

OPTIMASI PORTOFOLIO DENGAN KENDALA *ROUNDLOT* MENGUNAKAN METODE *ARTIFICIAL BEE COLONY* (ABC)

Luthfiyati Nurafifah¹⁾, Kuntjoro Adji Sidarto²⁾

¹⁾Universitas Wiralodra, Jl. Ir.H.Djuanda Km 3, Indramayu: elena.nachi@gmail.com

²⁾Institut Teknologi Bandung, Jl. Ganesha No. 10, Bandung: sidarto@math.itb.ac.id

Abstrak

Metode Optimasi yang terinspirasi oleh perilaku makhluk hidup telah diketahui mampu menyelesaikan masalah yang kompleks secara komputasi, terutama ketika metode tradisional tak mampu melakukannya. Metode *Artificial Bee Colony* (ABC) adalah sebuah metode yang terinspirasi oleh perilaku lebah madu dalam mencari makanan. Dalam penelitian ini, penulis menggunakan metode ABC untuk menyelesaikan masalah optimasi portofolio Markowitz dengan kendala *Roundlot*. Pertama-tama, penulis menggunakan uji coba metode ABC untuk optimasi fungsi *benchmark*, kemudian menyelesaikan masalah optimasi portofolio model Markowitz. Hasilnya metode ABC mampu menyelesaikan masalah optimasi portofolio model Markowitz dengan kendala *Roundlot*.

Kata Kunci: *Optimasi Portofolio, Metode Artificial Bee Colony, kendala Roundlot*

1. Pendahuluan

Bisnis saham merupakan salah satu bentuk investasi yang banyak diminati saat ini. Sebagaimana bentuk inventasi lain, investasi saham memiliki kelebihan dan kekurangan tersendiri. Salah satu kelebihan investasi saham adalah memberikan potensi *return* yang tinggi dan berkesinambungan. Namun kelebihan ini disertai dengan kekurangan nya, yaitu potensi *return* yang tinggi pada saham kadang juga diiringi potensi rugi yang besar akibat salah pilih saham. Sehingga yang menarik perhatian penulis adalah

Bagaimana cara memilih saham agar *return* yang diperoleh seoptimal mungkin.

Biggs dan Kanne (2007) telah mengkaji teori portofolio Markowitz dengan menambahkan kendala *buy-in threshold* dan *roundlot*, kemudian menyelesaikan dengan metode Direct. Chang, Meade, Beasley, dan Sharaiha (1999) telah mengkaji metode heuristic untuk *cardinality constrained portfolio optimization*. *Artificial Bee Colony* adalah salah satu metode Heuristic yang dikemukakan oleh Dervis Karaboga pada tahun 2005. Pada tulisan ini, akan digunakan Metode *Artificial Bee Colony* untuk menyelesaikan masalah optimasi portofolio dengan kendala *roundlot*.

Bartholomew-Biggs (2005) telah menuliskan formula optimasi untuk portofolio dengan n asset dalam m periode. Misalkan modal sebesar M akan didistribusikan untuk n investasi dengan masing-masing memiliki mean return $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_n$ dan investasi sebesar $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$

$$\text{Dengan syarat } \hat{y}_1 + \hat{y}_2 + \dots + \hat{y}_n = M \quad (1)$$

$$\text{Sehingga } y_1 + y_2 + \dots + y_n = 1 \quad (2)$$

Selanjutnya $y_i = \frac{\hat{y}_i}{M}$ untuk $i = 1, 2, \dots, n$ disebut sebagai proporsi besarnya investasi untuk aset. Nilai-nilai y_i mendefinisikan suatu portofolio. Sehingga $S = \sum_{i=1}^n y_i = 1$ dan $R = \sum_{i=1}^n \bar{r}_i y_i$.

