PENERAPAN *MULTI-OBJECTIVE BAT ALGORITHM* (*MOBA*) PADA OPTIMASI FUNGSI *MULTI-OBJECTIVE* BERKENDALA

SKRIPSI

Oleh

Wafiyatul Khusna
121810101047

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PENERAPAN *MULTI-OBJECTIVE BAT ALGORITHM* (*MOBA*) PADA OPTIMASI FUNGSI *MULTI-OBJECTIVE* BERKENDALA

SKRIPSI

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

Oleh

Wafiyatul Khusna
121810101047

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang telah memberikan rahmat serta hidayah-Nya.
2. Kedua orang tua saya, Ibunda Masfufah dan Ayahanda Hadib Daini serta kakak saya Yahya Baidhowi yang selalu memberikan kasih sayang dan do’a tiada henti.
3. Guru-guru sejak jaman taman kanak-kanak sampai perguruan tinggi.
4. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

"*The greatest accomplishment is not never falling, but in rising again after you fall*."

(Vince Lombardi)*)

"Maka seyogyanya manusia jangan sampai lengah diri dari hal-hal yang bermanfaat dan berbahaya di dunia dan akhirat."

(Burhanuddin az-Zarnuji)**)

*) https://www.brainyquote.com

**) http://yurirobithoh.blogspot.co.id/2011/05/terjemahan-ta-muta.html

PERNYATAAN

Saya yang bertanda-tangan dibawah ini:

Nama : Wafiyatul Khusna

NIM : 121810101047

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada Optimasi Fungsi *Multi-Objective* Berkendala" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah disebutkan sumbernya, belum pernah diajukan dalam instanti manapun serta bukan karya jiplakan. Saya bertanggungjawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat saksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Mei 2017
Yang menyatakan,

Wafiyatul Khusna
NIM 121810101047

SKRIPSI

PENERAPAN *MULTI-OBJECTIVE BAT ALGORITHM* (*MOBA*) PADA OPTIMASI FUNGSI *MULTI-OBJECTIVE* BERKENDALA

Oleh

Wafiyatul Khusna
NIM 121810101047

Pembimbing

Dosen Pembimbing Utama : M. Ziaul Arif, S.Si., M.Sc.
Dosen Pembimbing Anggota : Drs. Rusli Hidayat, M.Sc.

PENGESAHAN

Skripsi yang berjudul "Penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada Optimasi Fungsi *Multi-Objective* Berkendala" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji,

Ketua,	Sekretaris,
M. Ziaul Arif, S.Si., M.Sc. NIP 198501112008121002	Drs. Rusli Hidayat, M.Sc. NIP 196610121993031001
Penguji I,	Penguji II,
Kosala Dwidja Purnomo, S.Si., M.Si. NIP 196908281998021001	Ahmad Kamsyakawuni, S.Si., M.Kom. NIP 197211291998021001

Mengesahkan,

Dekan,

Drs. Sujito, Ph.D.
NIP 196102041987111001

RINGKASAN

Penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada Optimasi Fungsi *Multi-Objective* Berkendala; Wafiyatul Khusna, 121810101047; 2017; 93 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Multi-Objective Bat Algorithm (*MOBA*) merupakan pengembangan dari *Bat Algorithm* yaitu algoritma metaheuristik yang terinspirasi dari perilaku kelelawar. Kelelawar merupakan satu-satunya binatang mamalia yang memiliki sayap untuk terbang dan mempunyai kemampuan canggih dalam ekolokasi. Namun *Multi-Objective Bat Algorithm* (*MOBA*) merupakan algoritma optimasi yang digunakan hanya untuk penyelesaian permasalahan *multi-objective*.

Tujuan dari penulisan tugas akhir ini yaitu menerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada optimasi fungsi *multi-objective* berkendala. Selain itu juga mengetahui konvergensi ke daerah *feasible* solusi pareto optimal yang didapatkan dari penerapan *Multi-objective Bat Algorithm* (*MOBA*) pada fungsi *multi-objective* berkendala.

Penelitian dilakukan melalui beberapa langkah, yaitu diawali dengan Studi literatur, menentukan permasalahan, kemudian menentukan masukan dan keluaran untuk pembuatan program. Selanjutnya membuat program sesuai dengan algoritma *MOBA* menggunakan *sofware* Matlab. Setelah pembuatan program yaitu implementasi *Multi-Objective Bat Algorithm* (*MOBA*), kemudian analisis hasil yang didapatkan dari implementasi *Multi-objective Bat Algorithm*. Terakhir yaitu pembuatan kesimpulan berdasarkan dari analisis hasil.

Hasil penelitian yang didapatkan menunjukkan bahwa Algoritma *MOBA* merupakan agoritma yang cukup baik diterapkan dalam proses optimasi fungsi *multi-objective* berkendala. Meskipun membutuhkan waktu komputasi sedikit lama dalam setiap kali percobaan, kelebihan dari algoritma *MOBA* adalah solusi

yang didapatkan dari setiap permasalahan seluruhnya konvergen memenuhi daerah *feasible*.

PRAKATA

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat serta hidayahNya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Penerapan *Multi-Objective Bat algorithm* (*MOBA*) pada Fungsi *Multi-Objective* Berkendala". Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

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

1. Drs. Sujito, Ph D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam dan Kusbudiono, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
2. M. Ziaul Arif, S.Si., M.Sc. selaku dosen pembimbing utama serta dosen pembimbing akademik, Drs. Rusli Hidayat, M.Sc. selaku dosen pembimbing anggota, yang telah memberikan bimbingan serta pengarahan dalam penyusunan skripsi ini;
3. Kosala Dwidja Purnomo, S.Si., M.Si. dan Ahmad Kamsyakawuni, S.Si., M.Kom. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun dalam penyusunan skripsi ini;
4. Seluruh staf pengajar Program Sarjana (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember yang telah memberikan ilmunya selama perkuliahan.
5. Kedua orang tua saya, Ibunda Masfufah dan Ayahanda Hadib Daini serta kakak saya Yahya Baidhowi yang selalu memberikan do'a hingga terselesaikannya skripsi ini;
6. Segenap keluarga dan sahabat di PALAPA, BATHICS'12, Kosan pak ribut dan Alfia.

7. Semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam menyusun skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran dari semua pihak demi proyeksi untuk skripsi ini. Semoga skripsi ini dapat bermanfaat bagi pembaca.

Jember, Mei 2017

Penulis

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Permasalahan optimasi (*optimization problem*) dalam bidang matematika umumnya terdiri dari fungsi-fungsi tujuan (*objective functions*) dan fungsi-fungsi kendala (*constraints*). Pada permasalahan optimasi terdapat beberapa kasus, yang pertama yaitu kasus dimana fungsi tujuan tunggal dengan kendala maupun tanpa kendala. Sedangkan pada kasus yang lain yaitu *multi-objective* dengan kendala maupun tanpa kendala. Kasus *multi-objective* adalah kasus dimana jumlah fungsi tujuannya lebih dari satu, maka solusi optimum dari *multicriteria optimization problem* juga lebih dari satu.

Optimasi *multi-objective* (*multicriteria optimization/ pareto-optimal*) adalah cara dalam pengambilan keputusan, yang berkaitan dengan masalah optimasi matematika. Optimasi ini melibatkan lebih dari satu fungsi tujuan yang akan dioptimalkan secara bersamaan. Optimasi *multi-objective* telah diterapkan di berbagai bidang ilmu pengetahuan, dimana keputusan yang optimal perlu diambil dengan adanya pemilihan antara dua tujuan atau lebih yang saling bertentangan.

Sarker (2001) menyatakan bahwa pada hampir semua masalah optimasi fungsi *multi-objective* tidak akan didapatkan solusi optimum tunggal, tapi berupa kumpulan solusi alternatif. Jika semua fungsi obyektif dipertimbangkan, maka tidak akan didapatkan sebuah solusi yang lebih baik terhadap solusi lain. Hal ini biasa disebut solusi *pareto-optimal*. Solusi *pareto-optimal* memberikan keleluasaan kepada manusia sebagai pengambil keputusan untuk menentukan tujuannya berdasarkan domain pengetahuan yang dimiliki.

Konsep optimasi *multi-objective* sangat melekat dalam kehidupan sehari-hari. Sebagai contohnya dalam pengambilan keputusan ketika seseorang ingin membeli mobil. Ketika akan membeli mobil, seseorang akan memikirkan tingkat kenyamanan ketika digunakan untuk berkendara, selain itu juga memikirkan biaya dan tingkat konsumsi bahan bakar. Dalam bidang ekonomi konsep optimasi *multi-*

objective berkendala juga banyak diterapkan di industri-industri. Bahkan konsep optimasi *multi-objective* digunakan untuk perancangan alat dalam bidang fisika.

Permasalahan optimasi tersebut begitu rumit untuk dipecahkan menggunakan operasi matematika biasa, terlebih jika permasalahan non linear. Suatu permasalahan non linear adalah fungsi tujuan atau kendalanya mempunyai bentuk non linear pada salah satu atau kedua fungsinya. Optimasi non linear ditinjau dari segi matematis dapat diselesaikan secara analitik, misalnya menggunakan metode *Lagrange Multiplier*. Hal tersebut efisien jika diterapkan untuk menyelesaikan bentuk permasalahan *multi-objective* non linear yang sederhana. Namun untuk menyelesaikan bentuk permasalahan *multi-objective* non linear yang kompleks (misalkan memiliki fungsi kendala berbentuk trigonometri atau permasalahan yang memiliki lebih dari 3 fungsi tujuan), maka solusinya dapat dicari dengan metode numerik optimasi. Metode numerik optimasi digunakan untuk memformulasikan persoalan matematik sehingga dapat dipecahkan dengan operasi perhitungan secara komputasi.

Beberapa peneliti telah menemukan beberapa algoritma metaheuristik untuk penyelesaian optimasi. Yang (2011) menyatakan bahwa *bat algorithm* merupakan algoritma yang sangat menjanjikan untuk memecahkan masalah optimasi. Novian (2015) melakukan penelitian tentang pencarian solusi pada permasalahan sistem persamaan non linear menggunakan metode *bat algorithm*. Hasil penelitian Novian (2015) menyatakan bahwa *bat algorithm* dapat digunakan dalam mencari solusi optimum persamaan nonlinear.

Selain itu, Yang (2012) berhasil merumuskan algoritma baru untuk optimasi *multi-objective*. Namun algoritma yang dikembangkan oleh Yang (2012) hanya fokus pada optimasi *multi-objective* tanpa kendala. Pada kasus *multi-objective* berkendala, Mahmudy (2011) telah menerapkan algoritma genetik adaptif untuk menyelesaikannya. Hasilnya ditunjukkan dengan tercapainya solusi pareto dengan waktu yang lebih singkat dibandingkan dengan algoritma genetik biasa.

Dari latar belakang diatas maka penulis bermaksud untuk menerapkan *Multi-Objective Bat Algorithm* (*MOBA*) pada penemuan solusi pareto. Khususnya yaitu untuk kasus *multi-objective* yang berkendala. Sehingga peneliti berharap bahwa solusi yang akan didapatkan bisa melanjutkan kasus berkendala sebelumnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, adapun rumusan masalah yang akan dibahas dalam skripsi ini adalah sebagai berikut:

a. Bagaimanakah penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada optimasi fungsi *multi-objective* berkendala?
b. Bagaimanakah konvergensi ke daerah *feasible* solusi pareto optimal yang didapatkan dari penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada fungsi *multi-objective* berkendala?

1.3 Batasan Masalah

Adapun batasan masalah dalam skripsi ini adalah sebagai berikut:

a. Fungsi yang digunakan berupa fungsi aljabar, fungsi logaritma dan fungsi trigonometri
b. Fungsi dua variabel dan maksimal berorde tiga

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

a. Mengetahui penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada optimasi fungsi *multi-objective* berkendala.
b. Mengetahui konvergensi ke daerah *feasible* solusi pareto optimal yang didapatkan dari penerapan *Multi-Objective Bat Algorithm* (*MOBA*) pada fungsi *multi-objective* berkendala.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah memberikan wawasan dan alternatif baru kepada pembaca tentang penyelesaian permasalahan *multi-objective* yaitu menggunakan *Multi-Objective Bat Algorithm* (*MOBA*). Khususnya pada permasalahan fungsi *multi-objective* berkendala.

BAB 2. TINJAUAN PUSTAKA

2.1 Sistem Persamaan Non linear

Dalam matematika sistem persamaan ada dua, yaitu sistem persamaan linear dan sistem persamaan non linear. Suatu persamaan linear yang mengandung *n* peubah $x_1, x_2, \dots, x_n$ dinyatakan dalam bentuk $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$ dengan $a_1, a_2, \dots, a_n, b$ adalah konstanta riil. Sedangkan sistem persamaan linear adalah himpunan berhingga dari persamaan linear.

Misalkan: $x + y = 2$

$$2x + 2y = 6$$

(Yuliant, 2002).

Persamaaan non linear memiliki variabel berderajat tidak sama dengan satu atau mengandung nilai fungsi non linear seperti logaritma dan sebagainya. Sedangkan sistem persamaaan non linear adalah kumpulan dari beberapa persamaan non linear yang saling berhubungan untuk mencapai tujuan tertentu.

Misalkan: $2xy + 6y = 4\log(x)$

$$x + 2y^{1/2} - z = 2x^2$$

Penyelesaian optimasi non linear menurut Bronson (1997) terbagi dalam tiga bagian, yaitu:

1. Masalah optimasi non linear satu *variable*

   Dimisalkan *x* adalah *variable* penentu dan *f*(*x*) adalah fungsi tujuan dari suatu masalah. Metode optimasi untuk menyelesaikan masalah tersebut adalah dengan maximum *f*(*x*) atau minimum *f*(*x*)
2. Masalah optimasi non linear *multi-variable* tanpa kendala

   Optimasi tanpa kendala adalah optimasi suatu fungsi tanpa adanya syarat-syarat tertentu yang membatasinya.
3. Masalah optimasi non linear *multi-variable* dengan kendala.

   Jika suatu fungsi terikat oleh satu atau lebih syarat maka optimasi tersebut berkendala.

Dilihat dari bentuk kendalanya, masalah optimasi non linear dengan kendala terbagi menjadi 3 bagian, yaitu:

1. Fungsi kendala berbentuk persamaan

   Secara umum permasalahan *multi-objective optimization* diformulasikan ke dalam bentuk permasalahan matematika sebagai berikut:

   Minimize/Maximize $f_m(x)$, m = 1, 2, . . ., M;

   Subject to $g_j(x) = 0$, $j = 1,2,\ldots,J$;

   $h_k(x) = 0$, $k = 1,2,\ldots,K$;

   $x_i^{(L)} \le x_i \le x_i^{(U)}$ $i = 1,2,\ldots,n$.

2. Fungsi kendala berbentuk pertidaksamaan

   Optimasi *multi-variable* dengan kendala pertidaksamaan mempunyai bentuk umum sebagai berikut:

   minimumkan $f = f(X)$ degan $X = \{x_1, x_2, \ldots, x_n\}^t$

   kendala $g_j(X)\{\ge, >, <, \le\}0$, dengan $j = 1, 2, \ldots, m$

3. Fungsi kendala berbentuk campuran persamaan dan pertidaksamaan

   Merupakan gabungan gabungan dari fungsi kendala yang berbentuk persamaan dan petidaksamaan. Bentuk umum dari permasalahan ini adalah sebagai berikut:

   minimumkan $f = f(X)$ degan $X = \{x_1, x_2, \ldots, x_n\}^t$

   kendala $g_j(x)\{=, \ge, >, <, \le\}b_j$ dengan $j = 1, 2, \ldots, m$

Jika dilihat dari jumlah fungsi tujuannya, permasalahan optimasi sistem persamaan non linear dibagi menjadi dua bentuk yaitu:

1. *Single objective function*

   Permasalahan optimasi yang hanya memiliki satu fungsi tujuan.

2. *Multi-objective function*

   Permasalahan optimasi yang memliki lebih dari satu fungsi tujuan.

