

HYBRID ARTIFICIAL BEE COLONY : PENYELESAIAN BARU POHON RENTANG BERBATAS DERAJAT

Abidatul Izzah¹⁾, Ratih Kartika Dewi²⁾, Siti Mutrofin³⁾

Program Pasca Sarjana Teknik Informatika FTIF ITS Surabaya

Gedung Teknik Informatika Jl. Teknik Kimia ITS – Surabaya 60111

email : aza.syaifa@gmail.com¹⁾, ratihkartikadewi@yahoo.com²⁾, imut@siti-mutrofin.web.id³⁾

Abstrak

Swarm Intelligence (SI) adalah sebuah teknik kecerdasan buatan yang terinspirasi oleh sekelompok hewan di alam karena kecerdasan yang mereka miliki, seperti koloni lebah maupun koloni semut dalam mencari makanan, dan lain sebagainya. Penelitian terkait SI ini sudah banyak dilakukan, seperti Ant Colony Optimization (ACO) Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), dll.

Algoritma ABC adalah algoritma yang terinspirasi oleh perilaku koloni lebah madu dalam mencari makanan. Kelebihan ABC jika dibandingkan dengan algoritma lain yaitu sangat sederhana dan fleksibel. Namun, jika dimensi masalah meningkat, pertukaran informasi masih terbatas pada satu dimensi. Kelemahan ABC inilah yang dijadikan peluang untuk melakukan pengembangan ABC yakni, Hybrid Artificial Bee Colony (HABC) dengan menambahkan operator crossover dari algoritma genetika. Untuk menguji keunggulan HABC, dalam paper ini dilakukan pemecahan masalah Degree Costrained Minimum Spanning Tree (DCMST) menggunakan HABC. Dari hasil uji coba HABC diperoleh hasil yang lebih baik dibandingkan dengan GA, NGA, dan ABC.

Kata kunci :

Artificial Bee Colony, Degree Costrained Minimum Spanning Tree, Hybrid Artificial Bee Colony, Genetic Algorithm

1. Pendahuluan

Swarm Intelligence (SI) adalah sebuah teknik kecerdasan buatan yang terinspirasi oleh perilaku cerdas sekelompok hewan di alam, seperti koloni semut, kawanan burung, koloni lebah, kawanan bakteri, dan lain sebagainya. Dalam beberapa tahun terakhir ini, banyak algoritma SI yang telah diusulkan, seperti Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Immune Algorithm (IA), Bacterial Foraging Optimization (BFO), Cat Swarm Optimization (CSO), Firefly algorithms (FA), Artificial Bee Colony (ABC), dll. [10]

Algoritma ABC adalah algoritma baru yang terinspirasi oleh perilaku koloni lebah madu dalam mencari makanan. Algoritma ini pertama kali diperkenalkan oleh Karaboga pada tahun 2005 [9]. Dalam perkembangannya, algoritma ABC telah digunakan

untuk pemecahan permasalahan diskrit seperti Travelling Salesman Problem (TSP), Quadratic Minimum Spanning Tree (Q-MST), dan Degree Constrained Minimum Spanning Tree (DC-MST).

Algoritma yang digunakan dalam paper ini adalah Hybrid Artificial Bee Colony (HABC), yakni algoritma gabungan antara algoritma konvensional GA dengan konvensional ABC. Operator crossover milik GA disisipkan pada algoritma ABC guna meningkatkan pertukaran informasi antar lebah. Algoritma HABC ini ini pertama kali digunakan dalam permasalahan data clustering [9]. Untuk menguji keunggulan algoritma HABC, maka dalam paper ini dilakukan pemecahan masalah DCMST menggunakan algoritma HABC. Dalam paper ini HABC yang digunakan disisipi pula fase local search demi menghindari optimum lokal.

Penelitian ini bertujuan untuk:

- 1) Mengetahui implementasi algoritma HABC dalam pemecahan masalah DCMST.*
- 2) Mengetahui performa algoritma HABC dibandingkan dengan algoritma ABC dalam kasus pemecahan masalah DCMST.*

Di sisi lain masalah DCMST ini sudah pernah dipecahkan menggunakan algoritma genetika (Genetic Algorithm, GA) konvensional pada [1], [2], [8], Novel GA pada [5] dan dan ABC pada [3]. Dalam penelitian ini, digunakan data yang sama pada penelitian sebelumnya yakni data jarak 9 kota yang diambil dari Workshop Integer Programming 2004 [8] guna membandingkan keoptimalan antara hasil yang diperoleh sebelumnya. Dari penelitian ini, kontribusi yang diberikan oleh penulis adalah mengetahui perbandingan kinerja dari algoritma ABC dengan algoritma HABC pada kasus yang sama. Pada bagian akhir dari paper ini, akan dibandingkan hasil yang diperoleh dengan hasil yang sudah ada pada penelitian sebelumnya.