Tabel 1. Return n-aset dalam m-periode

Periode	1	2	3	...	M
Aset 1	r_{11}	r_{12}	r_{13}	...	r_{1m}
Aset 2	r_{21}	r_{22}	r_{23}	...	r_{2m}
Aset 3	r_{31}	r_{32}	r_{33}	...	r_{3m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Aset n	r_{n1}	r_{n2}	r_{n3}	...	r_{nm}

Resiko suatu portofolio ditentukan dari variansi dan kovariansi yang diperoleh dari data *return* historis aset ke- i pada periode ke- j , r_{ij} , untuk $i=1, 2, \dots, n$ dan $j=1, 2, \dots, m$.

Variansi aset_i dinyatakan dengan

$$\sigma_i^2 = \frac{1}{m} \sum_{j=1}^m (r_{ij} - \bar{r}_i)^2 \quad (3)$$

Covariansi aset_i dan aset_k dinyatakan dengan

$$\sigma_{ik} = \frac{1}{m} \sum_{j=1}^m (r_{ij} - \bar{r}_i)(r_{kj} - \bar{r}_k) \quad (4)$$

Variansi-covariansi dari portofolio dengan proporsi investasi $y_1, y_2, \dots, y_n$ adalah

$$V = \sum_{i=1}^n \sigma_i^2 y_i^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sigma_{ij} y_i y_j \quad (5)$$

Penulisan *return* dalam bentuk matriks menjadi

$$R = \vec{r}^T \vec{y} \quad (6)$$

dengan proporsi investasi $\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$ dan mean *return* $\vec{r} = \begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_n \end{pmatrix}$

Syarat pada persamaan (2)

$$\vec{e}^T \vec{y} = 1 \text{ dimana } \vec{e}^T = (1 \ 1 \ \dots \ 1) \quad (7)$$

Penulisan dalam bentuk matriks

$$\text{Expected return } R = \vec{r}^T \vec{y} \quad (8)$$

$$\text{Resiko } V = \vec{y}^T Q \vec{y} \quad (9)$$

$$Q = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nn} \end{pmatrix}$$

Dimana $\sigma_{ii} = \sigma_i^2$ dan $\sigma_{ij} = \sigma_{ji}$

Meminimumkan variansi-covariansi sama artinya dengan meminimumkan resiko. Optimasi portofolio Markowitz memiliki dua fungsi objektif, yakni memaksimumkan *return* dan meminimumkan resiko. Dalam optimasi portofolio Markowitz, penggunaan *ε-constraint method* adalah sebagai berikut:

Menargetkan Besar *Return* yang Ingin Diperoleh (R_p)

Masalah yang dihadapi dengan menggunakan metode ini disebut Minrisk 1m. Permasalahan optimasi portofolio menjadi meminimumkan resiko dengan target *return* yang telah ditetapkan. Secara matematis, ditulis:

$$\text{Min } V = \vec{y}^T Q \vec{y} \quad (10)$$

$$\text{Kendala } \sum_{i=1}^n \bar{r}_i y_i = R_p \text{ dan } \sum_{i=1}^n y_i = 1 \quad (11)$$

Masalah diatas memungkinkan diperoleh solusi $y_i < 0$, artinya mungkin terjadi *short-selling* dalam pelaksanaan portofolio tersebut. *Short-selling* adalah menjual saham yang tidak dimiliki (meminjam dari pemiliknya) dengan maksud membelinya kembali di masa yang akan datang (untuk mendapatkan keuntungan), kemudian mengembalikannya kepada pemilik saham semula (Higham,2004). Untuk menghindari hal tersebut, perlu ditambahkan kendala $y_i \geq 0$, untuk $i = 1, 2, \dots, n$.

