

# Penerapan Algoritma *Bee Colony Optimization* Dalam Pencarian Rute Tercepat Rumah Sakit di Kota Bogor

**Danar Julian Pratama**

Industri Kreatif dan Telematika, Teknik Informatika, Universitas Trilogi, Jakarta, Indonesia  
Jl. Taman Makam Pahlawan Kalibata, Jakarta Selatan, 12760, Indonesia  
Email: danarpratama@trilogi.ac.id

## Abstrak

Rumah sakit merupakan bagian penting dari kesehatan dan berhubungan dengan hidup mati seseorang. Pertolongan pertama dalam menolong seseorang yang sedang sakit adalah hal yang sangat penting. Maka dibutuhkan sebuah sistem optimasi untuk membantu para pasien dalam mencari rute terpendek menuju rumah sakit sehingga pasien yang sedang sakit dapat sampai lebih efisien. Penelitian ini menggunakan algoritma *bee colony* karena memiliki kemampuan optimasi secara efisien untuk menyelesaikan permasalahan pencarian rute terpendek. Hasil yang didapat dari penelitian ini yaitu algoritma *bee colony* dapat digunakan dalam menemukan rute terpendek. Semakin banyak lebah yang dilepaskan semakin besar peluang ditemukannya rute terpendek.

**Kata Kunci:** Kota Bogor, Rumah Sakit, Rute terpendek, Algoritma *Bee Colony Optimization*.

## Abstract

Hospitals are an important part of health and are related to someone's dead life. First aid in pawing at someone who is sick is very important. Then an optimization system is needed to help patients find the shortest route to the hospital so that patients who are sick can be more efficient. This study uses the Bee Colony algorithm because it has an efficient optimization capability to solve the problem of finding the shortest route. The results obtained from this study, namely the Bee Colony algorithm can be used to find the shortest route. The more bees released the greater the chance of finding the shortest route.

**Keywords:** Bogor City, Hospitals, Shortest Route, Bee Colony Optimization Algorithm.

## 1. PENDAHULUAN

Kota Bogor merupakan kota yang cukup luas dengan seluas 118,50 km<sup>2</sup> dengan jumlah penduduknya 1.030.720 jiwa. Kota Bogor merupakan kota yang dijuluki dengan kota penghujan dengan curah hujan rata-rata setiap tahunnya 3.500 – 4.000 mm dan lokasinya yang berada diperbukitan pulau jawa. Kota yang berdekatan dengan Ibukota Jakarta ini merupakan kota yang cukup ramai didatangi oleh rumah sakitwan ketika liburan sedang berlangsung, dikarenakan kotanya yang cukup sejuk dan banyak tempat-tempat rumah sakit yang disediakan. Banyaknya rumah sakitwan berbanding lurus dengan banyaknya jumlah kecelakaan yang terjadi dikota bogor dan kota bogor merupakan kota yang sering terjadinya bencana alam longsong yang dikarenakan seringnya hujan. Maka rumah sakit merupakan bagian penting dari sistem kesehatan untuk menolong para korban bencana alam.

Dalam sistem pelayanan kesehatan, apalagi ini berhubungan dengan hidup mati seseorang. Maka dari itu penelitian ini ingin membuat suatu pencari rute terpendek dalam menuju rumah sakit sehingga dapat membantu para pasien agar sampai dengan cepat dan membantu para perkerja rumah sakit agar lebih efektif. Rute tependek merupakan salah satu topik yang menjadi pembahasan pada masalah ini. Rute yang digunakan pada penelitian ini adalah rute terpendek untuk pencarian rumah sakit yang ada di Kota Bogor.

## 2. METODE PENELITIAN

### 2.1 Swarm

Swarm intelligence merupakan sebuah metode penyelesaian masalah yang memanfaatkan perilaku sekumpulan agen yang saling bekerja sama. Setiap upaya untuk merancang algoritma atau di distribusikan pemecahan perangkat terinspirasi oleh perilaku kolektif dari serangga sosial koloni dan masyarakat hewan lainnya.