Pembahasan dalam skripsi ini akan difokuskan pada permasalahan optimasi fungsi *multi-objective* yang memiliki kendala pertidaksamaan.

2.2 *Multi-Objective Optimization*

Multi-objective Optimization merupakan sebuah proses yang dilakukan secara simultan untuk optimalisasi dua atau lebih fungsi tujuan (objektif). Proses pencarian dari dua atau lebih variabel keputusan sekaligus yang memenuhi semua kendala, dan mengoptimalkan vektor fungsi *objective* dengan memetakan variabel keputusan untuk dua atau lebih *objective*.

$$\text{Minimum/maksimum } \left(f_k(s)\right), \forall k \in [1, K]$$

Setiap vektor keputusan atau solusi $\boldsymbol{s} = \{(s_1, s_2, \dots, s_m)\}$ menunjukkan ketepatan kualitas numerik dari *multi-objective optimization*. Kumpulan dari semua vektor keputusan merupakan ruang keputusan (*decision space*). Kumpulan dari dua atau lebih variabel keputusan sekaligus yang memenuhi semua kendala disebut *feasible set* (*F*). Vektor fungsi objektif (***f***) memetakan ruang vektor dari ruang keputusan kedalam K-dimensi objektif ruang.

Pareto optimality adalah sebuah konsep untuk menangani permasalahan *multi-objective optimization*. Konsep tersebut dikemukakan oleh Villtredo federico damaso pareto dalam bukunya yang terkenal *Manual Policival Economy* yang ditulis dalam bahasa Perancis tahun 1896. Konsep pareto berbunyi: *An optimal allocation of resources is achieved when it is not possible to make anyone better off without making someone else worse off*. Secara matematis didefinisikan:
x^* disebut *pareto optimal* jika dan hanya jika tidak ada solusi lain yang lebih layak dari x sehingga $f_i(x) \leq f_i(x^*), \forall i \in \{1,2, \dots, n\}$.

Solusi optimal yang diperoleh dari *pareto optimal* bukan berupa satu titik melainkan kumpulan beberapa titik, yang kesemuanya memenuhi konsep *Pareto optimality*. Kumpulan titik-titik tersebut dinamakan *pareto set*, atau disebut juga *pareto frontier*, karena semua titik tersebut mendominasi titik-titik yang lain.

2.3 *Bat Algorithm*

Bat Algorithm (*BA*) merupakan algoritma metaheuristik yang terinspirasi dari perilaku kelelawar. *Bat Algorithm* (*BA*) diperkenalkan oleh Xin She Yang pada tahun 2010. Kelelawar merupakan satu-satunya binatang mamalia yang memiliki sayap untuk terbang dan mempunyai kemampuan canggih dalam ekolokasi.

Kelelawar menggunakan tipe sonar yang disebut ekolokasi untuk mendeteksi makanan, menghindari rintangan dan mencari sarangnya dalam kegelapan (Suharto, 2015).

Beberapa asumsi yang harus diterapkan dalam mengimplementasikan *Bat Algorithm* (*BA*):

1. Semua kelelawar menggunakan ekolokasi untuk mengindra jarak, serta ekolokasi juga berfungsi untuk membedakan antara mangsa dengan penghalang.
2. Kelelawar terbang secara random dengan kecepatan v_i serta pada posisi x_i dengan frekuensi f_{min}. Selain itu kelelawar juga memiliki panjang gelombang serta *loudness* yang bervariasi dalam mencari mangsa. Setiap kelelawar secara otomatis mengatur besaran frekuensi serta *pulse rate* dengan besaran $r \in [0,1]$ yang mana nilai tersebut memiliki ketergantungan dengan jauh dekatnya jarak antar kelelawar dengan mangsanya.
3. Meskipun secara alamiah tingkatan *loudness* dapat bervariasi, namun pada prakteknya, nilai *loudness* diasumsikan hanya berkisar antara A_0 yang bernilai positif hingga mencapai nilai minimum konstan A_{min}

2.3.1 *Bat Motion*

Setiap iterasi, nilai kelelawar didefinisikan dengan posisi $x_i(x,y)$, kecepatan $v_i(x,y)$ di d-dimensi untuk setiap pencarian jarak terbaru. Solusi terbaru dari $x_i^t(x,y)$ dan kecepatan $v_i^t(x,y)$ pada waktu *t* didefinisikan:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \qquad 2.1$$

$$v_i^{t+1}(x,y) = v_i^t(x,y) + (x_i^t(x,y) - x_*(x,y))f_i \qquad 2.2$$

$$x_i^{t+1}(x,y) = x_i^t(x,y) + v_i^t(x,y) \qquad 2.3$$

$\beta \in [0,1]$ merupakan nilai random yang dihasilkan dari distribusi uniform. Nilai x_* merupakan nilai solusi terbaik (*global optimal*) sementara setelah membandingkan solusi untuk semua *bat n* pada setiap iterasi *t*. $\lambda_i f_i$ digunakan untuk meningkatkan kecepatan. f_i atau λ_i dapat digunakan untuk mengatur perubahan kecepatan, sementara untuk memperbaiki atau menstabilkan λ_i atau f_i tersebut dapat menggunakna f_i atau λ_i yang belum digunakan tergantung dari permasalahan yang ada. Nilai f_{min} atau f_{max} diasumsikan $f_{min} = 0$ dan $f_{max} = 1$ sesuai dengan permasalahan. Sedangkan inisialisasi nilai f untuk setiap *bat* dihasilkan secara random menggunakan distribusi poisson mengikuti $[f_{min}, f_{max}]$.

Pencarian jarak terdekat dalam algoritma ini menggunakan metode *local search*. Metode *local search* adalah metode pencarian iteratif yang menjadi dasar bagi metode-metode pencarian lebih lanjut. *Local search*, untuk setiap *bat* dengan solusi terbaik yang terpilih, solusi baru untuk *bat* tersebut dihasilkan menggunakan *Random Walk*. Yaitu:

$$x_{new}(x, y) = x_{old}(x, y) + \epsilon A^t \qquad 2.4$$

dimana ϵ merupakan *random number* yang nilainya antara $[-1, 1]$. Sedangkan $A^t = \langle A_i^t \rangle$ merupakan rata-rata *loudness* dari seluruh *bat* pada iterasi *t*.

2.3.2 *Loudness* dan *Pulse Rate*

Setiap iterasi ke t, nilai *loudness* A_i serta nilai *pulse* r_i harus di *update* secara berlanjutan. Secara teori, *bat* mengurangi nilai *loudness* serta meningkatkan nilai *pulse* ketika sudah mendapatkan mangsanya.

$$A_i^{t+1} = \propto A_i^t \qquad 2.5$$

$$r_i^t = r_i^0[1 - exp(-\gamma t)] \qquad 2.6$$

$A_o = 100$ dan $A_{min} = 1$ atau disederhanakan $A_o = 1$ dan $A_{min} = 0$ (Asumsikan $A_{min} = 0$ ketika bat sudah menemukan mangsa dan menghentikan ekolokasinya). Sedangkan nilai $\propto$ dan γ merupakan nilai konstan, $0 < \propto < 1$ dan $\gamma > 0$. Dalam simulasinya, dapat digunakan $\propto = \gamma = 0.9$. Sehingga diketahui:

$$A_i^t \to 0, \qquad r_i^t \to r_i^0, \qquad atau \quad t \to \infty$$

(Yang, 2011).

2.4 *Multi-Objective Bat Algorithm*

Multi-Objective Bat Algorithm (*MOBA*) merupakan pengembangan dari Algoritma *BA* (*Bat Algorithm*). *MOBA* diperkenalkan oleh Xin She Yang pada tahun 2012. Berbeda dengan *Bat Algorithm*, optimasi *multi-objective* merupakan permasalahan yang lebih kompleks dari pada optimasi *single objective*. Sehingga pada *multi-objective bat algorithm* untuk pendekatannya digunakan *optimality fronts*. Sebagai solusi non dominan, *pareto front* dalam *multi-objective* didefinisikan sebagai berikut:

$$PF = \{s \in S | \nexists s' \in S : s' < s\}$$

Sedangkan dalam pencarian jarak didefinisikan:

$$PF' = \{x \in F | \nexists x' \in F : f(x') < f(x)\}$$

dimana $f = (f_1, \dots, f_K)^T$.

Pada algoritma *MOBA* setiap fungsi tujuan dikombinasikan menjadi *single objective* campuran. Hal ini dapat diselesaikan dengan menggunakan berbagai metode atau cara. Metode yang digunakan pada MOBA adalah penjumlahan bobot untuk mengkombinasikan semua fungsi tujuan f_k menjadi fungsi tujuan tunggal.

$$f = \sum_{k=1}^{K} \omega_k f_k \ , \qquad \sum_{k=1}^{K} \omega_k = 1 \qquad 2.7$$

Dimana ω_k diturunkan secara acak dari distribusi uniform, hal tersebut sesuai agar dapat merubah bobot dengan nilai keragaman yang cukup sehingga *pareto front* dapat didekati dengan tepat (Yang, 2012).

2.5 *Lagrange Multiplier*

Lagrange multiplier merupakan metode yang melibatkan modifikasi dari fungsi tujuan melalui penambahan parameter yang menggambarkan kendala. Fungsi tujuan $f(x)$ ditambah dengan $\sum \lambda g(x)$, dimana λ merupakan parameter lagrange. Fungsi lagrange didefinisikan:

$$J_A(x_1, x_2, \dots x_n, \lambda_1, \lambda_2, \dots, \lambda_m) = f(x_1, x_2, \dots x_n) + \sum_{j=1}^{m} \lambda_j g_j\,(x_1, x_2, \dots x_n)$$

(Gavin, 2016).

Titik minimum dalam optimasi multi variabel yang memiliki kendala pertidaksamaan dapat diselesaikan dengan metode lagrange menggunakan syarat Kuhn - Tucker. Misalkan permasalahan:

Minimumkan $f = f(X)$ dengan $X = \{x_1, x_2, \dots, x_n\}^t$

kendala $g_j(X) \leq 0$, dengan $j = 1,2, \dots, m$

Adapun syarat kuhn – tucker:

$\frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{g_j}{\partial x_i} = 0 \qquad i = 1,2, \dots, n$

$\lambda_j g_i = 0 \qquad j = 1,2, \dots, m$

$g_i \leq 0 \qquad j = 1,2, \dots, m$

$\lambda_j \geq 0 \qquad j = 1,2, \dots, m$

(Luknanto, 2000).

2.6 *Mean Squared Error* (*MSE*)

Nilai tengah kesalahan kuadrat atau yang lebih dikenal dengan *Mean Squared Error* (*MSE*) adalah metode untuk mengukur tingkat keakuratan suatu model peramalan. Mendenhall, dkk (1990) menyatakan bahwa semakin kecil nilai *MSE* yang dihasilkan maka semakin baik penduga yang dihasilkan. Jika menurut Gaspersz (2005), akurasi peramalan akan semakin tinggi apabila nilai – nilai *MSE* semakin kecil. *MSE* didefinisikan dengan rumus sebagai berikut:

$$MSE = \sum_{t=1}^{n} \frac{(X_t - F)^2}{n} \qquad 2.8$$

Keterangan:

X_t : Data real pada periode t (penduga)

F : Nilai ramalan

n : Banyaknya data real

(Lehmann, 1998).

2.7 Matlab

Matlab merupakan bahasa pemrograman dengan kemampuan tinggi dalam bidang komputasi. Memiliki kemampuan mengintegrasikan komputasi, visualisasi, dan pemrograman. Dalam memvisualisasikan sebuah obyek, matlab memiliki kemampuan merotasi obyek tanpa mengubah programnya. Program Matlab dikembangkan oleh Mathworks Inc. Matlab merupakan sebuah singkatan dari

Matrix Laboratory, yang pertama kali dikenalkan oleh University of New Mexico dan University of Stanford pada tahun 1970. Software ini pertama kali memang digunakan untuk keperluan analisis numerik, aljabar linier dan teori tentang matriks.

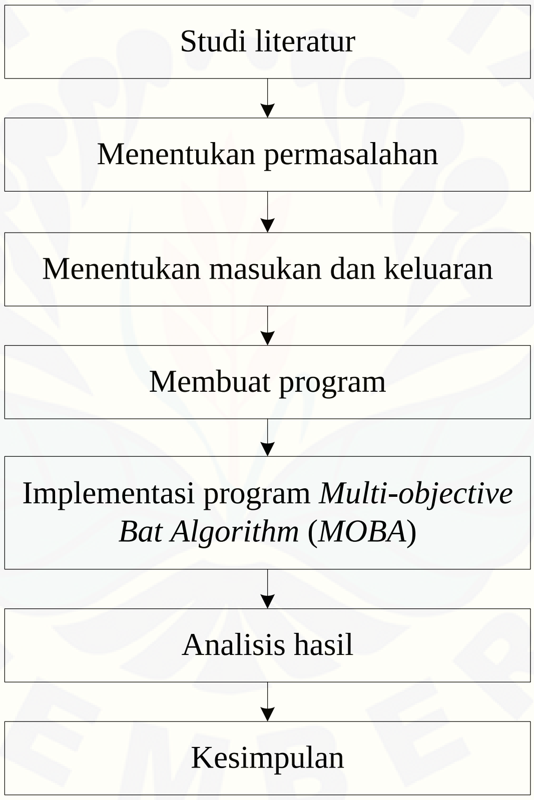
Beberapa manfaat yang didapatkan dari Matlab antara lain:

1. Perhitungan Matematika
2. Komputasi numerik
3. Simulasi dan pemodelan
4. Visualisasi dan analisis data
5. Pembuatan grafik untuk keperluan sains dan teknik
6. Pengembangan aplikasi, misalnya dengan memanfaatkan GUI.

(Supardi, 2010).

BAB 3. METODE PENELITIAN

Pada penelitian ini, penulis akan menerapkan *Multi-objective Bat Algorithm* (*MOBA*) pada optimasi fungsi *multi-objective* berkendala. Permasalahan yang digunakan adalah permasalahan fungsi *multi-objective* yang memiliki kendala non linear. Langkah-langkah yang akan dilakukan dalam penelitian ini, dapat digambarkan dengan diagram alir seperti berikut:



Gambar 3.1 Skema metode penelitian

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

a. Studi literatur

   Peneliti melakukan studi literatur yaitu dengan mempelajari beberapa hal dari buku, jurnal, skripsi guna mendapatkan informasi mengenai optimasi fungsi *multi-objektif*, *MOBA* dan fungsi berkendala non linear.

b. Menentukan permasalahan

   Masalah yang akan dibahas dalam skripsi ini adalah penyelesaian optimasi fungsi *multi-objective* berkendala. Permasalahan fungsi diambil dari beberapa jurnal yang berisi tentang optimasi fungsi *multi-objective* yang memiliki kendala non linear.

c. Menentukan masukan dan keluaran

   Sebelum pembuatan program, peneliti terlebih dahulu menentukan *input* dan *output* dari program yang akan dibuat.

   1. *Input*

      Sistem persamaan non linear, fungsi kendala, nilai pareto, jumlah *bat*, iterasi maksimal, jumlah dimensi, batas minimal dan maksimal posisi, serta parameter-parameter dalam *MOBA*. Sesuai dengan jurnal Yang (2012), adapun parameter-parameter yang akan digunakan dalam algoritma *MOBA* adalah:

      - Nilai *pulse* = 0,5
      - Nilai *loudness* = 0,5
      - $\beta \in [0,1]$
      - $f_{min} = 0$ dan $f_{max} = 2$
      - Nilai ϵ dalam pencarian jarak terdekat yaitu $[-1,1]$

   2. *Output*

      Adapun *output* yang akan dihasilkan dari program skripsi ini adalah berupa grafik konvergensi pareto optimal kedalam daerah *feasible* serta nilai konvergensi yang memenuhi kendala.

d. Membuat program

Program dibuat dengan menggunakan *sofware* Matlab. Langkah awal dengan menuliskan *script* dan kemudian dijalankan untuk menyelesaikan permasalahan secara komputasi.

e. Implementasi *Multi-objective Bat Algorithm* (*MOBA*)

Program yang telah dibuat akan digunakan untuk menyelesaikan permasalahan optimasi fungsi *multi-objective* berkendala.