Menentukan Besar Resiko yang Dapat Diterima (V_a)

Dengan menentukan besar resiko yang dapat diterima, masalah optimasi portofolio multi-objektif ini menjadi masalah optimasi *single-objektif* yang disebut Maxret 1m. Ditulis:

$$\text{Min } R = -\vec{r}^T \vec{y} \quad (12)$$

$$\text{Kendala } \vec{y}^T Q \vec{y} = V_a \text{ dan } \vec{e}^T \vec{y} = 1 \quad (13)$$

Sama halnya dengan masalah Minrisk 1m, masalah (12,13) memungkinkan diperolehnya nilai proporsi yang negatif, untuk mengatasinya, ditambahkan kendala seperti pada kasus Minrisk 1m.

Kendala *Roundlot*

Jobst dkk (2001) mendefinisikan *roundlot* sebagai bilangan diskrit yang menyatakan jumlah asset yang diambil sebagai satuan dasar dalam investasi. Transaksi model roundlot mensyaratkan seorang investor bertransaksi dalam satuan lot biasanya berisi 10 atau 100 saham atau sejumlah lain. Biggs-Kane (2007) telah merumuskan formula untuk menyelesaikan optimasi portofolio dengan kendala roundlot. Formula tersebut kemudian dimodifikasi oleh Jamaliatul Badriyah (2014) untuk mencegah adanya saham yang dibuang. Berikut formula yang digunakan dalam tulisan ini.

Misalkan seorang investor ingin menginvestasikan dana sebesar M dalam n saham, Jamaliatul Badriyah (2014) menyatakan bobot untuk tiap saham i dalam model dapat pula didefinisikan sebagai berikut :

$$y_i = L_i t_i \frac{p_i}{M} \quad (14)$$

dengan L_i menyatakan lot size dari saham i , t_i adalah bilangan bulat yang menyatakan banyaknya lot dengan t_i tidak melebihi $\left\lfloor \frac{M}{L_i p_i} \right\rfloor$ dan p_i menyatakan harga saham i .

Sehingga masalah meminimumkan resiko dapat ditulis menjadi

$$\text{Min } \mathbf{y}^T \mathbf{Q} \mathbf{y} \text{ dengan kendala } \mathbf{r}^T \mathbf{y} = R p, \mathbf{e}^T \mathbf{y} = 1, y_i = L_i t_i \frac{p_i}{M} \quad (15)$$

$$\text{dan } t_i \text{ adalah bilangan bulat dengan } 0 \leq t_i \leq \left\lfloor \frac{M}{L_i p_i} \right\rfloor$$

sehingga didapatkan hasil $y_1, y_2, \dots, y_n$ yang menyatakan bobot tiap-tiap saham dan $t_1, t_2, \dots, t_n$ yang menyatakan banyaknya lot saham i . Sehingga fungsi penalti minrisk 1u menjadi:

$$F = \vec{y}^T \mathbf{Q} \vec{y} + \rho \left(\frac{\vec{r}^T \vec{y}}{R p} - 1 \right)^2 + \rho (\vec{e}^T \vec{y} - 1)^2 + \mu \sum_{i=1}^n (\min(0, y_i))^2 \quad (16)$$

$$\text{dengan } y_i = L_i t_i \frac{p_i}{M}$$

Sedangkan masalah memaksimumkan *return* menjadi

$$\text{Min } -\mathbf{r}^T \mathbf{y} \text{ dengan kendala } \mathbf{y}^T \mathbf{Q} \mathbf{y} = V a, \mathbf{e}^T \mathbf{y} = 1, y_i = L_i t_i \frac{p_i}{M} \quad (17)$$

$$\text{dan } t_i \text{ adalah bilangan bulat dengan } 0 \leq t_i \leq \left\lfloor \frac{M}{L_i p_i} \right\rfloor$$

fungsi penaltinya menjadi

$$F = -\vec{r}^T \vec{y} + \rho (\vec{e}^T \vec{y} - 1)^2 + \rho \left(\frac{\vec{y}^T \mathbf{Q} \vec{y}}{V a} - 1 \right)^2 + \mu \sum_{i=1}^n (\min(0, y_i))^2 \quad (18)$$

$$\text{dengan } y_i = L_i t_i \frac{p_i}{M}$$

Metode *Artificial Bee Colony* (ABC)

