

Optimasi Rute Terpendek Jalur Distribusi Pupuk Menggunakan Algoritma *Artificial Bee Colony* (Studi Kasus: PT Bungo Dani Mandiri Utama)

Rio Liusman¹, Rara Sandhy Winanda²

^{1,2}Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Padang (UNP)

Article Info

Article history:

Received July 24, 2023
Revised August 07, 2023
Accepted December 20, 2023

Keywords:

Distribution
Travelling Salesman Problem
Artificial Bee Colony

Kata Kunci:

Distribusi
Travelling Salesman Problem
Artificial Bee Colony

ABSTRACT

Product distribution involves planning and storing information related to product storage until the product is delivered. PT Bungo Dani Mandiri Utama is a fertilizer distributor that must visit ten retailers in its product distribution. This research aims to test whether the route currently used by PT Bungo Dani Mandiri Utama is optimal or needs improvement. This research is an applied study that uses the Artificial Bee Colony algorithm to solve the fertilizer distribution problem modeled as a Traveling Salesman Problem. From the analysis, the optimal route is obtained, starting from the warehouse, passing Lubuk Beringin, Limbur, Kerakap Island, Rantau Ikil, Mangun Jayo, Tanjung Menanti, Sungai Binjai, SPA Unit 1 Market, Tirta Mulya, Senamat, and back to the warehouse, with a total distance of 330 km. This optimal route is 48 km shorter than the usual route used by PT Bungo Dani Mandiri Utama which covers 378 km.

ABSTRAK

Pendistribusian produk melibatkan perencanaan dan pengelolaan informasi terkait penyimpanan produk sebelum produk tersebut dikirimkan. PT Bungo Dani Mandiri Utama adalah distributor pupuk yang harus mengunjungi sepuluh pengecer dalam distribusi produknya. Penelitian ini bertujuan untuk menguji apakah rute yang saat ini digunakan oleh PT Bungo Dani Mandiri Utama sudah optimal atau perlu perbaikan. Penelitian ini merupakan studi terapan yang menggunakan algoritma *Artificial Bee Colony* untuk menyelesaikan masalah distribusi pupuk yang dimodelkan sebagai *Travelling Salesman Problem*. Dari hasil analisis diperoleh rute optimal yaitu, dimulai dari Gudang, melewati Lubuk Beringin, Limbur, Pulau Kerakap, Rantau Ikil, Mangun Jayo, Tanjung Menanti, Sungai Binjai, Pasar SPA Unit 1, Tirta Mulya, Senamat, dan kembali Gudang, dengan total jarak 330 km. Rute optimal ini lebih pendek 48 km dibandingkan rute biasa yang digunakan oleh PT Bungo Dani Mandiri Utama yang mencakup 378 km.

This is an open access article under the [CC BY-SA](#) license.



Penulis pertama/sesuai:

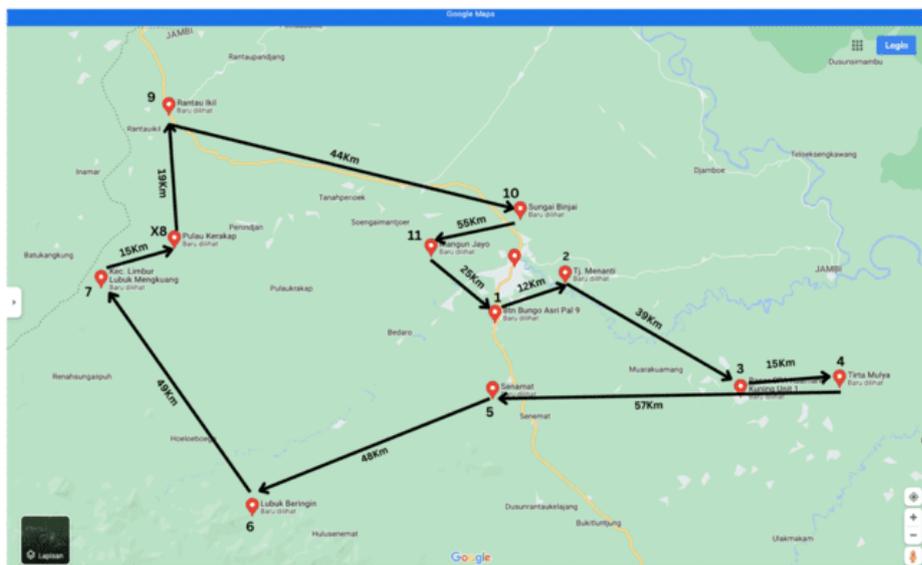
(Rio Liusman)

Prodi Matematika, Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Padang, Jl. Prof. Dr. Hamka, Air Tawar barat, Padang Utara, Padang, 25171 Padang, Sumatera Barat
Email: hazarinmorelaynt@gmail.com

1. PENDAHULUAN

Pendistribusian produk merupakan sebuah proses yang umumnya melibatkan perencanaan dan penyimpanan informasi terkait stok produk hingga produk tersebut berhasil dikirimkan [1]. Pendistribusian produk dapat dianggap sebagai bentuk permasalahan *Travelling Salesman Problem* [2]. Traveling Salesman Problem adalah jenis masalah optimasi yang mencari rute perjalanan dimulai dari titik awal, melalui sejumlah lokasi yang telah ditentukan, dan kembali ke titik awal, dengan tujuan mencari rute terpendek yang melibatkan setiap lokasi hanya sekali kunjung [3].