2. Tinjauan Pustaka

Algoritma ABC memiliki beberapa kelebihan yaitu konsep dan implementasi yang mudah, sedikit parameter serta sangat sederhana dan fleksibel jika dibandingkan dengan algoritma lain. Namun, ABC juga memiliki beberapa kelemahan diantaranya [10]:

- 1) Jika dimensi meningkat, pertukaran informasi masih terbatas pada satu dimensi.
- 2) Persekitaran dan dimensi dipilih secara acak, sehingga sumber makanan dengan *fitness* yang lebih tinggi memungkinkan untuk tidak dipilih.

Kelemahan yang dimiliki oleh algoritma ABC tersebut dijadikan peluang untuk melakukan sebuah penelitian dengan mengembangkan algoritma ABC yakni HABC dengan tambahan operator *crossover* dari algoritma genetika, algoritma hibrid ini bertujuan untuk meningkatkan optimasi algoritma ABC dan pertukaran informasi antar lebah [10].

2.1 Algoritma Artificial Bee Colony (ABC)

Algoritma *Artificial Bee Colony* (ABC) diinspirasi dari perilaku lebah dalam mencari makanan. Dalam algoritma ABC, beberapa fenomena alam disimulasikan dalam ruang pencarian agar dapat merepresentasikan proses pencarian makanan koloni lebah secara riil. Contohnya, lokasi pencarian solusi disimulasikan dengan lokasi lebah mencari makanan. Setiap solusi yang diperoleh disebut dengan sumber makanan (*food source*). Nektar dari sumber makanan tersebut direpresentasikan sebagai nilai kebugaran (*fitness*) [10].

Dalam koloni lebah, terdapat tiga jenis lebah yaitu, *employee*, *onlooker*, dan *scout*. *Employee* bertugas untuk mencari sumber makanan (*food source*) dan menghitung nektar. Selanjutnya lebah tersebut memberi informasi pada lebah *onlooker* dengan cara menari-nari di *weggle dance*, yaitu area dansa yang diperuntukkan untuk bertemunya para lebah. Kemudian *onlooker* bertugas menerima informasi tentang kualitas *food source* dan memilih *food source* yang terbaik. *Food source* yang memiliki nektar lebih banyak memiliki peluang semakin besar untuk dipilih *onlooker*. Setelah itu, *employee* yang berada pada setiap *food source* pergi untuk mencari *food source* baru di sekitarnya (*neighborhood*). Kemudian saat proses pencarian *food source* baru *employee* berubah menjadi *scout* [9].

Salah satu parameter dari algoritma ABC adalah jumlah populasi lebah (yang terdiri dari *employee* dan *onlooker*). Setiap *food source* ditempati oleh setiap *employee*, sehingga jumlah *food source* sama dengan jumlah *employee* dan jumlah *onlooker* [9]. Berikut ini adalah *pseudocode* algoritma ABC:

```

1. Initialization
   Initialize the food source and evaluate the nectar amount
   of food source (fitness).
   Send the employed bees to the current food source.
   iterasi = 0
2. Do while (the termination condition are not met)
3. // Employed Bee
   For each employed bee
     Find a new food source in its neighborhood
     Evaluate the fitness of the new food source
     Apply greedy selection of the new food source
    
```

```

End for
4. Calculate the probability P for each food source
5. //Onlooker Bee
   For each onlooker bee
     while (Current number of onlooker Bee i <= Sn/2)
       if (rand() < P)
         Send onlooker bees to the food source of i employe
         bee
         Find a new food source in its neighborhood
         Apply greedy selection of the new food source
         i=i+1
       else
         i=i+1
         i=mod((i-1),Sn/2)+1
     End if
   End while
6. //Scout Bee
7. Memorize the best solution
   Iteration = iteration+1;
   End while
8. Output the best solution
    
```

Gambar 1. Pseudocode ABC

2.2 Algoritma Genetika

Algoritma Genetika (*Genetic Algorithm*, GA) adalah algoritma evolusi klasik yang dikemukakan pada tahun 1975 oleh Holland. Algoritma ini diinspirasi oleh fenomena evolusi alam [10]. GA merupakan metode untuk menyelesaikan permasalahan-permasalahan dengan menggunakan algoritma yang diinspirasi oleh proses evolusi. Operator yang terdapat dalam GA adalah *crossover* dan mutasi. *Crossover* adalah proses penggabungan (persilangan) antara dua induk untuk mendapatkan individu baru yang mewarisi sifat dari kedua induknya. Sedangkan mutasi merupakan proses perubahan sebagian sifat individu secara random yang menghasilkan struktur genetik baru. [7].