Tereshko mengembangkan sebuah model dari kelakuan koloni lebah dalam mencari makanan berdasarkan persamaan reaksi difusi (Karaboga dan Akay,

2009). Lebah madu menunjukkan tingkat intelegensi yang menarik pada saat mencari sumber makanan. Model sederhana untuk pencarian sumber makanan ini terdiri atas tiga komponen penting, sumber makanan (*food source*), lebah pekerja (*employed bee*), dan lebah bukan pekerja (*unemployed bee*). Model ini menjelaskan dua kelakuan lebah madu, yakni pemilihan sumber makanan (nektar) dan pencampakkan sumber makanan yang sudah tidak dapat dieksploitasi lagi.

1. Sumber makanan (*food source*)

Nilai keberhargaan sebuah sumber makanan tergantung pada banyak factor seperti jaraknya terhadap sarang lebah, kekayaan yang tergantung dalam sumber (jumlah nektar), rasa dari nektar, dan tingkat kesulitan atau kemudahan dalam mengeskplotasi nektar.

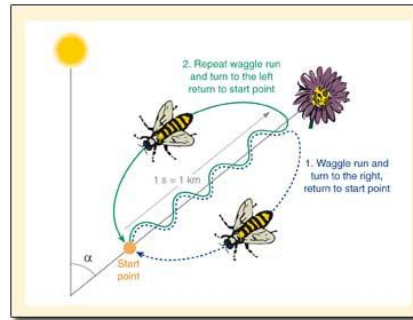
2. Lebah Pekerja (*Employed Bee*)

Lebah pekerja bertugas untuk menangani sumber makanan dimana ia bekerja. Ia membawa informasi mengenai sumber makanan tertentu dan membagi nya dengan lebah lain yang ada di sarang. Informasi ini menyangkut jarak dan arah dari sarang, dan potensi (keuntungan yang dimiliki) sumber makanan.

3. Lebah bukan Pekerja (*Unemployed Bee*)

Lebah ini mencari sumber makanan yang bisa dieksploitasi. Terdapat dua jenis lebah bukan pekerja, lebah perekrut (*scout bee*) yang bertugas mencari sumber makanan baru di sekitar sarang dan lebah pencari (*onlooker bee*) yang bertugas menunggu di sarang dan mengeksplor daerah persekitaran *food source* yang diperoleh dari *employed bee*.

Pertukaran informasi antar lebah merupakan bagian terpenting dalam formasi pengumpulan pengetahuan. Ketika meneliti keseluruhan sarang ada kemungkinan akan dikenali beberapa bagian yang ada di semua sarang. Bagian terpenting dari sarang yang berkaitan dengan pertukaran informasi adalah *dancing area*. Komunikasi antar lebah sehubungan dengan kualitas sumber makanan terjadi di *dancing area*. Tarian tersebut disebut *waggle dance*. *Onlooker bee* melihat *waggle dance* yang dilakukan oleh *employed bee* kemudian memutuskan sumber makanan mana yang akan dieksploitasi berdasarkan peluang tertentu. Ada kemungkinan *onlooker* akan memilih sumber makanan yang lebih menjanjikan (memiliki nektar yang banyak).



Gambar 1. Waggle Dance

2. Metode

Metode penelitian yang digunakan dalam menyusun tesis ini adalah studi literatur dengan mengumpulkan informasi dari buku, jurnal, makalah ilmiah, dan artikel yang berkaitan dengan masalah optimasi portofolio, algoritma *Artificial Bee Colony*, dan *Mix integer Non-Linear Programming*.

Selain studi literatur, penulis juga menggunakan simulasi numerik mengenai permasalahan yang dihadapi dengan menggunakan program MATLAB versi (R2013b) 64-bit. Program dijalankan menggunakan komputer lab dengan spesifikasi prosessor Inter Core i5 RAM 4 GB dibawah sistem operasi Windows 8.