PT Bungo Dani Mandiri Utama (PT BDMU) merupakan perusahaan distributor pupuk yang terletak di Kabupaten Bungo. Dalam mendistribusikan produknya, PT BDMU harus berkeliling ke sepuluh pengecer yang sudah terdaftar. Setiap dua bulan sekali PT BDMU berkeliling mengantar pupuk ke sepuluh pengecer. Jarak tempuh rute yang biasa digunakan PT BDMU dalam mendistribusikan pupuk sejauh 378 km. Rute yang sering digunakan PT BDMU terdapat pada Gambar 1.



Gambar 1. Peta Rute Distribusi PT BDMU

Pada penelitian ini, akan diuji apakah rute yang biasa digunakan oleh PT BDMU sudah optimal atau memerlukan perbaikan. Untuk memecahkan *Travelling Salesman Problem*, digunakan algoritma *Artificial Bee Colony* (ABC) [4]. Penelitian terkait dapat ditemukan dalam studi oleh Furqan, yang mengenai algoritma *Artificial Bee Colony* digunakan untuk mencari rute terpendek dalam jalur distribusi pipa air bersih [5]. Penelitian tersebut menggunakan data dari delapan titik pemasangan rumah pelanggan dan satu titik Instalasi Pengolahan Air (IPA). Hasil penelitian menunjukkan bahwa algoritma ABC mampu menyelesaikan permasalahan dan menghasilkan jalur terpendek [6].

Selain itu, Chaerunnissa juga melakukan penelitian sejenis mengenai penyelesaian *Travelling Salesman Problem* dengan menggunakan algoritma ABC untuk mendapatkan rute terpendek di PT X Shipping Process [7]. Penelitian ini bertujuan untuk menentukan model *Travelling Salesman Problem* dalam pendistribusian barang. Studi tersebut dijadikan referensi karena masalah yang dibahas hampir sama, yaitu masalah distribusi.

Berdasarkan uraian di atas, dilakukan penelitian yang berhubungan dengan pencarian rute terpendek dalam jalur distribusi pupuk dengan judul "Optimasi Rute Terpendek Jalur Distribusi Pupuk Menggunakan Algoritma *Artificial Bee Colony* (Studi Kasus: PT Bungo Dani Mandiri Utama)".

2. METODE

Jenis data yang digunakan dalam penelitian ini adalah data primer melalui wawancara [8]. Data yang diperoleh berupa data lokasi pengecer pupuk pada PT BDMU dan rute yang digunakan PT BDMU dalam mendistribusikan barang. Adapun teknik pengolahan data yang dilakukan dengan menggunakan algoritma *Artificial Bee Colony* [9] adalah:

- 1) Menentukan data lokasi pengecer pupuk pada PT BDMU dan rute yang digunakan PT BDMU

dalam mendistribusikan barang.

- 2) Untuk proses algoritma *Artificial Bee Colony* ini akan dilakukan iterasi sebanyak 10 kali, untuk iterasi awal akan dilakukan proses inisialisasi yaitu menentukan solusi awal rute dengan mencari rute secara acak.
- 3) Selanjutnya, menghitung nilai *fitness* dari solusi awal dengan menjumlahkan total jarak pada rute awal.
- 4) Mencari solusi baru oleh *Employed Bee* menggunakan *swap operator*. Dengan contoh bilangan *random* berindeks (3,6) maka hasil *swap operator* (3,6) dijabarkan pada Tabel 1.

Tabel 1. Proses *Swap Operator*

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| ↓ | | | | | | |
| 1 | 2 | 6 | 4 | 5 | 3 | 7 |

- 5) Menghitung nilai *fitness* dari solusi baru.
- 6) Melakukan *greedy selection* terhadap nilai *fitness* awal dan baru.
- 7) Menghitung nilai probabilitas dari nilai *fitness* yang didapat pada fase *Employed Bee*.

$$P_i = \frac{fit_{x_i}}{\sum_{n=1}^{SN} fit_{x_i}} \quad (1)$$

Dimana:

P_i = nilai probabilitas

fit_i = nilai *fitness* solusi ke-i

$\sum_{n=1}^{SN} fit_{x_i}$ = jumlah dari nilai *fitness* ke-i sampai SN.

- 8) Memilih solusi untuk diperbarui *Onlooker Bee* dengan membandingkan nilai probabilitas dengan nilai *random*.
- 9) Mencari solusi baru oleh *Onlooker Bee* menggunakan *swap operator*.
- 10) Menghitung nilai *fitness* dari solusi *Onlooker Bee*.
- 11) Melakukan *greedy selection* terhadap nilai *fitness* *Employed Bee* dan *Onlooker Bee*.
- 12) Setelah melakukan *greedy selection* pada *Onlooker Bee*, dilakukan pengecekan apakah ada solusi yang tidak melakukan pembaruan jika ada, maka dilakukan fase *Scout Bee* untuk mencari solusi baru.
- 13) Setelah fase *Scout Bee* dilakukan maka dilakukan iterasi kedua sampai ke iterasi kesepuluh.
- 14) Selanjutnya dipilih solusi terendah dari setiap solusi.
- 15) Membandingkan rute optimal yang didapat menggunakan *Artificial Bee Colony* dengan rute yang biasa digunakan PT BDMU.

