

***BINARY CUCKOO SEARCH* UNTUK OPTIMASI PORTOFOLIO DENGAN KENDALA CARDINALITY**

Lilik Muzdalifah

Program Studi Matematika FMIPA Universitas PGRI Ronggolawe Tuban

Jl. Manunggal No. 61 Tuban

moezdalif_ah@yahoo.co.id

ABSTRAK

Cuckoo Search merupakan salah satu metode optimasi metaheuristik yang terinspirasi dari tingkah laku burung Cuckoo. Algoritma *Cuckoo Search* pertama kali dikembangkan oleh Yang dan Deb pada tahun 2009. Pada dasarnya algoritma ini dikembangkan untuk menyelesaikan masalah optimasi yang hanya melibatkan variabel kontinu (real). Akan tetapi, pada kenyataannya banyak masalah optimasi yang melibatkan tidak hanya variabel real, tetapi juga variabel diskrit, bulat, dan biner. Masalah optimasi portofolio dengan kendala *cardinality*, yaitu permasalahan membentuk komposisi dari berbagai aset dalam portofolio sehingga didapatkan hasil yang optimal dengan cara memilih k aset dari n aset yang tersedia. Masalah optimasi portofolio dengan kendala *cardinality* melibatkan variabel real dan biner (0-1). Pada variabel biner, nol artinya aset tersebut tidak dipilih sedangkan satu artinya aset tersebut dipilih untuk dimasukkan ke dalam portofolio. Pada penelitian ini penulis memodifikasi algoritma *Cuckoo Search* sehingga mampu mengatasi masalah optimasi yang melibatkan variabel real dan biner, yang selanjutnya disebut algoritma *Binary Cuckoo Search*. Kemudian algoritma tersebut diterapkan pada masalah meminimumkan risiko (Minrisk) dan memaksimumkan *return* (Maxret) dengan memilih 20 saham dari 31 saham Hangseng. Hasil penelitian menunjukkan bahwa *Binary Cuckoo Search* merupakan algoritma yang handal untuk mengatasi masalah optimasi portofolio dengan kendala *cardinality*.

Kata Kunci: algoritma *cockoosearch*, *binary cuckoo search*, optimasi portofolio, *cardinality*

ABSTRACT

Cuckoo Search is one of metaheuristic optimization method that inspired by the behavior of Cuckoo Bird. Cuckoo Search algorithm is developed by Yang and Deb, Firstly at 2009. Basically, this algorithm is developed to solve optimization problem that just include continue variable (real). However, in fact there are many optimization problems not only includes real variable but also discrete, integer, and binary variable. Portfolio optimization problem with cardinality constraint is a problem to set some assets composition in portfolio such that gained optimal solution, by choosing k assets from n available assets. Portfolio optimization problem with cardinality constraint include real and binary (0-1) variables. In binary, zero means the asset is not chosen and one means the asset is chosen to be portfolio input. In this paper, the researcher modify Cuckoo Search algorithm such that is able to handle optimization problem with real and binary variable, called Binary Cuckoo Search Algorithm. Hereafter, that algorithm is applied to risk minimization and return maximization problem by choosing 20 assets from 31 Hangseng assets. This paper result shows that Binary Cuckoo Search is reliable algorithm to handle portfolio optimization problem with cardinality constraint.

Key Words: cuckoo search algorithm, binary cuckoo search, portfolio optimization, cardinality

Pendahuluan

Cuckoo Search merupakan salah satu metode metaheuristik. *Cuckoo Search* terinspirasi dari perilaku unik burung *Cuckoo*. Yang dan Deb (2009) pertama kali mengembangkan metode ini dengan menggunakan asumsi:

- Setiap burung *Cuckoo* hanya meletakkan satu telur pada satu waktu dan meletakkannya pada sarang burung lain yang dipilih secara acak.
- Sarang terbaik dengan telur berkualitas tinggi (solusi), akan lolos menuju generasi selanjutnya.
- Banyaknya sarang yang tersedia tetap, dan peluang telur burung *Cuckoo* ditemukan oleh burung yang menjadi sasaran parasitnya adalah $Pa \in [0,1]$. Jika telur burung *Cuckoo* ditemukan, maka burung pemilik sarang akan meninggalkan sarangnya dan membuat sarang baru.