Pertama-tama penulis menguji metode *Artificial Bee Colony* (ABC) untuk fungsi kontinu, fungsi benchmark. Dalam hal ini, penulis mengambil fungsi rosenbrock dan schwefel. Kemudian penulis menerapkan metode ABC yang telah dimodifikasi sehingga dapat melibatkan bilangan bulat untuk menyelesaikan optimasi portofolio dengan kendala *roundlot*. Penulis menggunakan data 20 saham LQ 45 pada periode 24 Agustus 2009-24 Agustus 2014 dengan *return* tahunan (<http://people.brunel.ac.uk/~mastjib/jeb/orlib/files/>). Tabel 2 menyajikan data nama perusahaan, *return*, dan harga saham perusahaan LQ 45 yang penulis ambil.

Data tersebut akan diolah dengan menggunakan algoritma ABC. Dalam algoritma ABC, posisi sumber makanan merepresentasikan solusi yang mungkin bagi masalah optimasi dan jumlah nektar yang dimiliki tiap sumber makanan berkorespondensi dengan kualitas (fitness) dari solusi

tersebut. Jumlah *employed bee* atau *onlooker bee* sama dengan jumlah solusi dalam suatu populasi. Algoritma ABC secara umum terdiri atas empat tahapan utama, tahap inialisasi, tahap *employed bee*, tahap *onlooker bee*, dan tahap *scout bee*.

Tabel 2. Data 20 saham perusahaan LQ 45

Nama Perusahaan	Return	Harga Saham	Nama Perusahaan	Return	Harga Saham
AALI	0.006702	26150	BSDE	0.025844	1640
ADRO	0.003642	1255	GGRM	0.026464	55500
AKRA	0.036225	5050	INTP	0.019283	25350
ASII	0.002785	7725	JSMR	0.023499	6350
ASRI	0.033838	540	KLBF	0.021268	1690
BBCA	0.018395	11800	MNCN	0.049813	2915
BBNI	0.019708	5350	PGAS	0.012456	5975
BBRI	0.014368	11275	PTBA	0.004737	13625
BMRI	0.018875	10525	SMGR	0.020182	16800
BMTR	0.038128	2020	UNVR	0.020163	31600

3.1 Tahap Inialisasi

Di tahapan ini, ABC membangkitkan populasi awal (initial population) secara acak sebanyak SN solusi awal. SN adalah jumlah *employed bee* atau *onlooker bee*. Masing-masing solusi (x_{ij}) adalah vektor berdimensi D, dengan D adalah banyaknya parameter yang dioptimasi. Secara matematis, ditulis

$$x_{ij} = lb + rand(0,1)(ub - lb) \quad (19)$$

$i = 1, 2, \dots, SN$ dengan SN banyak sumber makanan, $SN = CS/2$

$j = 1, 2, \dots, D$ dengan D banyak peubah

lb = batas bawah suatu selang pada peubah i

ub = batas atas suatu selang pada peubah i

Akan diperoleh nilai x_{awal} berdimensi (SN,D). Setelah x_{awal} diperoleh, dicari nilai fungsi dari x_{awal} ($f(x_{awal})$) dan nilai fitness dari $f(x_{awal})$. Nilai $f(x_i)$ terbaik disimpan. Setelah tahap inialisasi, proses masuk kedalam cycle berulang hingga cycle maksimal (Maxcycle(MCN)). Dengan kata lain, ketiga tahapan lain berada dalam cycle.