Parameter dari GA antara lain jumlah individu, p_c , p_m , dan maksimum generasi [7]. Berikut ini adalah *pseudocode* algoritma GA :

```

1. Generate population randomly
   Generate n kromosoms and initialize parameter
   Generation = 0
2. Do while (the termination condition are not met)
3. //Evaluate
   Evaluate each kromosom on population
4. //New Population
   Create new population by this steps
   a. [Selection] Choose parent based on the
     probability of fitness. The better fitness will be
     chosed.
   b. [Crossover] Create offspring by crossing the parent
   c. [Mutation] Create offspring by inver the kromosom
5. //Update Population
   Update population by new kromosom based on
   objective function to be the next population.
   Generation = Generation + 1
   End while
6. Output the best solution
    
```

Gambar 2. Pseudocode GA

2.3 Algoritma Hybrid Artificial Bee Colony (HABC)

Dua kelemahan yang ada dalam algoritma memunculkan ide Yan, dkk. untuk menggabungkan operator *crossover* milik GA ke dalam ABC untuk memperbaiki kemampuan optimasi ABC dalam permasalahan *clustering* [10]. Selanjutnya algoritma ini disebut *Hybrid Artificial Bee Colony* (HABC). Secara umum prosedurnya HABC sama dengan ABC, tetapi disisipi tahapan *crossover* diantara *onlooker* dan *scout*. Berikut ini *pseudocode* HABC untuk menyelesaikan DCMST [10].

```

1. Initialization
   iterasi = 0
2. Do while (the termination condition are not met)
3. // Employed Bee
4. Calculate the probability P for each food source
5. //Onlooker Bee
6. //Crossover
   Produce parent population applying tournament
   selection
   For (each food source in original population P)
       select 2 parent randomly from P
       Produce new food source by crossing the selected
       parent
       Apply greedy selection on the original food source and
       the newly produced food source
   End for
7. //Scout Bee
8. Memorize the best solution
   Iteration = iteration+1;
   End while
9. Output the best solution
    
```

Gambar 3. Pseudocode HABC

Dari algoritma HABC yang diusulkan oleh Yan [10], dalam paper ini ditambahkan fase *local search* yang dilakukan pada akhir setiap iterasi yakni setelah fase *crossover*. *Local search* adalah pencarian solusi baru disekitar solusi yang ada agar solusi yang didapatkan tidak terjebak dalam lokal optimum. *Local search* ini dilakukan secara terus-meneus sampai solusi yang dihasilkan tidak lebih baik dari solusi sebelumnya.

2.4 Masalah Degree Constrained Minimum Spanning Tree (DCMST)

Pohon rentang (*Spanning Tree*) merupakan himpunan garis minimum pada E yang menghubungkan semua titik pada V sehingga setiap graf memiliki paling tidak satu *spanning tree*. Misalkan T^* adalah pohon rentang yang memiliki total bobot minimum. *Minimum spanning tree* (MST) dapat ditulis sebagai berikut [3]:

$$T^* = \min_T \sum_{e_{ij} \in E} W_{ij}$$

dengan T adalah himpunan seluruh *spanning tree* dari sebuah graf.

Konsep DCMST dapat digunakan untuk mendesain jaringan telekomunikasi, komputer, transportasi, jaringan energi, pipa, dll. Misalkan pada kasus transportasi, permasalahan akan muncul ketika ada 4 lintasan sebagai batasan yang diizinkan untuk bertemu di

satu persimpangan. Sedangkan pada jaringan komunikasi, DCMST berperan sebagai batasan derajat yang digunakan untuk memenuhi jumlah kabel yang akan bertemu di satu titik.

Misalkan $T_{b,c}^*$ adalah *minimum spanning tree* yang dibatasi dengan derajat di tiap titiknya maka permasalahan ini dapat ditulis sebagai:

$$T_{b,c}^* = \min_T \sum_{\substack{e_{ij} \in E \\ d_i \leq b}} w_{ij}$$

3. Metode Penelitian

Dalam metode penelitian ini, penulis membagi menjadi dua, yaitu metode HABC itu sendiri sebagai poin utamanya, dan metode dalam menguji kinerja atau performa setiap algoritma yang akan dibandingkan.

