikalikan dengan \mathbf{g}_{k+1}^T diperoleh

$$\mathbf{g}_{k+1}^T d_{k+1} = -\mathbf{g}_{k+1}^T \mathbf{g}_{k+1} + (1 - \theta_k) \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k$$

karena $0 < \theta_k < 1$ maka diperoleh

$$\begin{aligned} \mathbf{g}_{k+1}^T d_{k+1} &= -\mathbf{g}_{k+1}^T \mathbf{g}_{k+1} + (1 - \theta_k) \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k \\ &\leq -\mathbf{g}_{k+1}^T \mathbf{g}_{k+1} + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k + \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k \end{aligned}$$

atau

$$\begin{aligned} \mathbf{g}_{k+1}^T d_{k+1} &\leq -\|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k + \frac{\|\mathbf{g}_{k+1}\|^2}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k \\ \mathbf{g}_{k+1}^T d_{k+1} &\leq \left(-1 + \frac{\mathbf{g}_{k+1}^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \right) \|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k \\ \mathbf{g}_{k+1}^T d_{k+1} &\leq \left(-\frac{\mathbf{y}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} + \frac{\mathbf{g}_{k+1}^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \right) \|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k \quad (3.2) \end{aligned}$$

jika $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ maka pertidaksamaan (3.2) menjadi

$$\begin{aligned} \mathbf{g}_{k+1}^T d_{k+1} &\leq \frac{-(\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{S}_k + \mathbf{g}_{k+1}^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k \\ \mathbf{g}_{k+1}^T d_{k+1} &\leq \frac{-\mathbf{g}_{k+1}^T \mathbf{S}_k + \mathbf{g}_k^T \mathbf{S}_k + \mathbf{g}_{k+1}^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k \\ \mathbf{g}_{k+1}^T d_{k+1} &\leq \frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k \quad (3.3) \end{aligned}$$

Karena $\mathbf{y}_k^T \mathbf{S}_k > 0$ dari pertidaksamaan (2.4) dan $\mathbf{g}_k^T \mathbf{S}_k \leq 0$ maka

$$\frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 \leq 0. \text{ Dari persamaan (3.1):}$$

$$\begin{aligned}
\left| \frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \right| \|\mathbf{g}_{k+1}\|^2 &\geq \frac{(\mathbf{g}_{k+1}^T \mathbf{y}_k)(\mathbf{g}_{k+1}^T \mathbf{S}_k)}{\|\mathbf{g}_k\|^2} \\
-\left| \frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \right| \|\mathbf{g}_{k+1}\|^2 &\leq -\frac{(\mathbf{g}_{k+1}^T \mathbf{y}_k)(\mathbf{g}_{k+1}^T \mathbf{S}_k)}{\|\mathbf{g}_k\|^2} \\
\frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 &\leq -\frac{(\mathbf{g}_{k+1}^T \mathbf{y}_k)(\mathbf{g}_{k+1}^T \mathbf{S}_k)}{\|\mathbf{g}_k\|^2} \\
\frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 + \frac{(\mathbf{g}_{k+1}^T \mathbf{y}_k)(\mathbf{g}_{k+1}^T \mathbf{S}_k)}{\|\mathbf{g}_k\|^2} &\leq 0. \tag{3.4}
\end{aligned}$$

Dari pertidaksamaan (3.3) dan (3.4) diperoleh

$$\mathbf{g}_{k+1}^T \mathbf{d}_{k+1} \leq \frac{\mathbf{g}_k^T \mathbf{S}_k}{\mathbf{y}_k^T \mathbf{S}_k} \|\mathbf{g}_{k+1}\|^2 + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k \leq 0. \tag{3.5}$$

Pada pertidaksamaan (3.5) didapatkan $\mathbf{g}_{k+1}^T \mathbf{d}_{k+1} \leq 0$ dan \mathbf{d}_{k+1} adalah salah satu arah *descent*.

Teorema 3.2 (Kondisi *Descent*)

Diasumsikan bahwa $(\mathbf{g}_{k+1}^T \mathbf{y}_k)(\mathbf{g}_{k+1}^T \mathbf{S}_k) \leq 0$. Jika $0 < \theta_k < 1$ maka arah \mathbf{d}_{k+1} yang diberikan oleh persamaan (2.8) memenuhi syarat cukup kondisi *descent*

$$\mathbf{g}_{k+1}^T \mathbf{d}_{k+1} \leq -\left(1 - \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{S}_k}{\mathbf{g}_k^T \mathbf{y}_k}\right) \|\mathbf{g}_{k+1}\|^2.$$

Bukti:

Persamaan (2.8) dikalikan dengan \mathbf{g}_{k+1}^T didapatkan:

$$\mathbf{g}_{k+1}^T \mathbf{d}_{k+1} = -\mathbf{g}_{k+1}^T \mathbf{g}_{k+1} + (1 - \theta_k) \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{g}_k^T \mathbf{g}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k$$

$$\mathbf{g}_{k+1}^T \mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + (1 - \theta_k) \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\|\mathbf{g}_k\|^2} \mathbf{g}_{k+1}^T \mathbf{S}_k + \theta_k \frac{\|\mathbf{g}_{k+1}\|^2}{\mathbf{y}_k^T \mathbf{S}_k} \mathbf{g}_{k+1}^T \mathbf{S}_k$$

$$g_{k+1}^T d_{k+1} = - \left(1 - \theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} \right) \|g_{k+1}\|^2 + (1 - \theta_k) \frac{(g_{k+1}^T y_k)(g_{k+1}^T S_k)}{\|g_k\|^2}.$$

Jika $(g_{k+1}^T y_k)(g_{k+1}^T S_k) \leq 0$, $g_k^T S_k \leq 0$ dan $y_k^T S_k > 0$ (dari persamaan (2.4)) maka

$$\begin{aligned} g_{k+1}^T d_{k+1} &= - \left(1 - \theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} \right) \|g_{k+1}\|^2 + (1 - \theta_k) \frac{(g_{k+1}^T y_k)(g_{k+1}^T S_k)}{\|g_k\|^2} \\ &\leq - \left(1 - \theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} \right) \|g_{k+1}\|^2 \end{aligned}$$

atau

$$g_{k+1}^T d_{k+1} \leq - \left(1 - \theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} \right) \|g_{k+1}\|^2 \leq 0.$$

Harus ditunjukkan bahwa $1 - \theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} > 0$.

$$y_k^T S_k = (g_{k+1}^T - g_k^T) S_k$$

$$y_k^T S_k = g_{k+1}^T S_k - g_k^T S_k$$

$$g_{k+1}^T S_k = y_k^T S_k + g_k^T S_k$$

karena $y_k^T S_k > 0$ yang diperoleh dari persamaan (2.4) dan $g_k^T S_k \leq 0$ sehingga

$$g_{k+1}^T S_k = y_k^T S_k + g_k^T S_k < y_k^T S_k$$

$$g_{k+1}^T S_k < y_k^T S_k$$

$$1 < \frac{y_k^T S_k}{g_{k+1}^T S_k}. \quad (3.6)$$

Pada pertidaksamaan (3.6), jika $0 < \theta_k < 1$ maka $\theta_k < \frac{y_k^T S_k}{g_{k+1}^T S_k}$ atau

$\theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} < 1$ dan diperoleh bahwa $1 - \theta_k \frac{g_{k+1}^T S_k}{y_k^T S_k} > 0$, teorema terbukti.

Parameter θ_k dipilih sedemikian rupa sehingga kondisi konjugasi $y_k^T d_{k+1} = 0$ terpenuhi pada setiap iterasi dan terpisah dari *line search*. Dari $y_k^T d_{k+1} = 0$ dan persamaan (2.8) yang dikalikan dengan y_k^T diperoleh:

$$\begin{aligned}
 y_k^T d_{k+1} &= -y_k^T g_{k+1} + (1 - \theta_k) \frac{g_{k+1}^T y_k}{g_k^T g_k} y_k^T S_k + \theta_k \frac{\|g_{k+1}\|^2}{y_k^T S_k} y_k^T S_k \\
 0 &= -y_k^T g_{k+1} + \frac{g_{k+1}^T y_k}{g_k^T g_k} y_k^T S_k - \theta_k \frac{g_{k+1}^T y_k}{g_k^T y_k} y_k^T S_k + \theta_k \|g_{k+1}\|^2 \\
 0 &= -y_k^T g_{k+1} + \frac{(g_{k+1}^T y_k)(y_k^T S_k)}{g_k^T g_k} + \theta_k \left(-\frac{(g_{k+1}^T y_k)(y_k^T S_k)}{g_k^T g_k} + \|g_{k+1}\|^2 \right) \\
 0 &= \frac{-(y_k^T g_{k+1})(g_k^T g_k) + (y_k^T g_{k+1})(y_k^T S_k)}{g_k^T g_k} + \theta_k \left(-\frac{(y_k^T g_{k+1})(y_k^T S_k)}{g_k^T g_k} + \|g_{k+1}\|^2 \right) \\
 \theta_k \left(\frac{(y_k^T g_{k+1})(y_k^T S_k) - \|g_k\|^2 \|g_{k+1}\|^2}{\|g_k\|^2} \right) &= \left(\frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{\|g_k\|^2} \right) \\
 \theta_k &= \left(\frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{\|g_k\|^2} \right) \times \left(\frac{\|g_k\|^2}{(y_k^T g_{k+1})(y_k^T S_k) - \|g_k\|^2 \|g_{k+1}\|^2} \right) \\
 \theta_k &= \left(\frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T S_k) - \|g_k\|^2 \|g_{k+1}\|^2} \right). \tag{3.7}
 \end{aligned}$$

Parameter θ_k yang digunakan pada persamaan (2.7) harus berada pada selang $[0,1]$. Pada persamaan (3.7), nilai parameter θ_k dapat

keluar dari interval [0.1]. Jika nilai parameter $\theta_k \leq 0$, maka pada persamaan (2.7) ditetapkan $\theta_k = 0$ dan diperoleh $\beta_k^N = \beta_k^{PRP}$. Jika nilai parameter $\theta_k \geq 1$, maka pada persamaan (2.7) ditetapkan $\theta_k = 1$ dan diperoleh $\beta_k^N = \beta_k^{DY}$. Di sisi lain, jika $0 < \theta_k < 1$ maka β_k^N merupakan kombinasi β_k^{PRP} dan β_k^{DY} .

Arah d_{k+1} pada persamaan (2.8) merupakan kombinasi metode PRP dan metode DY berdasarkan aturan θ_k yang terpilih.

3.2 Studi Kasus

Ditinjau masalah optimasi sebagai berikut:

$$\text{minimalkan fungsi objektif } f(x_1, x_2) = x_1 e^{(-x_1^2 - x_2^2)}. \quad (3.8)$$

3.2.1 Penyelesaian Secara Analitik

Akan digunakan Definisi 2.9 dan Teorema 2.3 untuk memperoleh solusi minimum. Langkah pertama adalah mencari $\nabla f(x_1, x_2)$ yaitu dengan cara mencari turunan parsial fungsi objektif terhadap x_1 dan x_2 :

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f(x_1, x_2)}{\partial x_1} \\ \frac{\partial f(x_1, x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} e^{(-x_1^2 - x_2^2)} - 2x_1^2 e^{(-x_1^2 - x_2^2)} \\ -2x_1 x_2 e^{(-x_1^2 - x_2^2)} \end{bmatrix}.$$

Selanjutnya turunan parsial tersebut disamakan dengan nol, yaitu:

$$\frac{\partial f(x_1, x_2)}{\partial f(x_1)} = 0$$

$$e^{(-x_1^2 - x_2^2)} - 2x_1^2 e^{(-x_1^2 - x_2^2)} = 0$$

$$2x_1^2 e^{(-x_1^2 - x_2^2)} = e^{(-x_1^2 - x_2^2)}$$

$$2x_1^2 = 1$$

$$x_1^2 = \frac{1}{2}$$

$x_1 = \pm \frac{1}{2}\sqrt{2}$ dan $x_2 = x_2$ sehingga diperoleh penyelesaian:

$$\left\{ \left(x_1 = -\frac{1}{2}\sqrt{2}, x_2 = x_2 \right), \left(x_1 = \frac{1}{2}\sqrt{2}, x_2 = x_2 \right) \right\}. \quad (3.9)$$

$$\frac{\partial f(x_1, x_2)}{\partial f(x_2)} = 0$$

$$-2x_1 x_2 e^{(-x_1^2 - x_2^2)} = 0$$

$$x_1 = 0 \quad \text{atau} \quad x_2 = 0.$$

Nilai $x_1 = 0$ diperoleh $x_2 = x_2$ dan untuk nilai $x_2 = 0$ diperoleh $x_1 = x_1$, sehingga diperoleh penyelesaian:

$$\left\{ (x_1 = 0, x_2 = x_2), (x_1 = x_1, x_2 = 0) \right\}. \quad (3.10)$$

Himpunan penyelesaian (3.9) dan (3.10) dikombinasikan sehingga didapatkan titik ekstrim:

$$\left\{ \left(-\frac{1}{2}\sqrt{2}, 0 \right), \left(\frac{1}{2}\sqrt{2}, 0 \right), (0, 0) \right\}.$$