Dalam hal ini, telur pada sarang burung yang menjadi sasaran parasit burung *Cuckoo* merepresentasikan suatu solusi. Telur burung *Cuckoo* yang berkualitas tinggi kemudian akan menjadi solusi baru, yang akan menggantikan solusi yang kurang baik dalam sarang tersebut.

Algoritma dasar dari *Cuckoo Search* disajikan seperti pada gambar 1 (Yang 2010, Yang dan Deb 2009).

Cuckoo Search via Lévy Flights

```

begin
  Objective function  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ 
  Generate initial population of
     $n$  host nests  $\mathbf{x}_i$  ( $i = 1, 2, \dots, n$ )
  while ( $t < \text{MaxGeneration}$ ) or (stop criterion)
    Get a cuckoo randomly by Lévy flights
    evaluate its quality/fitness  $F_i$ 
    Choose a nest among  $n$  (say,  $j$ ) randomly
    if ( $F_i > F_j$ ),
      replace  $j$  by the new solution;
    end
    A fraction ( $p_a$ ) of worse nests
      are abandoned and new ones are built;
    Keep the best solutions
      (or nests with quality solutions);
    Rank the solutions and find the current best
  end while
  Postprocess results and visualization
end

```

Gambar 1. Algoritma *Cuckoo Search*

Algoritma tersebut dikombinasikan dengan *Lévy Flight*, yaitu random walk dengan langkah acak yang berdistribusi *Lévy*. Solusi baru $x^{(t+1)}$, untuk *cuckooi*, dibangkitkan dengan *Lévy Flight*,

$$x^{(t+1)} = x^{(t)} + \alpha \oplus \text{Lévy}(\lambda) \quad (1)$$

dimana $\alpha > 0$ adalah panjang langkah yang berkaitan dengan skala permasalahan (bisa digunakan $\alpha = 1$) dan \oplus merupakan perkalian *entrywise*. Dalam masalah ini digunakan panjang langkah, *step*, yang mengikuti algoritma mantegna,

$$\text{step} = \frac{u}{|v|^{1/\beta}}, \quad 0 < \beta \leq 2 \quad (2)$$

$$u \sim N(0, \sigma_u), \quad v \sim N(0, \sigma_v) \quad (3)$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad (4)$$

$$\sigma_v = 1 \quad (5)$$

Persamaan (1) adalah persamaan stokastik untuk *random walk*. Random walk dengan *Lévy Flight* telah terbukti efisien dalam mengeksplorasi daerah pencarian solusi (Yang dan Deb 2009).

Portofolio merupakan kumpulan investasi yang dimiliki oleh institusi atau perorangan. Masing-masing investasi memiliki imbal hasil (*return*) dan risiko yang berbeda. Masalah optimasi portofolio merupakan permasalahan membentuk komposisi dari berbagai aset dalam portofolio sehingga didapatkan hasil yang optimal. Markowitz (1952) telah mengembangkan teori portofolio modern yang melibatkan masalah memaksimalkan *return* (Maxret) dan meminimumkan risiko (Minrisk). Suatu portofolio disebut sebagai portofolio efisien jika portofolio tersebut menghasilkan *return* maksimum dengan risiko tertentu atau menghasilkan risiko minimum dengan target *return* tertentu.

Untuk keperluan monitoring dan kontrol, para investor membatasi banyaknya jenis aset dalam portofolionya. Mereka melakukan investasi hanya pada sejumlah k aset dari n aset yang tersedia. Sehingga perlu dilakukan pemilihan kombinasi k aset yang membentuk portofolio optimal. Masalah inilah yang disebut sebagai masalah optimasi portofolio dengan kendala *cardinality*. Untuk

masalah meminimumkan risiko (Minrisk) dengan diberikan target *return* sebesar Rp , masalah tersebut dirumuskan sebagai berikut

$$\text{Min } V = \vec{y}^T Q \vec{y} \quad (6)$$

Kendala

$$\vec{r}^T \vec{y} = Rp$$

$$\vec{e}^T \vec{y} = 1$$

$$y_{\min} z_i \leq y_i \leq y_{\max} z_i$$

$$i = 1, 2, \dots, n$$

$$z_i \in \{0,1\}$$

$$\sum_{i=1}^n z_i = k$$

Sedangkan masalah memaksimalkan *return* (Maxret) dengan risiko yang berani diterima sebesar Va , dirumuskan sebagai berikut