3. HASIL DAN PAMBAHASAN

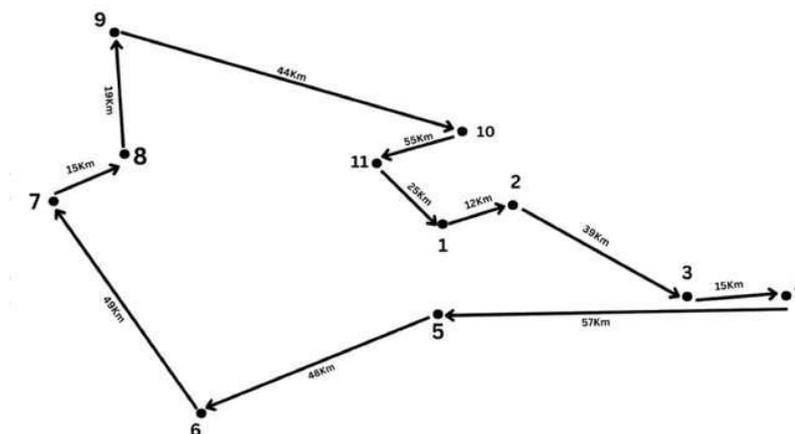
Data yang diperoleh dari wawancara dengan manager PT BDMU yaitu, PT BDMU memiliki satu Gudang dan sepuluh pengecer yang melakukan kegiatan distribusi. Data yang digunakan pada penelitian ini dijabarkan pada Tabel 2.

Tabel 2. Lokasi Pengecer Pupuk

| Nama | Node | Lokasi |
|------|------|--------|
|------|------|--------|

| | | |
|-------------|----|------------------|
| Gudang | 1 | Pal 9 Muarabungo |
| Pengecer 1 | 2 | Tanjung Menanti |
| Pengecer 2 | 3 | Pasar SPA unit 1 |
| Pengecer 3 | 4 | Tirta Mulya |
| Pengecer 4 | 5 | Senamat |
| Pengecer 5 | 6 | Lubuk Beringin |
| Pengecer 6 | 7 | Limbur |
| Pengecer 7 | 8 | Pulau Kerakap |
| Pengecer 8 | 9 | Rantau Ikil |
| Pengecer 9 | 10 | Sungai Binjai |
| Pengecer 10 | 11 | Mangun Jayo |

Tabel 2 berisi informasi tentang beberapa pengecer pupuk dan Gudang yang terdapat dalam wilayah tertentu. Gudang berlokasi di Pal 9 Muarabungo dengan nomor node 1, nomor node yang digunakan untuk menentukan titik simpul pada graf. Sementara itu, terdapat sepuluh pengecer lainnya dengan nomor node 2 hingga 11 yang tersebar di berbagai wilayah seperti Tanjung Menanti, Pasar SPA unit 1, Tirta Mulya, Senamat, Lubuk Beringin, Limbur, Pulau Kerakap, Rantau Ikil, Sungai Binjai, dan Mangun Jayo. Lokasi Gudang dan 10 pengecer divisualisasikan dalam bentuk graf berbobot. Bobot dalam graf ini merupakan jarak antara 2 node, sebagai contoh bobot untuk node 1 dan node 2 adalah 12 km karena jarak antara Gudang dengan pengecer 1 sejauh 12 km. graf berbobot untuk rute PT BDMU dilihat pada Gambar 2.



Gambar 2. Graf Jalur Distribusi Pupuk PT BDMU

Gambar 2, menunjukkan rute perjalanan yang biasa dilewati oleh PT BDMU. Dimana rute awal dalam pendistribusian pupuk ini diawali dengan Gudang, ke pengecer 1 ke pengecer 2 sampai ke pengecer dan kembali ke Gudang dengan total jarak 378 km.

Berdasarkan data yang ada, jarak gudang dengan pengecer - pengecer dilihat menggunakan *google maps* dan jalurnya sudah tersedia disusun dalam satuan kilometer, dan untuk lebih lanjut untuk melihat rute terbaik dilakukan pengukuran jarak setiap node menggunakan *google maps*. Sehingga datanya dijabarkan pada Tabel 3.

Tabel 3. Tabel Jarak Antarlokasi

| Node | 1 (Gudang) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------------|------------|----|-----|-----|----|----|-----|-----|-----|----|----|
| 1 (Gudang) | 0 | 12 | 42 | 57 | 11 | 41 | 74 | 68 | 57 | 15 | 25 |
| 2 | 12 | 0 | 39 | 53 | 23 | 53 | 83 | 67 | 56 | 12 | 25 |
| 3 | 42 | 39 | 0 | 15 | 42 | 82 | 117 | 102 | 90 | 46 | 59 |
| 4 | 57 | 53 | 15 | 0 | 57 | 96 | 124 | 116 | 105 | 61 | 74 |
| 5 | 11 | 23 | 42 | 57 | 0 | 48 | 95 | 79 | 67 | 26 | 36 |
| 6 | 41 | 53 | 82 | 96 | 48 | 0 | 49 | 50 | 67 | 53 | 63 |
| 7 | 74 | 83 | 117 | 124 | 95 | 49 | 0 | 15 | 29 | 72 | 57 |
| 8 | 68 | 67 | 102 | 116 | 79 | 50 | 15 | 0 | 19 | 56 | 42 |
| 9 | 57 | 56 | 90 | 105 | 67 | 67 | 29 | 19 | 0 | 44 | 40 |
| 10 | 15 | 12 | 46 | 61 | 26 | 53 | 72 | 56 | 44 | 0 | 55 |
| 11 | 25 | 25 | 59 | 74 | 36 | 63 | 57 | 42 | 40 | 55 | 0 |

Tabel 3, menunjukkan jarak antara 11 node dimana setiap elemen menunjukkan jarak antar dua node yang berbeda, dengan baris menunjukkan node asal dan kolom menunjukkan node tujuan.