Langkah kedua adalah mencari $\nabla^2 f(x_1, x_2)$ atau matriks *Hessian* yaitu:

$$\begin{aligned} \nabla^2 f(x_1, x_2) &= Hf(x_1, x_2) \\ &= \begin{bmatrix} \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} & \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} \end{bmatrix} \\ &= \begin{bmatrix} -6x_1 e^{(-x_1^2 - x_2^2)} + 4x_1^3 e^{(-x_1^2 - x_2^2)} & -2x_2 e^{(-x_1^2 - x_2^2)} + 4x_1^2 x_2 e^{(-x_1^2 - x_2^2)} \\ -2x_2 e^{(-x_1^2 - x_2^2)} + 4x_1^2 x_2 e^{(-x_1^2 - x_2^2)} & -2x_1 e^{(-x_1^2 - x_2^2)} + 4x_1 x_2^2 e^{(-x_1^2 - x_2^2)} \end{bmatrix}. \end{aligned}$$

Selanjutnya setiap titik ekstrim yang diperoleh, dicek menggunakan matriks *Hessian*, yaitu:

$$Hf\left(-\frac{1}{2}\sqrt{2}, 0\right) = \begin{bmatrix} 2.2028 & 0 \\ 0 & 2.0207 \end{bmatrix}$$

determinan submatriks utama 1 = $2.2028 \geq 0$ dan $2.0207 \geq 0$,

determinan submatriks utama 2 = $(2.2028 \times 2.0207) - (0 \times 0)$
 $= 4.4512 \geq 0$.

Karena determinan submatriks utama bernilai positif (≥ 0) maka matriks *Hessian* $Hf\left(-\frac{1}{2}\sqrt{2}, 0\right)$ definit semi positif, sehingga titik

$\left(-\frac{1}{2}\sqrt{2}, 0\right)$ merupakan titik minimum lokal.

$$Hf\left(\frac{1}{2}\sqrt{2}, 0\right) = \begin{bmatrix} -1.7155 & 0 \\ 0 & -0.8077 \end{bmatrix}$$

determinan submatriks utama 1 = $-1.7155 \leq 0$ dan $-0.8077 \leq 0$,

determinan submatriks utama 2 = $(-1.7155 \times -0.8077) - (0 \times 0)$
 $= 1.3856 \geq 0$.

Karena determinan submatriks utama 1 bernilai negatif (≤ 0) dan submatriks utama 1 bernilai positif (≥ 0) maka matriks *Hessian* $Hf(\frac{1}{2}\sqrt{2},0)$ definit semi negatif, sehingga titik $(\frac{1}{2}\sqrt{2},0)$ merupakan titik maksimum lokal.

$$Hf(0,0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

determinan submatriks utama 1 = 0 dan 0,

determinan submatriks utama 2 = $(0 \times 0) - (0 \times 0) = 0$.

Karena determinan submatriks utama 1 dan 2 bernilai nol maka matriks *Hessian* $Hf(0,0)$ tidak dapat disimpulkan, sehingga titik $(0,0)$ bukan titik maksimum lokal maupun titik minimum lokal.

Masing-masing titik ekstrim $\left\{ \left(-\frac{1}{2}\sqrt{2},0\right), \left(\frac{1}{2}\sqrt{2},0\right), (0,0) \right\}$

hanya didapatkan satu titik minimum lokal yaitu $(-\frac{1}{2}\sqrt{2},0)$,

sehingga titik $(-\frac{1}{2}\sqrt{2},0)$ merupakan titik minimum global.

Substitusikan $(x_1 = -\frac{1}{2}\sqrt{2}, x_2 = 0)$ ke dalam fungsi objektif persamaan (3.8):

$$\begin{aligned} f(x_1 = -\frac{1}{2}\sqrt{2}, x_2 = 0) &= x_1 e^{(-x_1^2 - x_2^2)} \\ &= \left(-\frac{1}{2}\sqrt{2}\right) \times e^{\left(-\left(\frac{1}{2}\sqrt{2}\right)^2 - (0)^2\right)} \\ &= \left(-\frac{1}{2}\sqrt{2}\right) \times e^{\left(-\frac{1}{2}\right)} \\ &= -0.4289, \end{aligned}$$

sehingga solusi minimum global dari fungsi $f(x_1, x_2) = x_1 e^{(-x_1^2 - x_2^2)}$

adalah -0.4289 terjadi di $(x_1, x_2) = \left(-\frac{1}{2}\sqrt{2}, 0\right)$.

3.2.2 Penyelesaian Numerik Menggunakan Hibrid Algoritma Genetika

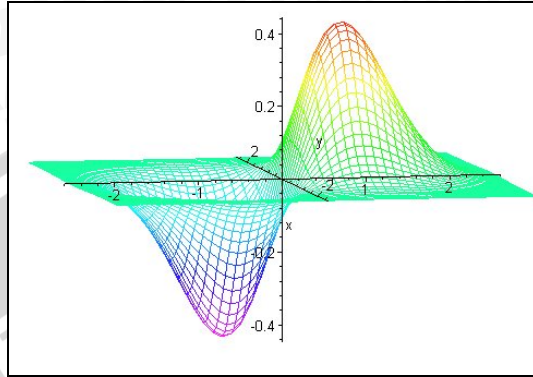
Secara umum, sistem penyelesaian numerik hibrid algoritma genetika dibuat dengan menggunakan sistem komputer dengan spesifikasi sebagai berikut:

1. Intel Pentium 4 Processor 524 (3.06 GHz, 533.2 MHz FSB, 1 MB L2 Cache),
2. memori 512 MB DDR2,
3. hard disk berukuran 80 GB.

Sedangkan sistem operasi yang digunakan adalah sistem operasi Microsoft Windows XP Professional Service Pack 3. Metode hibrid algoritma genetika diimplementasikan menggunakan perangkat lunak Matlab 7.

Hibrid algoritma genetika digunakan untuk memperoleh solusi minimum global fungsi sederhana dalam interval $-2 \leq x_1 \leq 2$ dan $-2 \leq x_2 \leq 2$. Jumlah gen yang digunakan adalah 6, jumlah individu = 10 dan maksimum generasi = 100. Dengan menggunakan beberapa peluang persilangan dan beberapa peluang mutasi yang dikombinasikan, diambil titik (x_1, x_2) dengan nilai $f(x_1, x_2)$ yang paling optimal untuk selanjutnya digunakan sebagai masukan awal algoritma *new conjugate gradient*. Nilai paling minimum yang diperoleh algoritma genetika adalah $f(x_1, x_2) = -0.1000$ dengan $(x_1, x_2) = (-0.59, -1.19)$. Setelah diselesaikan menggunakan algoritma *new conjugate gradient*, diperoleh nilai yang lebih optimal yaitu $f(x_1, x_2) = -0.4288$ dengan titik $(x_1, x_2) = (-0.7146, 0)$.

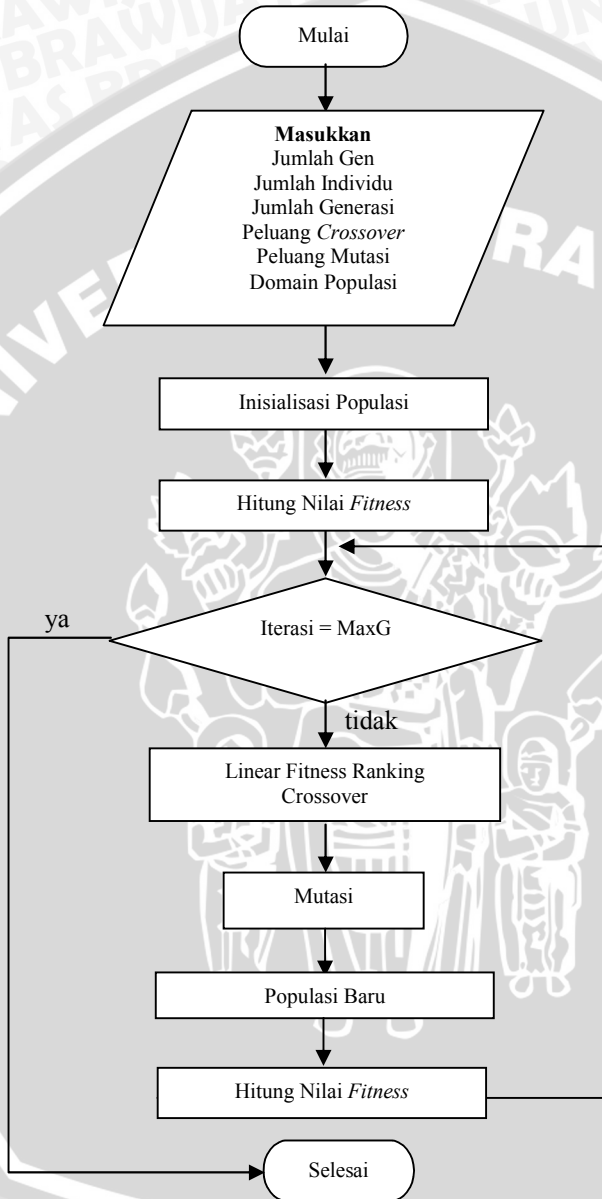
Proses perhitungan perolehan titik minimum global ini dapat dilihat pada lampiran. Untuk ilustrasi fungsi dapat dilihat pada Gambar 3.1.