Adapun algoritma dari *MOBA* adalah sebagai berikut:

a) Definisikan fungsi *objective,* fungsi kendala
b) Inisialisasikan nilai pareto, jumlah *bat*, iterasi maksimal, jumlah dimensi, batas minimal dan maksimal posisi, batas minimal dan maksimal frekuensi, nilai *loudness* dan *pulse rate*
c) Mengkombinasikan semua fungsi tujuan menjadi fungsi tujuan tunggal
d) Optimasi dengan *Bat Algorithm* (*bat motion*)
e) Menerima solusi baru, menambah *pulse rates* dan mengurangi *loudness*
f) Mencari nilai terbaik
g) Catat dan simpan nilai sebagai solusi

f. Analisis hasil

Analisis dilakukan untuk mengetahui hasil yang diperoleh. Peneliti akan menganalisis hasil yang diperoleh dari perhitungan analitik dengan metode *Lagrange Multiplier* dan juga menganalisis hasil dari perhitungan secara komputasi menggunakan program yang telah dibuat dari bantuan sofware matlab. Sesuai dengan implementasi *MOBA*, maka yang akan dianalisis adalah:

1. Analisis grafik kekonvergenan dan daerah *feasible*
2. Analisis nilai hasil penerapan *MOBA*
3. Analisis berdasarkan nilai *Mean Squared Error* (*MSE*)

g. Kesimpulan

Kesimpulan dibuat berdasarkan hasil analisis dan menyesuaikan dengan rumusan masalah.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan rumusan masalah yang telah dikemukakan, adapun kesimpulan yang dapat diambil dari skripsi ini adalah sebagai berikut:

a. Penerapan *Multi-objective Bat Algorithm* (*MOBA*) pada optimasi fungsi *multi-objective* berkendala yaitu dengan meggabungkan fungsi *multi-objective* menjadi fungsi *single objective*. Pada kasus optimasi fungsi *multi-objective* yang memiliki fungsi kendala berbentuk aljabar (kuadratik dan kubik), menghasilkan nilai *fitness* yang selalu konvergen di daerah *feasible* dan solusi yang didapatkan selalu mendekati solusi eksak. Pada kasus optimasi fungsi *multi-objective* yang memiliki fungsi kendala berbentuk fungsi trigonometri, menghasilkan daerah *feasible* yang lebih dari satu. Dalam beberapa percobaan pada kasus optimasi fungsi *multi-objective* yang memiliki fungsi kendala berbentuk fungsi logaritma, pada setiap percobaan menghasilkan nilai *fitness* yang selalu konvergen di daerah *feasible* dan solusi yang dihasilkan selalu berada di persekitaran solusi eksak.

b. Algoritma *MOBA* dibuktikan bahwa algoritma tersebut cukup baik diterapkan untuk kasus optimasi fungsi *multi-objective* berkendala. Hal tersebut dapat diketahui dari pembuktian nilai *MSE* dari kasus pertama, kasus ketiga dan kasus keempat yang memiliki nilai *MSE* mendekati nilai 0. Hal tersebut mengacu pada pernyataan Mendenhall, dkk (1990) dan Gaspersz (2005) yang salah satunya menyatakan bahwa semakin kecil nilai *MSE* yang dihasilkan maka semakin baik. Algoritma *MOBA* membutuhkan waktu komputasi sedikit lama namun kelebihan dari algoritma *MOBA* yaitu solusi yang didapatkan selalu konvergen memenuhi daerah *feasible*.

5.2 Saran

Program algoritma MOBA dalam skripsi ini masih terbatas pada persamaan yang berdimensi dua dan bentuk permasalahan yang terbatas. Sehingga diharapkan dapat dilakukan penelitian dengan mengembangkan program agar dapat menyelesaikan beberapa bentuk persamaan yang lebih luas.

DAFTAR PUSTAKA

Asghar, M. Bhatti. 2000. *Practical Optimization Methods with Mathematic Applications*. New York: Springer-Verlag.

Bansal, S., Singh, A.K. dan Gupta, N. 2016. *Optimal Golomb Ruler Sequences Generation for Optical WDM Systems: A Novel Parallel Hybrid Multi-objective Bat Algorithm*. *Journal Institution of Engineers India Ser.B.*

Bronson, R. 1997. *Operation Research, 2nd Edition*. New Delhi: McGraw Hill Professional.

Deb, K., Pratap, A., Meyarivan, T. 2002. *A Fast and Elitist Multiobjective Genetic Algorithm*: NSGA-II. *IEEE Transactions On Evolutionary Computation*, Vol. 6, No. 2, April 2002.

Gaspersz, Vincent. 2005. *Total Quality Management.* Jakarta: PT Gramedia Pustaka.

Gavin, H.P., Scruggs, J.T. 2016. *Constrained Optimization Using Lagrange Multipliers*. CEE 201L. *Uncertainty, Design, and Optimization*. Duke, Spring 2016.

Gill, C., Marquez, A., Banos, R., Montonya, M. G. dan Gomez, J. 2007. *A Hybrid Method For Solving Multi-Objective Global Optimization Problems*. *Journal Global Optimal*, 38:265–281.

Lehmann, E.L, Casella, G. 1998. *Theory of Point Estimation, Second Edition*. New York: Springer-Verlag.

Luknanto, D. 2000. Pengantar Optimasi Non Linear. Yogyakarta: Universitas Gajah Mada.

Mahmudy, W.F. dan Rahman, M.A. 2011. Optimasi Fungsi Multi-Obyektif Berkendala Menggunakan Algoritma Genetika Adaptif Dengan Pengkodean *Real*. Jurnal Ilmiah, 6 (1): 0216 – 0544.

Mendenhall, W., Scheaffer, R.L, dan Wackerly, D.D. 1990. *Mathematical Statistic with Application Fourth Edition.* California: Wasworth.

Novian, T. 2015. Pencarian Solusi Pada Permasalahan Sistem Persamaan Nonlinear Menggunakan Metode Bat Algorithm. *Skripsi*. Jakarta: UIN Syarif Hidayatullah.

Sibaroni, Yuliant. 2002. Buku Ajar Aljabar Linear. Bandung: STT Telkom.

Soelaeman, R. dan Chasiani, N. (2009). Penerapan Optimasi Chaos Dan Metode BFGS (Broyden, Broyden, Fletcher, Goldfarb, And Shanno) Pada Penyelesaian Permasalahan Sistem Persamaan Nonlinier. Surabaya: ITS.

Suharto., Robandi, I. dan Priyadi, A. 2015. Penalaan *Power System Stabilizer* (PSS) untuk Perbaikan Stabilitas Dinamik pada Sistem Tenaga Listrik Menggunakan *Bat Algorithm* (BA). Jurnal Teknik ITS, 4 (1): 337-3539.

Supardi. 2010. Pemrograman Matlab. Yogyakarta: Universitas Negeri Yogyakarta.

Yang, Nien-Che. dan Le, Minh-Duy. 2015. *Optimal Design Of Passive Power Filters Based On Multi-Objective Bat Algorithm And Pareto Front*. *Journal Applied Soft Computing*.35(C): 257-266.

Yang, X. 2011. *A new metaheuristic bat-inspired algorithm*. *Nature Inspired Cooperative Strategies for Optimization*. Springer, 284: 65-74.

Yang, X. 2012. *Bat Algorithm For Multi-objective Optimization*. *International Journal Bio-Inspired Computation*, 3 (5): 267-274.

LAMPIRAN

A. KASUS 1 (FUNGSI ALJABAR)

A.1 Bentuk permasalahan kasus 1

Minimumkan $f_1(x,y), f_2(x,y)$:

$$f_1(x,y) = 4x^2 + 4y^2$$

$$f_2(x,y) = x^2 + (y-1)^2$$

Dengan kendala:

$$x^2 - 9y \leq 0$$

$$x + 2y \leq 0$$

Batas pencarian: $-15 \leq x \leq 15$ dan $-15 \leq y \leq 15$

Jumlah *bat* = 15 *bat*

Iterasi maksimal = 100 iterasi

Nilai pareto = 10 pareto

A.2 Hasil yang didapatkan dari percobaan pertama pada kasus 1

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,6	-0,28264	0,1036	0,362473	0,883418	1,245892	0,57085
2	0,3	-0,79503	0,3817	3,11107	1,014368	4,125438	1,6434
3	**0,7**	**-0,08037**	**0,039049**	**0,031933**	**0,929885**	**0,961819**	**0,30132**
4	0,8	-0,24486	0,094094	0,27524	0,880622	1,155863	0,39632
5	0,5	-0,86883	0,43108	3,762782	1,078536	4,841318	2,4207
6	0,1	-0,32637	0,14393	0,508933	0,839373	1,348306	0,80633
7	0,2	-0,84616	0,37223	3,418168	1,110082	4,52825	1,5717
8	0,4	-0,7429	0,32165	2,621437	1,012059	3,633496	1,6558
9	0,9	-0,40697	0,19902	0,820934	0,807194	1,628128	0,81958
10	1	-0,69851	0,07973	1,977092	1,334813	3,311905	1,9771

A.3 Hasil terbaik dari 10 kali percobaan pada kasus 1

Percobaan	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,7	-0,08037	0,039049	0,031933	0,929885	0,961819	0,30132
2	1	-0,1197	0,025736	0,059962	0,963518	1,02348	0,059959
3	0,5	-0,36693	0,17577	0,662131	0,813993	1,476124	0,73807
4	1	-0,2292	0,04406	0,217896	0,966354	1,18425	0,21789

5	0,7	-0,28798	0,017557	0,332963	1,048127	1,38109	0,54752
6	0,5	-0,14876	0,069324	0,107741	0,888287	0,996029	0,49801
7	0,8	-0,21144	0,069697	0,198258	0,910171	1,108429	0,34064
8	0,5	-0,36693	0,17577	0,662131	0,813993	1,476124	0,73807
9	1	-0,17126	0,064379	0,133899	0,904717	1,038615	0,13389
10	0,7	-0,11216	0,026304	0,053087	0,960664	1,013751	0,32538

B. KASUS 2

B.1 Bentuk permasalahan kasus 2

Minimumkan $f_1(x,y), f_2(x,y)$:

$$f_1(x,y) = (x-2)^2 + (y-1)^2 + 2$$

$$f_2(x,y) = 9x - (y-1)^2$$

Dengan kendala:

$$x^2 + y^2 - 255 \leq 0$$

$$x - 3y + 10 \leq 0$$

Batas pencarian: $-17 \leq x \leq 17$ dan $-15 \leq y \leq 15$

Jumlah *bat* = 15 *bat*

Iterasi maksimal = 100 iterasi

Nilai pareto = 10 pareto

B.2 Hasil dari perhitungan kasus 2

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,8	-2,0692	4,4267	30,30066	-30,3651	-0,06441	18,1676
2	0,9	-0,0645	4,0932	15,83015	-10,1486	5,681543	13,2321
3	1	-0,0639	7,2033	44,74079	-39,0564	5,684374	44,7411
4	0,6	-3,911	3,2301	41,91327	-40,1723	1,740921	9,0791
5	0,3	-8,4207	10,8619	207,8472	-173,043	34,80421	-58,776
6	**0,1**	**-11,883**	**10,3874**	**282,8526**	**-195,068**	**87,78506**	**-147,28**
7	0,4	-1,8913	15,4695	226,5083	-226,388	0,120564	-45,23
8	0,2	-7,1559	13,4922	241,8857	-220,458	21,4275	-127,99
9	0,5	-0,4706	15,3744	214,7272	-210,859	3,868513	1,9343
10	0,7	-4,5784	2,6329	47,94171	-43,872	4,069747	20,3975

C. KASUS 3 (FUNGSI TRIGONOMETRI)

C.1 Bentuk permasalahan kasus 3

Minimumkan $f_1(x,y), f_2(x,y)$:

$$f_1(x,y) = 2 + (x-2)^2 + (y-1)^2$$

$$f_2(x,y) = 9x - (y-1)^2$$

Dengan kendala:

$$\sin x + y \leq 0$$

$$\cos 3x - y \leq 0$$

Batas pencarian: $-9 \leq x \leq 2$ dan $-9 \leq y \leq 2$

Jumlah *bat* = 15 *bat*

Iterasi maksimal = 100 iterasi

Nilai pareto = 15 pareto

C.2 Hasil yang didapatkan dari percobaan pertama pada kasus 3

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,8	-0,9135	0,7848	10,535	-8,2682	2,2668	6,7744
2	0,3333	-7,3159	-0,9818	92,713	-69,771	22,943	-15,609
3	0,5333	-1,4835	0,6915	14,229	-13,447	0,7833	1,3142
4	0,8667	0,8985	-0,8238	6,5397	4,7599	11,299	6,3024
5	**0,0667**	**-8,9827**	**0,1525**	**123,338**	**-81,562**	**41,775**	**-67,902**
6	0,7333	-0,526	0,4379	8,6968	-5,0503	3,6465	5,0309
7	1	-1,0286	0,5945	11,3368	-9,4218	1,915	11,337
8	0,9333	-0,9947	0,1828	11,6362	-9,6202	2,0159	10,219
9	0,6	-1,2792	0,8593	12,773	-11,533	1,2403	3,0507
10	0,2	-7,6128	-0,3068	96,1137	-70,223	25,891	-36,956
11	0,4	-5,4303	-0,8101	60,486	-52,149	8,3366	-7,0951
12	0,6667	-0,5696	0,4754	8,8781	-5,4018	3,4764	4,1182
13	0,1333	-8,0131	0,9219	102,268	-72,124	30,144	-48,872
14	0,4667	-3,1736	-0,8977	32,367	-32,164	0,2037	-2,0492
15	0,2667	-7,5363	-0,6796	95,762	-70,648	25,114	-26,272

C.3 Hasil terbaik dari 10 kali percobaan pada kasus 3

Percobaan	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,0667	-8,9827	0,1525	123,338	-81,562	41,775	-67,902
2	0,0667	-8,9112	0,0511	121,955	-81,101	40,853	-67,564
3	0,0667	-8,9824	0,0611	123,494	-81,723	41,771	-68,041
4	0,0667	-8,9864	-0,085	123,877	-82,054	41,823	-68,325

5	0,0667	-8,9173	0,1619	121,889	-80,958	40,932	-67,435
6	0,0667	-8,974	0,3605	122,838	-81,175	41,663	-67,574
7	0,0667	-8,9827	0,1525	123,338	-81,562	41,775	-67,902
8	0,0667	-8,9342	-0,057	122,674	-81,525	41,149	-67,912
9	0,0667	-8,9515	0,1504	122,657	-81,285	41,374	-67,689
10	0,0667	-8,9857	-0,173	124,061	-82,246	41,814	-68,492

D. KASUS 4 (FUNGSI LOGARITMA)

D.1 Bentuk permasalahan kasus 4

Minimumkan $f_1(x,y), f_2(x,y)$:

$$f_1(x,y) = 4x^2 + 4y^2$$

$$f_2(x,y) = (x-5)^2 + (y-5)^2$$

Dengan kendala:

$$x^2 + y - 3 \leq 0$$

$$\log(\mathrm{x}+4) - \mathrm{y} - 7 \leq 0$$

Batas pencarian: $-7 \leq x \leq 4$ dan $-7 \leq y \leq 4$

Jumlah *bat* = 15 *bat*

Iterasi maksimal = 150 iterasi

Nilai pareto = 35 pareto

D.2 Hasil yang didapatkan dari percobaan pertama pada kasus 4

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,4	0,3247	1,2336	6,5089	36,044	42,553	24,229
2	0,2	0,9029	1,8094	16,356	26,966	43,323	24,844
3	0,0667	0,7613	1,8617	16,182	27,815	43,997	27,04
4	0,8667	0,0582	-0,4137	0,6983	53,729	54,428	7,7691
5	0,6	0,7584	0,4922	3,2698	38,311	41,581	17,286
6	0,2667	1,0881	1,1426	9,9579	30,182	40,14	24,789
7	0,4667	1,227	0,7879	8,5117	31,972	40,484	21,024
8	0,3333	0,7125	2,3398	23,929	25,459	49,387	24,949
9	0,9333	0,2694	0,2246	0,4922	45,183	45,675	3,4715
10	0,6667	0,5436	0,4031	1,8321	40,991	42,823	14,885
11	0,5333	1,1781	1,0614	10,058	30,119	40,177	19,419
12	0,8	0,55816	0,2467	1,4895	42,324	43,814	9,6565
13	0,7333	0,76892	0,5104	3,4069	38,058	41,465	12,647
14	**1**	**0,55571**	**-0,0826**	**1,2625**	**45,585**	**46,847**	**1,2626**
15	0,1333	-0,4055	1,9482	15,839	38,533	54,374	35,507

