# OPTIMASI MODEL MULTIOBJEKTIF MENGGUNAKAN GABUNGAN ALGORITMA KELELAWAR DENGAN *DIFFERENTIAL EVOLUTION (DE)* DAN *LEVY FLIGHT TRAJECTORY*

Veri Julianto[1)]
*[1)] Dosen Jurusan Teknik Informatika Politeknik Negeri Tanah Laut, Banjarmasin*

## ABSTRACT

Optimization is a process to maximize or minimize a particular problem in order to obtain optimal conditions. The optimization problem has various model functions. There are linear, non-linear, multi modal, multiobjektif and others. In everyday reality there are many muliobjective problems, in which there is more than one model function that must be executed simultaneously in a multi-objective model to optimize one function again. This results in difficulties in optimizing it. In this research, metaheuristic method is used to solve multiobjective problems. This method is easy to use and does not require information of the model to be optimized. One metaheuristic method is the Bat (BA) Algorithm. In this study BA, will be in hybrids using the method of Differential Evolution (DE) that has long existed in finding the optimal solution. Both of these algorithms are combined with local search (Levy Flight Trajectory) in BA with DE. Thus, hybrids from BA and DE will be tested to complete multi-purpose test functions.
**Keywords**: *multiobjective, bat algorithm , differential evolution, levy flight trajectory*

## 1. PENDAHULUAN

Optimasi merupakan proses mengoptimalkan suatu fungsi objektif (*objective function*) dengan masih memperhitungkan batasan-batasan (*constraint*). Proses mengoptimalkan fungsi objektif dapat berupa memaksimalkan dan meminimalkan. Dalam proses optimasi sebuah fungsi tujuan diperlukan teknik agar diperoleh hasil optimal dan waktu yang cepat. Metode yang digunakan untuk mengoptimasi suatu fungsi tujuan dari berbagai permasalahan ada banyak tergantung pada kerumitan maslah yang dihadapi. Apabila kasus yang dihadapi masih tergolong sederhana maka dapat digunakan metode eksak, akan tetapi apabila kasusnya rumit maka diperlukan metode komputasi dalam menyelesaikannya. Metode Komputasi dalam menyelesaikan permasalahan komputasi cukup banyak. Metode *gradient* atau metode turunan banyak digunakan sebelum metode metaheuristik diperkenalkan. Banyak yang menggunakan metode *gradient* dalam menyelesaikan permasalahan optimasi seperti mencari akar, memaksimalkan dan meminimalkan fungsi dan lain-lain. Metode ini memiliki kelebihan dalam menyelesaikan fungsi objektif yang sederhana, akan tetapi kelemahan metode ini yaitu menghendaki turunan yang kontinyu, harus memiliki tebakan awal yang tepat dan terkadang terjebak pada kondisi maksimum atau minimum lokal. Oleh karena itu diperlukan metode lain untuk menyelesaikan kelemahan tersebut, salah satunya yaitu metode metaheuristik.

Metode metaheuristik merupakan metode optimasi yang dilakukan dengan memperbaiki kandidat penyelesaian secara iteratif berdasarkan dengan fungsi objektifnya (Talbi, 2009). Metode ini mampu menghasilkan penyelesaian yang baik dalam waktu yang cepat (*acceptable*), tetapi tidak menjamin bahwa penyelesaian yang dihasilkan merupakan penyelesaian yang terbaik. Metode ini lebih menekankan pada proses eksplorasi (pencarian global) dan eksploitasi (pencarian lokal). Beberapa contoh metode metaheuristik yaitu *Algoritma Genetika, Simulated Annealing (SA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO)* dan Algoritma Kelelawar.

Salah satu algoritma yang akan digunakan dalam penelitian ini yaitu algoritma kelelawar yang diperkenalkan oleh X.S Yang pada 2010. Algoritma ini di inspirasi oleh salah satu jenis kelelawar yaitu *microbat* dalam proses mencari makan. Algoritma kelelawar akan dikombinasikan dengan metode *Differential Evolution (DE)* untuk membantu menemukan solusi permasalahan. Kedua metode diatas juga akan dikombinasikan dengan *Levy Flight* yang membantu untuk mengatasi konvergensi yang terlalu cepat dan hanya mendapatkan minimum lokal. Metode ini sudah pernah berhasil diuji oleh Xie J (2013) dalam menyelesaikan persamaan non linier dengan hasil yang lebih bagus dari algoritma kelelawar. Pada penelitian ini algoritma kelelawar yang digabung dengan DE dan menggunakan *local search* Levy Flight dalam konsep eksploitasinya dalam menyelesaikan fungsi-fungsi multiobjektif. Proses ini akan dilihat *pareto front* yang dihasilkan dan dianalisa dengan menggunkan *true pareto front* sesuai dengan fungsinya.