3.2 Tahap *Employed Bee*

Employed Bee menghasilkan modifikasi dari solusi (sumber makanan baru) berdasarkan informasi lokal yang ia miliki, kemudian mencari nilai fungsi dan nilai fitness nya. Cara mencari sumber makanan baru yang memiliki nektar yang lebih banyak di sekitar sumber makanan yang didapat di tahap inisialisasi (x_{ij}) adalah dengan rumusan

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{ik}) \quad (20)$$

v_{ij} = sumber makanan baru

ϕ_{ij} = nilai random di selang $[-1,1]$

x_{ij} = sumber makanan yang dipilih secara random di tahap inisialisasi

j = bilangan random $(1,D)$

k = bilangan random $(1,SN)$

Kemudian akan dihitung nilai fungsi dan fitness dari tiap sumber makanan baru dan dibandingkan fitness antara x_{ij} dan v_{ij} yang memiliki nilai fitness lebih baik disimpan. Cara menghitung nilai fitness dalam algoritma ABC yang penulis ambil adalah dengan rumusan:

$$fit_i(x_i) = \begin{cases} \frac{1}{(1+f_i(x_i))} & \text{untuk } f_i(x_i) \geq 0 \\ 1 + abs(f_i(x_i)) & \text{untuk } f_i(x_i) < 0 \end{cases} \quad (21)$$

Employed bee melakukan tarian di *dancing area* untuk mencari rekan kerja dalam mengeksplorasi sumber makanan. Lebah lain yang berada di sarang dan menyaksikan tarian, yang kemudian akan mendatangi sumber makanan dan ikut mengeksplorasinya disebut *onlooker bee*.

3.3 Tahap *Onlooker Bee*

Onlooker Bee melihat para *employed bee* menari, kemudian memutuskan sumber makanan dari *employed bee* yang mana yang akan ia eksploitasi. Sehingga dapat dikatakan *onlooker bee* mengeksplorasi daerah sumber makanan dengan peluang tertentu. Semakin banyak jumlah nektar yang dimiliki oleh sumber makanan, semakin besar peluangnya untuk terpilih

oleh *onlooker bee*. Cara mencari nilai peluang dapat ditentukan sendiri oleh user. Penulis menggunakan formula:

$$p_i = \frac{fit_i(x_i)}{\sum_{n=1}^{SN} fit_n(x_n)} \quad (22)$$

p_i = peluang individu (solusi) ke- i

$fit_i(x_i)$ = nilai fitness individu (x_i)

Proses eksploitasi yang dilakukan *onlooker bee* serupa dengan proses yang dilakukan *employed bee*.

3.4 Tahap Scout Bee

Employed Bee dengan sumber makanan yang tidak meningkat di tiap cycle setelah mencapai batas yang telah ditetapkan (limit) akan menjadi *Scout Bee* dan mulai mencari sumber makanan baru secara acak, dengan prosedur seperti tahap inisialisasi. Sumber makanan yang tak berkembang tersebut akan dibuang. Nilai limit dapat ditentukan sendiri oleh user. Penulis menggunakan kriteria:

$$Limit = (D \times CS)/2 \quad (23)$$

Asumsikan bahwa solusi yang dibuang adalah x_i dan $j \in \{1, 2, \dots, D\}$, kemudian *scout bee* menemukan sumber makanan baru yang akan menggantikan x_i . Sumber makanan baru tersebut dapat diperoleh dengan cara

$$x_i^j = lb_i^j + rand[0,1](ub_i^j - lb_i^j) \quad (24)$$

Dengan demikian, algoritma ABC memiliki tiga kontrol parameter, SN (jumlah sumber makanan), nilai limit, dan jumlah cycle maksimum (Maxcycle).

Berikut disajikan Pseudocode dari Algoritma ABC

1. Menginisialisasi populasi dari solusi x_i , $i=1, 2, \dots, SN$
2. Menghitung nilai fungsi dan fitness populasi solusi
3. Cycle = 1
4. Repeat
5. Mencari solusi baru v_i untuk *employed bee* dengan formula (20) kemudian menghitung nilai fungsi dan fitnessnya