D.3 Hasil terbaik dari 10 kali percobaan pada kasus 4

Percobaan	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	1	0,55571	-0,0826	1,2626	45,585	46,847	1,2626
2	1	0,15089	0,065177	0,108063	47,86635	47,97441	0,10807
3	0,0667	0,13617	0,12376	0,135435	47,43456	47,56999	3,6679
4	1	0,33984	-0,10321	0,504574	47,75984	48,26442	0,50457
5	0,0667	0,32057	0,24878	0,658626	44,47116	45,12978	5,4236
6	1	0,14626	0,12822	0,151329	47,29303	47,44436	0,15133
7	1	0,3769	0,063906	0,58455	45,73808	46,32263	0,58455
8	1	0,42757	0,16434	0,839295	44,29072	45,13002	0,8393
9	0,0667	0,19142	0,0865	0,176495	47,26492	47,44142	3,809
10	0,0667	0,03932	0,07003	0,025801	48,91295	48,93875	3,3572

E. KASUS 5

E.1 Bentuk permasalahan kasus 5

Minimumkan $f_1(x,y), f_2(x,y)$:

$$f_1(x,y) = 2 + (x-2)^2 + (y-1)^2$$

$$f_2(x,y) = 9x - (y-1)^2$$

$$f_3(x,y) = 4x^2 + 4y^2$$

Dengan kendala:

$$x^2 + y^2 - 225 \leq 0$$

$$x - 3y + 10 \leq 0$$

$$3x - y + 10 \leq 0$$

Batas pencarian: $-15 \leq x \leq 15$ dan $-15 \leq y \leq 15$

Jumlah *bat* = 15 *bat*

Iterasi maksimal = 100 iterasi

Nilai pareto = 10 pareto

E.2 Hasil dari perhitungan kasus 5

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	$f_3(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,4	-4,5379	2,6253	47,38574	-43,4827	109,9389	113,842	38,8915
2	0,2	-1,4334	5,9532	38,32243	-37,4348	149,9809	150,8685	52,6838
3	0,8	-3,5868	2,7517	36,28079	-35,3497	81,74795	82,67908	33,6644
4	0,7	-2,3108	4,4616	32,56567	-32,7799	100,9827	100,7685	33,0261
5	0,3	-4,6731	4,0999	56,13964	-51,6673	154,5882	159,0605	52,8648
6	**0,5**	**-3,3079**	**2,4874**	**32,38616**	**-31,9835**	**68,51744**	**68,92015**	**25,3264**
7	0,1	-3,2754	3,8513	37,95976	-37,6085	102,243	102,5943	32,8815

8	1	-4,8996	5,6096	70,85289	-65,3448	221,8948	227,4028	70,8529
9	0,6	-1,714	7,9798	64,5114	-64,1436	266,46	266,8278	79,1705
10	0,9	-2,8058	2,8641	28,57058	-28,7271	64,30233	64,14584	27,4921

F. KASUS 6

F.1 Bentuk permasalahan kasus 6

Minimumkan $f_1(x,y), f_2(x,y), f_3(x,y)$:

$$f_1(x,y) = 5 + (x-2)^2 + (y-1)^2$$
$$f_2(x,y) = 2x - (y-1)^2$$
$$f_3(x,y) = x^3 + 5y^2$$
$$f_4(x,y) = 2y - x^2 + (x+3)$$
$$f_5(x,y) = 4\log(x^2+17) - (14-y)^3 + 5$$

Dengan kendala:

$$(x-1)^2 + (y-1)^2 - 15 \le 0$$
$$(x+1)^2 + (y+1)^2 - 15 \le 0$$
$$(x+1)^2 + (y-1)^2 - 15 \le 0$$
$$(x-1)^2 + (y+1)^2 - 15 \le 0$$

Batas pencarian: $-15 \le x \le 15$ dan $-15 \le y \le 15$

Jumlah *bat* = 15 *bat*

Iterasi maksimal = 100 iterasi

Nilai pareto = 10 pareto

F.2 Hasil dari perhitungan kasus 6

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	$f_3(x,y)$	$f_4(x,y)$	$f_5(x,y)$	f_{total}	$f_{total(bobot)}$
1	1	1,8238	0,0927	5,81764	2,82448	6,2814	1,683	-2679,6	-2662,99	5,8542
2	0,7	0,83796	-2,158	13,405	-8,2983	117,03	-1,181	-4208,71	-4087,75	-302,657
3	0,5	0,31208	-1,509	6,49009	-5,6749	57,018	0,1951	-3721,01	-3662,98	-456,51
4	0,4	1,8597	-0,537	7,36111	1,35552	13,654	0,3262	-3062,09	-3039,39	-453,902
5	0,9	-0,2672	-2,567	6,07581	-13,263	164,82	-2,474	-4537,8	-4382,64	-92,2722
6	0,2	0,92399	-2,104	13,3878	-7,7856	111,43	-1,137	-4166,23	-4050,33	-825,989
7	0,6	-0,7884	-1,833	-8,6529	-9,6044	83,535	-2,077	-3959,32	-3896,12	-382,338
8	**0,1**	**-1,4379**	**-1,738**	**-28,138**	**-10,371**	**72,517**	**-3,981**	**-3887,72**	**-3857,69**	**-871,303**
9	0,3	-1,5208	-1,864	-30,438	-11,247	83,401	-4,563	-3982,75	-3945,59	-688,469
10	0,8	0,63686	-1,651	9,49541	-5,7546	68,412	-0,071	-3823,88	-3751,8	-179,359

G. UJI PARAMETER PADA KASUS 1

G.1 Percobaan 1

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

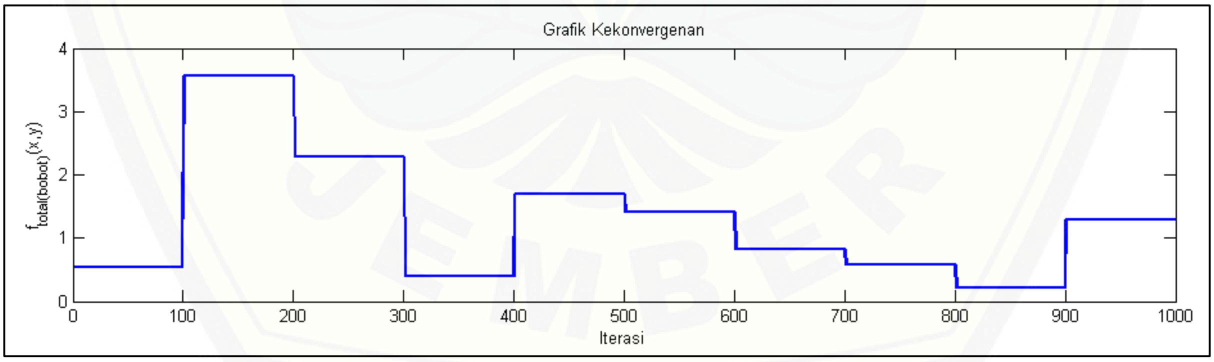
Pulse : 0,5

Loudness : 0,5

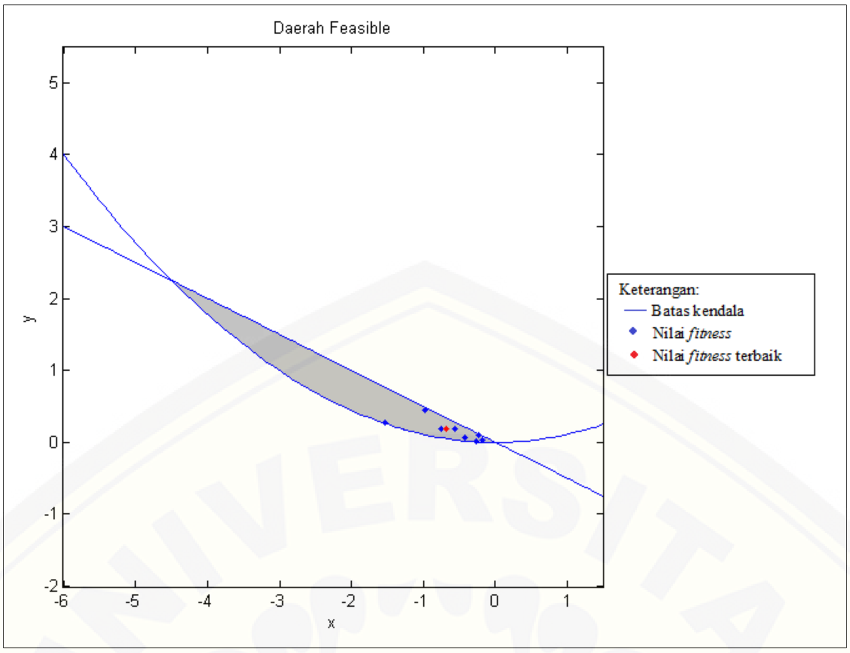
G.1.1 Tabel percobaan 1

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,5	-0,1748	0,04116	0,129054	0,760241	0,889295	0,5395
2	0,1	-1,5332	0,27927	9,714776	76,02531	85,74009	3,5547
3	0,3	-0,9805	0,45327	4,667493	13,65596	18,32345	2,2825
4	0,8	-0,2393	0,09661	0,266465	0,547406	0,813871	0,38785
5	0,4	-0,7504	0,18891	2,395269	1,982462	4,37773	1,6907
6	1	-0,5673	0,1818	1,419341	0,208898	1,628238	1,4194
7	0,7	-0,4236	0,07354	0,739349	0,073348	0,812697	0,82887
8	0,6	-0,2643	0,01925	0,280943	0,517414	0,798357	0,58126
9	**0,9**	**-0,1748**	**0,03224**	**0,126434**	**0,764157**	**0,890591**	**0,21051**
10	0,2	-0,6764	0,18545	1,967797	0,971023	2,93882	1,2904

G.1.2 Grafik kekonvergenan percobaan 1



G.1.3 Daerah *feasible* percobaan 1



G.2 Percobaan 2

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 150 iterasi

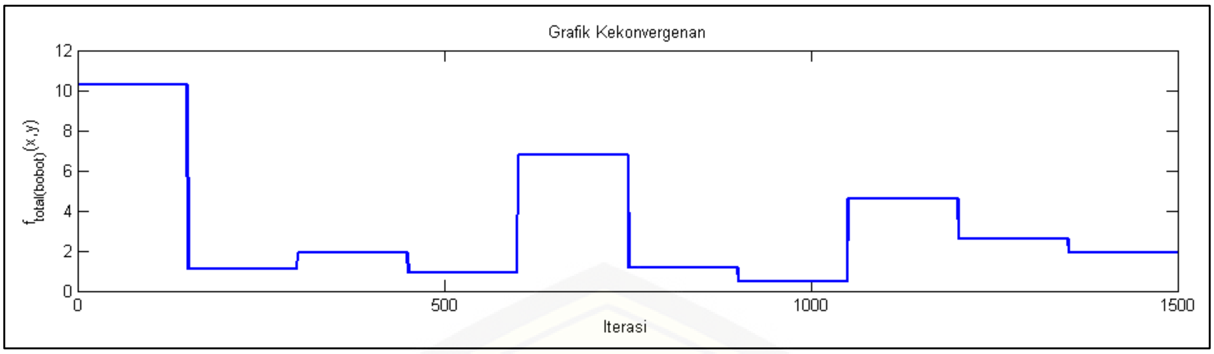
Pulse : 0,5

Loudness : 0,5

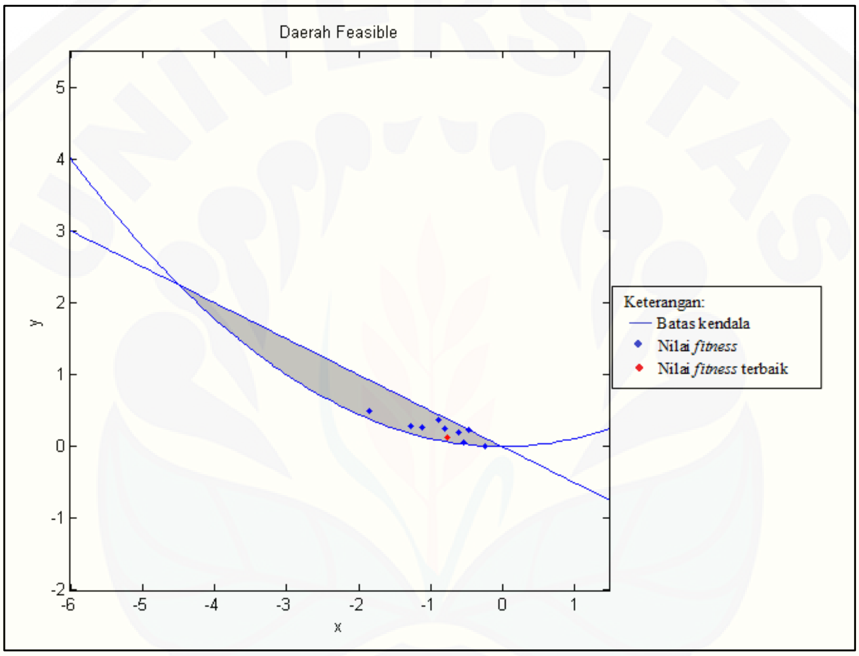
G.2.1 Tabel percobaan 2

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,6	-1,8548	0,48738	14,71129	3,703062	18,41435	10,3085
2	0,1	-0,61127	0,18944	1,638154	1,030659	2,668813	1,0914
3	0,3	-0,88647	0,37963	3,719792	1,170688	4,89048	1,9354
4	0,4	-0,4682	0,22688	1,082743	0,816926	1,899669	0,92326
5	1	-1,2732	0,28419	6,807209	2,133422	8,940631	6,8073
6	0,2	-0,54402	0,061441	1,198931	1,176851	2,375782	1,1813
7	**0,7**	**-0,23526**	**0,011156**	**0,221887**	**1,03316**	**1,255047**	**0,46527**
8	0,8	-1,1203	0,25994	5,290564	1,802761	7,093324	4,5927
9	0,9	-0,79201	0,25192	2,762974	1,186904	3,949878	2,6054
10	0,5	-0,77074	0,1224	2,436088	1,364222	3,80031	1,9001