Berdasarkan data jarak antara Gudang dan pengecer dilakukan perhitungan menggunakan algoritma ABC. Tahapan-tahapan penyelesaiannya adalah sebagai berikut:

3.1 Fase Inisialisasi

Fase inisialisasi berfungsi untuk mendapatkan rute awal menggunakan pencarian secara acak menggunakan nilai *random* [10]. Setelah mendapatkan rute awal, langkah berikutnya adalah menghitung nilai *fitness* (total jarak) dari rute awal dan nilai *fitness* ini dicatat dalam Tabel 4.

Tabel 4. Fase Inisialisasi

| Populasi | Solusi Fase Inisialisasi | | | | | | | | | | | Total Jarak (<i>Fitness</i>) | |
|----------|--------------------------|----|----|----|----|---|----|----|----|----|----|--------------------------------|-----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | 1 |
| Bee 1 | 1 | 6 | 4 | 9 | 10 | 5 | 7 | 11 | 2 | 3 | 8 | 1 | 698 |
| Bee 2 | 1 | 2 | 6 | 9 | 8 | 7 | 3 | 5 | 10 | 4 | 11 | 1 | 511 |
| Bee 3 | 1 | 10 | 8 | 11 | 9 | 5 | 2 | 4 | 7 | 6 | 3 | 1 | 593 |
| Bee 4 | 1 | 4 | 11 | 6 | 10 | 9 | 7 | 8 | 3 | 2 | 5 | 1 | 510 |
| Bee 5 | 1 | 11 | 8 | 3 | 7 | 9 | 10 | 5 | 4 | 2 | 6 | 1 | 589 |
| Bee 6 | 1 | 4 | 3 | 5 | 6 | 2 | 10 | 7 | 11 | 9 | 8 | 1 | 483 |
| Bee 7 | 1 | 4 | 2 | 8 | 7 | 6 | 10 | 3 | 11 | 9 | 5 | 1 | 562 |
| Bee 8 | 1 | 4 | 6 | 9 | 11 | 8 | 10 | 3 | 2 | 7 | 5 | 1 | 632 |
| Bee 9 | 1 | 3 | 8 | 9 | 10 | 6 | 2 | 7 | 4 | 5 | 11 | 1 | 638 |
| Bee 10 | 1 | 9 | 3 | 11 | 8 | 4 | 7 | 2 | 10 | 6 | 5 | 1 | 695 |
| Bee 11 | 1 | 10 | 2 | 4 | 3 | 5 | 6 | 11 | 9 | 7 | 8 | 1 | 400 |

Dalam Tabel 4, setiap baris mewakili satu lebah (Bee) dalam populasi. Setiap lebah mencoba menginisialisasi solusi awalnya sendiri dengan mengacak urutan kunjungan node-node tersebut. Dengan syarat rute dimulai dari node 1 dan berhenti di node 1.

Sebagai contoh Bee 1 mengikuti urutan kunjungan dari Node 1 (Gudang) ke node 6 ke node 4, dan seterusnya hingga kembali ke Node 1 (Gudang), maka total jarak yang ditempuh adalah 698 km. Karena fase inialisasi sudah selesai maka proses selanjutnya adalah melakukan yang namanya fase *Employed Bee*.

3.2 Fase *Employed Bee*

Setelah selesai dengan fase inialisasi, langkah selanjutnya adalah memasuki fase *Employed Bee*. Tujuan dari fase *Employed Bee* adalah untuk melakukan operasi *swap*, yaitu menukar dua posisi indeks secara acak dalam solusi [11]. Pada fase inialisasi sebelumnya, solusi baru telah diperoleh melalui proses *swap operator*, dan selanjutnya dilakukan perhitungan nilai *fitness* untuk mendapatkan hasil seperti yang dijelaskan dalam Tabel 5.

Tabel 5. Proses *Employed Bee*

| Populasi | Indeks | Solusi Fase <i>Employed Bee</i> | | | | | | | | | | | Total Jarak (<i>Fitness</i>) | |
|----------|---------|---------------------------------|----|----|----|----|----|----|----|----|----|----|-----------------------------------|-----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | |
| Bee 1 | (3,9) | 1 | 6 | 2 | 9 | 10 | 5 | 7 | 11 | 4 | 3 | 8 | 1 | 631 |
| Bee 2 | (10,11) | 1 | 2 | 6 | 9 | 8 | 7 | 3 | 5 | 10 | 11 | 4 | 1 | 537 |
| Bee 3 | (4,8) | 1 | 10 | 8 | 4 | 9 | 5 | 2 | 11 | 7 | 6 | 3 | 1 | 637 |
| Bee 4 | (9,10) | 1 | 4 | 11 | 6 | 10 | 9 | 7 | 8 | 2 | 3 | 5 | 1 | 494 |
| Bee 5 | (7,9) | 1 | 11 | 8 | 3 | 7 | 9 | 4 | 5 | 10 | 2 | 6 | 1 | 609 |
| Bee 6 | (4,5) | 1 | 4 | 3 | 6 | 5 | 2 | 10 | 7 | 11 | 9 | 8 | 1 | 493 |
| Bee 7 | (3,8) | 1 | 4 | 5 | 8 | 7 | 6 | 10 | 3 | 11 | 9 | 2 | 1 | 523 |
| Bee 8 | (2,8) | 1 | 3 | 6 | 9 | 11 | 8 | 10 | 4 | 2 | 7 | 5 | 1 | 632 |
| Bee 9 | (4,9) | 1 | 3 | 8 | 4 | 10 | 6 | 2 | 7 | 9 | 5 | 11 | 1 | 667 |
| Bee 10 | (5,7) | 1 | 9 | 3 | 11 | 7 | 4 | 8 | 2 | 10 | 6 | 5 | 1 | 694 |
| Bee 11 | (2,6) | 1 | 5 | 2 | 4 | 3 | 10 | 6 | 11 | 9 | 7 | 8 | 1 | 416 |

Pada Tabel 5, terlihat bahwa Bee 1 yang terdapat pada Tabel 4 akan melakukan *swap* dengan indeks (3,9) artinya akan menukarkan posisi 3 menjadi posisi 9 dan posisi 9 menjadi posisi 3.