### 2.2. Algoritma Bee Colony

Algoritma BCO (*Bee Colony Optimization*) merupakan satu dari sekian banyak algoritma optimasi yang diadaptasikan dari konsep *Swarm Intelligence* (SI) yang mendapat perhatian dari banyak kalangan peneliti. Algoritma ini pertama diusulkan oleh Dervis Karaboga pada tahun 2005. Algoritma *Bee Colony Optimization* menirukan cara kerja suatu koloni lebah dalam mencari nektar. Dalam mencari sumber makanan kemampuan koloni lebah dibagi menjadi tiga kelompok yaitu lebah pengintai, lebah penjelajah, dan lebah pekerja. Ketiga kelompok lebah ini berkerja untuk menentukan besar dan letak sumber nektar lalu mengingat dan membandingkan dengan sumber lain. Pada fungsi akhir dipilih nektar dengan jarak yang optimal (Karaboga, D. & Basturk, B., 2007). *Algoritma bee colony* juga merupakan salah satu algoritma yang digunakan untuk pencarian jalur, yang merupakan algoritma optimasi yang berdasarkan pada tingkah laku kumpulan lebah madu

dalam sebuah koloni untuk menemukan sumber makanan. Kemungkinan solusi dilambangkan dengan posisi sumber makanan, sedangkan nilainya dilambangkan dengan jumlah nektar yang terdapat dalam sumber makanan tersebut. Pendekatan yang dilakukan pada algoritma *bee colony optimization* ini adalah *population-based metaheuristic*, dimana pendekatan ini terinspirasi oleh perilaku kawanan lebah madu dalam mencari makanan. Ada 3 tahapan, yaitu:

1. Menghasilkan inisial solusi dari sumber makanan secara acak. Untuk memperbarui solusi yang mungkin di setiap *employed bee* memilih calon posisi sumber makanan baru, yang dimana posisi tersebut berbeda dengan sebelumnya.
2. Pada setiap *onlooker bee* memilih salah satu sumber makanan yang diperoleh dari *employed bee*. Setelah memilih sumber makanan, *onlooker bee* pergi ke sumber makanan yang dipilih dan memilih sumber calon makanan baru.
3. Terdapat limit yang telah ditetapkan. Pada tahapan ini, limit adalah batasan yang telah ditetapkan dalam siklus algoritma *bee colony* dan mengendalikan banyaknya solusi tertentu yang tidak di perbarui. Setiap sumber makanan yang tidak meningkat melawati limit akan ditinggalkan dan diganti dengan posisi baru dan *employed bee* menjadi *scout bee*.

Yang mana jarak diperoleh dari:

$$C_{ik} + C_{kj} - C_{ij}$$

Dimana:

$C_{ik}$  adalah jarak dari kota i ke kota k

$C_{kj}$  adalah jarak dari kota k ke kota j

$C_{ij}$  adalah jarak dari kota i ke kota j

### 3. ANALISA DAN PEMBAHASAN

Algoritma *bee colony optimization* berkerja berdasarkan banyaknya lebah yang dilepaskan. Untuk lebah yang pertama (bee-1) penelusuran yang dilakukan adalah (*forward*) penelusuran maju. Hal ini berbeda dengan lebah berikutnya. Penelusuran ini dilakukan dengan cara mengetahui posisi akhir dari lebah sebelumnya (bee-1), kemudian dilakukan penelusuran mundur (*backward*). Setelah menemukan titik yang memiliki pilihan jalur lebih dari 1 dan ada jalur yang belum dilalui oleh lebah-lebah sebelumnya maka titik asal yang baru ditemukan. Titik asal yang baru ini dijadikan posisi awal untuk penelusuran maju bagi lebah yang bersangkutan.