---

[1] Koresponding : Veri Julianto, Telp 0811518669, verijulianto@gmail.com

## 2. METODE PENELITIAN

Proses penelitian ini dilakukan dengan berorientasi mendapatkan hasil sesuai dengan indicator keberhasilan. Untuk mencapai indikator tersebut, tahapa awal dalam penelitian ini ialah studi literatur yang berkaitan dengan *diferential equation*. *Differential Evolution* (DE) adalah sebuah metode optimasi yang dikembangkan oleh Kenneth Price dan dipublikasikan pada Oktober 1994 dalam majalah Dr. Dobb's Journal (Price et al., 2005). Metode ini merupakan metode optimasi matematis fungsi multidimensional dan termasuk dalam kelompok *evolutionary algorithm*. Sebuah populasi dapat diinisialisasi, *upper* dan *lower bounds* untuk setiap parameter harus ditentukan, yaitu dengan vector inisialisasi dimensi $b_L$dan $b_U$. L menunjukan *lower* dan U menunjukan *upper*. Berikutnya adalah membangitkan bilangan acak untuk setiap parameter $j$ dan vektor $\boldsymbol{i}$ pada integrasi $g$. Misal nilai inisial $(g = 0)$ :

$$x_{j,i,0} = r\quad d_j(0,1).\left(b_{j,U} - b_{j,L}\right) + b_{j,L}$$

Bilangan acak di atas dibangkitkan berasarkan distribusi uniform pada rentang $[0,1)$ atau $0 \leq r\quad d_j(0,1) < 1$ . Setelah diinisialisasi, DE akan memutasi dan me-rekombinasi populasi awal untuk menghasilkan populasi baru. Mutasi pada beberapa kamus bahasa menunjukkan pengertian berubah dan dalam konteks genetika mutasi berarti perubahan dengan elemen acak. Berikut ini adalah persamaan yang menunjukkan bagaimana membentuk vektor mutan, $v_{i,g}$:

$$\boldsymbol{v_{i,g}} = \boldsymbol{x_{r\ ,g}} + \boldsymbol{F}.\left(\boldsymbol{x_{r\ ,g}} - \boldsymbol{x_{r\ ,g}}\right)$$

Dimana $r0, r1, r2$ adalah indeks acak, integer, dan berbeda. Indeks basis vektor $r0$, dapat ditentukan dengan berbagai cara antara lain acak, permutasi, stokastik, dan acak offset. Sedangkan untuk $r1$ dan $r2$ dipilih secara acak sekali untuk setiap mutan. Untuk melengkapi strategi pencarian *differential mutation*, DE menggunakan *crossover* dengan tujuan meningkatkan diversitas parameter populasi. *Crossover* membangun vektor uji dari nilai parameter yang telah dikopi dari dua vektor yang berbeda. Persamaan untuk vektor uji adalah sebagai berikut:

$$u_{i,g+1} = \left(u_{1i,g+1}, u_{2i,g+1}, \dots, u_{n\ ,g+1}\right) \tag{1}$$

Dimana :

$$u_{i,g+1} = \begin{cases} v_{j,i,g+1}\ i\ (r\quad d_j(0,1) \leq C\quad o\quad j = j_r \\ \\ x_{j,i,g}\ y\qquad l\qquad\quad j = 1,2,3,\dots,n \end{cases} \tag{2}$$

Menurut Price et al. (2005), pada dasarnya ada dua tahapan dalam proses evolusi yang menggunakan seleksi yaitu *parent selection* dan *survivor selection*. Berikut ini adalah penjelasan mengenai kedua seleksi tersebut: 1) *parent selection,* vektor yang terpilih ditandai dengan nilai fungsi terbaik dan probabilitas seleksi tertinggi. Metode ini dalam memberikan probabilitas seleksi membutuhkan tambahan asumsi tentang bagaimana menggambarkan nilai fungsi tujuan menjadi probabilitas. 2) s*urvivor selection,* metode ini juga bias disebut *repelcment*. Untuk mengetahui apakah vector menjadi anggota generasi $g + 1$, maka vector uji $u_{i,g+1}$ dibandingkan dengan vector target $x_{i,g}$menggunakan kriteria *greedy.* Jika vektor $u_{i,g+1}$ menghasilkan fungsi biaya yang lebih kecil daripada $x_{i,g}$ maka $x_{i,g+1}$ akan diatur menjadi $u_{i,g+1}$, dan bila sebaliknya maka nilai $x_{i,g}$ yang lama dipertahankan. Apabila penjelasan diatas ditunjukan dalam persamaan, hasilnya dapat dilihat berikut ini.