G.2.2 Grafik kekonvergenan percobaan 2



G.2.3 Daerah *feasible* percobaan 2



G.3 Percobaan 3

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 300 iterasi

Pulse : 0,5

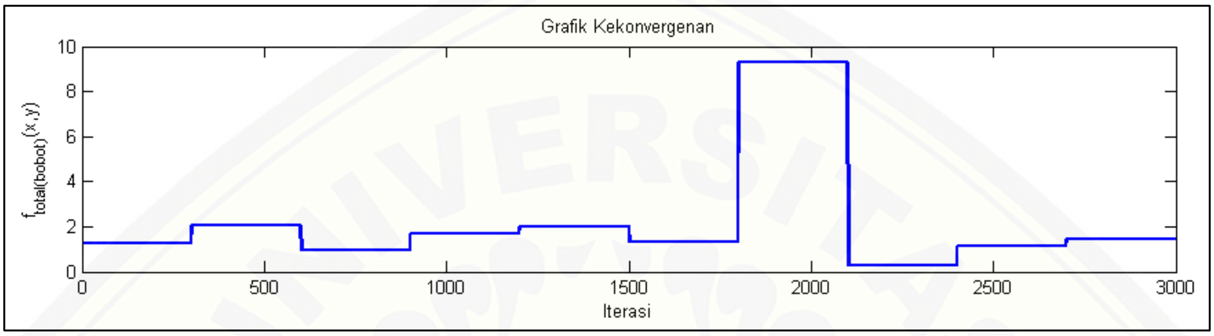
Loudness : 0,5

G.3.1 Tabel percobaan 3

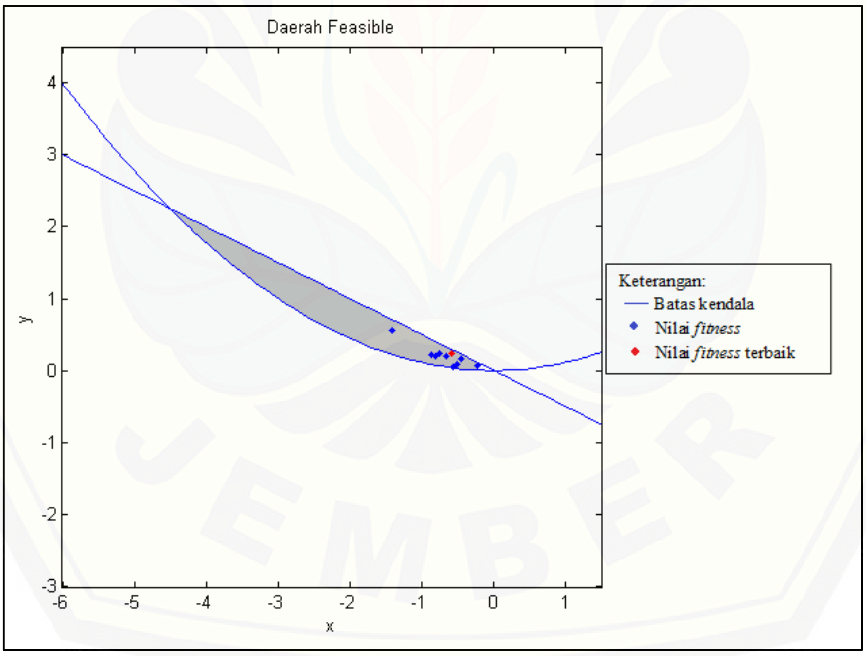
Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,6	-0,56482	0,046181	1,284617	1,228792	2,51341	1,2623
2	0,4	-0,85726	0,21474	3,124032	1,351528	4,47556	2,0605
3	0,3	-0,45787	0,15581	0,935687	0,922302	1,857988	0,92631

4	0,8	-0,65056	0,20432	1,8599	1,056335	2,916235	1,6992
5	0,5	-0,80455	0,19332	2,738693	1,298033	4,036727	2,0184
6	0,1	-0,75202	0,23098	2,475543	1,156926	3,632469	1,2888
7	1	-1,4148	0,56547	9,285661	2,190475	11,47614	9,2856
8	**0,9**	**-0,21744**	**0,061821**	**0,204408**	**0,92746**	**1,131868**	**0,27671**
9	0,2	-0,51569	0,079297	1,088897	1,11363	2,202527	1,1087
10	0,7	-0,59055	0,24185	1,628963	0,923541	2,552504	1,4173

G.3.2 Grafik kekonvergenan percobaan 3



G.3.3 Daerah *feasible* percobaan



G.4 Percobaan 4

Pareto : 10 pareto

Bat : 25 *bat*

Iterasi : 100 iterasi

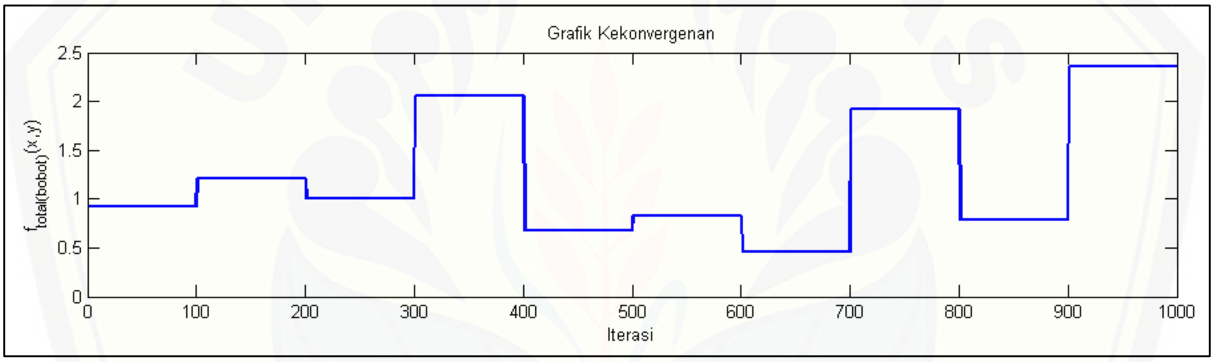
Pulse : 0,5

Loudness : 0,5

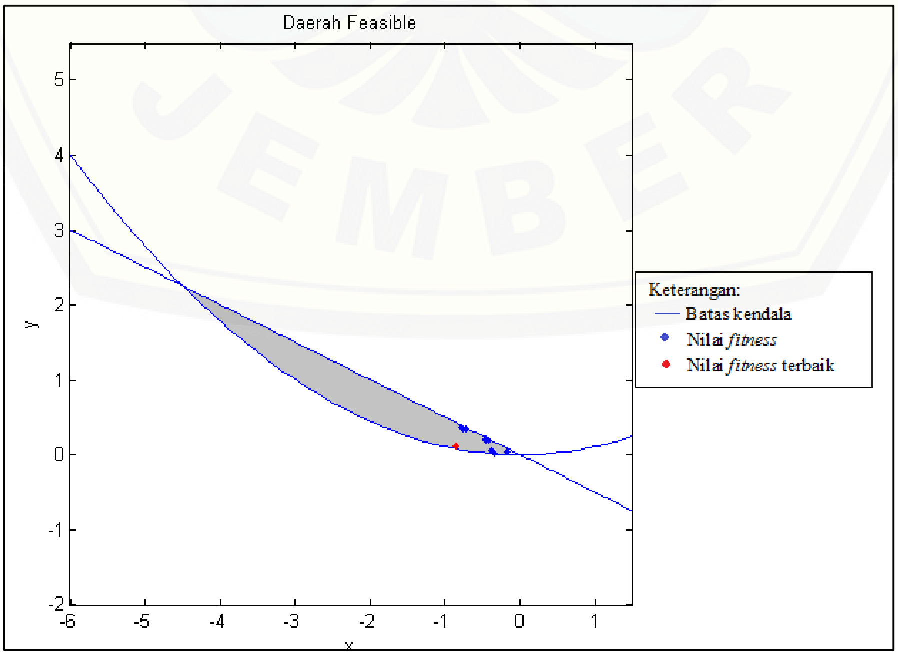
G.4.1 Tabel percobaan 4

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,4	-0,46341	0,20096	1,020535	0,853214	1,873749	0,92014
2	0,1	-0,78566	0,37096	3,019492	1,012953	4,032445	1,2136
3	0,9	-0,45967	0,20714	1,016814	0,839923	1,856737	0,99913
4	0,7	-0,71931	0,3366	2,522826	0,957506	3,480332	2,0532
5	0,8	-0,38123	0,062118	0,59678	1,024959	1,621739	0,68241
6	0,3	-0,41368	0,19825	0,841737	0,813934	1,655671	0,82228
7	**1**	**-0,33946**	**0,028223**	**0,464119**	**1,059584**	**1,523702**	**0,46413**
8	0,5	-0,76394	0,34907	2,821817	1,007314	3,829131	1,9146
9	0,2	-0,18115	0,044067	0,139029	0,946623	1,085652	0,78511
10	0,6	-0,8487	0,12421	2,942879	1,4873	4,430179	2,3607

G.4.2 Grafik kekonvergenan percobaan 4



G.4.3 Daerah *feasible* percobaan 4



G.5 Percobaan 5

Pareto : 10 pareto

Bat : 45 *bat*

Iterasi : 100 iterasi

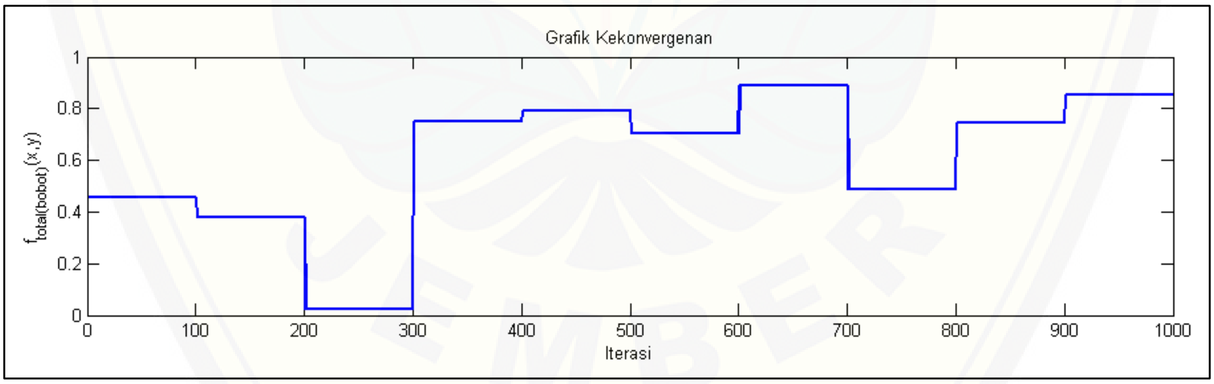
Pulse : 0,5

Loudness : 0,5

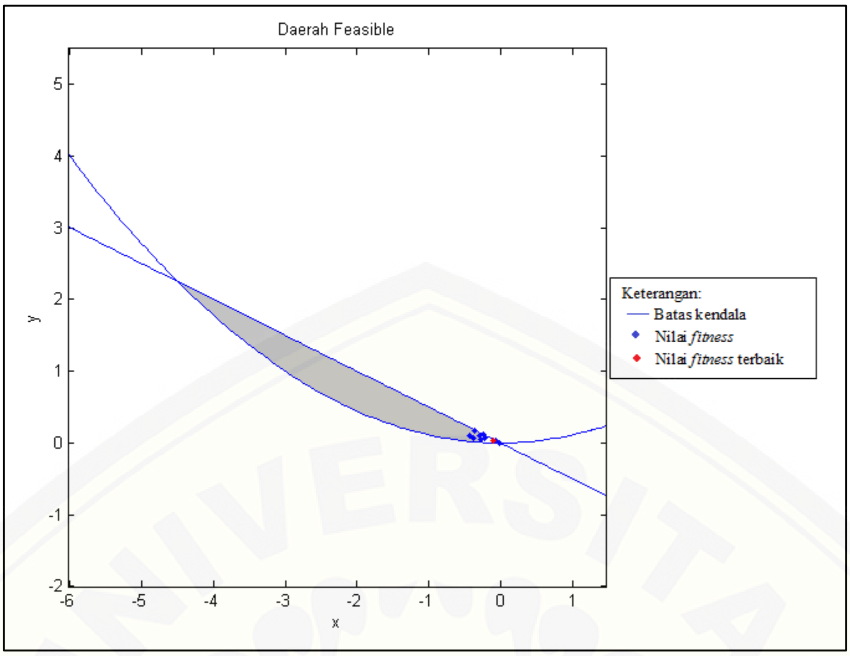
G.5.1 Tabel percobaan

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,7	-0,24356	0,12038	0,295251	0,833053	1,128304	0,45658
2	0,9	-0,2759	0,047762	0,313608	0,982878	1,296486	0,38053
3	**1**	**-0,07429**	**0,026038**	**0,024789**	**0,954121**	**0,97891**	**0,024789**
4	0,6	-0,37205	0,064375	0,570261	1,013815	1,584077	0,74769
5	0,2	-0,01874	0,004588	0,001489	0,991197	0,992686	0,79326
6	0,3	-0,21895	0,070187	0,211461	0,912491	1,123953	0,70218
7	0,5	-0,43151	0,10058	0,785269	0,995157	1,780426	0,89022
8	0,8	-0,29363	0,10716	0,390807	0,883382	1,274189	0,48932
9	0,4	-0,35892	0,16545	0,624789	0,825297	1,450086	0,7451
10	0,1	-0,09876	0,03419	0,043691	0,942543	0,986233	0,85266

G.5.2 Grafik kekonvergenan percobaan 5



G.5.3 Daerah *feasible* percobaan 5



G.6 Percobaan 6

Pareto : 25 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

Pulse rate : 0,5

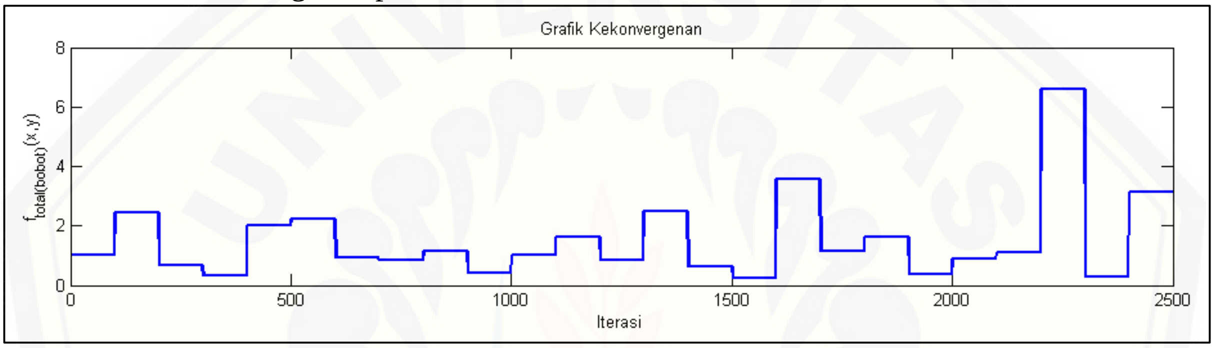
Loudness : 0,5

G.6.1 Tabel percobaan 6

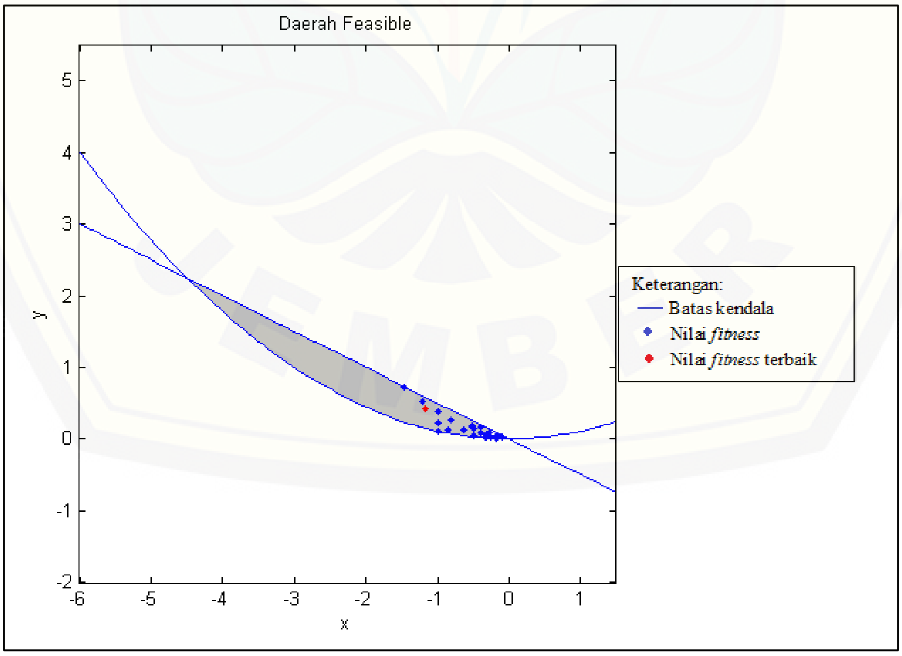
Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,04	-0,3395	0,02822	0,464119	1,059584	1,523702	1,0358
2	0,72	-0,8117	0,26699	2,920432	1,196128	4,11656	2,4376
3	0,48	-0,2854	0,07954	0,351165	0,928709	1,279875	0,65149
4	0,76	-0,1812	0,04407	0,139029	0,946623	1,085652	0,33285
5	0,12	-0,9901	0,11389	3,972917	1,765449	5,738367	2,0304
6	0,52	-0,8487	0,12421	2,942879	1,4873	4,430179	2,2442
7	0,2	-0,3165	0,03013	0,404244	1,040803	1,445047	0,91349
8	0,28	-0,3965	0,16119	0,732873	0,860838	1,593711	0,82501
9	0,68	-0,5228	0,17575	1,216873	0,952718	2,169592	1,1324
10	0,56	-0,0915	0,03941	0,039715	0,931115	0,970829	0,43193
11	0,64	-0,4962	0,15072	1,075605	0,967461	2,043066	1,0367
12	0,08	-0,9858	0,38354	4,475303	1,351746	5,827049	1,6016
13	0,44	-0,3894	0,08404	0,63487	0,990648	1,625518	0,8341
14	0,36	-0,9915	0,23605	4,155088	1,566672	5,72176	2,4985