Selanjutnya akan dilakukan *greedy selection* antara nilai *fitness* pada fase inialisasi dan nilai *fitness* pada fase *Employed Bee*. Proses *greedy selection* pada fase *Employed Bee* dilakukan dengan membandingkan nilai *fitness* dari fase inialisasi dengan fase *Employed Bee* untuk setiap Bee. Solusi yang dipilih adalah solusi dengan nilai *fitness* terendah. Jika nilai *fitness* dari fase *Employed Bee* lebih rendah dari nilai *fitness* fase inialisasi, maka solusi dari fase *Employed Bee* dipilih sebagai solusi pada proses *greedy selection* tersebut. Hasil perbandingan antara nilai *fitness* fase inialisasi dan nilai *fitness* fase *Employed Bee* dijabarkan pada Tabel 6.

Tabel 6. Hasil *Greedy Selection* Nilai *Fitness* Fase Inialisasi dan Fase *Employed*

| Populasi | <i>Fitness</i> | <i>Fitness</i> | Hasil | <i>trial</i> |
|----------|-----------------|--------------------------|-------------------------|--------------|
| | Fase Inialisasi | Fase <i>Employed Bee</i> | | |
| Bee 1 | 698 | 631 | 631 | 0 |
| Bee 2 | 511 | 537 | 511 | 1 |
| Bee 3 | 593 | 637 | 593 | 1 |
| Populasi | <i>Fitness</i> | <i>Fitness</i> | Hasil | <i>trial</i> |
| | Fase Inialisasi | Fase <i>Employed Bee</i> | <i>Greedy Selection</i> | |
| Bee 4 | 510 | 494 | 494 | 0 |
| Bee 5 | 589 | 609 | 589 | 1 |
| Bee 6 | 483 | 493 | 483 | 1 |

| | | | | |
|---------------|-----|-----|-----|---|
| Bee 7 | 562 | 523 | 523 | 0 |
| Bee 8 | 632 | 632 | 632 | 0 |
| Bee 9 | 638 | 667 | 638 | 1 |
| Bee 10 | 695 | 694 | 694 | 0 |
| Bee 11 | 400 | 416 | 400 | 1 |

Berdasarkan *greedy selection* dapat disimpulkan jika nilai *fitness* pada fase *Employed Bee* lebih kecil sama daripada nilai *fitness* pada fase inialisasi maka *trial* akan di *reset* 0. Tetapi jika nilai *fitness* pada fase *Employed Bee* lebih besar daripada fase inialisasi maka nilai *trial* akan ditambahkan 1. Sebagai contoh, nilai *fitness* Bee 1 pada fase inialisasi yaitu 698 sedangkan pada fase *Employed Bee* 631 karena pada fase *Employed Bee* nilai *fitness* Bee 1 lebih rendah dari nilai *fitness* pada fase inialisasi maka nilai *trial*nya direset menjadi 0. Pada Bee 2 ternyata nilai *fitness* *Employed Bee* lebih tinggi dari pada nilai *fitness* fase inialisasi maka nilai *trial*nya menjadi 1, dan seterusnya. Setelah fase *Employed Bee* selesai maka dilakukan fase *Onlooker Bee* untuk mendapatkan solusi baru.

3.3 Fase *Onlooker Bee*

Pada fase *Onlooker Bee* nilai *fitness* yang didapatkan dari fase *Employed Bee* akan dilakukan perhitungan nilai probabilitas dari nilai *fitness* dari hasil *greedy selection* pada fase *Employed Bee* untuk memilih satu solusi terbaik berdasarkan peluang dengan persamaan 1 [12]. dengan dibangkitkan nilai variabel *random* bernilai antara [0,1] maka hasilnya didapat disajikan pada Tabel 7.

Tabel 7. Pemilihan Populasi yang akan Diperbarui

| Populasi | Probabilitas | Nilai <i>Random</i> |
|---------------|--------------|---------------------|
| Bee 1 | 0,1020 | 0,0962 |
| Bee 2 | 0,0826 | 0,2234 |
| Bee 3 | 0,0958 | 0,0326 |
| Bee 4 | 0,0798 | 0,1739 |
| Bee 5 | 0,0952 | 0,0643 |
| Bee 6 | 0,0781 | 0,0918 |
| Bee 7 | 0,0845 | 0,0217 |
| Bee 8 | 0,1021 | 0,0557 |
| Bee 9 | 0,1031 | 0,0197 |
| Bee 10 | 0,1122 | 0,0456 |
| Bee 11 | 0,0646 | 0,0821 |

Tabel 7 merupakan proses seleksi populasi yang akan diperbarui melalui perbandingan nilai probabilitas dengan nilai *random* yang didapat. Jika nilai *random* didapat lebih kecil daripada nilai probabilitas maka akan dicari solusi baru oleh populasi. Tetapi jika nilai *random* didapat lebih besar sama maka tidak akan dilakukan pencarian solusi baru oleh *Onlooker Bee*. Sebagai contoh, nilai *random* dari Bee 1 lebih kecil dari nilai probabilitasnya maka Bee 1 melakukan pencarian solusi baru dengan *swap operator*. Bee 2 tidak melakukan pencarian solusi baru dikarenakan nilai *random* dari Bee 2 lebih besar dari nilai probabilitasnya, dan seterusnya. Didapat 7 populasi yang melakukan pencarian solusi baru yaitu Bee 1, Bee 3, Bee 5, Bee 7, Bee 8, Bee 9, dan Bee 10.