$$x_{i,g+1} = \begin{cases} u_{i,g}\ if\ f\left(u_{i,g}\right) \leq f\left(x_{i,g}\right) \\ \\ x_{i,g}, y\qquad l \end{cases} \tag{3}$$

Studi literatur yang lain ialah yang berkaitan dengan algoritma kelelawar. Konsep dasar pengembangan algoritma kelelawar ialah semua kelelawar menggunakan kemampuan ekolokasi untuk mengetahui jarak, mereka juga dapat membedakan antara mangsa dan benda-benda di sekitar mereka. Kelelawar terbang secara acak dengan kecepatan $\mathbf{v_i}$ dan dengan posisi $\mathbf{x_i}$ dengan frekuensi tetap $f_m$ , dengan variasi panjang gelombang $\lambda$ dan kenyaringan $A_0$ saat mencari mangsa. Mereka secara otomatis dapat menyesuaikan panjang geombang dari sinyal yang merka pancarkan dan menyesuaikan tingkat sinyal

$r \in [0,1]$, dan tergantung pada target mereka. Tingkat kekerasan suara diasumsikan bervariasi dari $A_m$ hingga $A_0$.

Dalam jurnal yang ditulis oleh Yang (2010) membahas mengenai pergerakan kelelawar dalam menghasilkan sebuah algoritma sehingga dapat dijadikan sebagai metode pencarian solusi suatu fungsi objektif. Pada jurnal tersebut dikatakan bahwa kelelawar terbang dengan kecepatan $\mathbf{v_i}$ pada posisi $\mathbf{x_i}$ di ruang pencarian pada dimensi d. Posisi ($\mathbf{x_i^t}$) dan kecepatan ($\mathbf{v_i^t}$) yang baru pada waktu t diberikan oleh:

$$\mathrm{f_i = f_m + (f_m - f_m)\beta}, \tag{4}$$

$$\mathbf{v_i^t} = \mathbf{v_i^{t-1}} + (\mathbf{x_i^t} - \mathbf{x}_*)\mathrm{f_i}, \tag{5}$$

$$\mathbf{x_i^t} = \mathbf{x_i^{t-1}} + \mathbf{v_i^t}, \tag{6}$$

dengan $\beta \in [0,1]$ merupakan vektor acak dari distribusi uniform dan $\mathbf{x}_*$ merupakan solusi global terbaik yang diperoleh dengan membandingkan dengan seluruh solusi di antara $n$ kelelawar. Dalam pencarian lokal, setiap satu solusi didapatkan diantara solusi terbaik saat itu. Solusi yang baru untuk setiap kelelawar dibangkitkan secara lokal menggunakan *random walk.*

$$\mathbf{x_b} = \mathbf{x_{la}} + \epsilon A^t, \tag{7}$$

dengan $\epsilon \in [-1,1]$ merupakan suatu bilangan acak dan $A^t = \langle A_i^t \rangle$ adalah rata-rata dari tingkat Kekerasan suara dari seluruh kelelawar pada waktu $t$. Pada penelitian ini *random walk* di rubah dengan menggunakan hasil penelitian Yuanbin, M. ,Xinquan, Z. , dan Shujian, X. (2013) berikut.

$$\mathbf{x_b} = \mathbf{x_{la}} + c \cdot r \cdot (\mathbf{x}_* - \mathbf{x_{la}}) \tag{8}$$

dengan $\mathbf{x}_*$ adalah $\mathbf{x}$ terbaik sekarang untuk kelelawar ke-i dan nilai $c$ adalah konstanta positif . Sedangkan $r \in [0,1]$ merupakan bilangan acak.