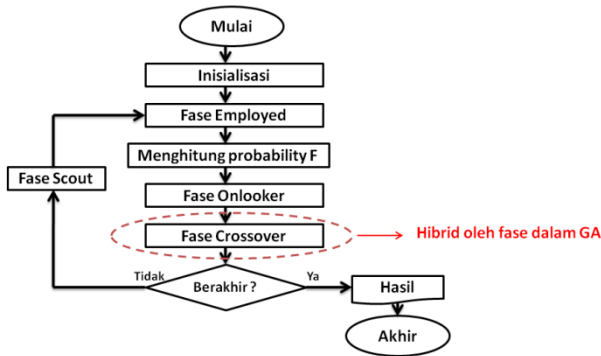
3.1. Metode HABC

Berikut ini metode yang dilakukan untuk memecahkan masalah DCMST dengan pendekatan algoritma HABC. Langkah-langkahnya sebagai berikut:

1. Inisialisasi parameter yaitu *NBee* (jumlah lebah yang terdiri dari *employee* dan *onlooker*), p_c (probabilitas terpilihnya induk untuk *crossover*, *dc* (*degree constrained*) dan *MaxIterasi*.
2. Pembangkitan solusi (*food source*) secara acak sejumlah lebah *employee*.
3. Masing-masing *employee* menuju *food source* dan menghitung nektar (*fitness*).
4. Menghitung probabilitas untuk setiap *food source* dengan rumusan

$$P_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j}$$
5. *Onlooker* memilih *food source* terbaik. Pemilihan ini menggunakan bilangan acak yang dibangkitkan dimana $0 < r < 1$.
6. Melakukan seleksi untuk memilih calon induk *crossover* dan melakukan *crossover* pada induk terpilih.
7. *Employee* berubah menjadi *scout* dan mencari *food source* baru di sekitar solusi (*neighborhood*).
8. Lebah menyimpan memori tentang solusi terbaik. Jika iterasi terpenuhi, solusi terbaik yang diperoleh adalah solusi akhir.

Secara sederhana metode HABC dapat digambarkan dalam bentuk *flowchart* seperti pada gambar 4.



Gambar 4. Flowchart HABC

3.2. Metode Pengujian Performa Algoritma HABC

Pada pengujian ini dilakukan uji coba dengan langkah-langkah sebagai berikut:

1. Menguji dan menganalisis hasil pengujian algoritma HABC pada kasus dimana algoritma GA, NGA, dan ABC telah diujikan dengan kriteria parameter yang digunakan sama, yaitu meliputi : NBee (jumlah kromosom), p_c , dc, dan MaxIterasi (jumlah generasi).
2. Membandingkan proses dan rata-rata hasil dari pengujian algoritma HABC sehingga dapat diketahui konvergensi dan nilai yang terbaik dari setiap algoritma.
3. Membuat kesimpulan dari perbandingan performa algoritma HABC dan penelitian sebelumnya yang terkait.

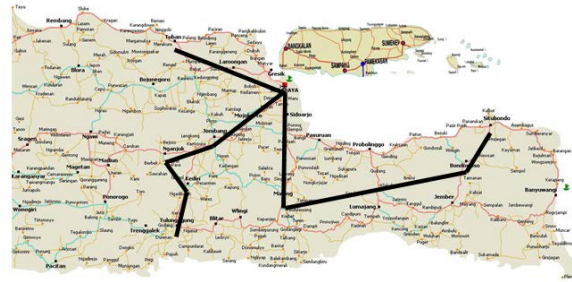
4. Hasil dan Pembahasan

Dalam paper ini dilakukan pendekatan masalah DCMST menggunakan HABC dengan parameter NBee=20, $p_c=0.6$, dc=3, MaxIterasi=100. Pemilihan parameter ini didasarkan pada penelitian sebelumnya agar hasil yang diperoleh dari perbandingan ini tidak terjadi bias pada parameteranya. Perbandingan ini dilakukan dengan GA konvensional yang telah dilakukan sebelumnya pada [1], [2], [6] yang masing-masing menggunakan operator crossover dan mutasi yang berbeda, Novel Genetic Algorithm (NGA) pada [5] dan ABC konvensional pada [3]. Parameter yang digunakan dalam GA [1], [2], [5], [6] adalah jumlah kromosom=10, dc=3, $p_m=0.1$, $p_c=0.6$, dan maksimum generasi=100. Sedangkan dalam ABC [3] digunakan parameter NBee=20, dc=3, $p_{bt}=0.8$, dan maksimum trial=100.