15	0,6	-0,3131	0,06919	0,411249	0,964438	1,375687	0,63252
16	**0,88**	**-0,186**	**0,00574**	**0,138516**	**1,02315**	**1,161666**	**0,24467**
17	0,16	-1,4612	0,71427	10,58115	2,216747	12,7979	3,5551
18	0,8	-0,5166	0,19454	1,219051	0,915683	2,134734	1,1584
19	0,84	-0,6377	0,12955	1,693676	1,164319	2,857995	1,609
20	0,96	-0,2717	0,08854	0,326641	0,90458	1,231221	0,34976
21	0,24	-0,3207	0,02356	0,413666	1,056291	1,469957	0,90206
22	0,4	-0,5078	0,04777	1,040369	1,164548	2,204917	1,1149
23	0,92	-1,2176	0,52062	7,01438	1,712355	8,726735	6,5905
24	1	-0,2632	0,02677	0,280026	1,016476	1,296502	0,28003
25	0,32	-1,1689	0,41779	6,163503	1,705296	7,868798	3,1321

G.6.2 Grafik kekonvergenan percobaan 6



G.6.3 Daerah *feasible* percobaan 6



G.7 Percobaan 7

Pareto : 45 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

Pulse : 0,5

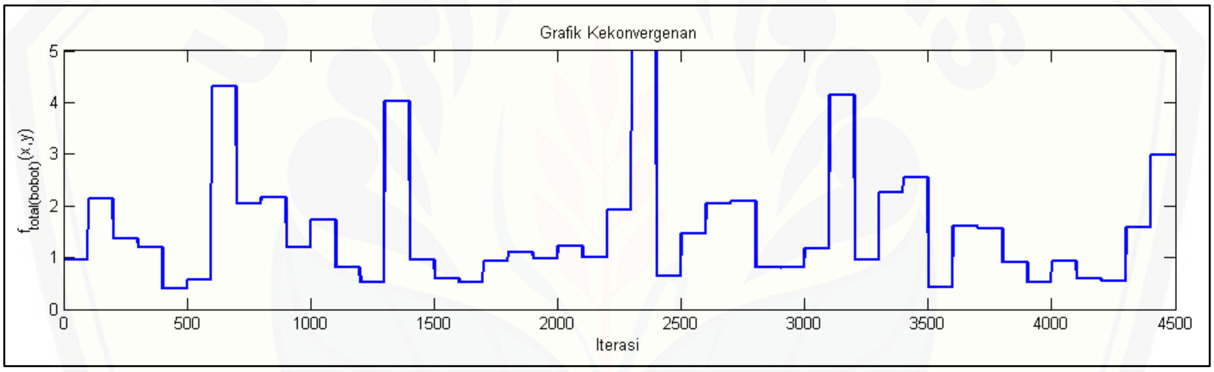
Loudness : 0,5

G.7.1 Tabel percobaan 7

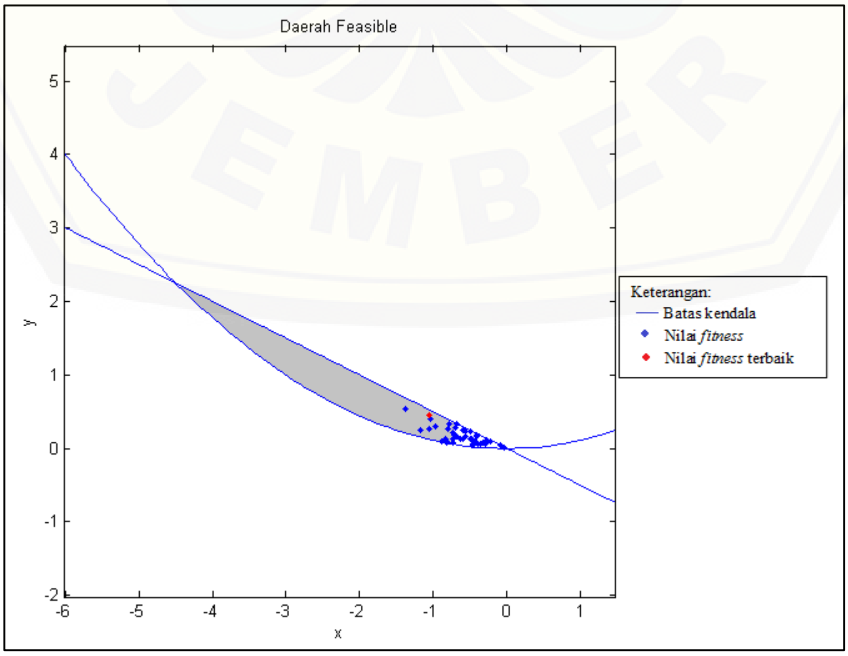
Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,0667	-0,5572	0,22493	1,444261	0,911205	2,355467	0,94674
2	0,5556	-0,80729	0,082015	2,633774	1,494414	4,128188	2,1274
3	0,2	-0,72587	0,21383	2,290442	1,144951	3,435393	1,3741
4	0,6222	-0,55124	0,16282	1,321504	1,004736	2,326239	1,2018
5	**0,8444**	**-0,25811**	**0,087976**	**0,297442**	**0,898409**	**1,195851**	**0,39092**
6	0,5778	-0,27676	0,10552	0,350922	0,876691	1,227613	0,57291
7	0,3333	-1,3742	0,54236	8,73032	2,09786	10,82818	4,3085
8	0,4667	-0,81325	0,094147	2,680957	1,481945	4,162902	2,0415
9	0,9778	-0,73333	0,079633	2,176457	1,384848	3,561306	2,1589
10	0,1556	-0,69621	0,28718	2,268723	0,992821	3,261544	1,1913
11	0,7556	-0,67321	0,16003	1,915285	1,158761	3,074047	1,7304
12	0,7333	-0,40525	0,17846	0,784302	0,839156	1,623458	0,79892
13	0,6444	-0,27104	0,10975	0,342031	0,866008	1,208039	0,52834
14	0,9556	-0,9749	0,29671	4,153867	1,445047	5,598914	4,0335
15	0,1778	-0,42641	0,094915	0,763337	1,001004	1,764342	0,95875
16	0,4222	-0,21519	0,10217	0,226982	0,852405	1,079387	0,58834
17	0,6889	-0,27692	0,10602	0,3517	0,875885	1,227585	0,51478
18	0,6	-0,45158	0,049132	0,825354	1,108074	1,933428	0,93845
19	0,2444	-0,59684	0,25632	1,687672	0,909278	2,59695	1,0995
20	0,2222	-0,47724	0,13693	0,986031	0,972648	1,958679	0,97562
21	0,0222	-0,71574	0,17755	2,175231	1,188708	3,363939	1,2106
22	0,8667	-0,48769	0,12812	1,017025	0,998016	2,015041	1,0145
23	0,7111	-0,72637	0,13228	2,180446	1,280551	3,460997	1,9205
24	0,8222	-1,1641	0,24165	5,654094	1,930224	7,584318	4,9917
25	0,3556	-0,0276	0,004509	0,003127	0,991765	0,994892	0,64025
26	1	-0,5877	0,133	1,452321	1,09708	2,549401	1,4523
27	0,1333	-1,0481	0,27167	4,689273	1,628978	6,318251	2,0372
28	0,3111	-0,88219	0,096027	3,149922	1,595426	4,745348	2,079
29	0,4444	-0,40124	0,15614	0,741493	0,873093	1,614586	0,8146
30	0,5333	-0,38549	0,068164	0,612995	1,016921	1,629916	0,8015
31	0,0444	-0,62093	0,12253	1,602271	1,155508	2,757778	1,1754

32	0,7778	-1,0298	0,40896	4,910945	1,409816	6,320762	4,133
33	0,5111	-0,45533	0,068081	0,847842	1,075798	1,92364	0,9593
34	0,9111	-0,73688	0,21419	2,355478	1,160489	3,515967	2,2493
35	0,8	-0,78345	0,33826	2,912855	1,051694	3,964549	2,5406
36	0,8889	-0,2916	0,057299	0,353255	0,973716	1,326971	0,4222
37	0,4889	-0,68225	0,33996	2,324151	0,901118	3,225269	1,5968
38	0,0889	-0,83601	0,13873	2,872635	1,440699	4,313334	1,568
39	0,1111	-0,32026	0,073866	0,432091	0,960291	1,392381	0,9016
40	0,6667	-0,27637	0,1112	0,354983	0,866346	1,221329	0,52543
41	0,2889	-0,48788	0,23801	1,178703	0,818656	1,997358	0,92267
42	0,3778	-0,08893	0,037672	0,037307	0,933983	0,97129	0,59524
43	0,9333	-0,34932	0,069971	0,507682	0,986978	1,49466	0,53964
44	0,2667	-0,78854	0,26461	2,767255	1,162594	3,929849	1,5905
45	0,4	-1,0598	0,44641	5,289832	1,429638	6,71947	2,9736

G.7.2 Grafik kekonvergenan percobaan 7



G.7.3 Daerah *feasible* percobaan 7



G.8 Percobaan 8

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

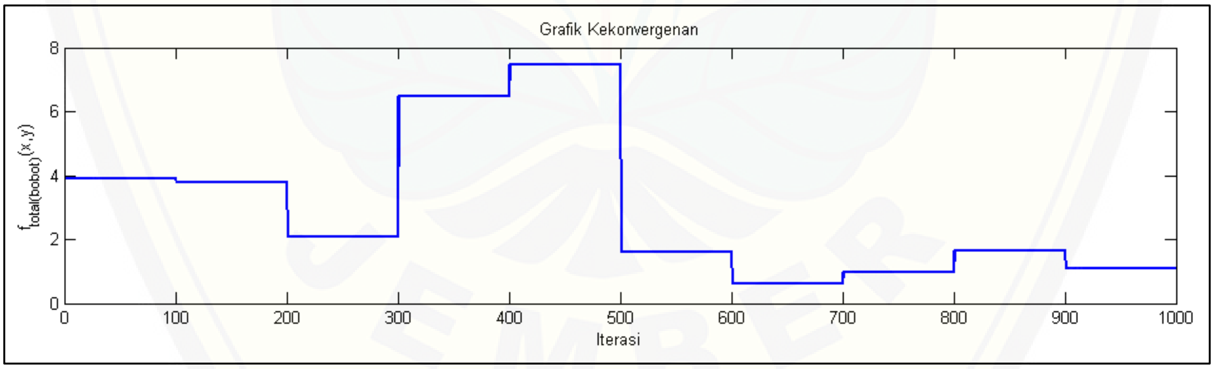
Pulse : 0,9

Loudness : 0,9

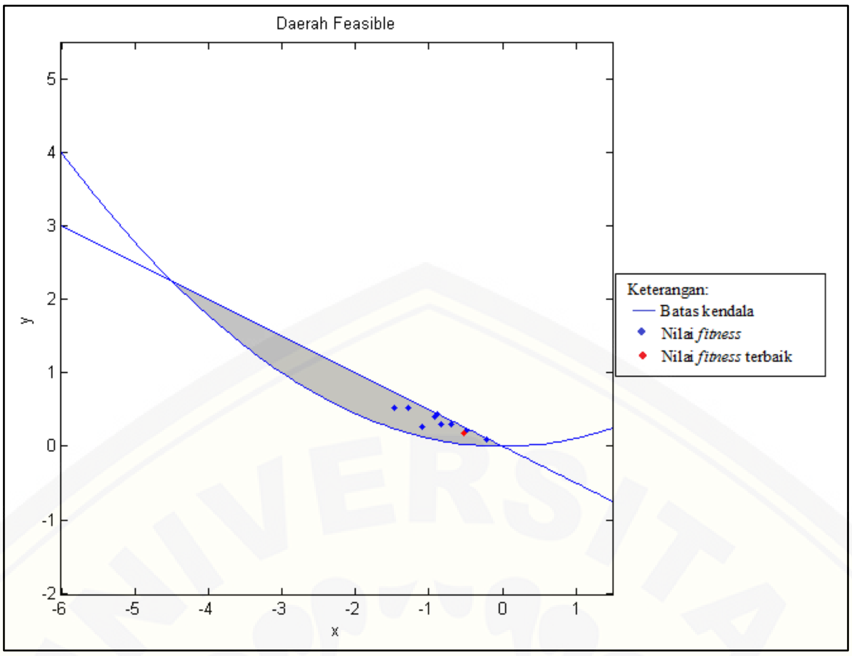
G.8.1 Tabel percobaan 8

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	1	-0,88216	0,43931	3,884798	1,09258	4,977378	3,8848
2	0,9	-0,92011	0,40695	4,048843	1,198311	5,247154	3,7638
3	0,1	-1,0991	0,27486	5,134275	1,733849	6,868124	2,0738
4	0,8	-1,274	0,5315	7,622273	1,842568	9,464841	6,4664
5	0,7	-1,4632	0,52671	9,673511	2,364958	12,03847	7,4811
6	0,2	-0,84167	0,30938	3,216497	1,185364	4,401862	1,5916
7	**0,4**	**-0,21983**	**0,094612**	**0,229107**	**0,868053**	**1,097159**	**0,61247**
8	0,3	-0,49594	0,20911	1,158734	0,871463	2,030197	0,95764
9	0,5	-0,69422	0,29986	2,28743	0,972137	3,259567	1,6298
10	0,6	-0,51928	0,18697	1,218438	0,930669	2,149107	1,1033