$$\text{Min } R = -\vec{r}^T \vec{y} \quad (7)$$

Kendala

$$\vec{y}^T Q \vec{y} = Va$$

$$\vec{e}^T \vec{y} = 1$$

$$y_{\min} z_i \leq y_i \leq y_{\max} z_i$$

$$i = 1, 2, \dots, n$$

$$z_i \in \{0,1\}$$

$$\sum_{i=1}^n z_i = k$$

dimana V dan R secara berturut-turut adalah risiko dan *return* yang didapatkan.

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \text{ adalah proporsi}$$

investasi masing-masing aset ($y_i \in$

$$(0,1)) \text{ dan } Q = \begin{pmatrix} \sigma_{11} & \sigma_{12} \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} \dots & \sigma_{2n} \\ \vdots & \vdots \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} \dots & \sigma_{nn} \end{pmatrix}$$

adalah matriks variansi-kovariansi.

$$\vec{r} = \begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_n \end{pmatrix} \text{ merupakan mean return}$$

dari masing-masing aset. Sedangkan

$$\vec{e} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \text{ adalah matriks satuan. } y_{min} \text{ dan}$$

y_{max} merupakan batas bawah dan batas atas proporsi investasi dan $z_i \in \{0,1\}$ adalah variabel biner yang menyatakan suatu aset dipilih atau tidak untuk dimasukkan ke dalam portofolio.

Masalah (6) dan (7) dapat diselesaikan dengan cara mengubahnya menjadi masalah optimasi tanpa kendala, yaitu dengan menggunakan fungsi penalti. Fungsi penalti yang akan diminimumkan menjadi

$$V = \vec{y}^T Q \vec{y} + \rho (\vec{e}^T \vec{y} - 1)^2 + \rho \left(\frac{\vec{r}^T \vec{y}}{Rp} - 1 \right)^2 + \rho (\vec{e}^T \vec{z} - k)^2 \tag{8}$$

$$R = -\vec{r}^T \vec{y} + \rho (\vec{e}^T \vec{y} - 1)^2 + \rho \left(\frac{\vec{y}^T Q \vec{y}}{Va} - 1 \right)^2 + \rho (\vec{e}^T \vec{z} - k)^2 \tag{9}$$

dimana ρ adalah parameter penalti yang merupakan bilangan positif yang cukup besar.

Masalah (6)/(8) dan (7)/(9) merupakan masalah optimasi yang melibatkan variabel real (y_i) dan biner (z_i). Pada penelitian ini akan dilakukan modifikasi pada algoritma *Cuckoo Search* (Gambar 1) sehingga algoritma tersebut mampu mengatasi masalah optimasi yang melibatkan variabel real dan biner, selanjutnya disebut algoritma *Binary Cuckoo Search*. Kemudian algoritma tersebut diterapkan pada masalah (8) dan (9) dengan memilih 20 saham dari 31 saham Hangseng.

Metode Penelitian

Tujuan *utama* dari penelitian ini adalah untuk menyelesaikan masalah optimasi portofolio dengan kendala *cardinality* menggunakan algoritma *Binary Cuckoo Search*. Data yang digunakan adalah 31 saham Hangseng, kemudian dipilih 20 saham untuk dimasukkan ke dalam portofolio ($k = 20$). Data yang digunakan adalah data mingguan pada periode Maret 1992 - September 1997^(2.44) (<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/>), dengan *mean return* pada tabel 1.