Gambar 3.1 Fungsi $f = x_1 e^{-(x_1^2 - x_2^2)}$

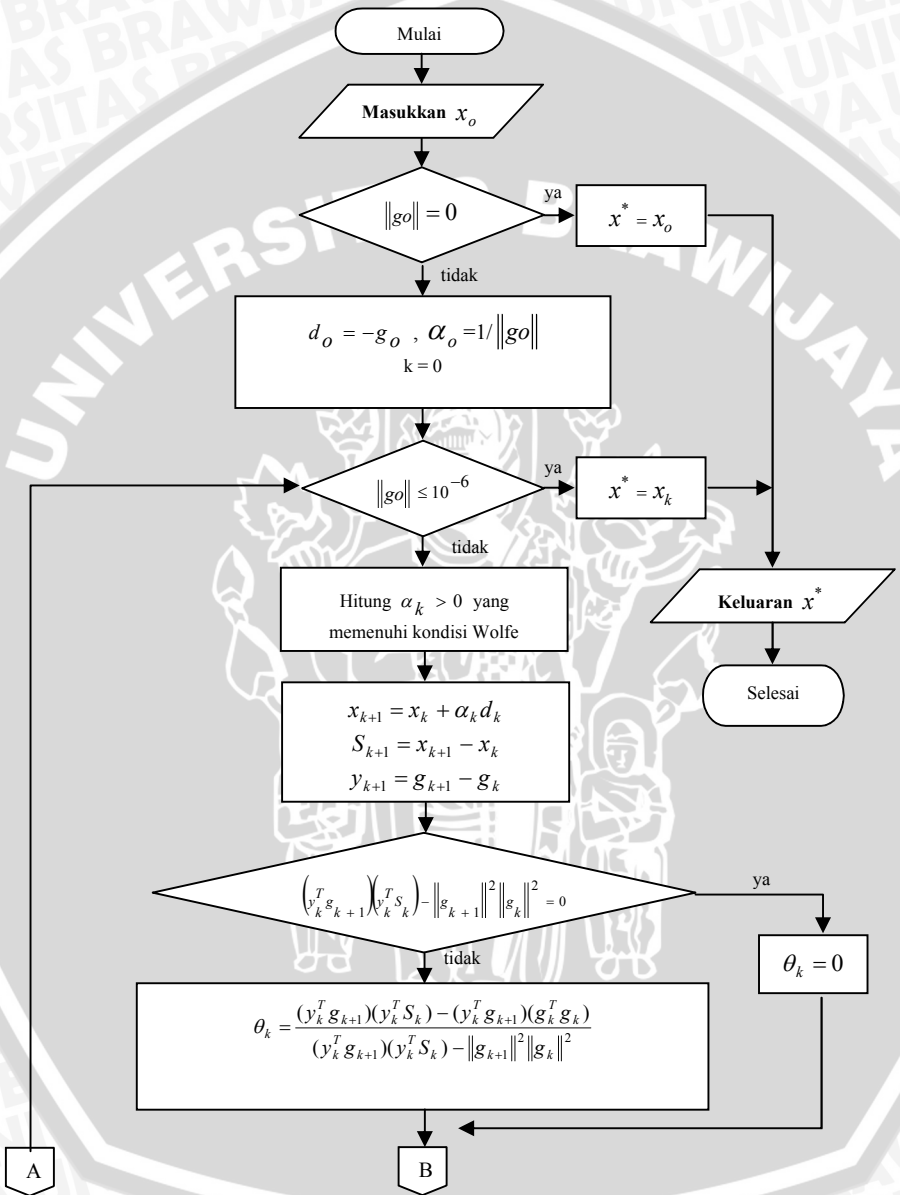


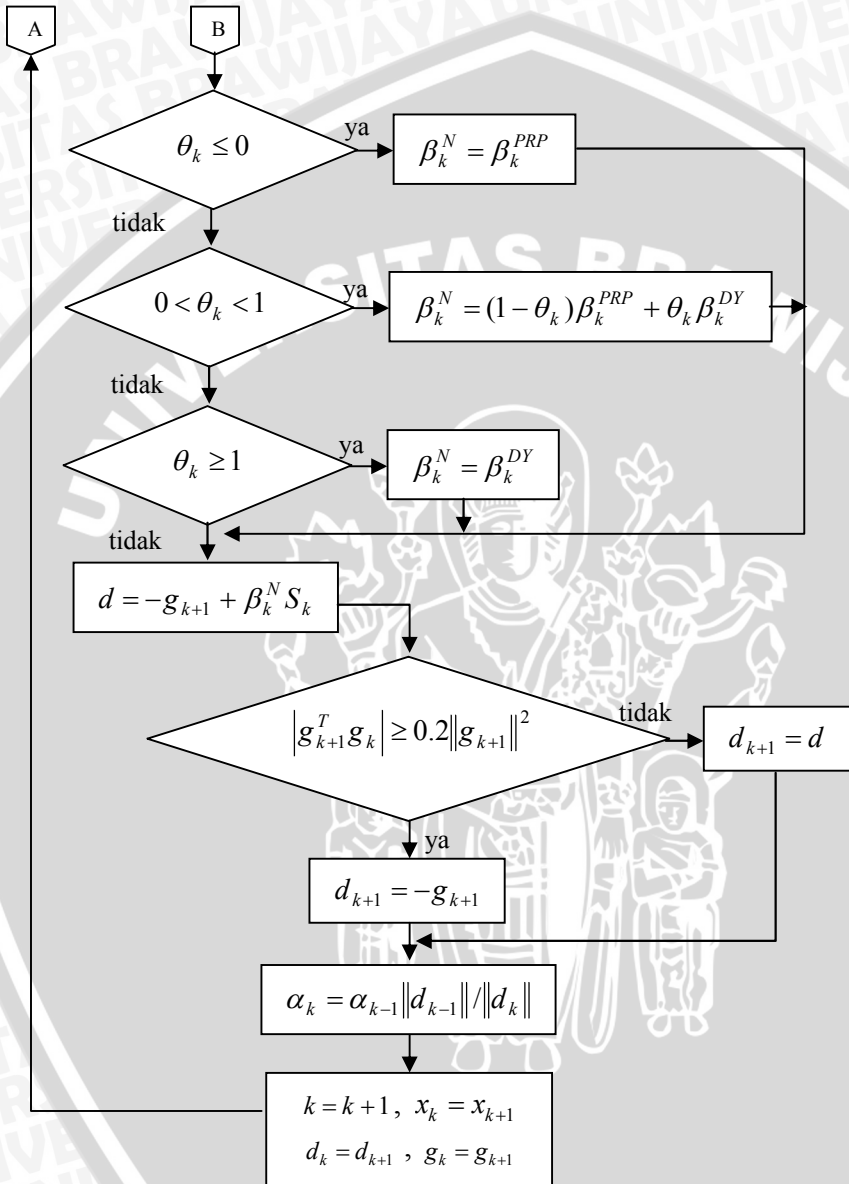
3.3 Diagram Alir Algoritma Genetika



Gambar 3.2 Diagram Alir Algoritma Genetika

3.4 Diagram Alir New Conjugate Gradient





Gambar 3.3 Diagram Alir New Conjugate Gradient

3.5 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Paralel Algoritma Genetika Menggunakan Beberapa Tes Fungsi

Pada studi kasus dalam skripsi ini akan dibandingkan solusi minimum metode hibrid algoritma genetika dengan paralel algoritma genetika untuk beberapa tes fungsi dua variabel termasuk waktu komputasi dari kedua metode tersebut. Selain itu, dalam skripsi ini akan dicari pengaruh parameter ukuran populasi terhadap solusi dan waktu komputasi.

Parameter peluang persilangan yang digunakan adalah $\{0.95, 0.85, 0.75\}$ dan peluang mutasi $\{0.01, 0.02, 0.03, 0.04\}$. Kedua metode ini menggunakan parameter maksimum generasi sejumlah 50. Metode hibrid algoritma genetika diujikan pada tes fungsi unimodal yaitu tes fungsi De Jong dan beberapa tes fungsi multimodal. Karakteristik dari masing-masing fungsi tersebut dapat dilihat pada lampiran. Ditinjau dari solusi numerik, diperoleh data sebagai berikut:

a. tes fungsi De Jong

Fungsi De Jong dengan solusi eksak minimum global $f(x^*) = 0$.

Tabel 3.1 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi De Jong

Uk Pop	Hibrid AG					Paralel AG				
	P.S	P.M	$f(\bar{x})$ ($\times 10^{-13}$)	E ($\times 10^{13}$)	t	P.S	P.M	$f(\bar{x})$	E	t
50	0.85	0.04	2.296	2.296	19.142	0.75	0.04	18.692	18.692	19.063
100	0.85	0.02	2.411	2.411	50.969	0.95	0.01	49.477	49.477	50.906
150	0.95	0.04	1.573	1.573	99.078	0.85	0.03	10.719	10.719	99.000
200	0.95	0.02	2.193	2.193	162.126	0.85	0.02	3.853	3.853	162.094
250	0.75	0.04	2.443	2.443	226.109	0.75	0.03	4.311	4.311	226.062

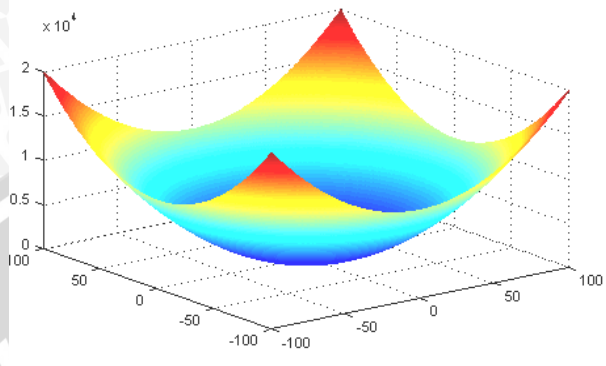
Perbandingan solusi numerik tes fungsi De Jong dapat dilihat pada nilai kesalahan mutlak (E). Semakin kecil nilai kesalahan mutlak atau semakin mendekati nol, semakin baik pula solusi

numerik yang dihasilkan. Pada ukuran populasi 50, nilai kesalahan mutlak metode hibrid algoritma genetika sebesar 2.2962×10^{-13} sedangkan pada metode paralel algoritma genetika, nilai kesalahan mutlak sebesar 18.6920. Dengan membandingkan nilai kesalahan mutlak, metode hibrid algoritma genetika jauh lebih baik dibandingkan dengan metode paralel algoritma genetika. Perbandingan nilai kesalahan mutlak secara keseluruhan pada Tabel 3.1, metode hibrid algoritma genetika jauh lebih baik daripada menggunakan metode paralel algoritma genetika.

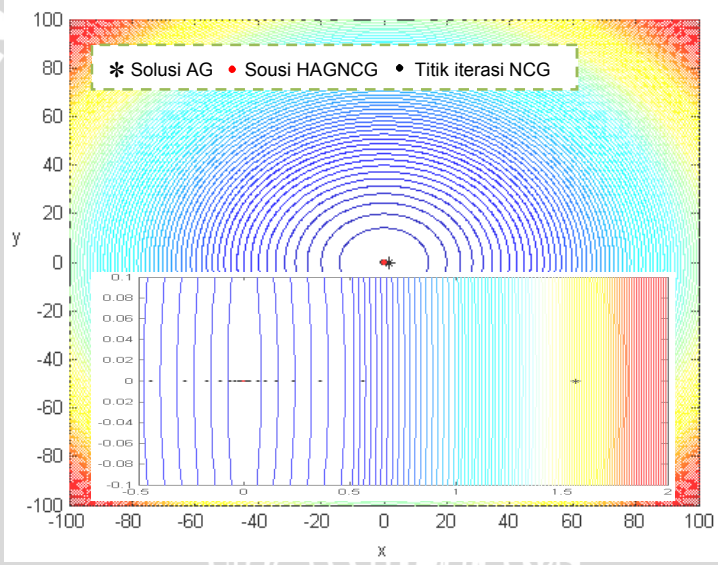
Sedangkan untuk nilai kesalahan mutlak pada masing-masing metode jika dibandingkan berdasarkan ukuran populasinya, maka tidak dapat dipastikan bagaimana pengaruh ukuran populasi terhadap solusi. Hal ini dikarenakan sebagian besar komponen algoritma genetika berbasis pada bilangan acak, sehingga solusi yang dihasilkan berbeda-beda.

Perbandingan waktu komputasi tes fungsi De Jong dapat dilihat pada kolom t . Semakin kecil nilai t , semakin sedikit pula waktu komputasi yang diperlukan untuk mendapatkan solusi optimal. Pada ukuran populasi 50, waktu yang dibutuhkan metode hibrid algoritma genetika untuk mendapatkan solusi optimal adalah 19.142 detik, sedangkan pada metode paralel algoritma genetika, waktu yang dibutuhkan adalah 19.063 detik. Dengan membandingkan waktu komputasi yang diperlukan masing-masing metode pada Tabel 3.1, metode hibrid algoritma genetika memerlukan waktu yang sebanding dengan metode paralel algoritma genetika. Dalam hal ini, waktu komputasi yang harus dibayar oleh hibrid algoritma genetika sebanding dengan paralel algoritma genetika dan mampu menghasilkan solusi minimum yang jauh lebih optimal. Sedangkan, jika waktu komputasi pada masing-masing metode dibandingkan berdasarkan ukuran populasinya, maka semakin besar ukuran populasi semakin lama pula waktu komputasi yang diperlukan.

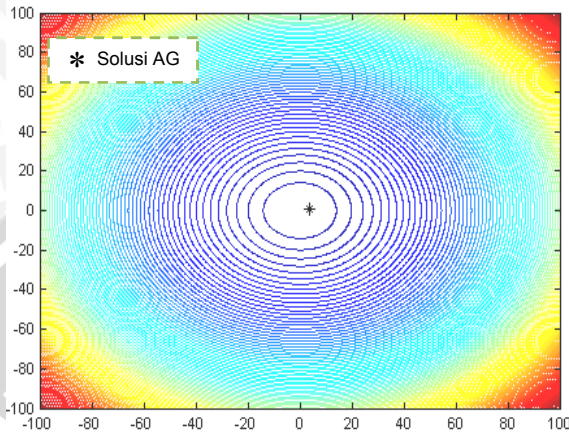
Pada Gambar 3.4 sampai Gambar 3.6 di bawah ini adalah contoh visualisasi perbandingan solusi numerik menggunakan metode hibrid algoritma genetika dan metode paralel algoritma untuk tes fungsi De Jong pada maksimum generasi = 50 dan ukuran populasi = 250.



Gambar 3.4 Fungsi De Jong



Gambar 3.5 *Contour* dan Titik-titik Solusi Tes Fungsi De Jong dengan Metode Hibrid Algoritma Genetika $f(x^*) = 2.4431 \times 10^{-13}$ dan $(x^*) = (-3.2548 \times 10^{-7}, -3.7198 \times 10^{-7})$



Gambar 3.6 *Contour* dan Titik-titik Solusi Tes Fungsi De Jong dengan Metode Paralel Algoritma Genetika
 $f(x^*) = 4.3106$ dan $(x^*) = (-1.3672, -1.5625)$

b. tes fungsi Ackley

Fungsi Ackley dengan solusi eksak minimum global $f(x^*) = 0$.