6. Melakukan proses pemilihan nilai fitness terbaik (*greedy selection*) untuk tahap *employed bee*
7. Menghitung nilai peluang p_i untuk tiap solusi x_i oleh formula (22)
8. Mencari solusi baru v_i pada tahap *onlooker bee* bergantung pada nilai p_i , kemudian menghitung nilai fungsi dan fitnessnya
9. Melakukan *greedy selection* untuk tahap *onlooker bee*
10. Menentukan solusi yang dibuang pada tahap *scout bee*, jika ada, dan menggantinya dengan solusi baru yang dipilih secara random berdasarkan formula (24)
11. Menyimpan solusi terbaik akhir yang diperoleh
12. Cycle=cycle+1
13. Until cycle=MCN

4. Pembahasan

Fungsi rosenbrock (disebut pula *banana function*) adalah fungsi yang memiliki sebuah lembah yang panjang dan grafiknya berbentuk parabola. Karena lembah panjang ini, terkadang titik optimum global fungsi rosenbrock sulit ditemukan. Namun demikian, fungsi ini sering digunakan untuk menguji performa suatu metode untuk menyelesaikan masalah optimasi. Formula fungsi rosenbrock

$$f(x) = \sum_{i=1}^D 100(x_{i-1}^2 - x_i)^2 + (x_{i-1} - 1)^2, \quad -30 \leq x_i \leq 30 \quad (25)$$

Fungsi rosenbrock memiliki nilai minimum 0 pada titik $x = (1, 1, \dots, 1)$. Fungsi Schwefel merupakan fungsi *multimodal* yang memiliki banyak lembah dan nilai minimum lokal. Fungsi ini memiliki titik minimum global yang jauh secara geometri, jauh dari titik minimum lokal. Sehingga memungkinkan kekonvergenan algoritma dalam arah yang keliru. Formula matematis untuk fungsi schwefel

$$f(x) = \sum_{i=1}^D -x_i \sin(\sqrt{|x_i|}), \quad -500 \leq x_i \leq 500 \quad (26)$$

Fungsi schwefel memiliki nilai minimum ($D \times (-418.9829)$) pada titik $x = (420.9687, \dots, 420.9687)$. Hasil uji coba metode ABC untuk fungsi rosenbrock dan schwefel disajikan dalam tabel 3.

Tabel 3. Hasil Uji Coba Fungsi Rosenbrock dan Fungsi Schwefel Dengan Metode ABC

<i>Variabel</i>	<i>Hasil Program Fungsi Rosenbrock</i>	<i>Hasil Program Fungsi Schwefel</i>
Jumlah Source Number	20	20
BatasBawah/BatasAtas	-30/30	-500/500
MaxCycle	1500	150
Nilai Minimum	4.2064×10^{-16}	-837.9658
Titik Optimum	$x_1=1$	$x_1=420.9587$
	$x_2=1$	$x_2=420.9687$
Waktu	7.940247	0.409466
D	2	2

Hasil program metode ABC untuk kendala *roundlot* dengan fungsi penalti pada persamaan (38) dan input parameter SN=1000, Maxcycle 5000, D=20, Limit= SNxD,n= 150 (kriteria cauchy), L=10 (lot saham), Rp = 0.013, dan M= 10^6 dari tujuh kali running diambil satu terbaik disajikan pada tabel 4.

Hasil program metode ABC untuk kendala *roundlot* dengan fungsi penalti pada persamaan (39) parameter SN=1000, Maxcycle 5000, D=20, Limit= SNxD,n= 150 (kriteria cauchy), L=10 (lot saham), Va = 0.007, dan M= 10^6 dari tujuh kali running diambil satu terbaik disajikan pada tabel 4.