Tabel 1. *Mean Return* 31 Saham Hangseng

| Saha m | Mean Return | Saha m | Mean Return |
|-----------|----------------|-----------|----------------|
|-----------|----------------|-----------|----------------|

| | | | |
|----|----------|----|----------|
| 1 | 0.001309 | 21 | 0.002699 |
| 2 | 0.004177 | 22 | 0.001879 |
| 3 | 0.001487 | 23 | 0.004656 |
| 4 | 0.004515 | 24 | 0.003842 |
| 5 | 0.010865 | 25 | 0.002690 |
| 6 | 0.001759 | 26 | 0.004793 |
| 7 | 0.002594 | 27 | 0.003286 |
| 8 | 0.004950 | 28 | 0.002338 |
| 9 | 0.007115 | 29 | 0.005817 |
| 10 | 0.003186 | 30 | 0.001993 |
| 11 | 0.002093 | 31 | 0.002380 |
| 12 | 0.005202 | | |
| 13 | 0.004489 | | |
| 14 | 0.003642 | | |
| 15 | 0.003960 | | |
| 16 | 0.000141 | | |
| 17 | 0.000282 | | |
| 18 | 0.000392 | | |
| 19 | 0.005294 | | |
| 20 | 0.004801 | | |

Data tersebut dimodelkan dalam portofolio Markowitz kemudian diolah untuk dicari solusi optimumnya menggunakan alat bantu Matlab R2013a dengan algoritma *BinaryCuckoo Search*. Data diproses dengan melakukan *running* sebanyak 20 kali dan solusi yang diambil adalah solusi terbaik (optimum).

Algoritma *Binary Cuckoo Search* pada dasarnya tidak mengurangi atau menghilangkan sifat dasar dari *Cuckoo Search*. Asumsi dan langkah yang digunakan masih sama dengan asumsi dan langkah dari algoritma *Cuckoo Search*. Hanya saja, variabel yang digunakan tidak hanya variabel real tetapi juga variabel biner, $[y_i, z_i]$. Variabel biner (z_i) didapatkan dengan cara mengkonversi variabel real (y_i)

menggunakan fungsi sigmoid yang didefinisikan sebagai berikut,

$$z_i = \begin{cases} 1 & \text{jika } sig(y_i) \geq 0,5 \\ 0 & \text{jika yang lainnya} \end{cases} \quad (10)$$

dengan

$$sig(y) = \frac{1}{(1 + e^{-y})}$$

(Hongqing Zheng, Yongquan Zhou, Sucai He, Xinxin Ouyang 2012, Yanhong Feng, Ke Jia, dan Yichao He 2014). Kriteria pemberhentian yang digunakan adalah *maximum generation* (iterasi maksimum) dan kriteria pemberhentian *Cauchy*. Kriteria pemberhentian *Cauchy* digunakan untuk mengontrol efisiensi waktu dalam pencarian solusi terbaik, tanpa menunggu sampai iterasi maksimum berakhir. Kriteria pemberhentian ini dilakukan setelah iterasi mencapai setengah dari iterasi maksimum dan dilakukan pengecekan selama 50 iterasi secara berturut-turut. Dalam penelitian ini, digunakan asumsi bahwa jika selama 50 iterasi secara berturut-turut tidak ada perubahan solusi ($\delta_i \leq eps, i = k, k + 1, \dots, k + 49$) maka proses selesai.

dimana,

$$\delta_i = |fbest_i - fbest_{i-1}|, i = k: k + 49$$

k adalah indeks iterasi yang dimulai dari $\frac{1}{2} \text{iterasi max} + 50$, $fbest_i$ adalah nilai fungsi solusi terbaik pada iterasi ke- i , dan eps adalah batas toleransi pemberhentian

(biasanya dipilih 10^{-12}). Algoritma *Binary Cuckoo Search* dirumuskan sebagai berikut,

Begin

Fungsi objektif $f(y, z)$, $y = (y_1, \dots, y_d)^T$, $z = (z_1, \dots, z_d)^T$

Bangkitkan populasi awal sebanyak n sarang $[y_i, z_i], i = 1, 2, \dots, n$

while($t < MaxGeneration$)dan

(*kriteria pemberhentian Cauchy*)

Bangkitkan sarang baru dengan

Lévy flight dan evaluasi

kualitas/*fitness*-nya, F_i

Pilih satu sarang dari n sarang yang ada secara acak (katakan, j)

if($F_i < F_j$)

ganti j dengan solusi baru

end

Sarang terburuk dengan peluang ditemukan $> Pa \in [0,1]$,

digantikan dengan sarang baru

Simpan solusi baru

Urutkan semua solusi dan tentukan solusi terbaik saat ini.

end while

Hasil dan visualisasi

End

Hasil dan Pembahasan

Pada Bab ini akan dibahas mengenai hasil penerapan *Binary Cuckoo Search* untuk masalah optimasi portofolio dengan kendala cardinality, yaitu masalah (8) dan (9). Adapun input parameter yang

digunakan adalah, $d = 31$, $n = 10d$, $Pa = 0.1$, $N_{itermax} = 200000$, $eps = 10^{-12}$, $rho = 10^2$, dan $mu = 10^5$.