Tabel 3.2 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Ackley

Uk Pop	Hibrid AG					Paralel AG				
	P.S	P.M	$f(\bar{x})$ ($\times 10^{-9}$)	E ($\times 10^{-9}$)	t	P.S	P.M	$f(\bar{x})$	E	t
50	0.95	0.03	7.9814	7.9814	19.032	0.85	0.02	3.2526	3.2526	18.985
100	0.75	0.04	6.5488	6.5488	49.719	0.75	0.01	0.8616	0.8616	49.672
150	0.85	0.04	6.4478	6.4478	91.954	0.95	0.04	2.8906	2.8906	91.907
200	0.75	0.04	6.5300	6.5300	148.344	0.95	0.03	0.3756	0.3756	148.250
250	0.75	0.03	6.5116	6.5116	215.750	0.75	0.03	0.4395	0.4395	215.656

c. tes fungsi Drop Wave

Fungsi Drop Wave dengan solusi eksak minimum global $f(x^*) = -1.5$.

Tabel 3.3 Perbandingan Solusi Numerik Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Drop Wave

Uk Pop	Hibrid AG					Paralel AG				
	P.S	P.M	$f(\bar{x})$	E ($\times 10^{-15}$)	t	P.S	P.M	$f(\bar{x})$	E	t
50	0.95	0.02	-1.5	4.4409	19.030	0.95	0.01	-1.499	0.0003	18.984
100	0.75	0.01	-1.5	3.9968	49.843	0.85	0.01	-1.469	0.0306	49.765
150	0.85	0.01	-1.5	4.4409	93.109	0.85	0.02	-1.499	0.0003	93.031
200	0.95	0.03	-1.5	10.6580	157.641	0.75	0.04	-1.469	0.0309	157.500
250	0.75	0.03	-1.5	3.9968	219.453	0.95	0.03	-1.487	0.0124	219.375

d. tes fungsi Shekel

Fungsi Shekel dengan solusi eksak minimum global $f(x^*) = -10.5786$.

Tabel 3.4 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Shekel

Uk Pop	Hibrid AG					Paralel AG				
	P.S	P.M	$f(\bar{x})$	E ($\times 10^{-5}$)	t	P.S	P.M	$f(\bar{x})$	E	t
50	0.95	0.02	-10.5786	1.984	19.688	0.85	0.01	-8.356	2.2221	19.547
100	0.75	0.04	-10.5786	1.984	50.157	0.95	0.04	-10.217	0.3616	50.016
150	0.75	0.01	-10.5786	1.984	94.063	0.75	0.01	-9.422	1.1563	92.922
200	0.95	0.02	-10.5786	1.984	150.094	0.85	0.03	-10.455	0.1227	149.938
250	0.75	0.02	-10.5786	1.984	235.407	0.95	0.01	-10.594	0.3187	235.235

- e. tes fungsi Rastrigin
 Fungsi Rastrigin dengan solusi eksak minimum global $f(x^*) = 0$.

Tabel 3.5 Perbandingan Solusi Numerik dan Waktu Komputasi Metode Hibrid Algoritma Genetika dengan Metode Paralel Algoritma Genetika pada Tes Fungsi Rastrigin

Uk Pop	Hibrid AG					Paralel AG				
	P.S	P.M	$f(\bar{x})$ ($\times 10^{-9}$)	E ($\times 10^{-9}$)	t	P.S	P.M	$f(\bar{x})$	E	t
50	0.75	0.02	0	0	19.031	0.75	0.03	1.157	1.157	18.984
100	0.75	0.03	0	0	50.297	0.75	0.04	0.791	0.791	50.250
150	0.85	0.04	0	0	93.078	0.75	0.02	0.170	0.170	93.031
200	0.75	0.02	0	0	149.359	0.95	0.01	0.414	0.414	149.296
250	0.85	0.02	0	0	219.437	0.85	0.04	1.057	1.057	219.375

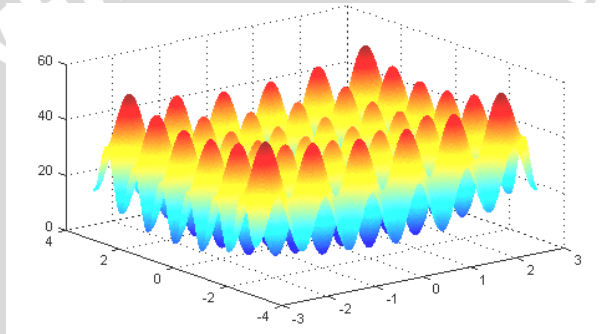
Pada beberapa tes fungsi multimodal (kasus b,c,d dan e), nilai kesalahan mutlak (E) metode hibrid algoritma genetika jauh lebih baik daripada metode paralel algoritma genetika. Hal ini dapat dilihat dari nilai kesalahan mutlak metode hibrid algoritma genetika yang sangat kecil atau mendekati nol daripada nilai kesalahan mutlak metode paralel algoritma genetika yang masih terlalu besar.

Sama halnya pada tes fungsi unimodal De Jong, nilai kesalahan mutlak beberapa tes fungsi multimodal pada masing-masing metode jika dibandingkan berdasarkan ukuran populasinya, tidak dapat dipastikan bagaimana pengaruh ukuran populasi terhadap solusi. Hal ini karena sebagian besar komponen algoritma genetika berbasis pada bilangan acak, sehingga solusi yang dihasilkan berbeda-beda.

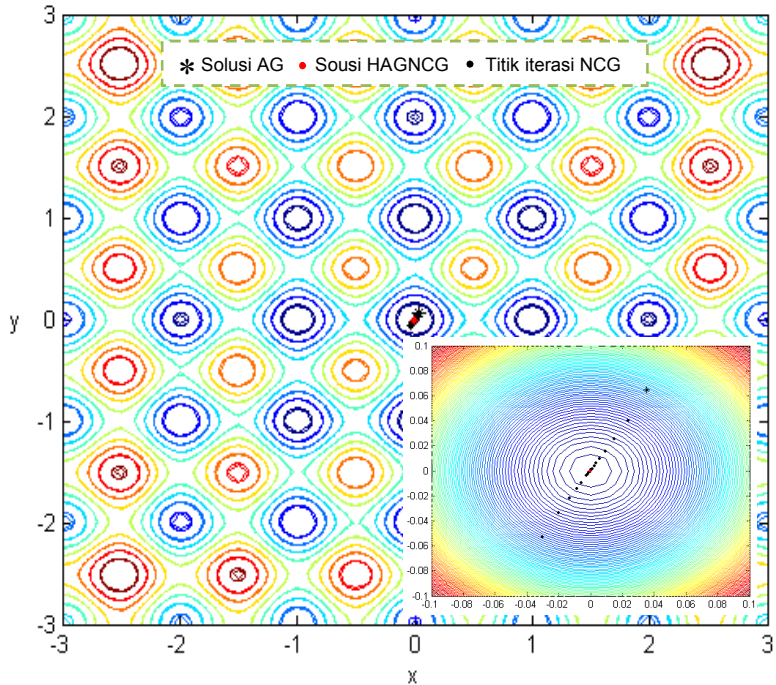
Pada beberapa kasus yang diuji dengan tes fungsi multimodal (kasus b,c,d dan e), waktu komputasi (t) metode hibrid algoritma genetika sebanding dengan metode paralel algoritma genetika. Hal ini sama halnya pada tes fungsi unimodal De Jong, waktu komputasi antara kedua metode mempunyai selisih yang sangat kecil. Dari data percobaan didapatkan bahwa hibrid algoritma genetika dapat memperoleh solusi yang optimal dengan waktu komputasi yang sebanding dengan waktu komputasi metode paralel algoritma genetika.

Sama halnya pada tes fungsi unimodal De Jong, semakin banyak ukuran populasi yang digunakan, semakin lama pula waktu yang dibutuhkan untuk memperoleh solusi pada masing-masing metode. Baik itu metode hibrid algoritma genetika maupun metode paralel algoritma genetika.

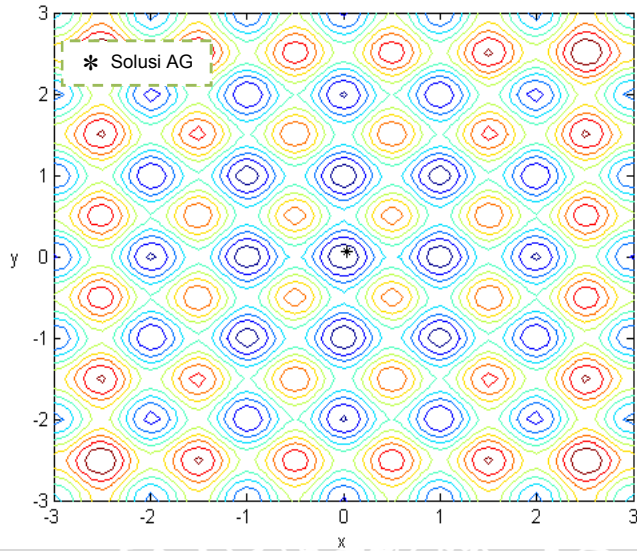
Pada Gambar 3.7 sampai Gambar 3.9 di bawah ini adalah contoh visualisasi perbandingan solusi numerik menggunakan metode hibrid algoritma genetika dan metode paralel algoritma untuk tes fungsi Rastrigin pada maksimum generasi = 50 dan ukuran populasi = 250.



Gambar 3.7 Fungsi Rastrigin



Gambar 3.8 *Contour* dan Titik-titik Solusi Tes Fungsi Rastrigin dengan Metode Hibrid Algoritma Genetika $f(x^*) = 0$ dan $(x^*) = (-8.6388 \times 10^{-10}, -1.4125 \times 10^{-9})$



Gambar 3.9 *Contour* dan Titik-titik Solusi Tes Fungsi Rastrigin dengan Metode Paralel Algoritma Genetika
 $f(x^*) = 1.0572$ dan $(x^*) = (0.035156, 0.064453)$

UNIVERSITAS BRAWIJAYA



BAB IV PENUTUP

4.1 Kesimpulan

Berdasarkan pembahasan dapat diambil beberapa kesimpulan antara lain:

1. d_{k+1} pada metode *new conjugate gradient* adalah vektor dengan arah *descent*, sehingga metode *new conjugate gradient* dapat menentukan nilai minimum global.

$$\text{Parameter } \theta_k = \left(\frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T S_k) - \|g_k\|^2 \|g_{k+1}\|^2} \right) \text{ mem-}$$

pengaruhi nilai β_k^N . Nilai parameter θ_k dapat keluar dari interval $[0,1]$. Jika nilai parameter $\theta_k \leq 0$, maka pada persamaan (2.7) ditetapkan $\theta_k = 0$ dan diperoleh $\beta_k^N = \beta_k^{PRP}$. Jika nilai parameter $\theta_k \geq 1$, maka pada persamaan (2.7) ditetapkan $\theta_k = 1$ dan diperoleh $\beta_k^N = \beta_k^{DY}$.

Arah d_{k+1} pada persamaan (2.8) merupakan kombinasi dari metode PRP dan metode DY berdasarkan aturan untuk θ_k yang terpilih;

2. solusi numerik yang dihasilkan oleh hibrid algoritma genetika jauh lebih baik dibandingkan dengan solusi numerik yang dihasilkan oleh paralel algoritma genetika. Hal ini dapat dilihat dari nilai kesalahan mutlak;
3. waktu komputasi yang dihasilkan oleh hibrid algoritma genetika sebanding dengan waktu komputasi paralel algoritma genetika. Dengan waktu komputasi yang sebanding, metode hibrid algoritma genetika menghasilkan solusi yang jauh lebih optimal. Besar ukuran populasi juga berpengaruh pada waktu komputasi, semakin besar ukuran populasi, semakin lama waktu komputasi yang dibutuhkan.

4.2 Saran

Untuk memperoleh solusi optimal menggunakan hibrid algoritma genetika atau parallel algoritma genetika, disarankan melakukan banyak percobaan. Hal ini dikarenakan metode algoritma genetika banyak menggunakan bilangan acak, sehingga setiap menjalankan program akan memperoleh solusi yang sedikit berbeda.

Penggunaan parameter maksimum generasi dan ukuran populasi disarankan sesuai dengan kondisi setiap fungsi dan batas interval fungsi. Penggunaan parameter yang sesuai ditujukan agar bilangan acak yang dibangkitkan dapat mencapai solusi paling optimal.