Tabel 4. Minrisk 1m dan Maxret 1m dengan Kendala *Roundlot*

Saham	Minrisk 1m		Maxret 1m	
	Proporsi	Banyak Lot	Proporsi	Banyak Lot

ALI	0	0	0	0
ADRO	0.01255	1	0	0
AKRA	0	0	0	0
ASII	0.07725	1	0.23175	3
ASRI	0	0	0	0
BBCA	0	0	0	0
BBNI	0.0535	1	0	0
BBRI	0	0	0	0
BMRI	0	0	0	0
BMTR	0.1212	6	0.2828	14
BSDE	0	0	0.246	15
GGRM	0	0	0	0
INTP	0	0	0	0
JSMR	0.1905	3	0.0635	1
KLBF	0	0	0	0
MNCN	0	0	0.1749	6
PGAS	0	0	0	0
PTBA	0.545	4	0	0
SMGR	0	0	0	0
UNVR	0	0	0	0
Jumlah	1	16	0.99895	39
Resiko	0.00442			
Return	0.02799			
Iterasi	168		414	
Investasi	1000000		998950	
Waktu	287.03		697.955	

5. Kesimpulan dan Saran

Metode *Artificial Bee Colony* (ABC) merupakan suatu metode yang dapat digunakan untuk menyelesaikan masalah optimasi portofolio. Masalah optimasi portofolio yang dapat diselesaikan oleh metode ini mencakup optimasi yang melibatkan variabel kontinu, variabel diskrit, dan variabel campuran kontinu-diskrit (*Mix-integer continu programming*). Pada awalnya, metode ABC hanya di desain untuk menyelesaikan masalah yang melibatkan variabel kontinu saja. Kemudian penulis memodifikasi metode

ABC agar dapat digunakan untuk menyelesaikan masalah yang melibatkan variabel diskrit, dan *mix-integer continu variable*.

Dari hasil ujicoba fungsi benchmark, metode ABC telah berhasil menentukan nilai optimum global fungsi kontinu. Kemudian penulis menggunakan metode ABC yang telah dimodifikasi untuk masalah *Mix-Integer Non Linear Programming (MINLP)* untuk menyelesaikan masalah optimasi portofolio 20 saham LQ 45 dengan kendala *roundlot* dan diperoleh hasil langsung dalam satuan lot.

Kelemahan dari metode ABC adalah waktu *running* yang cukup lama. Hal ini kemungkinan disebabkan oleh *looping* pada algoritma ABC yang sangat panjang. Ketika percobaan pertama belum menemukan nilai optimum, algoritma ABC akan memulai lagi pencarian dari awal, sehingga memakan waktu yang tidak sedikit. Bagi pembaca yang tertarik akan metode ini, disarankan untuk memodifikasi kembali metode ini supaya memiliki waktu kerja yang lebih efisien.

Daftar Pustaka

- Badriyah,J.2014. *Optimasi Portofolio dengan kendala Buy-in Threshold, Roundlot, dan Cardinality Menggunakan Differential Evolution*. Bandung: Institut Teknologi Bandung.
- Bartholomew, M.C , Biggs. 2005. *Nonlinear Optimization With Financial Applications*. Boston: Kluwer Academic Publishers.
- Biggs, M.C. Bartholomew dan Kane, S.J. 2007. *A Global Optimization Problem in Portfolio Selection*.Berlin: Springer-Velberg.
- Chang,T.J, Meade,N., Beasley,J.E., and Sharaiha,Y.M.,1999 .*Heuristics for Cardinality Canstained Portofolio Optimization*,Comput. Oper. Res., vol. 27,pp.1271-1302,2000. London: Canary Wharf.
- Higham,D.J.,2004. *An Introduction to Financial Option Valuation*. New York: Cambridge Univercity Press.

Jobst, N.D. dkk. 2001. Computational Aspects of Alternative Portfolio Selection Models in The Presence of Discrete Asset Choice Constraint. *Quantitative Finance*, **1**, 1-13.

Karaboga, D. 2005. *An Idea based on honey bee swarm for numerical optimization*. Tech. Report TR06, Comput.eng. dept/., Turkey: Erciyes University.

Karaboga D, and Akay B. 2009. A Comparative Study of Artificial Bee Colony Algorithm. *Applied Mathematics and Computation*, **214**, 108-132.
<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files>. (diakses tanggal 19 Juni 2014)