Untuk masalah Minrisk, digunakan target *return*, $Rp = 0.006$. Sedangkan untuk masalah Maxret, besar risiko yang berani diterima, $Va = 0.0025$. Syarat batas minimum tiap saham adalah $y_{min} = 0.0075$ dan syarat batas maksimum tiap saham adalah $y_{max} = 1$. Dari 31 saham Hangseng tersebut akan dipilih 20 saham ($k = 20$).

Masalah optimasi portofolio seperti yang didefinisikan pada persamaan (8) dan (9), dikerjakan menggunakan algoritma *Binary Cuckoo Search* dan setelah dilakukan proses *running* sebanyak 20 kali, didapatkan hasil terbaik sebagai berikut,

Tabel 2. Hasil Minrisk dan Maxret dengan Kendala *Cardinality* dipilih 20 Saham dari 31 saham Hangseng

| Saha m | Minrisk dengan kendala <i>cardinality</i> Pers. (2.55) | | Maxret dengan kendala <i>cardinality</i> Pers. (2.56) | |
|-----------|---|-------|--|-------|
| | y_i | z_i | y_i | z_i |
| 1 | 0.0000 | 0 | 0.0075 | 1 |
| 2 | 0.0075 | 1 | 0.0000 | 0 |
| 3 | 0.0075 | 1 | 0.0000 | 0 |
| 4 | 0.0075 | 1 | 0.0000 | 0 |
| 5 | 0.2561 | 1 | 0.4477 | 1 |
| 6 | 0.0000 | 0 | 0.0000 | 0 |
| 7 | 0.0000 | 0 | 0.0075 | 1 |
| 8 | 0.0223 | 1 | 0.0000 | 0 |
| 9 | 0.0794 | 1 | 0.0466 | 1 |
| 10 | 0.0075 | 1 | 0.0075 | 1 |
| 11 | 0.0000 | 0 | 0.0000 | 0 |
| 12 | 0.0000 | 0 | 0.0075 | 1 |
| 13 | 0.0619 | 1 | 0.0000 | 0 |
| 14 | 0.0075 | 1 | 0.0075 | 1 |
| 15 | 0.0727 | 1 | 0.0000 | 0 |

| | | | | |
|--------------|--------|--------------|--------|---|
| 16 | 0.0000 | 0 | 0.0075 | 1 |
| 17 | 0.0075 | 1 | 0.0075 | 1 |
| 18 | 0.0075 | 1 | 0.0075 | 1 |
| 19 | 0.0000 | 0 | 0.3576 | 1 |
| 20 | 0.0075 | 1 | 0.0075 | 1 |
| 21 | 0.0075 | 1 | 0.0000 | 0 |
| 22 | 0.0000 | 0 | 0.0075 | 1 |
| 23 | 0.0000 | 0 | 0.0000 | 0 |
| 24 | 0.0000 | 0 | 0.0075 | 1 |
| 25 | 0.0075 | 1 | 0.0075 | 1 |
| 26 | 0.1446 | 1 | 0.0125 | 1 |
| 27 | 0.0075 | 1 | 0.0075 | 1 |
| 28 | 0.1326 | 1 | 0.0075 | 1 |
| 29 | 0.1165 | 1 | 0.0231 | 1 |
| 30 | 0.0315 | 1 | 0.0075 | 1 |
| 31 | 0.0000 | 0 | 0.0000 | 0 |
| $V = 0.0010$ | | $R = 0.0076$ | | |