Pada saat bersamaan terjadi sebuah proses penyalinan jalur yang telah dilewati oleh lebah sebelumnya (bee-1). Penyalinan jalur dimulai dari titik asal yang baru sampai ke titik asal bagi lebah sebelumnya atau posisi asal perjalanannya. Penelusuran maju *forward* dilakukan hingga kriteria penghentian ditemukan.

Pertukaran informasi antara lebah satu dengan yang lainnya sangatlah penting. Pada saat memeriksa semua sarang, lebah dapat membedakan beberapa bagian yang terdapat dalam sarangnya. Hal yang sangat penting dalam sarang ialah melakukan sebuah pertukaran informasi yang disebut dengan *dancing area* atau disebut juga dengan *waggle dance*. Lebah yang telah selesai melakukan turnya masing-masing disebut dengan *waggle dance*, *waggle dance* ini dilakukan oleh lebah-lebah yang berhasil menemukan titik tujuan. Untuk lebah-lebah yang tidak menemukan titik tujuan tidak diperuntukan untuk melakukan *waggle dance*. Batas waktu *waggle dance* yang menunjukkan bahwa rute yang dilalui lebah yang bersangkutan adalah rute terpendek.

Algoritma *bee colony optimization* dimulai dengan sebuah populasi yang terdiri dari kumpulan agen lebah. Berdasarkan tugasnya, agen lebah ini terbagi menjadi tiga jenis, yaitu:

**a. Scout Bees**

*Scout bees* merupakan agen lebah yang bertugas mencari posisi sumber makanan di lingkungan sekitar sarang secara acak. Informasi posisi sumber makanan yang telah didapat akan diteruskan ke agen lebah berikutnya yaitu *Employed bees*.

**b. Employed Bees**

*Employed bees* merupakan agen lebah yang berhubungan langsung dengan sumber makanan yang sebelumnya ditemukan oleh *scout bees*. Tugas dari agen ini adalah menyimpan informasi yang berhubungan dengan setiap sumber makanan, baik berupa informasi mengenai jarak dan arah dari sarang, informasi tingkat profitabilitas atau kekayaan dari sumber makanan, maupun nilai kepantasan informasi sumber makanan tersebut untuk disebarluaskan. Maka dari itu, jumlah dari *employed bees* harus ekuivalen dengan jumlah sumber makanan yang ditemukan.

Dalam menyebarkan informasi mengenai sumber makanan, *employed bees* ini melakukan tarian yang bernama *waggle dance* yang bertempat pada pusat dari sarang lebah dengan *Onlooker Bees* yang menonton dan memilih sumber makanan yang akan dieksploitasi.

**c. Onlooker Bees**

*Onlooker bees* merupakan agen lebah yang bertugas untuk memilih dan mengeksploitasi sumber makanan yang informasinya disimpan oleh agen lebah *employed bees*.

### 3.1 Fase *Employeed Bee*

*Swap Operator* (SO) atau *Random Swap* digunakan untuk menentukan 2 Rumah Sakit dari solusi yang akan di tukar, dimana Rumah Sakit1 dan Rumah Sakit2 yang ditukar menggunakan teknik random dan bilangan yang dihasilkan untuk Rumah Sakit1  $\neq$  Rumah Sakit2. Sebagai contoh bilangan *random* yang dihasilkan adalah SO (2, 4). Maka solusi baru yang dihasilkan dari *initial solution* dengan SO adalah sebagai berikut:

$$Bee_i = Bee_i + SO \quad (1)$$

Keterangan :

$Bee_i$  = Individu *Bee* ke  $i$

SO = *Swap Operator* (Rumah Sakit1, Rumah Sakit2) Rumah Sakit1  $\neq$  Rumah Sakit2.

Sebagai contoh untuk  $bee_1$ , secara *random* nilai dari *swap operator* adalah SO(2, 4).