Setelah ditentukan populasi mana yang akan dilakukan pencarian solusi baru, maka selanjutnya dilakukan *swap operator* seperti pada fase *Employed Bee* untuk populasi yang terpilih yaitu Bee 1,

Bee 3, Bee 5, Bee 7, Bee 8, Bee 9, dan Bee 10. Untuk populasi yang tidak melakukan perubahan solusi maka populasinya tetap, artinya indeksnya (0,0) tidak dilakukan perubahan prosesnya. Setelah mendapatkan solusi baru dari *swap operator*, selanjutnya akan dilakukan perhitungan nilai *fitness*, dan didapatkan hasil yang akan dijabarkan pada Tabel 8.

Tabel 8. Proses Fase *Onlooker Bee*

| Populasi | Indeks | Solusi Fase <i>Onlooker Bee</i> | | | | | | | | | | | Total Jarak (<i>Fitness</i>) | |
|----------|--------|---------------------------------|-----------|-----------|----|----------|----------|----|----------|----------|----------|----------|-----------------------------------|-----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | |
| Bee 1 | (2,8) | 1 | 11 | 2 | 9 | 10 | 5 | 7 | 6 | 4 | 3 | 8 | 1 | 601 |
| Bee 2 | (0,0) | 1 | 2 | 6 | 9 | 8 | 7 | 3 | 5 | 10 | 4 | 11 | 1 | 511 |
| Bee 3 | (5,10) | 1 | 10 | 8 | 11 | 6 | 5 | 2 | 4 | 7 | 9 | 3 | 1 | 585 |
| Bee 4 | (0,0) | 1 | 4 | 11 | 6 | 10 | 9 | 7 | 8 | 2 | 3 | 5 | 1 | 494 |
| Bee 5 | (11,5) | 1 | 11 | 8 | 3 | 6 | 9 | 10 | 5 | 4 | 2 | 7 | 1 | 655 |
| Bee 6 | (0,0) | 1 | 4 | 3 | 5 | 6 | 2 | 10 | 7 | 11 | 9 | 8 | 1 | 483 |
| Bee 7 | (11,6) | 1 | 4 | 5 | 8 | 7 | 2 | 10 | 3 | 11 | 9 | 6 | 1 | 556 |
| Bee 8 | (2,10) | 1 | 7 | 6 | 9 | 11 | 8 | 10 | 4 | 2 | 3 | 5 | 1 | 534 |
| Bee 9 | (2,8) | 1 | 7 | 8 | 9 | 10 | 6 | 2 | 3 | 4 | 5 | 11 | 1 | 430 |
| Bee 10 | (3,9) | 1 | 9 | 10 | 11 | 7 | 4 | 8 | 2 | 3 | 6 | 5 | 1 | 700 |
| Bee 11 | (0,0) | 1 | 10 | 2 | 4 | 3 | 5 | 6 | 11 | 9 | 7 | 8 | 1 | 400 |

Pada tabel 8 proses yang dilakukan sama dengan pada fase *Employed Bee* yaitu menggunakan *swap operator*, tetapi pada fase *Onlooker Bee* yang dilakukan pencarian solusi baru yaitu populasi yang telah dipilih melalui seleksi perbandingan nilai probabilitas dengan nilai *random*.

Selanjutnya dilakukan *greedy selection* antara nilai *fitness* pada fase *Employed Bee* dan nilai *fitness* pada fase *Onlooker Bee*. Proses *greedy selection* dilakukan dengan membandingkan nilai *fitness* dari fase *Employed Bee* dan fase *Onlooker Bee* untuk setiap Bee. Solusi yang dipilih adalah solusi dengan nilai *fitness* terendah. Jika nilai *fitness* dari fase *Onlooker Bee* lebih rendah dari nilai *fitness* fase *Employed Bee*, maka solusi dari fase *Onlooker Bee* dipilih sebagai solusi pada proses *greedy selection* tersebut. Hasil perbandingan antara nilai *fitness* fase *Employed Bee* dan nilai *fitness* fase *Onlooker Bee* dijabarkan pada Tabel 9.