DAFTAR PUSTAKA

- Anton, H. 2000. *Dasar-Dasar Aljabar Linear*. Interaksa. Batam.
- Boyd, S. dan V. Lieven. 2004. *Convex Optimization*. Cambridge University Press. United Kingdom.
- Bronshtein, I.N., K.A. Semendyayev, G. Musiol dan H. Muehlig. 2007. *Handbook of Mathematics*. Springer. Berlin.
- Chen, W.W.L. 1997. *Multivariable and Vector Analysis*. Macquarie University. Sidney.
- Dai, Y.H. dan Y. Yuan. 2001. *Nonlinear Conjugate Gradient Methods*. Shanghai Science and Technology. Cina.
- Gen, M. dan R. Cheng. 1997. *Genetic Algorithms and Engineering Design*. John Wiley and Sons. New York.
- Grossman, S.I. 1995. *Multivariable Calculus, Linear Algebra, and Differential Equations*. Saunders College Publishing. London.
- Gudla, P.K. dan R. Ganguli. 2005. *An Automated Hybrid Genetic – Conjugate Gradient Algorithm for Multimodal Optimization Function*. Indian Institute of Science. India.
- Leader, J.J. 2004. *Numerical Analysis and Scientific Computation International Edition*. Pearson Addison-Weasley. New York.
- Leonardo, I.V. 2003. *Pengunaan Algoritma Genetik untuk Rute Perencanaan Rute Perjalanan di Jawa Timur*. Universitas Kristen Petra. Surabaya.
http://dewey.petra.ac.id/spektra/module/catalog/docs/library/search_simple.php?keyword=sampah&source=title&mode=display&result_per_page=20&npage=1. Tanggal akses: 13 Januari 2009.
- Luknanto, D. 2000. *Pengantar Optimasi Nonlinier*. Jurusan Teknik Sipil Fakultas Teknik Universitas Gajah Mada. Yogyakarta.

- Mathews, J.H. dan K.D. Fink. 2004. *International Edition Numerical Methods Using Matlab Fourth Edition*. Pearson Education International. USA.
- McKinney, H.W. 2008. *Linear Equation Problem*. <http://ce.utexas.edu/HW8/>. Tanggal akses 30 Desember 2008.
- Neculai, A. 2008. *New Hybrid Conjugate Gradient Algorithms as a Convex Combination of PRP and DY for Unconstrained Optimization*. Research Institute for Informatics. Romania.
- Nocedal, J. dan S.J. Wright. 1999. *Numerical Optimization*. Springer-Verlag. New York.
- Obitko, M. 1998. *Introduction to Genetic Algorithm*, <http://cs.felk.cvut.cz/~xobitko/ga/>. Tanggal akses: 30 Desember 2008.
- Polak, E. dan G. Ribiere. 1969. *Note sur la Convergence de Directions Conjugues*. Recherche Operationelle. Perancis.
- Polyak, B.T. 1969. *The Conjugate Gradient Method in Extreme Problem*. USSR Computational Mathematics and Mathematical Physic.
- Setiawan, K. 2003. *Paradigma Sistem Cerdas: Jaringan Saraf Tiruan, Logika Fazi, dan Algoritma Genetika*. Bayumedia Publishing. Malang.
- Suyanto. 2005. *Algoritma Genetika dalam Matlab*. Andi. Yogyakarta.
- Tomizuka, H. dan H. Yabe. 2004. *Hybrid Conjugate Gradient Method for Unconstrained Optimization*. <http://ism.ac.jp/tomizuka04/>. Tanggal akses: 30 Desember 2008.

LAMPIRAN

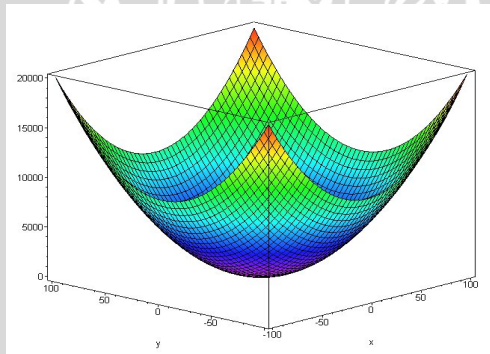
Lampiran A: Tes Fungsi yang Digunakan

1. fungsi De Jong

Fungsi De Jong merupakan fungsi kontinu, konveks dan unimodal. Dengan persamaan:

$$f(\bar{x}) = \sum_{i=1}^n x_i^2.$$

Batas daerah asal fungsi adalah $-100 \leq x_i \leq 100$. Fungsi De Jong menghasilkan nilai global minimum $f(x) = 0$ untuk $x_i = 0, i = 1, 2, \dots, n$.



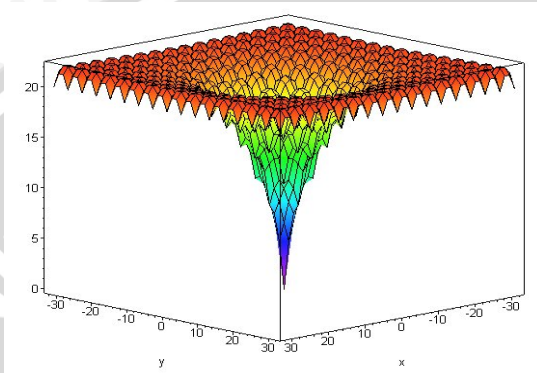
Gambar A.1 Fungsi De Jong dengan $n = 2$

2. fungsi Ackley

Fungsi Ackley terkenal sebagai tes fungsi multimodal. Dengan persamaan:

$$f(\bar{x}) = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1)$$

di mana nilai $a = 20, b = 0.2, c = 2\pi$. Batas daerah asal fungsi adalah $-20 \leq x_i \leq 20$. Fungsi Ackley menghasilkan nilai global minimum $f(x) = 0$ untuk $x_i = 0, i = 1, 2, \dots, n$.



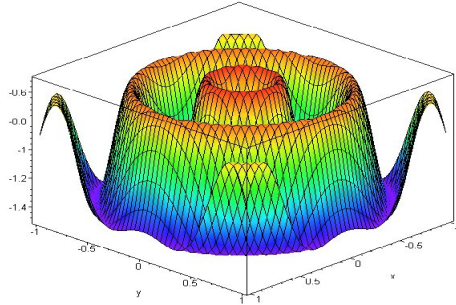
Gambar A.2 Fungsi Ackley dengan $n = 2$

3. fungsi Drop Wave

Fungsi Drop Wave merupakan tes fungsi multimodal. Fungsi ini hanya mempunyai dua variabel dengan persamaan:

$$f(x_1, x_2) = \frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{\frac{1}{2}(x_1^2 + x_2^2) + 2}$$

Batas daerah asal fungsi adalah $-1 \leq x_1 \leq 1$ dan $-1 \leq x_2 \leq 1$. Fungsi Drop Wave menghasilkan nilai global minimum $f(x_1, x_2) = 0$ untuk $x_1 = x_2 = 0$.



Gambar A.3 Fungsi Drop Wave dengan $n = 2$

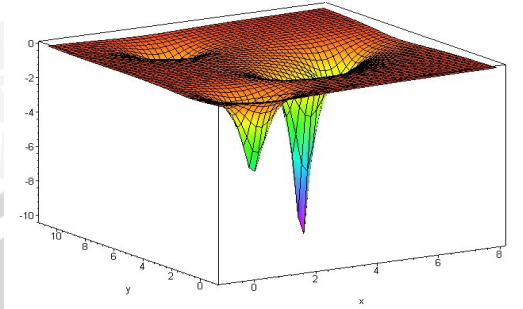
4. fungsi Shekel

Fungsi Shekel merupakan tes fungsi multimodal dengan persamaan:

$$f(\bar{x}) = -\sum_{i=1}^5 \left(\sum_{j=1}^n \left[(x_j - a_{ij})^2 + c_j \right] \right)^{-1}$$

di mana $a_{ij} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 \\ 6 & 6 & 6 & 6 & 6 \\ 2 & 9 & 2 & 9 & 4 \\ 5 & 5 & 3 & 3 & 9 \end{bmatrix}$ dan $c_i = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.4 \\ 0.6 \\ 0.7 \end{bmatrix}$.

Batas daerah asal fungsi adalah $1 \leq x_i \leq 8$. Fungsi Shekel menghasilkan nilai global minimum $f(\bar{x}) = -10.5786$ untuk $x_i = 4.001, i = 1, 2, \dots, n$.



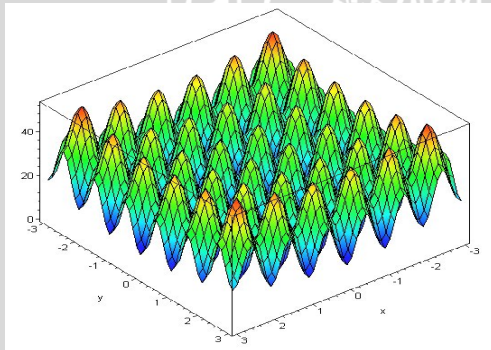
Gambar A.4 Fungsi Shekel dengan $n = 2$

5. fungsi Rastrigin

Fungsi Rastrigin adalah pengembangan dari fungsi De Jong dengan penambahan modulasi *cosinus* untuk menghasilkan fungsi minimum lokal yang berulang dengan persamaan:

$$f(\bar{x}) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$$

Batas daerah asal fungsi adalah $-3 \leq x_i \leq 3$. Fungsi Rastrigin menghasilkan nilai global minimum $f(x) = 0$ untuk $x_i = 0, i = 1, 2, \dots, n$.



Gambar A.5 Fungsi Rastrigin dengan $n = 2$

Lampiran B: Pseudocode Hibrid Algoritma Genetika dan New Conjugate Gradient

- Input** : himpunan peluang persilangan (Psilang) $\{Pc_1, Pc_2, \dots, Pc_m\}$, himpunan peluang mutasi (Pmutasi) $\{Pm_1, Pm_2, \dots, Pm_n\}$, batas atas domain (Ra), batas bawah domain (Rb), maksimum generasi (MaxG), ukuran populasi (UkPop) dan maksimum iterasi (MaxIter).
- Output** : \bar{x} akhir, $f(\bar{x})$ akhir, kesalahan mutlak (E), waktu komputasi (t), iterasi akhir.

% algoritma genetika

1. tic
2. for i \leftarrow 1 to length(Psilang)
 - a. for j \leftarrow 1 to length(Pmutasi)
 - 1) r \leftarrow r+1
 - 2) Populasi \leftarrow InisialisasiPopulasi(UkPop, JumGen)
 - 3) for generasi \leftarrow 1 to MaxG
 - a) x \leftarrow DekodekanKromosom(Populasi(1,:), Nvar, Nbit, Ra, Rb)
 - b) FF(1) \leftarrow fungsi(fungsi, x(1), x(2))
 - c) Min \leftarrow FF(1)
 - d) for ii \leftarrow 2 to UkPop
 - (1) Kromosom \leftarrow Populasi(ii,:)
 - (2) x \leftarrow DekodekanKromosom(Kromosom, Nvar, Nbit, Ra, Rb)
 - (3) FF(ii) \leftarrow fungsi(fungsi, x(1), x(2))
 - (4) if (FF(ii) < Min)
 - (a) Min \leftarrow FF(ii)
 - e) for ii \leftarrow 1 to UkPop
 - (1) FF(ii) \leftarrow FF(ii) - Min
 - f) x \leftarrow DekodekanKromosom(Populasi(1,:), Nvar, Nbit, Ra, Rb)
 - g) Fitness(1) \leftarrow EvaluasiIndividu(FF(1), BilKecil)
 - h) MaxF \leftarrow Fitness(1)