Tingkat kekerasan $A_i$ dan laju emisi gelombang suara $r_i$ harus diperbaharui di setiap iterasi. Tingkat kekerasan suara biasanya menurun seiring kelelawar menemukan mangsanya sementara laju emisi gelombang suara meningkat. $A$ dapat bervariasi dari $A_0 = 1$ hingga $A_m = 0$. Untuk setiap iterasi, maka $A_i$ dan $r_i$ diperbaharui dengan formula berikut.

$$A_i^{t+1} = \alpha A_i^t,\ r_i^{t+1} = r_i^0[1 - \exp(-\gamma\ )], \tag{9}$$

dengan $0 < \alpha < 1$ dan $\gamma > 0$. untuk $t \to \infty$ kita perhatikan bahwa :

$$A_i^t \to 0, \quad \text{dan} \quad r_i^t \to r_i^0.$$

Hal lain yang dilakukan berkaitan dengan literature ialah multiobjektif algortima kelelawar. Pada kasus multiobjektif dengan menggunakan algoritma kelelawar akan lebih rumit jika dibandingkan dengan menyelesaikan satu fungsi tujuan (*single objective*). Untuk menyelesaikan permasalahan multiobjektif dengan menggunakan Algoritma Kelelawar Multiobjektif atau *Multiobjective Bat Algorithm* (MOBA) diperlukan konsep pareto optimal dalam menemukan himpunan solusi-solusinya. Berikut ini adalah algoritma MOBA dengan menggunakan metode bobot jumlah untuk membuat fungsi tujuannya menjadi tunggal.

Levy Flight adalah random walk dengan panjang langkah diambil dari distribusi Levy, yang direpresntasikan dengan formula pangkat sederhana $L(s) \sim |s|^{-1-\beta}$ dengan $0 < \beta < 2$ adalah suatu indeks. Distribusi Levy merupakan distribusi yang non negatif dan berekor tebal (*hevy tail*) Berikut ini adalah fungsi kepadatan peluang dari ditribusi levy.

$$L(x,y,\mu) = \begin{cases} \frac{\sqrt{\gamma}}{2\pi}\left(\frac{1}{(x-\mu)^{\frac{3}{2}}}\right)\exp\left(-\frac{\gamma}{2(x-\mu)}\right) \\ 0 \end{cases}$$

Dengan $\mu > 0$ adalah langka minimum 20 dengan $\mu > 0$ adalah langkah minimum dan $\gamma$ adalah parameter scalar. Dalam Yang (2010), disebutkan bahwa Levy Flight lebih efisien daripada Gerak Brown dalam mengeksplorasi ruang pencarian bersekala besar yang tidak di kenal. Salah satu argumentasi dari pernyataan ini adalah bahwa variansi dari Levy Flight meningkat jauh lebih cepat daripada variansi gerak Brown. Berkaitan dengan simulasi numerik, dalam melakukan simulasi untuk menyelesaikan fungsi-fungsi multiobjektif maka digunakan Softwere MATLAB (Version:R2013a), Precessor Intel Core i5-7200U @2.50GHz RAM 4 GB , dan OS Windows 64 bit. Beberapa fungsi yang akan digunakan untuk simulasi ini yaitu Fungsi Fonseca dan Felming (FON), Fungsi Schafer (SCH) dan ZDT 1. Simulasi yang dilakukan yaitu dengan merubah fungsi multiobjektif menjadi fungsi skalar dengan bantuan bobot. Kemudian *pareto front* yang dihasilkan oleh fungsi-fungsi tersebut akan dibandingkan dengan true pareto front, dan akan dilihat tingkat erornya.

## 3. HASIL DAN PEMBAHASAN

Hasil uji coba penggabungan algoritma *Differensial Evolution (DE)* dan Algoritma Kelelawar dalam mendapatkan solusi optimal untuk kasus multiobjektif.

*Fungsi objektif* $f_1(\mathbf{x}), \dots, f_p(\mathbf{x}),\ \mathbf{x} = (x_1, \dots, x_n)^T$
*Inisiasi populasi kelelawar* $\mathbf{x_i}$ $(i = 1,2, \dots, n)$ *dan* $v_i$
*Definisikan frekuensi* $f_i$ *p* $\mathbf{x_i}$
*Inisiasi laju emisi gelombang* $r_i$ *d ti kе* $A_i$
***For j=1 :n*** *(titik-titik pareto front)*
    *Bangkitkan p bobot* $w_k \geq 0, \sum_{k=1}^{p} w_k = 1$
    *Bentuk fungsi tunggal* $f = \sum_{k=1}^{p} w_k f_k$
***while*** *(t < i ra m )*
    *Bangkitkan solusi baru dengan mengatur frekuensi*
    *Perbaharui kecepatan dan lokasi*
    [p (4)sc (6)]
   ***if (r > r_i)***
      *pilih solusi diantara solusi terbaik*
      ***membangkitkan solusi local dengan Differential Evolution* dan *Levy Flight.***
   ***end***
   ***if (r*** $< \boldsymbol{A_i}$ & $f(\mathbf{x_i}) < f(\mathbf{x_*})$***)***
      *terima solusi yang baru*
      *perbaharui* $r_i$ *dan* $A_i$ *persamaan* (8)
   ***end***
*Urutkan setiap kelelawar dan pilih* $\mathbf{x_*}$ *yang baru*
***End while***
*Simpan* $\mathbf{x}^*$ *sebagai non dominated solution*
***end***