Tabel 9. Hasil *Greedy Selection* Nilai *Fitness* Fase *Employed* dan Fase *Onlooker*

| Populasi | <i>Fitness</i> | <i>Fitness</i> | Hasil | <i>trial</i> |
|----------|--------------------------|--------------------------|-------------------------|--------------|
| | Fase <i>Employed Bee</i> | Fase <i>Onlooker Bee</i> | <i>Greedy Selection</i> | |
| Bee 1 | 631 | 601 | 601 | 0 |
| Bee 2 | 511 | 511 | 511 | 1 |
| Bee 3 | 593 | 585 | 585 | 1 |
| Bee 4 | 494 | 494 | 494 | 0 |
| Bee 5 | 589 | 655 | 589 | 1 |
| Bee 6 | 483 | 483 | 483 | 1 |
| Populasi | <i>Fitness</i> | <i>Fitness</i> | Hasil | <i>trial</i> |
| | Fase <i>Employed Bee</i> | Fase <i>Onlooker Bee</i> | <i>Greedy Selection</i> | |
| Bee 7 | 523 | 556 | 523 | 0 |
| Bee 8 | 632 | 534 | 534 | 0 |
| Bee 9 | 638 | 430 | 430 | 1 |
| Bee 10 | 694 | 700 | 694 | 0 |

| | | | | |
|---------------|-----|-----|-----|---|
| Bee 11 | 400 | 400 | 400 | 1 |
|---------------|-----|-----|-----|---|

Berdasarkan *greedy selection* dapat disimpulkan jika nilai *fitness* pada fase *Onlooker Bee* lebih kecil sama daripada nilai *fitness* pada fase *Employed Bee* maka *trial* akan di *reset* 0. Tetapi jika nilai *fitness* pada fase *Onlooker Bee* lebih besar daripada fase *Employed Bee* maka nilai *trial* akan ditambahkan 1. Sebagai contoh, nilai *fitness* Bee 1 pada fase *Employed Bee* yaitu 631 sedangkan pada fase *Onlooker Bee* 601 karena pada fase *Onlooker Bee* nilai *fitness* Bee 1 lebih rendah dari nilai *fitness* pada fase *Employed Bee* maka nilai *trial*nya direset menjadi 0. Pada Bee 2 ternyata nilai *fitness* *Onlooker Bee* lebih tinggi dari pada nilai *fitness* fase *Employed Bee* maka nilai *trial*nya menjadi 1, dan seterusnya. Setelah fase *Onlooker Bee* selesai maka dilakukan pencarian solusi mana yang tidak melakukan pembaruan, jika ada solusi yang tidak melakukan pembaruan solusi maka akan dilakukan fase *Scout Bee*.

3.4 Fase *Scout Bee*

Setelah melewati dua fase peningkatan solusi yaitu fase *Employed Bee* dan fase *Onlooker Bee*, selanjutnya akan masuk ke fase *Scout Bee* [12]. Jumlah *Scout Bee* bersifat dinamis tergantung pada jumlah populasi yang memiliki percobaan (*trial*) lebih dari batas maksimum yang telah ditetapkan [13]. Jika ada populasi yang memiliki nilai percobaan melebihi batas maksimum, maka solusi dari populasi tersebut akan dihapus dan digantikan dengan solusi baru yang dihasilkan melalui teknik pengacakan populasi secara acak [14]. Populasi yang mengalami perubahan tersebut akan digunakan untuk memperbarui jarak yang dihasilkan dan nilai percobaan akan diatur ulang kembali menjadi 0. Pada iterasi ini, "*Scout Bee*" akan melakukan tugasnya dengan memperbarui populasi dari "Bee 2", "Bee 5", "Bee 6", dan "Bee 11", karena populasi dari lebah-lebah tersebut memiliki nilai percobaan yang melebihi batas maksimum yang telah ditetapkan. Detail pembaruan solusi pada fase *Scout Bee* dijelaskan dalam Tabel 10.

Tabel 10. Fase *Scout Bee*

| Populasi | Solusi Fase <i>Scout Bee</i> | | | | | | | | | | | Total Jarak (<i>Fitness</i>) | |
|---------------|------------------------------|----|----|----|----|----|----|---|----|----|----|-----------------------------------|------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | |
| Bee 1 | 1 | 11 | 2 | 9 | 10 | 5 | 7 | 6 | 4 | 3 | 8 | 1 | 601 |
| Bee 2 | 1 | 5 | 6 | 7 | 8 | 9 | 3 | 2 | 10 | 4 | 11 | 1 | 443 |
| Bee 3 | 1 | 10 | 8 | 11 | 6 | 5 | 2 | 4 | 7 | 9 | 3 | 1 | 585 |
| Bee 4 | 1 | 4 | 11 | 6 | 10 | 9 | 7 | 8 | 2 | 3 | 5 | 1 | 494 |
| Bee 5 | 1 | 5 | 11 | 7 | 9 | 10 | 4 | 6 | 2 | 8 | 3 | 1 | 598 |
| Bee 6 | 1 | 11 | 9 | 7 | 8 | 10 | 4 | 2 | 3 | 6 | 5 | 1 | 459 |
| Bee 7 | 1 | 4 | 5 | 8 | 7 | 6 | 10 | 3 | 11 | 9 | 2 | 1 | 523 |
| Bee 8 | 1 | 7 | 6 | 9 | 11 | 8 | 10 | 4 | 2 | 3 | 5 | 1 | 534 |
| Bee 9 | 1 | 7 | 8 | 9 | 10 | 6 | 2 | 3 | 4 | 5 | 11 | 1 | 430 |
| Bee 10 | 1 | 9 | 3 | 11 | 7 | 4 | 8 | 2 | 10 | 6 | 5 | 1 | 694 |
| Bee 11 | 1 | 10 | 11 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 5 | 1 | 377 |

Tabel 10, merupakan hasil dari Fase *Scout Bee*. Pada *Scout Bee*, jika terdapat populasi yang mendapatkan solusi dengan nilai *fitness* yang sama atau lebih buruk dari solusi terbaik sebelumnya, maka populasi tersebut akan menjadi "*scout*" dan mencari solusi baru dengan cara menginisialisasi kembali solusinya. Sebagai contoh, Bee 2 nilai *fitness* awalnya 511 pada saat fase

Employed Bee didapat 531 tidak melakukan pembaruan solusi karena solusi awal lebih baik, dan pada saat fase *Onlooker Bee* 2 tidak melakukan pembaruan maka *Bee 2* menjadi *scout*. Maka *Bee 2* mencari solusi baru dengan cara menginisialisasi kembali solusinya. Didapat solusi baru yaitu rute 1 – 5 – 6 – 7 – 8 – 9 – 3 – 2 – 10 – 4 – 11 – 1, dengan nilai *fitness* 443 km.