*Swap sequence* merupakan kumpulan dari *swap operator*, solusi baru untuk tiap *Employeed Bee* dihasilkan berdasarkan *tour* sebelumnya dan *Swap sequence*

$$\begin{aligned} Bee_i &= Bee_i + SS. \\ &= Bee_i + (SO_1, SO_2, SO_3, \dots, SO_n) \\ &= (Bee_i + SO_1) + SO_2 + SO_3 \dots + SO_n \end{aligned} \quad (2)$$

Keterangan:

SS = *Swap Sequence*

SO $_n$  = *Swap Operator* ke- $n$

$n$  = Banyaknya *Swap Sequence*

Setelah kriteria berhenti dari proses *Employeed Bee* telah terpenuhi, maka dilakukan perhitungan probabilitas dengan menggunakan persamaan untuk tiap *Employeed Bee*. Perhitungan ini akan digunakan pada fase berikutnya.

$$P_{ij} = \frac{F(\theta_i)}{\sum_{k=1}^S F(\theta_k)} \quad (3)$$

Keterangan:

$P_{ij}$  = Kemungkinan memilih *Employeed Bee* ke- $i$

$F(\theta_i)$  = *Fitness Value* dari *Employeed Bee* ke- $i$

$S$  = Jumlah *Employeed Bee*

$\theta_i$  = Posisi dari *Employeed Bee*

### 3.2 Fase *Onlooker Bee*

*Insert Operator* (IO) atau *Random Insertion* digunakan untuk menentukan sebuah Rumah Sakit dari solusi yang akan di tukar posisinya, dimana Rumah Sakit awal dan posisi tujuan yang ditukar menggunakan teknik random dimana bilangan random yang dihasilkan untuk Rumah Sakit1  $\neq$  tujuan, dan kemudian sisa urutan solusi akan digeser. Sebagai contoh bilangan random yang dihasilkan adalah IO(2, 4). Maka solusi baru yang dihasilkan dari *initial tour* dengan IO seperti pada persamaan dibawah ini:

$$Bee_i = Bee_i + IO \quad (4)$$

Keterangan :

$Bee_i$  = Individu *Bee* ke  $i$

IO = *Insert Operator* (Rumah Sakit1, Tujuan)

Sebagai contoh untuk  $bee_1$ , secara random nilai dari *insert operator* adalah IO(2, 4).

*Insert sequence* merupakan kumpulan dari *insert operator*, solusi baru untuk tiap *Employeed Bee* dihasilkan berdasarkan *tour* sebelumnya dan *insert sequence*.

$$\begin{aligned} Bee_i &= Bee_i + IS \\ &= Bee_i + (IO_1, IO_2, IO_3, \dots, IO_n) \\ &= (Bee_i + IO_1) + IO_2 + IO_3 \dots + IO_n \end{aligned} \quad (5)$$

Keterangan :

IS = *Insert Sequence*

IO $_n$  = *Insert Operator* ke- $n$

$n$  = Banyaknya *Insert Sequence*

Setelah kriteria berhenti dari proses *Onlooker Bee* telah terpenuhi, selanjutnya memasuki fase *Scout Bee*.

### 3.3 Fase Scout Bee

Setelah melalui dua fase *improvement solution*, fase *Employeed Bee* dan fase *Onlooker Bee*, maka akan dilakukan perhitungan kualitas dari masing-masing *Employeed Bee*. Jumlah *Scout Bee* disini bersifat dinamis, tergantung pada jumlah *Employeed Bee* yang telah melebihi *limit*. Apabila *limit* dari *bee* yang melakukan *improvement solution* melebihi *maximum limit* yang ditetapkan, maka solusi dari *bee* tersebut akan dihilangkan dan diganti dengan solusi baru dengan menggunakan teknik *random*, memperbarui jarak yang dihasilkan, dan menyatel ulang *limit* kembali menjadi 0.