Data yang digunakan adalah data yang sama dengan penelitian sebelumnya yakni data jarak 9 kota di Jawa Timur yang diambil dari Workshop Integer Programming 2004. Data olahan ini dibentuk sebuah matriks simetri dengan indeks adalah 9 kota sebagai berikut: Tuban, Surabaya, Situbondo, Nganjuk, Kediri, Tulungagung, Jombang, Malang, dan Bondowoso.

Hasil dari implementasi algoritma HABC dengan parameter NBee=20, dc=3, $p_c=0.6$, dan maksimum

iterasi=100 diperoleh jarak optimal adalah 626 dengan rute seperti pada gambar 5.



Gambar 5. Rute Terpendek

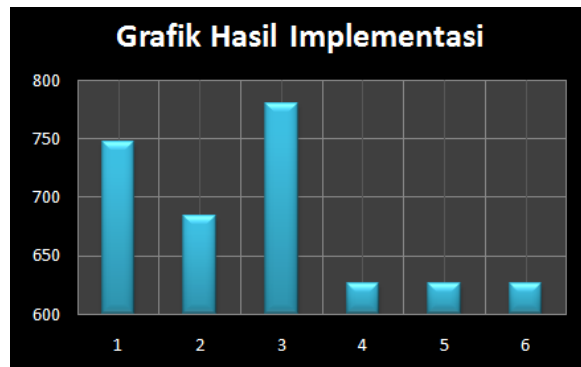
Kemudian jika ditinjau dari penelitian sebelumnya, diperoleh perbandingan seperti pada tabel 1.

Tabel 1. Perbandingan Hasil

Algoritma	Hasil
GA oleh Indrawati [1]	747
GA oleh Indriana [2]	684
GA oleh Nuriavani [6]	780
NGA oleh Muazaroh [5]	626
ABC oleh Izzah [3]	626
HABC	626

Jika data tersebut disajikan dalam bentuk diagram batang, akan terlihat seperti pada gambar 6.

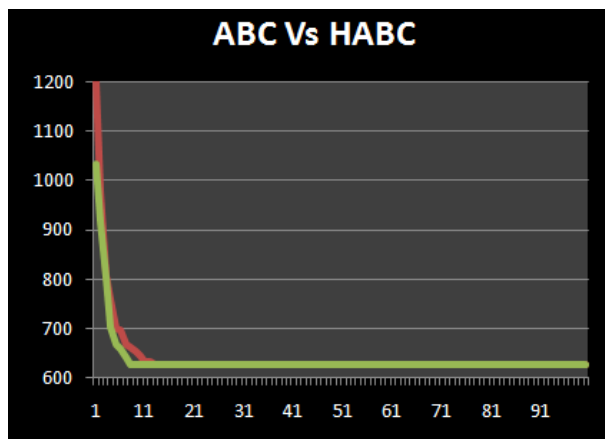
Pada gambar 6, indeks (1) adalah penyelesaian DCMST menggunakan GA oleh Indrawati, indeks (2) adalah GA oleh Indriana, indeks (3) adalah GA oleh Nuriavani, indeks (4) adalah Novel GA oleh Muazaroh, dan indeks (5) adalah penyelesaian dengan ABC oleh Izzah. Dari hasil yang diperoleh dapat dilihat bahwa algoritma ABC memang sudah lebih optimal daripada GA konvensional dan sama dengan Novel GA. Kemudian, untuk kasus ini HABC diimplementasikan dan diperoleh solusi yang sama yakni yang memiliki bobot 626.



Gambar 6. Grafik Implementasi

Walaupun pada hasil akhir diperoleh hasil yang sama, jika dilihat dari kecepatan konvergensi atau perolehan nilai optimal, HABC cenderung lebih cepat dari pada ABC. Algoritma ABC konvergen pada iterasi ke-13

sedangkan HABC sudah menemukan solusi optimal cukup pada iterasi ke-7 (mengalami konvergensi).



Gambar 7. Kekonvergenan ABC dan HABC

Kecepatan kekonvergenan yang dialami ABC dan HABC untuk memperoleh solusi sebesar 626 dapat dilihat pada grafik kekonvergenan dapat dilihat pada gambar 7. Kecepatan konvergensi ini akan menjadi sangat berpengaruh pada efektifitas algoritma jika kondisi terminatornya adalah kekonvergenan.