- i) $\text{MinF} \leftarrow \text{Fitness}(1)$
j) $\text{IndeksIndividuTerbaik} \leftarrow 1$
k) **for** ii $\leftarrow 2$ to UkPop
 (1) $\text{Kromosom} \leftarrow \text{Populasi}(\text{ii},:)$
 (2) $x \leftarrow \text{DekodekanKromosom}(\text{Kromosom}, \text{Nvar}, \text{Nbit}, \text{Ra}, \text{Rb})$
 (3) $\text{Fitness}(\text{ii}) \leftarrow \text{EvaluasiIndividu}(\text{FF}(\text{ii}), \text{BilKecil})$
 (4) **if** $(\text{Fitness}(\text{ii}) > \text{MaxF})$
 (a) $\text{IndeksIndividuTerbaik} \leftarrow \text{ii}$
 (b) $\text{MaxF} \leftarrow \text{Fitness}(\text{ii})$
 (c) $\text{BestX} \leftarrow x$
 (5) **if** $(\text{Fitness}(\text{ii}) < \text{MinF})$
 (a) $\text{MinF} \leftarrow \text{Fitness}(\text{ii})$
l) $\text{TemPopulasi} \leftarrow \text{Populasi}$
m) **if** $\text{mod}(\text{UkPop}, 2) = 0$
 (1) $\text{IterasiMulai} \leftarrow 3$
 (2) $\text{TemPopulasi}(1,:) \leftarrow \text{Populasi}(\text{IndeksIndividuTerbaik},:)$
 (3) $\text{TemPopulasi}(2,:) \leftarrow \text{Populasi}(\text{IndeksIndividuTerbaik},:)$
n) **else**
 (1) $\text{IterasiMulai} \leftarrow 2$
 (2) $\text{TemPopulasi}(1,:) \leftarrow \text{Populasi}(\text{IndeksIndividuTerbaik},:)$
o) $\text{LinearFitness} \leftarrow \text{LinearFitnessRangking}(\text{UkPop}, \text{Fitness}, \text{MaxF}, \text{MinF})$
p) **for** jj $\leftarrow \text{IterasiMulai}$:2 to UkPop
 (1) $\text{IP1} \leftarrow \text{RouletteWheel}(\text{UkPop}, \text{LinearFitness})$
 (2) $\text{IP2} \leftarrow \text{RouletteWheel}(\text{UkPop}, \text{LinearFitness})$
 (3) **if** $(\text{rand} < \text{Psilang}(i))$
 (a) $\text{Anak} \leftarrow \text{PindahSilang}(\text{Populasi}(\text{IP1},:), \text{Populasi}(\text{IP2},:), \text{JumGen})$
 (b) $\text{TemPopulasi}(\text{jj},:) \leftarrow \text{Anak}(1,:)$
 (c) $\text{TemPopulasi}(\text{jj}+1,:) \leftarrow \text{Anak}(2,:)$
 (4) **else**
 (a) $\text{TemPopulasi}(\text{jj},:) \leftarrow \text{Populasi}(\text{IP1},:)$
 (b) $\text{TemPopulasi}(\text{jj}+1,:) \leftarrow \text{Populasi}(\text{IP2},:)$
q) **for** kk $\leftarrow \text{IterasiMulai}$ to UkPop
 (a) $\text{TemPopulasi}(\text{kk},:) \leftarrow \text{Mutasi}(\text{TemPopulasi}(\text{kk},:), \text{JumGen}, \text{Pmutasi}(j))$

- r) populasi \leftarrow TemPopulasi
- 4) $Aiy1(r) \leftarrow BestX(1);$
 - 5) $Aiy2(r) \leftarrow BestX(2);$
 - 6) $FAiy(r) \leftarrow fungsiii(Aiy1(r),Aiy2(r))$
3. $MinFAiy \leftarrow FAiy(1)$
 4. $nn \leftarrow 1$
 5. **for** $i \leftarrow 2$ to r
 - a. **if** $(FAiy(i) < MinFAiy)$
 - 1) $MinFAiy \leftarrow FAiy(i)$
 - 2) $nn \leftarrow i$

% new conjugate gradient

6. $wo \leftarrow [Aiy1(nn) Aiy2(nn)]$
7. $go \leftarrow grad(fungsi,wo(1),wo(2))$
8. **if** $norma(go(1),go(2)) < epsilon$
 - a. **break**
9. $do \leftarrow -go$
10. $Alpha \leftarrow 1/(norma(go(1),go(2)) + BilKecil)$
11. $l \leftarrow 0$
12. **while** $((norma(go(1),go(2)) > epsilon) \& l < MaxIter)$
 - a. $HS \leftarrow go*(-go')$
 - b. $HSL \leftarrow wo + (Alpha*do)$
 - c. **while** $(fungsiii(fungsi,HSL(1),HSL(2)) > fungsiii(fungsi,wo(1),wo(2)) + (epsilon*Alpha*HS))$
 - 1) $Alpha \leftarrow Alpha * Pengalpa$
 - 2) $HSL \leftarrow wo + (Alpha*do)$
 - d. $w1 \leftarrow wo + (Alpha*do)$
 - e. $FAiy1 \leftarrow fungsiii(fungsi,w1(1),w1(2))$
 - f. $g1 \leftarrow grad(fungsi,w1(1),w1(2))$
 - g. $so \leftarrow w1 - wo$
 - h. $yo \leftarrow g1 - go$
 - i. **if** $(yo*g1')*(yo*so') - (norma(g1(1),g1(2))^2 *(norma(go(1),go(2))^2)) = 0$
 - 1) $tetao \leftarrow 0$
 - j. **else**
 - 1) $tetao \leftarrow ((yo*g1')*(yo*so') - (yo*g1')*(go*go')) / ((yo*g1')*(yo*so') - (norma(g1(1),g1(2))^2*(norma(go(1),go(2))^2))$

k. **if** tetao > 0 & tetao < 1

1) Betha ← $((1 - \text{tetao}) * (g1 * yo') / (go * go')) + (\text{tetao} * (g1 * g1') / (yo * so'))$

l. **else if** tetao >= 1

1) Betha ← $(g1 * g1') / (yo * so')$

m. **else**

1) Betha ← $(g1 * yo') / (go * go')$

n. **d** ← $-g1 + (\text{Betha} * so)$

o. **if** $\text{abs}(g1 * go') \geq 0.2 * \text{norma}(g1(1), g1(2))^2$

1) **d1** ← $-g1$

p. **else**

1) **d1** ← **d**

q. **Alpha** ← $\text{Alpha} * \text{norma}(do(1), do(2)) / \text{norma}(d1(1), d1(2))$

r. **wo** ← **w1**

s. **go** ← **g1**

t. **do** ← **d1**

u. **l** ← **l+1**

13. **t** ← **toc**

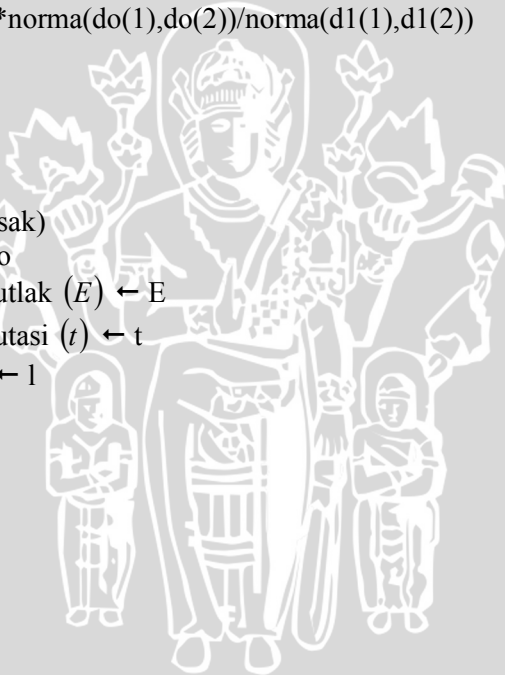
13. **E** ← $\text{abs}(\text{FAiy1} - \text{Eksak})$

14. Cetak \bar{x} akhir ← **wo**

15. Cetak kesalahan mutlak (**E**) ← **E**

16. Cetak waktu komputasi (**t**) ← **t**

17. Cetak iterasi akhir ← **l**



Lampiran C: Pseudocode Function yang Digunakan

1. Function Inisialisasi Populasi

function Populasi \leftarrow InisialisasiPopulasi(UkPop, JumGen)

1. **for** k \leftarrow 1 to UkPop

a. **for** g \leftarrow 1 to JumGen

1) a \leftarrow rand

2) **if** (a < 0.5)

a) Populasi(k,g) \leftarrow 0

3) **else**

a) Populasi(k,g) \leftarrow 1

2. Function Dekode Kromosom

function x \leftarrow DekodekanKromosom(Kromosom, jumlahvar, jumlahbit, batasbawah, batasatas)

1. **for** i \leftarrow 1 to jumlahvar

a. x(i) \leftarrow 0

1) **for** j \leftarrow 1 to jumlahbit

a) x(i) \leftarrow x(i) + Kromosom((i-1)*jumlahbit + j)*2^(-j)

2) x(i) \leftarrow batasbawah + (batasatas - batasbawah)*x(i)

3. Function Evaluasi Individu

function fitness \leftarrow EvaluasiIndividu(x, K)

1. fitness \leftarrow 1/(x+K)

4. Function Linear Fitness Ranking

function LFR \leftarrow LinearFitnessRangking(UkPop, Fitness, MaxF, MinF)

1. [SF, IndF] \leftarrow sort(Fitness)

2. **for** rr \leftarrow 1 to UkPop

a. LFR(IndF(UkPop-rr+1)) \leftarrow MaxF - (MaxF - MinF)*((rr-1)/(UkPop-1))

5. Function Roulette Wheel

function Pindex \leftarrow RouletteWheel(UkPop, LinearFitness)

1. JumFitness \leftarrow sum(LinearFitness)

2. Kumulatif $\leftarrow 0$

3. RN \leftarrow rand

4. ii $\leftarrow 1$

5. While ii \leq UkPop

a. Fitnessperjum(ii) \leftarrow LinearFitness(ii)/JumFitness

b. Kumulatif \leftarrow Kumulatif+ Fitnessperjum(ii)

c. if (Kumulatif) $>$ RN

1) Pindex \leftarrow ii

2) break

d. ii \leftarrow ii + 1

6. *Function* Pindah Silang

function Anak \leftarrow PindahSilang(Papa,Mama,JumlahGen)

1. TP $\leftarrow 1 + \text{fix}(\text{rand}*(\text{JumlahGen}-1))$

2. Anak(1,:) \leftarrow [Papa(1:TP) Mama(TP+1:JumlahGen)]

3. Anak(2,:) \leftarrow [Mama(1:TP) Papa(TP+1:JumlahGen)]

7. *Function* Mutasi

function MutKrom \leftarrow Mutasi(Kromosom,JumlahGen,Probmutasi)

1. MutKrom \leftarrow Kromosom

2. **for** i $\leftarrow 1$ to JumlahGen

a. **if** (rand < Probmutasi)

1) **if** Kromosom(i) = 0

a) MutKrom(i) $\leftarrow 1$

2) **else**

a) MutKrom(i) $\leftarrow 0$

8. *Function* Fungsii

function Faiy \leftarrow fungsii(fs,Aiy1,Aiy2)

1. FAiy \leftarrow Aiy1*exp(-Aiy1²-Aiy2²)

9. *Function* Gradien

function go=grad(fs,x,y)

1. h $\leftarrow 0.01$

2. go \leftarrow [(fungsii(x+h,y)-fungsii(x-h,y))/(2*h) (fungsii(x,y+h)-fungsii(x,y-h))/(2*h)]

10. *Function* Norma

function nor=norma(x,y)

1. nor \leftarrow sqrt(x²+y²)

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



Lampiran D: Proses Perhitungan Hibrid Algoritma Genetika Inialisasi Kromosom

Proses awal mencari nilai optimum pada algoritma genetika adalah memodelkan atau merepresentasikan ke dalam bentuk kromosom. Kromosom diperoleh dengan mengacak nilai setiap gen menggunakan *binary encoding* untuk membangkitkan domain populasi. Sebagai contoh dapat dilihat pada Tabel 4.1 dengan jumlah gen = 6, jumlah individu = 10. Pembangkitan data pada batas bawah interval = -2 dan batas atas interval = 2.

Tabel D.1 Kromosom dalam Satu Populasi

Kromosom ke-	x_1			x_2		
1	0	1	0	1	1	1
2	1	1	0	1	0	0
3	0	1	1	0	1	0
4	0	1	1	0	1	1
5	1	0	1	1	1	0
6	0	1	1	1	0	1
7	1	0	0	0	0	1
8	0	1	0	0	1	0
9	1	1	1	1	1	1
10	0	0	1	0	0	0

Dari Tabel D.1 di atas, akan dihitung nilai (x_1, x_2) menggunakan *binary encoding*. Sebagai contoh untuk kromosom pertama:

$$x_j = r_b + (r_a - r_b)(g_1 \times 2^{-1} + \dots + g_i \times 2^{-i} + K + g_N \times 2^{-N})$$

$$x_{11} = -2 + (2 - (-2))(0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}) = -1$$

$$x_{12} = -2 + (2 - (-2))(1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}) = 1.5$$

untuk kromosom pertama diperoleh titik $\bar{x}_1 = (x_{11}, x_{12}) = (-1, 1.5)$. Selengkapnya diperoleh hasil pada Tabel D.2.