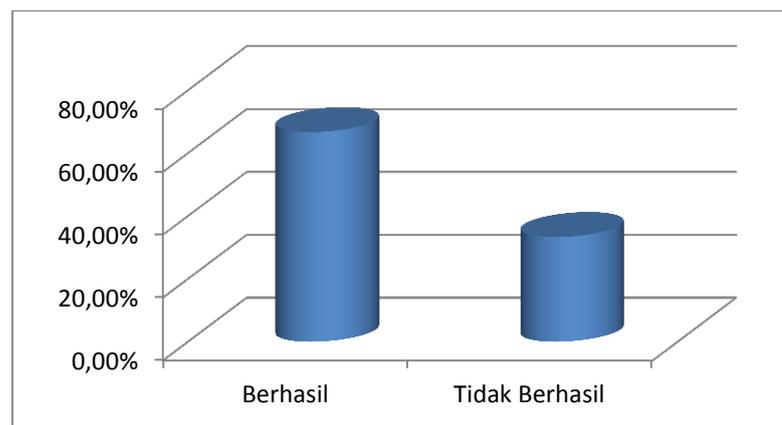
## 4. IMPLEMENTASI

### 4.1 Pengujian Berdasarkan Jumlah Lebah

Pengujian berdasarkan jumlah lebah dilakukan untuk melihat pengaruh jumlah lebah yang dilepas dalam menemukan rute terpendek menuju tujuan rumah sakit.

Pengujian pertama menggunakan parameter yang sama yaitu  $\alpha = 1$ ,  $\beta = 2$ ,  $\lambda = 0.7$ , dan  $K = 1$ . Pengujian pertama dilakukan dengan 1 ekor lebah. Pengujian berikutnya dengan 10, 50, 100, dan 300 ekor lebah. Secara keseluruhan pengujian dilakukan sebanyak 25 kali dengan 1 tujuan rumah sakit sebanyak 10 kali dan 2 tujuan rumah sakit sebanyak 10 kali dan 3 tujuan rumah sakit 5 kali.

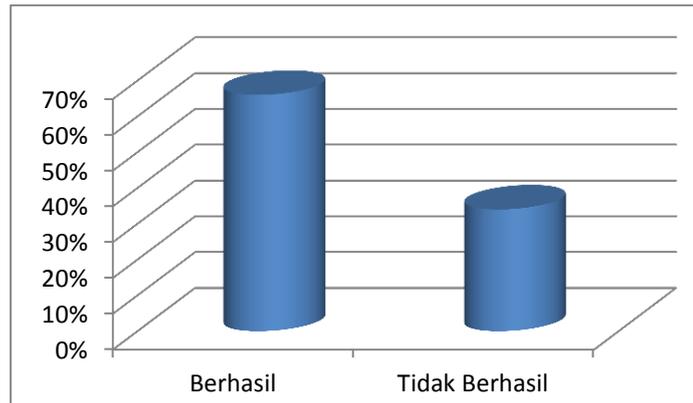
Tujuan rumah sakit yang berjumlah 1 memiliki tingkat keberhasilan sebesar 80%, hal ini ditandai dengan 8 percobaan yang berhasil dari 10 percobaan yang dilakukan. Tujuan rumah sakit yang berjumlah 2 memiliki tingkat keberhasilan sebesar 80%, hal ini ditandai dengan 8 percobaan yang berhasil dari 10 percobaan yang dilakukan. Tujuan rumah sakit yang berjumlah 3 memiliki tingkat keberhasilan sebesar 40%, hal ini ditandai dengan 2 percobaan yang berhasil dari 5 percobaan yang dilakukan. Rata-rata keberhasilan sebesar 66,7%. Tingkat keberhasilan rata-rata menemukan tujuan rumah sakit dibawah pengaruh jumlah lebah dapat dilihat pada Gambar 1.



Gambar 1 Hasil pengujian berdasarkan jumlah lebah.