Gambar 1 *Peseoudocode Hybrid* Algoritma

### *Uji Coba Hybrid Algoritma*

Hasil pembuatan program dengan penggabungan Algoritma *Differental Evolution* dan Algoritma Kelelawar akan diujicobakan kepada fungsi fonseca dan fleming (FON). Fungsi fonseca dan felming (FON) adalah fungsi yang mempunyai 2 fungsi tujuan dengan batas yaitu [-6,6]. Bentuk fungsi ini yaitu sebagai berikut.

$$\text{Minimum FON} = \begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^{d}\left(x_i - \frac{1}{\sqrt{d}}\right)^2\right) \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^{d}\left(x_i + \frac{1}{\sqrt{d}}\right)^2\right) \end{cases}$$

$$\text{dengan } -6 \leq x_i \leq 6$$

Dengan menggunakan algoritma *Hybrid* maka didapatkan solusi yaitu sebagai berikut ini, solusi ini akan diperoleh suatu himpunan *non-dominated* yang berbentuk *pareto front* seperti Gambar 2. Selain itu, dilakukan pula uji coba fungsi schafer (SCH). Fungsi schafer (SCH) adalah salah satu fungsi multiobjektif sederhana dengan fungsi pertamanya adalah fungsi kuadrat dan fungsi keduanya adalah fungsi kuadrat juga dengan titik minimum yaitu di (2,0). Berikut ini adalah bentuk umum dari persamaan SCH.

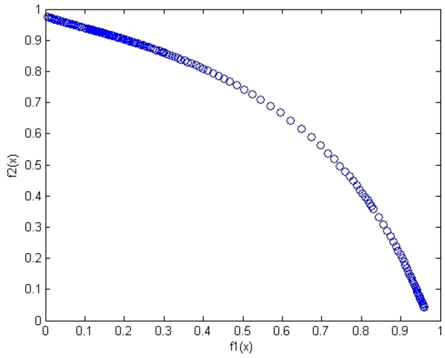
$$\text{Minimum:} \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases}, \text{dengan } -4 \leq x \leq 4$$

Selain uji coba di atas, dilakukan pula **ZDT 1.** Fungsi ZDT memiliki 6 fungsi diantaranya yaitu ZDT1. Fungsi ini mengenengahkan masalah meminimum 2 fungsi yaitu minimum $f_1(x)$, minimum $f_2(x) = g(x)h\big(f_1(x), g(x)\big)$. Pada fungsi ZDT1 ini terdapat 30 variabel (n=30) yang membentuk parito optimal yang konvek (Zitzler dkk. ,1999). Fungsinya disajikan berikut ini.

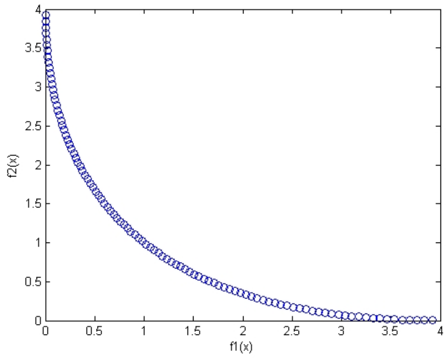
$$m \quad -\min Z \quad 1: \begin{cases} f_1(x) = x_1 \\ g(x) = 1 + \dfrac{9}{n-1}\displaystyle\sum_{i=2}^{3} x_i \\ f_2(f_1, g) = 1 - \sqrt{\dfrac{f_1}{g}} \end{cases},$$