Tabel D.2 Nilai $\bar{x} = (x_1, x_2)$ dalam Satu Populasi

Kromosom ke-	x_1	x_2
1	-1.0	1.5
2	1.0	0
3	-0.5	-1.0
4	-0.5	-0.5
5	0.5	1.0
6	-0.5	0.5
7	0	-1.5
8	-1.0	-1.0
9	1.5	1.5
10	-1.5	-2.0

Evaluasi Individu

Fungsi evaluasi individu bertujuan untuk menghitung nilai *fitness* dari suatu individu \bar{x} . Fungsi ini sangat bergantung pada masalah yang akan diselesaikan. Solusi yang dicari adalah meminimumkan sebuah fungsi $f = x_1 e^{(-x_1^2 - x_2^2)}$. Fungsi *fitness* yang digunakan adalah $fitness = \frac{1}{(f + a)}$ dengan a bilangan kecil positif

untuk menghindari pembagian dengan nol misalkan 0.001 dan nilai f harus positif. Pada kromosom pertama, $\bar{x}_1 = (x_{11}, x_{12}) = (-1, 1.5)$ diperoleh nilai f :

$$f_1 = -1 \times e^{(-1)^2 - (1.5)^2} = -0.0388.$$

Nilai f selengkapnya dapat dilihat pada Tabel D.3.

Tabel D.3 Nilai f Setiap Individu

Kromosom ke-	f	Kromosom ke-	f
1	-0.0388	6	-0.3033
2	0.3679	7	0
3	-0.1433	8	-0.1353
4	-0.3033	9	0.0167
5	-0.1433	10	-0.0029

Dalam hal ini nilai *fitness* tidak boleh negatif karena akan digunakan dalam perhitungan probabilitas untuk menentukan individu yang akan diseleksi. Oleh karena itu dicari nilai *f* terendah (*min f*) yaitu pada kromosom ke-4 dengan nilai -0.3033. Kemudian semua nilai *f* dalam satu populasi tersebut dikurangi dengan nilai *f* terendah, sehingga didapatkan nilai *f* yang baru (*f_b*), yaitu nilai *f* yang positif dalam satu populasi. Pada kromosom pertama didapatkan nilai *f_b* sebagai berikut:

$$f_{b1} = f_1 - \{\min f\}$$

$$f_{b1} = -0.0388 - (-0.3033) = 0.2645.$$

Nilai *f_b* selengkapnya dapat dilihat pada Tabel D.4.

Tabel D.4 Nilai *f_b* Setiap Individu

Kromosom ke-	<i>f_b</i>	Kromosom ke-	<i>f_b</i>
1	0.2645	6	0
2	0.6712	7	0.3033
3	0.1600	8	0.1680
4	0	9	0.3200
5	0.1600	10	0.3004

Langkah selanjutnya akan dicari nilai *fitness* dari masing-masing individu pada Tabel D.4 menggunakan fungsi $fitness = \frac{1}{(f + a)}$. Pada kromosom pertama didapatkan nilai *fitness* sebagai berikut:

$$fitness_1 = \frac{1}{f_{b1} + a}$$

$$fitness_1 = \frac{1}{0.2645 + 0.001} = 3.7665.$$

Nilai *fitness* selengkapnya dapat dilihat pada Tabel D.5.

Tabel D.5 Nilai *Fitness* Setiap Individu

Kromosom ke-	<i>fitness</i>	Kromosom ke-	<i>fitness</i>
1	3.7665	6	1000
2	1.4877	7	3.2862
3	6.2112	8	5.9172
4	1000	9	3.1153
5	6.2112	10	3.3179

Agar individu bernilai *fitness* tertinggi tidak hilang selama evolusi, perlu dilakukan *elitisme*, yaitu dengan cara membuat satu atau dua salinan individu bernilai *fitness* tertinggi. Nilai *fitness* tertinggi dan terendah berturut-turut disimpan dalam variabel *MaxF* dan *MinF*, sedangkan indeks dari individu bernilai *fitness* tertinggi disimpan dalam variabel *IndeksIndividuTerbaik*.

Pada Tabel D.5 diperoleh $MaxF = 1000$ dan $MinF = 1.4877$ sedangkan $IndeksIndividuTerbaik = 4$. Individu terbaik dari populasi disalin satu kali jika ukuran populasinya ganjil dan disalin dua kali jika ukuran populasinya genap.

Linear Fitness Ranking

Linear fitness ranking digunakan untuk menghindari kecenderungan konvergen pada optimum lokal. Langkah pertama adalah menskalakan nilai *fitness* seperti pada persamaan 2.10. Sebelum dikenakan fungsi *linear fitness ranking*, nilai *fitness* pada satu populasi diurutkan dari nilai *fitness* terkecil hingga terbesar seperti pada Tabel D.6.

Tabel D.6 Nilai *Fitness* yang Telah Diurutkan

Urutan ke-	Kromosom ke-	<i>Fitness</i>	Urutan ke-	Kromosom ke-	<i>Fitness</i>
1	2	1.4877	6	8	5.9172
2	9	3.1153	7	3	6.2112
3	7	3.2862	8	5	6.2112
4	10	3.3179	9	4	1000
5	1	3.7665	10	6	1000

Sebagai contoh, *linear fitness ranking* pada nilai *fitness* kromosom pertama adalah:

$$LFR(\text{kromosom_ke}(\text{urut_ke}(j))) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(j) - 1}{n - 1} \right)$$

$$LFR(\text{kromosom_ke}(\text{urut_ke}(10))) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(10) - 1}{n - 1} \right)$$

$$LFR(\text{kromosom_ke}(6)) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(10) - 1}{n - 1} \right)$$

$$LFR(6) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{1 - 1}{n - 1} \right)$$

$$f(6) = 1000 - (1000 - 1.4877) \left(\frac{1 - 1}{10 - 1} \right) = 1000,$$

untuk perhitungan selengkapnya, dapat dilihat pada Tabel D.7 dan Tabel D.8.

Tabel D.7 Nilai *Linear Fitness Ranking*

Kromosom ke- / <i>LFR ke-</i>	<i>Linear Fitness</i> <i>Ranking</i>	Kromosom ke- / <i>LFR ke-</i>	<i>Linear Fitness</i> <i>Ranking</i>
2	1.4877	8	556.2168
9	112.4335	3	667.1626
7	223.3793	5	778.1084
10	334.3251	4	889.0542
1	445.2709	6	1000.000

Tabel D.8 Nilai *Linear Fitness Ranking* yang telah diurutkan

Kromosom ke- / <i>LFR ke-</i>	<i>Linear Fitness</i> <i>Ranking</i>	Kromosom ke- / <i>LFR ke-</i>	<i>Linear Fitness</i> <i>Ranking</i>
1	445.2709	6	1000.0000
2	1.4877	7	223.3793
3	667.1626	8	556.2168
4	889.0542	9	112.4335
5	778.1084	10	334.3251

Seleksi Orang Tua

Seleksi orang tua adalah pemilihan dua buah kromosom sebagai orang tua yang akan dipindahsilangkan menggunakan metode *Roulette Wheel Selection*. Berdasarkan Tabel D.8 di atas, langkah pertama adalah menghitung total nilai *LFR* (persamaan 2.12) dari semua kromosom. Langkah selanjutnya adalah menghitung nilai kumulatif *LFR* masing-masing kromosom dibagi total nilai *LFR* dari semua kromosom (persamaan 2.13). Sebuah kromosom akan terpilih jika bilangan acak yang dibangkitkan kurang dari nilai kumulatif *LFR* dibagi dengan total nilai *LFR* (persamaan 2.14). Sebagai contoh perhitungan nilai kumulatif *LFR* dapat dilihat pada Tabel D.9.

Tabel D.9 Nilai Probabilitas q_k

Kromosom ke-	<i>Linear Fitness Ranking</i>	$p_k = \frac{LFR(k)}{Tot.LFR}$	$q_k = \sum_{j=1}^k p_k$	Interval q_k
1	445.2709	0.0889	0.0889	[0 , 0.0889]
2	1.4877	0.0003	0.0892	(0.0889 , 0.0892]
3	667.1626	0.1332	0.2225	(0.0892 , 0.2225]
4	889.0542	0.1775	0.4000	(0.2225 , 0.4000]
5	778.1084	0.1554	0.5554	(0.4000 , 0.5554]
6	1000.0000	0.1997	0.7551	(0.5554 , 0.7551]
7	223.3793	0.0446	0.7997	(0.7551 , 0.7997]
8	556.2168	0.1111	0.9108	(0.7997 , 0.9108]
9	112.4335	0.0225	0.9332	(0.9108 , 0.9332]
10	334.3251	0.0668	1.0000	(0.9332 , 1.0000]
Tot.LFR	5007.4385			

Misalkan bilangan acak yang dibangkitkan adalah 0.6 maka kromosom ke-6 terpilih sebagai orang tua.

Pindah Silang

Setelah diperoleh dua buah kromosom sebagai orang tua yang akan dipindahsilangkan, selanjutnya dibangkitkan satu bilangan acak untuk menentukan letak gen yang dipotong. Letak gen yang akan dipotong disimpan dalam variabel *TP* (titik potong) dengan perhitungan sebagai berikut:

$$TP = 1 + \text{fix}(\text{randomize} * (\text{JumGen} - 1)).$$

Misalkan terpilih dua kromosom orang tua dengan bilangan acak 0.4, $TP = 1 + \text{fix}(0.4 * (6 - 1)) = 3$. Proses persilangannya dapat dilihat pada Gambar D.1.

Titik potong ($TP = 3$)

Orang Tua 1	0	1	1	1	1	1
Orang Tua 2	0	0	1	0	0	0
Anak 1	0	1	1	0	0	0
Anak 2	0	0	1	1	1	1

Gambar D.1 Proses Pindah Silang

4.3.2.6 Mutasi

Mutasi dilakukan terhadap semua kromosom. Prosesnya adalah jika suatu bilangan random $[0,1)$ yang dibangkitkan kurang dari peluang mutasi, maka gen yang bersesuaian akan diganti dengan nilai kebalikannya (nilai 0 diubah menjadi 1 dan nilai 1 diubah menjadi 0).

Misalkan peluang mutasi adalah 0.05. Pada setiap gen, dibangkitkan bilangan acak seperti Gambar D.2 di bawah ini.

Kromosom 1	0	1	0	1	1	1
Bilangan Acak	0.5	0.3	0.7	0.04	0.9	0.8
Hasil Mutasi	0	1	0	0	1	1

Gambar D.2 Proses Mutasi

Oleh karena bilangan acak untuk gen keempat kurang dari 0.05, maka gen keempat berubah dari 1 menjadi 0.

Pemilihan Solusi Algoritma Genetika

Semua proses dari algoritma genetika ini berhenti sampai iterasi mencapai maksimum generasi. Penghitungan algoritma genetika ini menggunakan himpunan peluang persilangan dan peluang mutasi yang dikombinasikan.

Sebagai contoh, menggunakan himpunan parameter peluang persilangan = $\{0.7, 0.8, 0.9\}$ dan himpunan parameter peluang mutasi = $\{0.03, 0.05\}$ sehingga diperoleh pasangan parameter persilangan dan parameter mutasi sejumlah $nPsilang \times nPmutasi = 3 \times 2 = 6$ yaitu

$$\{(0.7, 0.03), (0.8, 0.03), (0.9, 0.03), (0.7, 0.05), (0.8, 0.05), (0.9, 0.05)\}.$$

Himpunan solusi yang diperoleh dari algoritma genetika dengan masing-masing kombinasi peluang di atas dipilih yang paling optimal dan akan digunakan sebagai masukan awal metode *new conjugate gradient* untuk mendapatkan solusi terbaik. Berikut ini merupakan solusi algoritma genetika yang dihasilkan dari kombinasi peluang di atas.

Tabel D.10 Solusi Algoritma Genetika

No	Peluang Persilangan	Peluang Mutasi	Solusi Optimal		
			x_1	x_2	$f(x_1, x_2)$
1	0.7	0.03	-0.22	-0.97	-0.0829
2	0.8	0.03	-0.61	1.22	-0.0953
3	0.9	0.03	-0.59	-1.19	-0.1000
4	0.7	0.05	-0.97	1.18	-0.0942
5	0.8	0.05	-0.79	1.21	-0.0974
6	0.9	0.05	-0.63	1.23	-0.0939

Solusi pada Tabel D.10, dipilih titik (x_1, x_2) dengan nilai $f(x_1, x_2)$ yang paling optimal. Pada kasus ini diambil yang paling minimum, yaitu nilai $f(x_1, x_2) = -0.1000$ dengan $(x_1, x_2) = (-0.59, -1.19)$ yang akan digunakan sebagai masukan awal metode *new conjugate gradient*.

Metode *New Conjugate Gradient*

Untuk mendapatkan solusi terbaik, digunakan metode *new conjugate gradient* dengan masukan awal adalah titik terbaik yang diperoleh dari kombinasi parameter algoritma genetika.