G.8.2 Grafik kekonvergenan percobaan 8



G.8.3 Daerah *feasible* percobaan 8



G.9 Percobaan 9

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

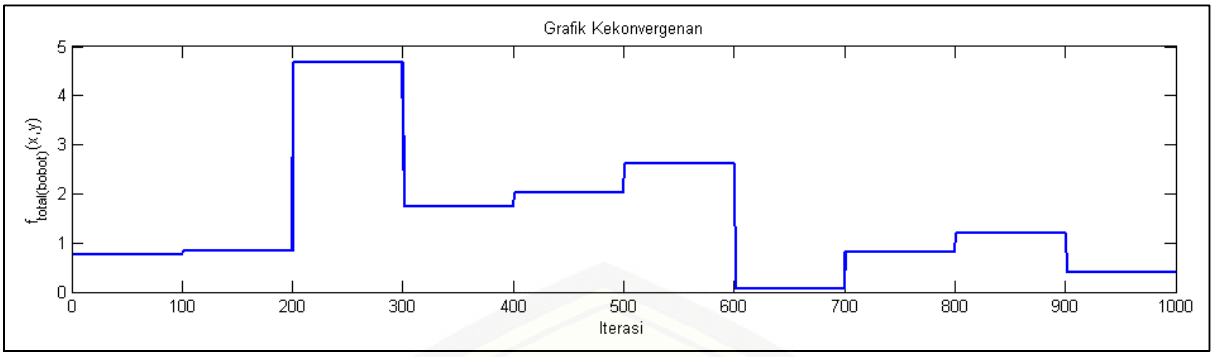
Pulse : 0,1

Loudness : 0,1

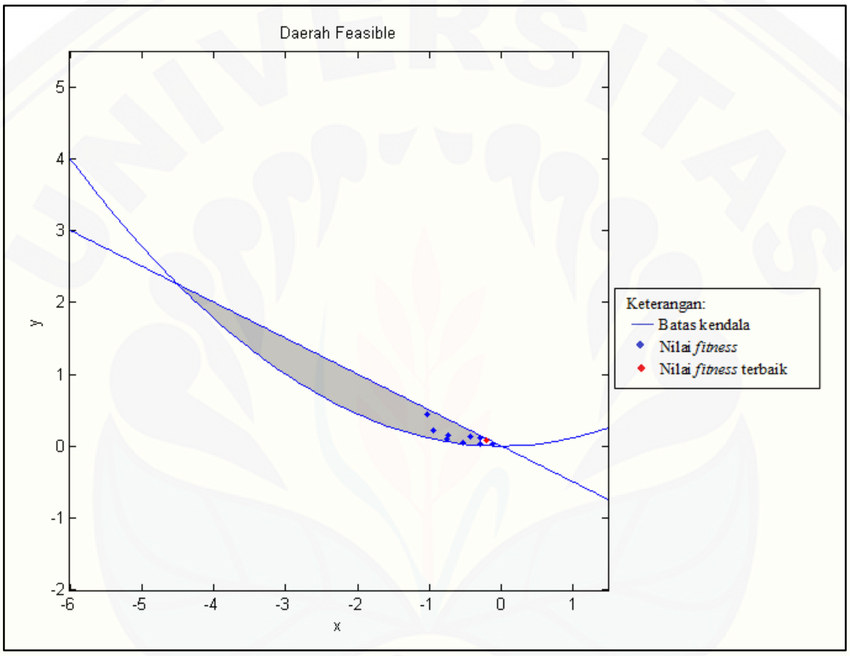
G.9.1 Tabel percobaan 9

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,2	-0,28526	0,12254	0,385557	0,851309	1,236867	0,75816
2	0,6	-0,42513	0,13989	0,801219	0,920525	1,721744	0,84895
3	0,9	-1,0273	0,45037	5,032714	1,357438	6,390152	4,6653
4	0,4	-0,74952	0,098894	2,286241	1,373772	3,660013	1,7388
5	0,8	-0,72913	0,15361	2,220906	1,248007	3,468913	2,0263
6	0,5	-0,93999	0,22554	3,737798	1,483369	5,221167	2,6106
7	**1**	**-0,1197**	**0,025736**	**0,059962**	**0,963518**	**1,02348**	**0,059959**
8	0,3	-0,29281	0,029802	0,346503	1,027022	1,373525	0,82287
9	0,1	-0,53066	0,045081	1,134529	1,19347	2,328	1,1876
10	0,7	-0,20693	0,082951	0,198804	0,883799	1,082602	0,4043

G.9.2 Grafik kekonvergenan percobaan 9



G.9.3 Daerah *feasible* percobaan 9



G.10 Percobaan 10

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

Pulse : 0,5

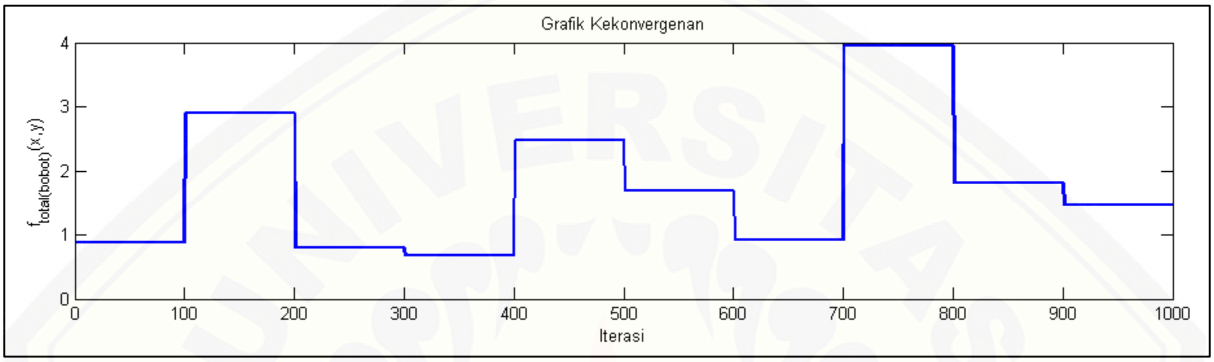
Loudness : 0,1

G.10.1 Tabel percobaan 10

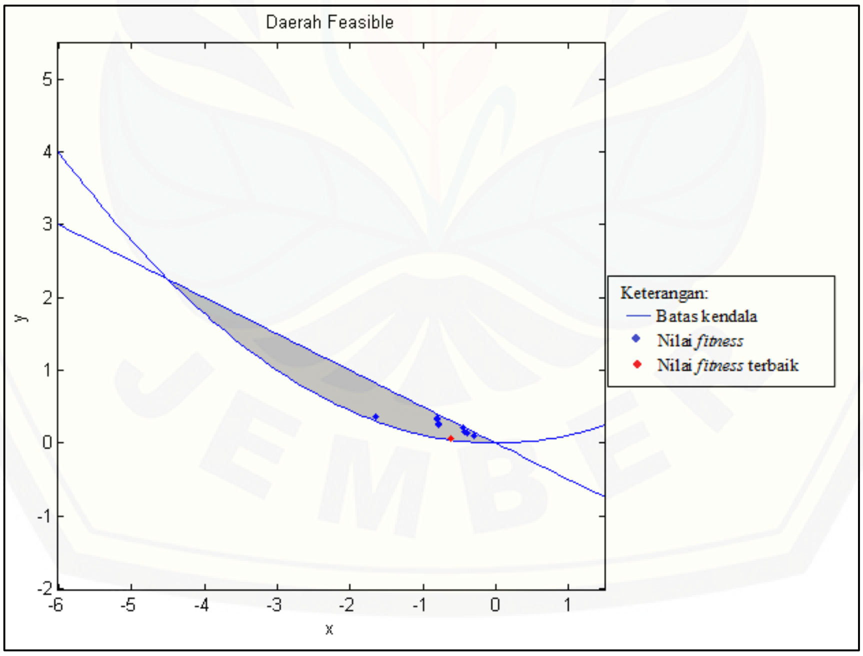
Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,5	-0,4358	0,15813	0,859742	0,898675	1,758417	0,8792
2	0,9	-0,8071	0,34943	3,094241	1,0747	4,168941	2,8923
3	0,2	-0,3048	0,10923	0,419337	0,886374	1,305711	0,79297

4	**1**	**-0,3861**	**0,14228**	**0,677298**	**0,884765**	**1,562063**	**0,67729**
5	0,8	-0,7952	0,26376	2,807331	1,174313	3,981644	2,4807
6	0,3	-0,8067	0,32843	3,034396	1,101739	4,136134	1,6815
7	0,6	-0,4497	0,20698	0,980139	0,831075	1,811214	0,92051
8	0,1	-1,6488	0,37357	11,43238	3,110956	14,54334	3,9432
9	0,4	-0,7898	0,24499	2,73509	1,193793	3,928883	1,8103
10	0,7	-0,6183	0,06123	1,544224	1,2636	2,807825	1,46

G.10.2 Grafik kekonvergenan percobaan 10



G.10.3 Daerah *feasible* percobaan 10



G.11 Percobaan 11

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

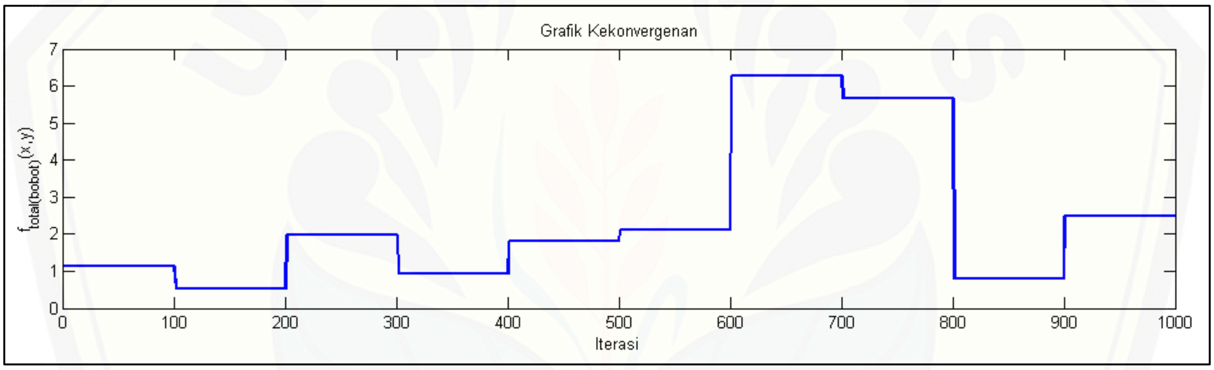
Pulse : 0,5

Loudness : 0,9

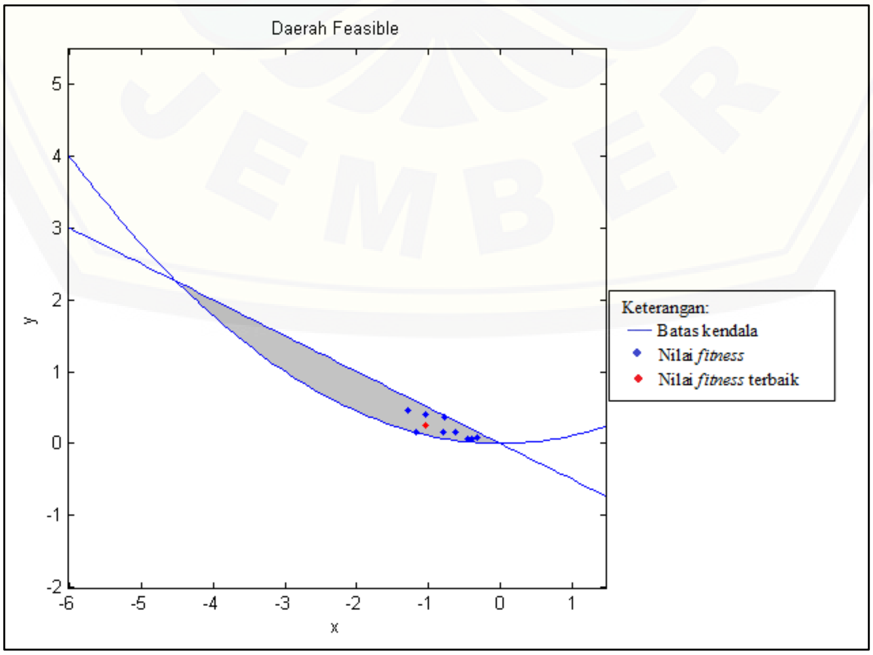
G.11.1 Tabel percobaan 11

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,1	-0,62498	0,16587	1,672451	1,086373	2,758824	1,145
2	**0,9**	**-0,32946**	**0,076637**	**0,457668**	**0,961143**	**1,418812**	**0,50801**
3	0,5	-0,78979	0,16046	2,598063	1,328596	3,926658	1,9633
4	0,7	-0,45539	0,058487	0,843203	1,093827	1,93703	0,91838
5	0,4	-0,7827	0,37042	2,999321	1,00899	4,008311	1,8051
6	0,2	-1,0319	0,4053	4,916343	1,418486	6,334829	2,1181
7	0,8	-1,2787	0,45487	7,367922	1,93224	9,300162	6,2809
8	1	-1,1796	0,15686	5,664245	2,102341	7,766586	5,6645
9	0,6	-0,40153	0,069797	0,664392	1,026504	1,690896	0,80924
10	0,3	-1,0334	0,26111	4,544376	1,613874	6,15825	2,4929

G.11.2 Grafik kekonvergenan percobaan 11



G.11.3 Daerah *feasible* percobaan 11



G.12 Percobaan 12

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

Pulse : 0,5

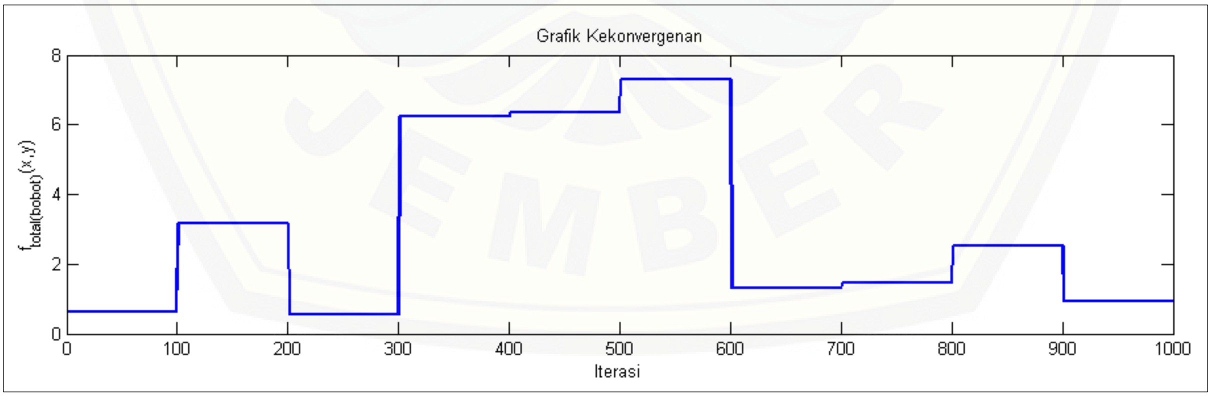
Loudness : 0,9

f_{maks} : 1

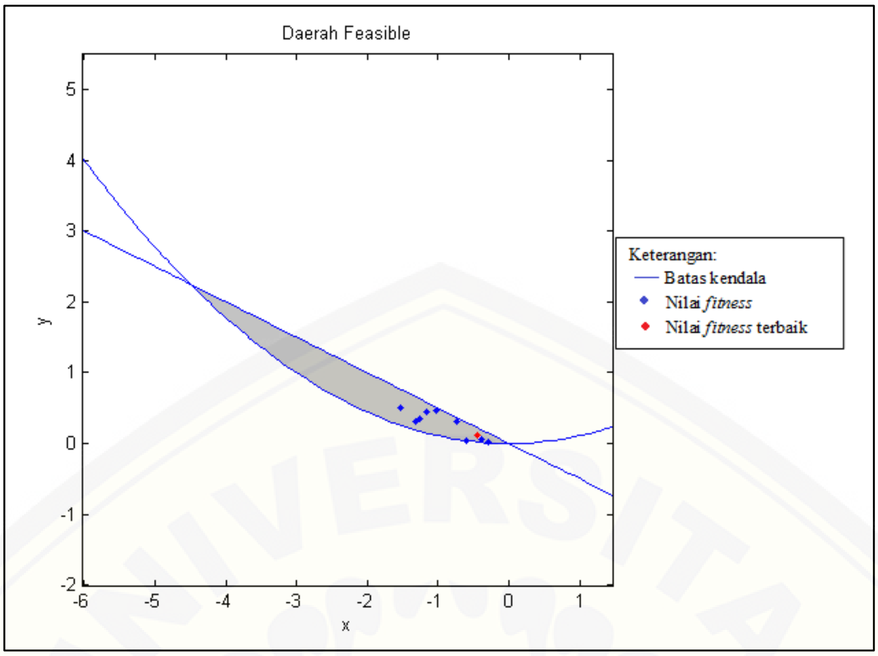
G.12.1 Tabel percobaan 12

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	1	-0,38904	0,059164	0,61941	1,036525	1,655935	0,61942
2	0,5	-1,0179	0,47433	5,044437	1,312449	6,356887	3,1785
3	**0,7**	**-0,28798**	**0,017557**	**0,332963**	**1,048127**	**1,38109**	**0,54752**
4	0,8	-1,312	0,30666	7,261537	2,202064	9,463602	6,2498
5	0,9	-1,2583	0,35524	6,838057	1,999034	8,837092	6,3541
6	0,6	-1,5313	0,51074	10,42294	2,584255	13,0072	7,2875
7	0,1	-0,60043	0,041748	1,449036	1,278763	2,727799	1,2958
8	0,3	-0,73369	0,30695	2,530077	1,018619	3,548697	1,4721
9	0,2	-1,1514	0,44146	6,082436	1,637689	7,720124	2,5267
10	0,4	-0,4483	0,12393	0,865326	0,968472	1,833798	0,92721