$$x_i \epsilon [0,1], i = 1, \dots, 30$$

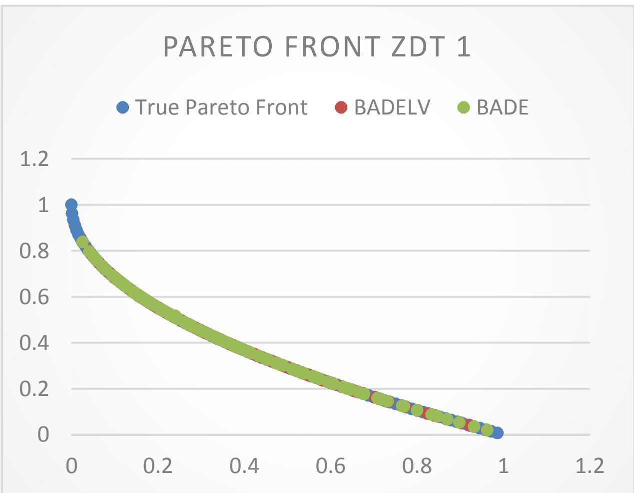
Dengan menggunakan algortima kelelawar maka didapatkan pareto optimalnya yaitu seperti Gambar 4.




Gambar 2 Solusi *Pareto Front* Fungsi FON




Gambar 3 Solusi *Pareto Front* Fungsi SCH




Gambar 4: Solusi *Pareto Front* Fungsi ZDT1

***Pembahasan***

Hasil yang diperoleh dari simulasi menggunkaan fungsi SCH, FON dan ZDT 1 yaitu seperti pada gambar 2, gambar 3 dan gambar 4. Pada uji coba fungsi SCH menggunakan parameter jumlah populasi 25, $A_0 = 0.9$ dan $r_0 = 0.95$ , F=0.9 . CR=0.5. diperoleh gambar seperi gambar 3. Pada simulasi fungsi SCH ini rata-rata pada iterasi 10-50 iterasi sudah konvergen terhadap solusinya. Pada uji coba fungsi FON yaitu dengan parameter jumlah populasi yaitu 25 individu, $A_0 = 0.9$ dan $r_0 = 0.95$, F=0.9, CR=0.5 didapatkan pareto front nya seperi gambar 2. Pada uji coba ketiga dengan menggunkan fungsi ZDT1 dengan dimensi 30, dan parameter uji untuk algoritma kelelawar jumlah populasi yaitu 25 $A_0 = 0.9$, $r_0 = 0.95$ dan $F = 0.9$ $C \ = 0.5$.

Pada Gambar 4 fungsi ZDT1 di uji dengan menggunakan Algoritma Kelelawar digabung dengan *diferential evolution* (DE) (BADE) dan dijuji dengan algoritma kelelawar digabung dengan DE serta untuk proses eksploitasinya/*random walk* menggunkan *Levy Flight Trajectory* (BADELV*)*. Dari Gambar 4, penggabungan anatara *true pareto front* ZDT1, *parto front* BADE dan *pareto front* BADELV. Dari proses analisa diperoleh selisih antara *true pareto front* ZDT1 dan BADELV kecil jika dibandingkan dengan BADE dengan true pareto front ZDT1 yaitu 0.7046576 dan 0.71102.

## 4. KESIMPULAN

Kesimpulan yang didapat dalam penelitian ini yaitu gabungan algoritma kelelawar dan *Differential Evolution* (DE) dan *Levy Flight Trajectory* dapat menyelesaikan fungsi-fungsi multiobjektif. Hasil simulasi menunjukan bahawa algoritma yang dibentuk menghasilkan hasil yang lebih bagus daripada tanpa ada *Levy Flight Trajectory*.

## 5. DAFTAR PUSTAKA

Deb K. 2001. *Multi-Objective Optimization Using Evolutionary Algorithm*. New York: John Wiley and Sons.

Fister et al. 2013. Hybrid Bat Algorithm*,* arxiv.org/pdf/1303.6310.

Price, K.V. et al. 2005. *Differential Evolution: A Practical Approach to Global Optimization.Natural Computing Series*. Berlin: Springer-Verlag.

Talbi. 2009. *Metaheuristics: From Design to Implementation*. New York: Wiley.

Xie, J. et all. 2013. Computational Intelligence and Neuroscience. *Hindawi Publishing Corporation*. Volume 2013: 13 pages

Yang. X.-S. 2010. A new Metaheuristic Bat-Inspired Algorithm: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010*)* (Eds. J.R. Gonzales et al.). *Studies in Computational Intelligence, Springer Berlin*, 284: 65-74

Yang, X. S. 2011. Bat Algorithm for Multiobjective Optimization. Int. *J. Bio-Inspired Com-putation*, III (5): 267 – 274.

*Yang, X. S. 2013. Multiobjective Cuckoo Search for Design Optimization. Int. J. Computers & Operations Research, Computers & Operations Research,* (40): 1616 –1624.