Data yang diambil adalah data jarak antar kota dengan mengabaikan faktor alam seperti adanya laut, dsb. Jika berdasar data acuan, jarak yang diperhitungkan adalah jarak kota yang terhubung langsung dengan melewati jalan lurus yang terhadap oleh laut. Jika rute ini memang diperuntukkan untuk pencarian rute terpendek pada hubungan antar kota, seharusnya diambil data riil yang merupakan jarak yang umum dilewati oleh alat transportasi.

5. Kesimpulan dan Saran

Dalam paper ini diperoleh solusi dari DCMST menggunakan algoritma HABC sebesar 626. Solusi ini diperoleh lebih awal dibandingkan dengan ABC.

Jika sebelumnya HABC telah membuktikan keoptimalannya dalam permasalahan optimasi dan data *clustering* maka pada penelitian selanjutnya diharapkan algoritma HABC dikembangkan pada aplikasi masalah sistem pengambilan keputusan atau permasalahan data *time series* agar algoritma ini dapat dimanfaatkan oleh dunia industri.

Daftar Pustaka

- [1]. Indrawaty, N., 2005, *Pendekatan Algoritma Genetik Bagi Persoalan DCMST (Degree-Constrained Minimum Spanning Tree)*, Departemen Matematika Fakultas Sains dan Teknologi Universitas Airlangga, Surabaya.
- [2]. Indriani, W., 2005, *Algoritma Genetika dengan Crossover Uniform Untuk Masalah DCMST (Degree-Constrained Minimum Spanning Tree)*, Departemen Matematika

Fakultas Sains dan Teknologi Universitas Airlangga, Surabaya.

- [3]. Izzah, A., 2012, *Artificial Bee Colony (ABC) sebagai Algoritma Pendekatan untuk Degree-Constrained Minimum Spanning Tre (DCMST)*, Departemen Matematika Fakultas Sains dan Teknologi Universitas Airlangga, Surabaya.
- [4]. Li, Shujun., Zheng, Xuan., 2002, *Cryptanalysis of a Chaotic Image Encryption Method*, Proceeding IEEE – ISACS, Vol.2, Scottsdale-Arizona.
- [5]. Muazaroh, N, 2012, *Novel Genetic Algorithm unruk Degree Constrained Minimum Spanning Tre*, Departemen Matematika Fakultas Sains dan Teknologi Universitas Airlangga, Surabaya.
- [6]. Nuriavani, M., 2005, *Algoritma Genetika dengan Crossover Berdasarkan Posisi Untuk Persoalan DCMST (Degree-Constrained Minimum Spanning Tree)*, Departemen Matematika Fakultas Sains dan Teknologi Universitas Airlangga, Surabaya.
- [7]. Obitko, M.,1998,*Genetic Algorithms*, Czech Technical University, Prague., Czech Republic
- [8]. Ross, K. *Traveling Along 100 Cities in Java, Indonesia, Workshop Integer Programming*, August 16-26, 2004, IPB, Bogor
- [9]. Sundar, S. and Singh, A., 2010, *A Swarm Intelligence Approach To The Quadratic Minimum Spanning Tree Problem*, Information Science, **180**, 3182-3191
- [10]. Yan, X., Zhu, Y., Zou, W., Wang, L., 2012, *A new approach for data clustering using hybrid artificial bee colony algorithm*, Neurocomputing 97, pp 241-250.

Biodata Penulis

Abidatul Izzah, memperoleh gelar Sarjana Sains (S.Si), Program Studi Matematika FSAINTEK UNAIR Surabaya – Jawa Timur, lulus tahun 2012. Saat ini tercatat sebagai mahasiswa program Magister Teknik Informatika ITS sekaligus pengajar di lembaga bimbingan belajar Sony Sugema College (SSC) Kediri-Jawa Timur.

Ratih Kartika Dewi, memperoleh gelar Sarjana Teknik (S.T), Program Studi Teknik Informatika UNIBRAW Malang - Jawa Timur, lulus tahun 2012. Saat ini tercatat sebagai mahasiswa program Magister Teknik Informatika ITS.

Siti Mutrofin, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Teknik Informatika Universitas Trunojoyo (Unijoyo) Bangkalan - Madura, lulus tahun 2009. Saat ini tercatat sebagai mahasiswa program Magister Teknik Informatika ITS sekaligus dosen di Universitas Pesantren Tinggi Darul ‘Ulum (Unipdu) Jombang - Jawa Timur.