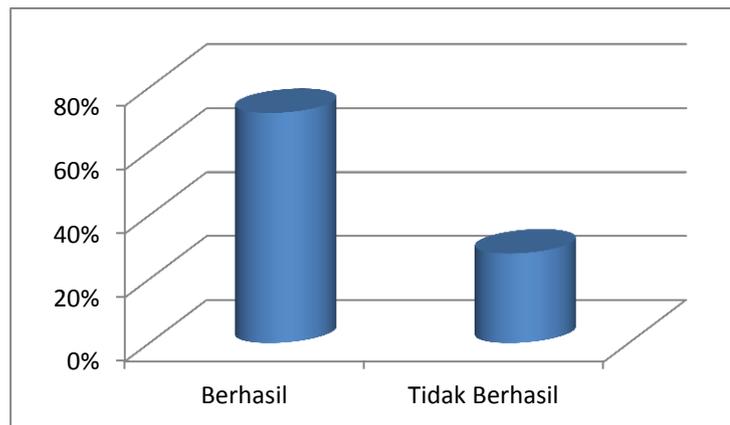
### 4.2 Pengujian Berdasarkan Jumlah Tujuan

Pengujian pertama dilakukan dengan 1 tujuan rumah sakit. Pengujian pertama ini menggunakan parameter yang sama yaitu jumlah lebah ( $N_{Bee}$ ) = 100,  $\alpha = 1$ ,  $\beta = 2$ ,  $\lambda = 0.7$ , dan  $K = 1$ . Secara keseluruhan pengujian dilakukan sebanyak 60 kali. Jumlah tujuan rumah sakit yang diuji sebanyak 10 tujuan rumah sakit dimana setiap tujuan rumah sakit dilakukan pengujian sebanyak 5 kali dengan posisi asal yang berbeda. Pengujian pertama terdapat 43 percobaan yang berhasil dan 17 yang tidak berhasil. Bila dipersentasekan ada 66% percobaan yang berhasil menemukan tujuan rumah sakit dan 34% yang tidak berhasil menemukan tujuan rumah sakit. Persentase tingkat keberhasilan diperlihatkan pada Gambar 2.



Gambar 2 Hasil pengujian dengan 1 tujuan

Pengujian kedua adalah pengujian dengan n tujuan rumah sakit. Pengujian dibagi menjadi dua yaitu dengan 2 tujuan rumah sakit dan 3 tujuan rumah sakit. Pengujian kedua ini menggunakan parameter yang sama yaitu jumlah lebah (NBee) = 100,  $\alpha = 1$ ,  $\beta = 2$ ,  $\lambda = 0.7$ , dan  $K = 1$ . Pengujian dengan 2 tujuan rumah sakit akan dilakukan sebanyak 35 kali. Jumlah kombinasi tujuan rumah sakit yang diuji sebanyak 5 kombinasi dengan 5 posisi asal yang berbeda untuk setiap kombinasinya. Pengujian kedua ini diperoleh hasil yaitu dari 35 percobaan terdapat 28 percobaan yang berhasil dan 7 yang tidak berhasil. Bila dipersentasekan ada 72% percobaan yang berhasil menemukan tujuan rumah sakit dan 28% yang tidak berhasil menemukan tujuan rumah sakit. Tingkat keberhasilan pengujian dengan 2 tujuan rumah sakit dapat dilihat dari grafik pada Gambar 3.



Gambar 3 Hasil pengujian dengan 2 tujuan

Hasil yang diperoleh dari pengujian berdasarkan jumlah lebah didapat hasil berupa peluang menemukan rute terpendek akan lebih besar apabila jumlah lebah yang dilepas semakin banyak. Semakin banyak lebah yang dilepas akan memperbesar peluang ditemukannya rute terpendek dari posisi awal ke posisi tujuan. Hal ini disebabkan sistem akan menelusuri sebanyak mungkin rute yang bisa dilalui sesuai dengan jumlah lebah yang dilepas. Jumlah lebah juga mempengaruhi waktu proses sistem. Semakin banyak jumlah lebah waktu proses akan semakin bertambah. Jumlah lebah yang ideal untuk dilepas adalah antara 10 sampai 100 lebah. Apabila pengujian berdasarkan jumlah tujuan wisata didapat hasil berupa algoritma *bee colony optimization* bisa diterapkan untuk menemukan posisi tujuan. Lebah akan mencari tujuan rumah sakit secara keseluruhan dimulai dari posisi asal menuju tujuan rumah sakit yang *diinputkan*.