Dari hasil masalah Minrisk dan Maxret dengan kendala *cardinality*, 20 saham yang terpilih untuk dimasukkan ke dalam portofolio adalah saham dengan nilai $z_i = 1$, dengan proporsi saham yang akan diinvestasikan sebesar y_i . Risiko dan *return* yang didapatkan secara berturut-turut adalah $V = 0.0010$ dan $R = 0.0076$. Dari tabel 2 terlihat bahwa proporsi investasi sudah memenuhi kendala yang diberikan, (1) Jumlah proporsi investasi semua saham adalah 1, (2) ketika $z_i = 1$, proporsi investasi selalu lebih dari batas bawah, $y_{min} = 0.0075$, dan kurang dari 1, (3) ketika $z_i = 0$ maka tidak ada investasi untuk saham terkait, $y_i = 0$, dan (4) tepat terpilih 20 saham yang membentuk portofolio optimum, $k = 20$. Hal ini membuktikan bahwa *Binary Cuckoo Search* mampu menyelesaikan masalah

optimasi portofolio dengan kendala *cardinality* dengan sangat baik.

Akan tetapi, penelitian ini hanya terbatas pada modifikasi algoritma *Cuckoo Search* menjadi *Binary Cuckoo Search* untuk kemudian diterapkan pada masalah optimasi portofolio dengan kendala *cardinality*. Solusi optimum yang didapatkan belum dibandingkan dengan solusi optimum dari penelitian lain dengan masalah optimasi yang sama dan algoritma berbeda. Hal ini dilakukan untuk membuktikan *Binary Cuckoo Search* lebih baik atau tidak lebih baik dari algoritma metaheuristik lain dalam menyelesaikan masalah optimasi portofolio dengan kendala *cardinality*.

Kesimpulan

Alhasil, algoritma *Cuckoo Search* yang pada dasarnya dikembangkan untuk menyelesaikan masalah optimasi yang hanya melibatkan variabel kontinu (real), dapat dimodifikasi untuk menyelesaikan masalah optimasi yang melibatkan variabel real dan biner, disebut algoritma *Binary Cuckoo Search*. Penerapan algoritma *Binary Cuckoo Search* untuk menyelesaikan masalah optimasi portofolio dengan kendala *cardinality*, memberikan hasil yang sangat baik. Dari 31 saham Hangseng terpilih tepat 20 saham yang membentuk portofolio optimum, dengan risiko dan *return* yang

didapatkan secara berturut-turut adalah $V = 0.0013$ dan $R = 0.0128$.

Lebih lanjut, hasil dari penelitian ini perlu dibandingkan dengan hasil penelitian lain dengan masalah optimasi yang sama dan algoritma berbeda. Hal ini ditujukan untuk membandingkan kehandalan *Binary Cuckoo Search* dibandingkan dengan algoritma metaheuristik lainnya dalam mengatasi masalah optimasi tersebut. Selain itu, algoritma *Binary Cuckoo Search* juga diharapkan dapat diterapkan pada masalah optimasi lainnya yang melibatkan variabel real dan biner. Lebih jauh lagi, algoritma *Cuckoo Search* diharapkan dapat dimodifikasi untuk menyelesaikan berbagaimasalah optimasi dengan berbagai macam variabel lainnya.

Pustaka

- Hongqing Zheng, Yongquan Zhou , Sucai He, Xinxin Ouyang. 2012. A Discrete Cuckoo Search Algorithm for Solving Knapsack Problems. *Advances in information Sciences and Service Sciences(AISS)*, Vol. 4.
- Markowitz, H. 1952. Portofolio Selection. *The Journal of Finance*, Vol. 7, pp. 77-91.
- Yang, X. S. 2010. *Nature-Inspired Metaheuristic Algorithms*, Second Edition, United Kingdom: Luniver Press.
- Yang, X. S. dan Deb, S. 2009. Cuckoo Search via Lévy Flights. *Proceeding of World Congres on Nature and Biologically Inspired Computing (NaBIC 2009, India)*, IEEE Publications, USA, pp.210-214.
- Yanhong Feng, Ke Jia, dan Yichao He. 2014. An Improved Hybrid Encoding Cuckoo Search Algorithm for 0-1 Knapsack Problems. *The Journal of Computational Intelligence and Neuroscience*, Vol. 2014, Article ID 970456, 9 pages.
- <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/>, Diakses pada tanggal 8 April 2016, pukul 15.00 WIB