G.12.2 Grafik kekonvergenan percobaan 12



G.12.3 Daerah *feasible* percobaan 12



G.13 Percobaan 13

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

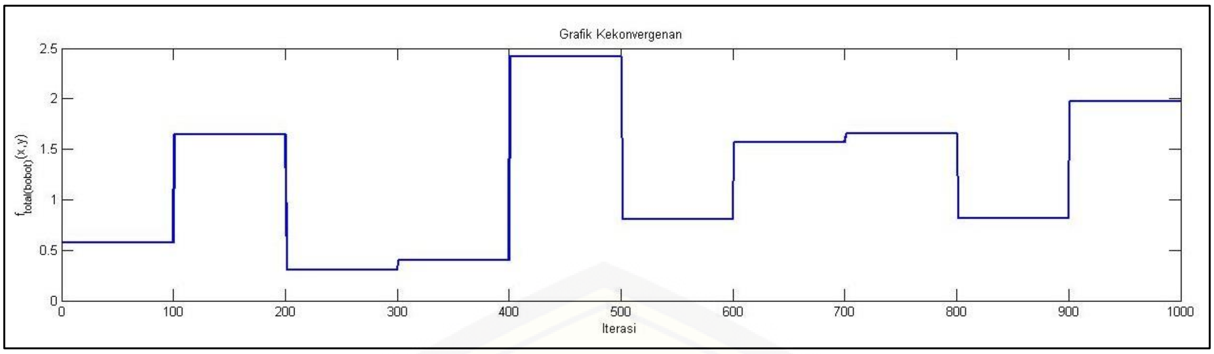
Pulse : 0,01

Loudness : 0,1

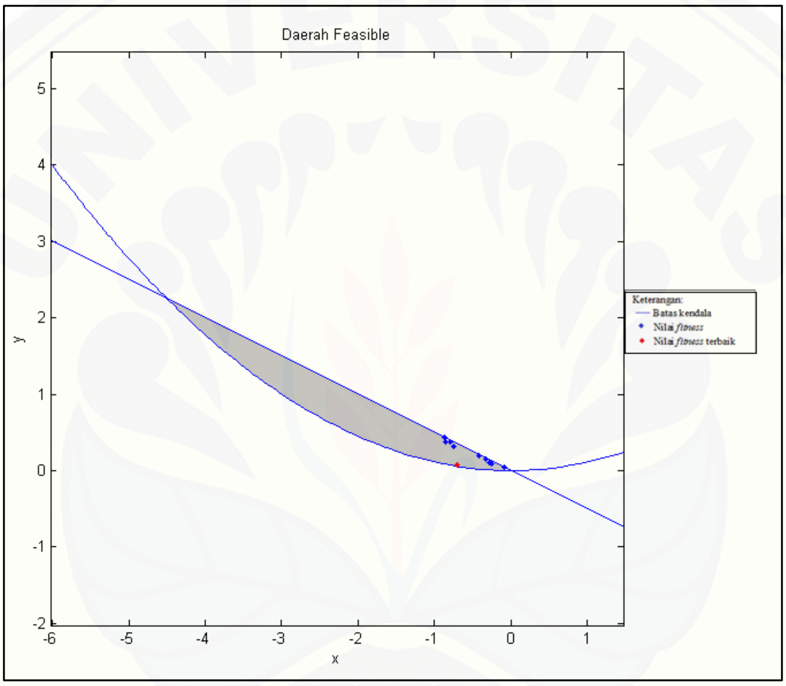
G.13.1 Tabel percobaan 13

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,6	-0,28264	0,1036	1,449893	0,883418	2,333312	0,57085
2	0,3	-0,79503	0,3817	12,44428	1,014368	13,45865	1,6434
3	**0,7**	**-0,0804**	**0,039049**	**0,127824**	**0,929891**	**1,057715**	**0,30132**
4	0,8	-0,24486	0,094094	1,100962	0,880622	1,981584	0,39632
5	0,5	-0,86883	0,43108	15,05113	1,078536	16,12966	2,4207
6	0,1	-0,32637	0,14393	2,035732	0,839373	2,875105	0,80633
7	0,2	-0,84616	0,37223	13,67267	1,110082	14,78275	1,5717
8	0,4	-0,7429	0,32165	10,48575	1,012059	11,49781	1,6558
9	0,9	-0,40697	0,19902	3,283737	0,807194	4,09093	0,81958
10	1	-0,69851	0,07973	7,908369	1,334813	9,243183	1,9771

G.13.2 Grafik kekonvergenan percobaan 13



G.13.3 Daerah *feasible* percobaan 13



G.14 Percobaan 14

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

Pulse : 0,001

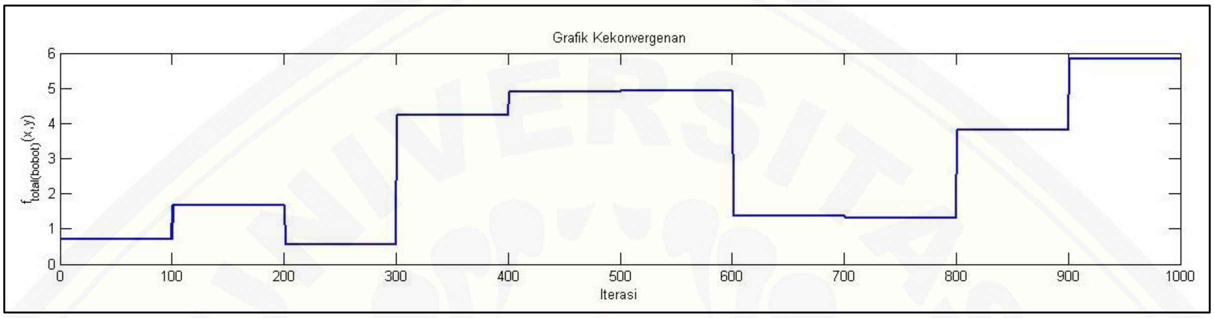
Loudness : 0,1

G.14.1 Tabel percobaan 14

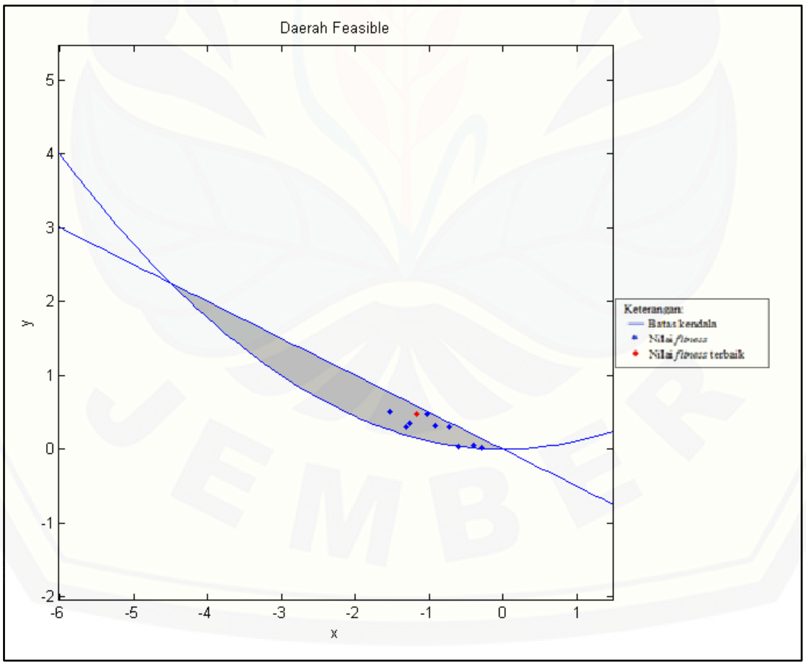
Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,8	-0,3890	0,059164	2.4776	1.0365	3.51416	0,70284
2	0,1	-1,0179	0,47433	20.177	1.3124	21.49	1,6857

3	**0,7**	**-0,2879**	**0,017557**	**1.33185**	**1.0481**	**2.3799**	**0,54752**
4	0,4	-1,312	0,30666	29.046	2.202	31.2482	4,226
5	0,6	-1,2583	0,35524	27.352	1.999	29.351	4,9024
6	0,3	-1,5313	0,51074	41.691	2.5842	44.276	4,9359
7	0,5	-0,6004	0,04175	5.79614	1.2787	7.0749	1,3639
8	0,2	-0,7336	0,30695	10.1203	1.0186	11.1389	1,3209
9	1	-0,9181	0,32805	15.206	1.2943	16.5012	3,8017
10	0,9	-1,1628	0,47349	25.221	1.6293	26.85	5,8376

G.14.2 Grafik kekonvergenan percobaan 14



G.14.3 Daerah *feasible* percobaan 14



G.15 Percobaan 15

Pareto : 10 pareto

Bat : 15 *bat*

Iterasi : 100 iterasi

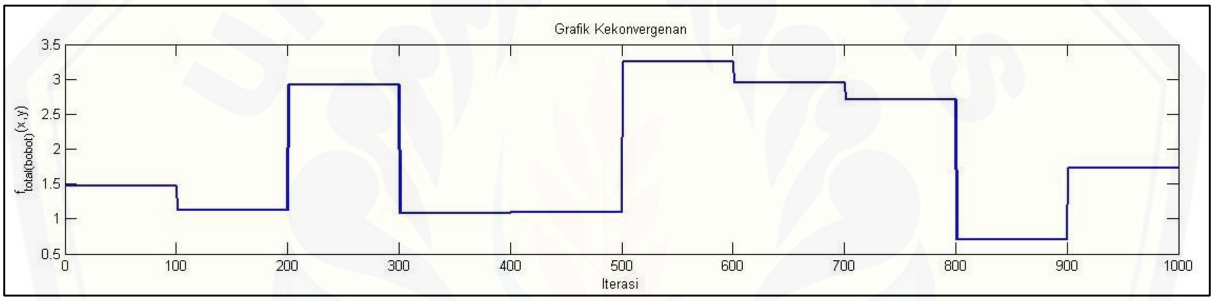
Pulse : 0,0001

Loudness : 0,01

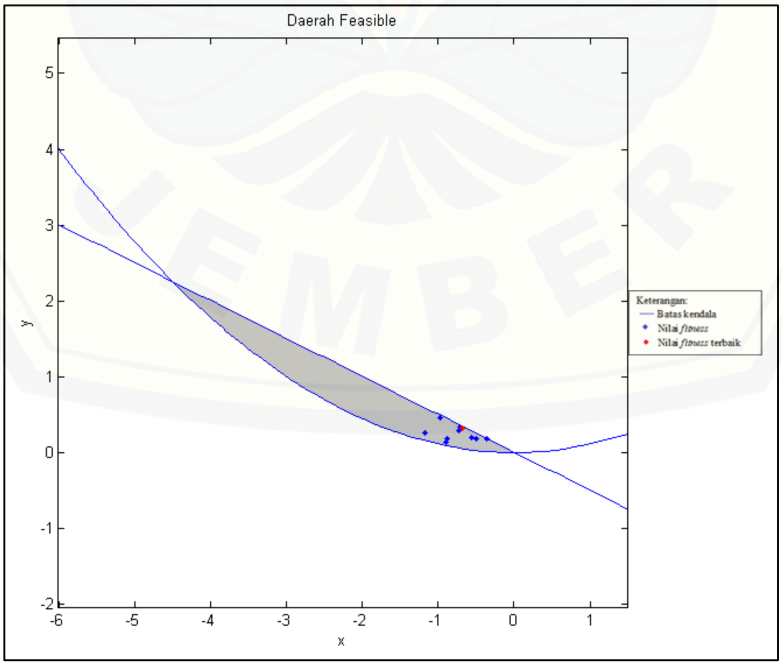
G.15.1 Tabel percobaan 15

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,3	-0,73219	0,28565	9.8831	1.0463	10.9295	1,4737
2	0,4	-0,55612	0,18937	5.5221	0.96639	6.4884	1,132
3	0,8	-0,88671	0,18717	13.1405	1.4469	14.5873	2,9175
4	0,9	-0,49363	0,1742	4.3842	0.9256	5.3098	1,079
5	0,1	-0,7082	0,32904	9.757	0.9517	10.708	1,1005
6	1	-0,89092	0,13656	12.998	1.5392	14.537	3,2496
7	0,5	-0,97529	0,45849	18.5824	1.2444	19.8268	2,945
8	0,2	-1,1751	0,25831	23.1613	1.9309	25.092	2,7029
9	**0,7**	**-0,36693**	**0,17577**	2.6485	0.8139	3.4625	**0,7077**
10	0,6	-0,67962	0,32454	9.0753	0.91812	9.9935	1,7286

G.15.2 Grafik kekonvergenan percobaan 15



G.15.3 Daerah *feasible* percobaan 15



H. TABEL NILAI TERBAIK DARI UJI PARAMETER PADA KASUS 3

Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	$f_{total}(x,y)$	$f_{bobot}(x,y)$	Pareto	Bat	Iterasi
0,06667	-8,0022	0,597	102,2064	-72,1822	30,0242	-60,5565	15	15	50
0,1	-7,9929	0,91334	101,8656	-71,9436	29,92195	-54,5624	10	20	100
0,06667	-7,8176	0,24141	98,96073	-70,9339	28,02687	-59,608	10	45	100

I. TABEL NILAI TERBAIK DARI UJI PARAMETER PADA KASUS 4

Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	$f_{total}(x,y)$	$f_{bobot}(x,y)$	Pareto	Bat	Iterasi
0,95	0,27171	-0,09341	1,320834	48,29957	49,62041	2,7287	20	35	300
0,06667	-0,2672	0,3041	0,655491	49,79487	50,45036	5,7669	15	35	100
1	-0,1219	-0,12164	0,118672	52,46557	52,58424	0,06	15	30	250
1	-0,1199	-0,12946	0,124572	52,52504	52,64962	0,12457	15	45	200
1	-0,2545	0,77602	2,668072	45,45262	48,12069	2,6681	15	10	300

J. HASIL DARI PERHITUNGAN MEMAKSIMUMKAN KASUS FUNGSI *MULTI-OBJECTIVE* BERKENDALA

Pareto	Bobot	x	y	$f_1(x,y)$	$f_2(x,y)$	f_{total}	$f_{total(bobot)}$
1	0,6	-3,7164	1,8032	-62,2526	-8,45676	-70,7094	-40,735
2	**0,1**	**-3,8858**	**1,8827**	**-68,576**	**-9,8786**	**-78,4546**	**-15,748**
3	1	-4,3006	2,0776	-85,2463	-13,6564	-98,9027	-85,248
4	0,9	-3,9901	1,9483	-72,8671	-10,8202	-83,6873	-66,664
5	0,4	-3,572	1,5255	-54,3453	-7,03533	-61,3807	-25,959
6	0,3	-4,4738	2,229	-93,9333	-15,5253	-109,459	-39,048
7	0,2	-4,2957	2,1386	-86,1066	-13,7494	-99,856	-28,222
8	0,8	-4,2367	2,0104	-81,9653	-12,9705	-94,9359	-68,165
9	0,5	-4,0077	1,9649	-73,69	-10,9927	-84,6827	-42,341
10	0,6	-3,7164	1,8032	-62,2526	-8,45676	-70,7094	-40,735