## 5. KESIMPULAN

Berdasarkan dari hasil penelitian yang dilakukan, diperoleh kesimpulan bahwa jumlah lebah yang dilepaskan mempengaruhi peluang untuk menemukan rute terpendek. Semakin banyak lebah yang dilepaskan maka memperbesar peluang ditemukannya rute terpendek dari posisi awal ke posisi tujuan. Dan pencarian rute terpendek menggunakan algoritma bee colony menghasilkan hasil yang optimal.

## REFERENCES

- [1] Paper, C., Chong, C. S., Sivakumar, A. I., Yoke, M., & Low, H. (2006). A Bee Colony Optimization Algorithm to Job Shop Scheduling, (December). <https://doi.org/10.1145/1218112.1218469>
- [2] Wong, L., Yoke, M., Low, H., & Chong, C. S. (2008). Bee Colony Optimization with Local Search for Traveling Salesman Problem, (August). <https://doi.org/10.1109/INDIN.2008.4618252>
- [3] Chong, C. S. (1999). USING A BEE COLONY ALGORITHM FOR NEIGHBORHOOD SEARCH IN JOB SHOP SCHEDULING PROBLEMS, (Pinedo 1995).
- [4] Pekanbaru, S. R., & Pekanbaru, S. R. (n.d.). Pencarian Rute Terbaik Pada Travelling Salesman Problem ( TSP ) Menggunakan Algoritma Genetika pada Dinas Kebersihan dan Pertamanan Kota Pekanbaru Lusiana.
- [5] Informasi, J. T., Rahmandha, E., Efendi, R., Puspitaningrum, D., Studi, P., Informatika, T., & Teknik, F. (2016). TERDEKAT MENGGUNAKAN METODE ARTIFICIAL BEE COLONY DI, 12, 141–154.
- [6] Matematika, P. S., Sains, F., Teknologi, D. A. N., Islam, U., Syarif, N., & Jakarta, H. (2013). MENGGUNAKAN MODIFIKASI ALGORITMA ARTIFICIAL BEE COLONY.
- [7] Nurdiana, D. (2015). TERPENDEK DENGAN MEMPERTIMBANGKAN HEURISTIK, 4.
- [8] Batmetan, J. R. (n.d.). Algoritma Ant Colony Optimization ( ACO ) untuk, 31–48.
- [9] Saidah, N. H., Er, M., Informasi, J. S., & Keputih, K. (n.d.). Implementasi algoritma optimasi bee colony untuk penjadwalan job shop.
- [10] Luthfi, N. I., & Handoko, S. (n.d.). OPTIMASI PENEMPATAN DISTRIBUTED GENERATION PADA IEEE 30 BUS SYSTEM MENGGUNAKAN BEE COLONY ALGORITHM Metode.
- [11] Sitorus, E. V. B., & Nugroho, A. (n.d.). OPTIMASI ECONOMIC DISPATCH PADA UNIT PEMBANGKIT PLTG DI PLTGU TAMBAK LOROK MENGGUNAKAN ALGORITMA ARTIFICIAL BEE COLONY Metode.
- [12] Herawati, Y., & Mahmudy, W. F. (2018). Optimasi Menu Makanan Bagi Pasien Gagal Ginjal Menggunakan Algoritme Lebah, 2(4), 1698–1703.
- [13] Yudhitiya, M. A. (2018). Penerapan Pencarian Rute Tercepat Aliran Air Menggunakan Algoritma Bee Colony Optimization, 5(4), 353–356.
- [14] Karaboga, D., & Basturk, B. (n.d.). Artificial Bee Colony ( ABC ) Optimization Algorithm for Solving Constrained Optimization, 789–790.