Telah dijelaskan sebelumnya bahwa setelah mendapatkan titik optimal dari algoritma genetika, dilanjutkan dengan mencari solusi paling optimal menggunakan metode *new conjugate gradient*. Berikut ini adalah contoh perhitungan manual dari metode *new conjugate gradient*:

Langkah 1

- Menetapkan nilai awal $x_0 \in \mathcal{R}^n$ dari algoritma genetika.

Contohnya telah diperoleh $x_0 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -0.59 \\ -1.19 \end{pmatrix}$ dan juga

parameter $0 < \rho < \sigma < 1$. Di mana $\rho = 10^{-6}$ dan $\sigma = 10^{-3}$.

- Menghitung nilai $f(x_0) = -1.000$.

- Menghitung nilai $g_0 = \nabla f(x_0) = \begin{pmatrix} 0.0521 \\ -0.2406 \end{pmatrix}$

- Menetapkan $d_0 = -g_0 = \begin{pmatrix} -0.0521 \\ 0.2406 \end{pmatrix}$.

- Menetapkan dugaan awal nilai α_0 yaitu

$$\alpha_0 = \frac{1}{\|g_0\|} = \frac{1}{\sqrt{(0.0521)^2 + (-0.2406)^2}} = 6.5767.$$

Langkah 2

- Untuk $k=0$ sampai dengan maksimum iterasi, melakukan **Langkah 3** sampai **Langkah 6**.

- Jika $\|g_k\|_\infty \leq 10^{-6}$ atau $k =$ maksimum iterasi = 10, maka iterasi dihentikan. Dalam hal ini,

$\|g_0\| = \sqrt{(0.0521)^2 + (-0.2406)^2} = 0.1521 > 10^{-6}$ sehingga iterasi tetap berjalan.

Langkah 3

- Menghitung nilai $\alpha_k > 0$ yang memenuhi kondisi pencarian arah Wolfe yaitu:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k \text{ dan}$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k$$

dalam hal ini nilai $\alpha_0 = 6.5767$ memenuhi kondisi Wolfe.

- Meng-update nilai $x_{k+1} = x_k + \alpha_k d_k$.

$$x_1 = x_0 + \alpha_0 d_0$$

$$x_1 = \begin{pmatrix} -0.59 \\ -1.19 \end{pmatrix} + 6.5767 \begin{pmatrix} -0.0521 \\ 0.2406 \end{pmatrix}$$

$$x_1 = \begin{pmatrix} -0.9323 \\ 0.3922 \end{pmatrix}$$

- Menghitung nilai $g_{k+1} = \nabla f(x_{k+1})$.

$$g_1 = \nabla f(x_1) = \begin{pmatrix} -0.2654 \\ 0.2629 \end{pmatrix}$$

- Menentukan nilai $S_k = x_{k+1} - x_k$.

$$S_0 = x_1 - x_0$$

$$S_0 = \begin{pmatrix} -0.9323 \\ 0.3922 \end{pmatrix} - \begin{pmatrix} -0.59 \\ -1.19 \end{pmatrix}$$

$$S_0 = \begin{pmatrix} -0.3423 \\ 1.5822 \end{pmatrix}$$

- Menentukan nilai $y_k = g_{k+1} - g_k$.

$$y_0 = g_1 - g_0$$

$$y_0 = \begin{pmatrix} -0.2654 \\ 0.2629 \end{pmatrix} - \begin{pmatrix} 0.0521 \\ -0.2406 \end{pmatrix}$$

$$y_0 = \begin{pmatrix} -0.3175 \\ 0.5034 \end{pmatrix}$$

Langkah 4

- Menghitung parameter θ_k .

♦ Jika $(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2 = 0$ maka, $\theta_k = 0$.

♦ Jika $(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2 \neq 0$ maka,

$$\theta_k = \frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2}$$

$$(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2 = (y_0^T g_1)(y_0^T S_0) - \|g_1\|^2 \|g_0\|^2$$

$$= (-0.3175 \quad 0.5034) \begin{pmatrix} -0.2654 \\ 0.2629 \end{pmatrix} - (0.1396)(0.0606)$$

$$= 0.1320 \neq 0$$

$$\theta_k = \frac{(y_k^T g_{k+1})(y_k^T S_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T S_k) - \|g_{k+1}\|^2 \|g_k\|^2}$$

$$\theta_0 = \frac{(y_0^T g_1)(y_0^T S_0) - (y_0^T g_1)(g_0^T g_0)}{(y_0^T g_1)(y_0^T S_0) - \|g_1\|^2 \|g_0\|^2}$$

$$\theta_0 = 0.9340$$

Langkah 5

- Menghitung parameter *conjugate gradient* β_k^N .

♦ Jika $\theta_k \leq 0$ maka,

$$\beta_k^N = \beta_k^{PRP} = \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{g}_k^T \mathbf{y}_k}.$$

♦ Jika $0 < \theta_k < 1$ maka

$$\beta_k^N = (1 - \theta_k) \beta_k^{PRP} + \theta_k \beta_k^{DY}$$

$$\beta_k^N = (1 - \theta_k) \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{g}_k^T \mathbf{y}_k} + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_k^T}{\mathbf{y}_k^T \mathbf{S}_k}.$$

♦ Jika $\theta_k \geq 1$ maka,

$$\beta_k^N = \beta_k^{DY} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_k^T}{\mathbf{y}_k^T \mathbf{S}_k}.$$

Karena nilai $\theta_0 = 0.9340$, $0 < \theta_0 < 1$ maka

$$\beta_k^N = (1 - \theta_k) \beta_k^{PRP} + \theta_k \beta_k^{DY}$$

$$\beta_k^N = (1 - \theta_k) \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{g}_k^T \mathbf{y}_k} + \theta_k \frac{\mathbf{g}_{k+1}^T \mathbf{g}_k^T}{\mathbf{y}_k^T \mathbf{S}_k}$$

$$\beta_0^N = (1 - \theta_0) \frac{\mathbf{g}_1^T \mathbf{y}_0}{\mathbf{g}_0^T \mathbf{y}_0} + \theta_0 \frac{\mathbf{g}_1^T \mathbf{g}_0^T}{\mathbf{y}_0^T \mathbf{S}_0}$$

$$\beta_0^N = 0.3800$$

Langkah 6

- Menghitung arah d_{k+1} .

- Menentukan nilai $d = -g_{k+1} + \beta_k^N S_k$.

$$d = -g_1 + \beta_0^N S_0$$

$$d = -\begin{pmatrix} -0.2654 \\ 0.2629 \end{pmatrix} + 0.3800 \begin{pmatrix} -0.3423 \\ 1.5822 \end{pmatrix}$$

$$d = \begin{pmatrix} 0.1353 \\ 0.3384 \end{pmatrix}$$

- Menghitung arah d_{k+1} .

◆ Jika $|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2$ maka,

$$d_{k+1} = -g_{k+1}$$

◆ Jika $|g_{k+1}^T g_k| < 0.2 \|g_{k+1}\|^2$ maka,

$$d_{k+1} = d$$

$$|g_{k+1}^T g_k|$$

$$|g_1^T g_0| = \left| \begin{pmatrix} -0.2654 & 0.2629 \end{pmatrix} \begin{pmatrix} 0.0521 \\ -0.2406 \end{pmatrix} \right|$$

$$|g_1^T g_0| = |-0.771| = 0.771$$

$$0.2 \|g_{k+1}\|^2$$

$$0.2 \|g_1\|^2 = 0.2 \left(\sqrt{(-0.2654)^2 + (0.2629)^2} \right)^2$$

$$0.2 \|g_1\|^2 = 0.2 \times 0.1396 = 0.0279$$

$$\text{Jika } \|g_1^T g_0\| \geq 0.2 \|g_{k+1}\|^2$$

$$0.771 \geq 0.0279$$

$$\text{maka } d_{k+1} = -g_{k+1}$$

$$d_1 = -g_1 = \begin{pmatrix} 0.2654 \\ -0.2629 \end{pmatrix}.$$

$$\text{- Menghitung nilai } \alpha_{k+1} = \alpha_k \frac{\|d_k\|}{\|d_{k+1}\|}$$

$$\alpha_1 = \alpha_0 \frac{\|d_0\|}{\|d_1\|}$$

$$\alpha_1 = 6.5767 \times \frac{0.7745}{0.3736}$$

$$\alpha_1 = 13.6340$$

$$\text{- Menetapkan nilai } k = k + 1$$

$$k = 0 + 1$$

$$k = 1$$

- Kembali ke **Langkah 2**. Jika salah satu kondisi telah terpenuhi, iterasi telah sampai maksimum iterasi atau $\|g_k\|_\infty \leq 10^{-6}$ maka proses iterasi dihentikan.

Proses di atas dilakukan terus-menerus sehingga menghasilkan nilai yang paling optimum untuk fungsi $f = x_1 e^{(-x_1^2 - x_2^2)}$. Iterasi selanjutnya disajikan dalam Tabel D.11.

Tabel D.11 Solusi dari Metode *New Conjugate Gradient*

Iterasi ke-	α	β	x_1	x_2	$f(x_1, x_2)$
0	6.5767	0.3800	-0.9323	0.3922	-0.3352
1	1.2897	0.4399	-0.5900	0.0531	-0.4154
2	1.0243	0.7228	-0.8091	0.0079	-0.4204
3	1.0911	0.7502	-0.6338	0.0007	-0.4241
4	1.0659	0.7763	-0.7740	0.0001	-0.4252
5	1.0305	0.7129	-0.6619	0.0000	-0.4271
6	1.1222	0.8217	-0.7516	0.0000	-0.4272
7	0.9736	0.6649	-0.6798	0.0000	-0.4282
8	0.9626	0.6860	-0.7258	0.0000	-0.4286
9	1.1662	0.8566	-0.6890	0.0000	-0.4286
10	0.9339	0.6524	-0.7184	0.0000	-0.4288

Karena maksimum iterasi yang diberikan adalah 10, sehingga solusi minimal untuk fungsi $f = x_1 e^{(-x_1^2 - x_2^2)}$ adalah -0.4288 dengan titik $(x_1, x_2) = (-0.7146, 0)$.

Setelah menyelesaikan fungsi $f = x_1 e^{(-x_1^2 - x_2^2)}$ secara numerik menggunakan hibrid algoritma genetika, selanjutnya mencari nilai kesalahan mutlak seperti pada persamaan (2.8).

$$\begin{aligned}
 |E| &= |y - \tilde{y}| \\
 &= |0.4289 - 0.4288| \\
 &= |0.0001| \\
 &= 0.0001
 \end{aligned}$$

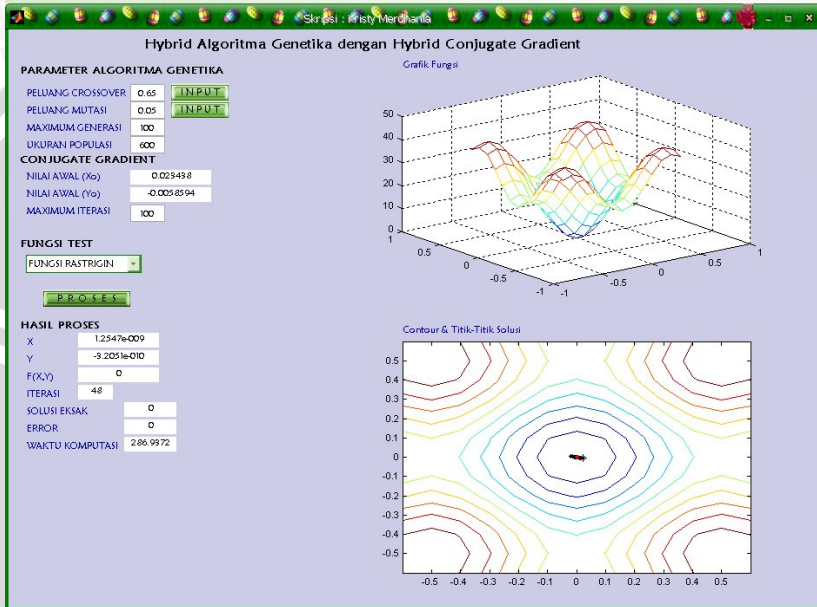
Semakin kecil nilai kesalahan mutlak (semakin mendekati nol) maka semakin baik solusi numerik yang dihasilkan.

UNIVERSITAS BRAWIJAYA



Lampiran E: Interface Program Hibrid Algoritma Genetika

Berikut ini adalah desain *interface* program hibrid algoritma genetika menggunakan software MATLAB 7 pada tes fungsi Rastrigin:



Gambar E.1 Desain *Interface* Program Hibrid Algoritma Genetika

UNIVERSITAS BRAWIJAYA