Setelah melakukan fase *Scout Bee* pada iterasi pertama, proses iterasi akan berlanjut kembali ke fase *Employed Bee*, *Onlooker Bee*, dan *Scout Bee* sampai iterasi kesepuluh.

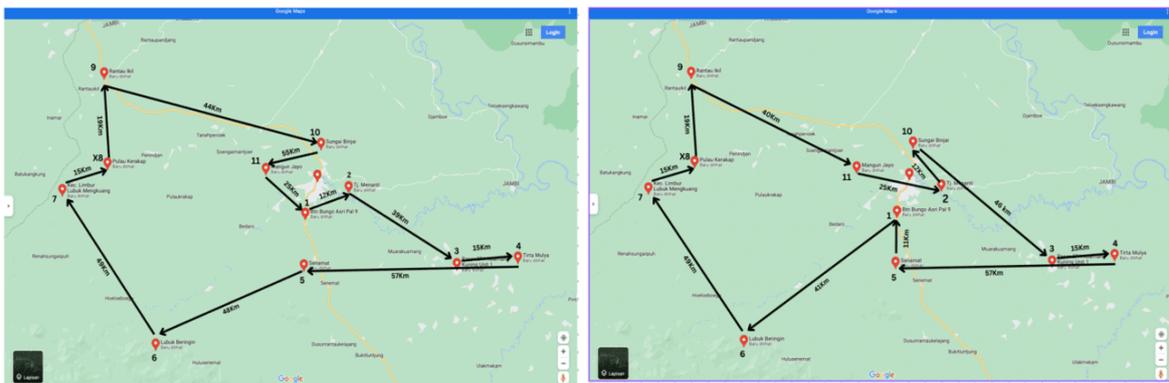
3.5 Pemilihan Sumber Makanan Terbaik

Setelah fase *Scout Bee* terlewati maka akan dilakukan kembali fase *Employed Bee* sampai memenuhi batas kriteria berhenti yaitu iterasi ke-10. Sumber makanan yang terbaik akan dipilih dan disimpan sebagai solusi terakhir untuk menghasilkan solusi – solusi lebih lanjut dalam upaya mencari solusi yang lebih mendekati solusi optimal [15]. Hasil dari proses 10 iterasi dapat dilihat pada Tabel 11.

Tabel 11. Hasil Proses Perhitungan 10 Iterasi ke-10

| Populasi | <i>Fitness ScoutBee</i> iterasi 1 | <i>Fitness ScoutBee</i> Iterasi 2 | <i>Fitness ScoutBee</i> iterasi 3 | <i>Fitness ScoutBee</i> iterasi 4 | <i>Fitness ScoutBee</i> iterasi 5 | <i>Fitness ScoutBee</i> iterasi 6 | <i>Fitness ScoutBee</i> iterasi 7 | <i>Fitness ScoutBee</i> iterasi 8 | <i>Fitness ScoutBee</i> iterasi 9 | <i>Fitness ScoutBee</i> iterasi 10 |
|----------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|------------------------------------|
| Bee 1 | 601 | 462 | 378 | 542 | 446 | 344 | 409 | 344 | 515 | 468 |
| Bee 2 | 443 | 443 | 393 | 501 | 415 | 370 | 370 | 370 | 509 | 501 |
| Bee 3 | 585 | 526 | 448 | 529 | 479 | 403 | 403 | 403 | 519 | 535 |
| Bee 4 | 494 | 352 | 330 | 507 | 451 | 429 | 397 | 397 | 396 | 499 |
| Bee 5 | 598 | 575 | 418 | 463 | 451 | 359 | 347 | 347 | 456 | 522 |
| Bee 6 | 459 | 393 | 346 | 455 | 338 | 338 | 419 | 419 | 398 | 438 |
| Bee 7 | 523 | 472 | 451 | 534 | 462 | 446 | 429 | 429 | 446 | 509 |
| Bee 8 | 534 | 461 | 448 | 534 | 462 | 433 | 365 | 365 | 477 | 576 |
| Bee 9 | 430 | 392 | 392 | 380 | 368 | 400 | 365 | 365 | 412 | 430 |
| Bee 10 | 694 | 660 | 580 | 428 | 412 | 402 | 397 | 344 | 428 | 663 |
| Bee 11 | 377 | 368 | 363 | 365 | 353 | 333 | 351 | 333 | 365 | 391 |

Tabel 11 memberikan informasi untuk setiap nilai *fitness* pada fase *Scout Bee* untuk 10 iterasi yang telah dilakukan, dimana Tabel 11 ini digunakan untuk menentukan sumber makanan atau solusi terbaik pada 10 iterasi yang telah dilakukan. Terlihat pada iterasi ketiga yaitu pencarian solusi oleh populasi 4 didapat nilai *fitness* paling rendah diantara semua nilai *fitness* yaitu 330 km, dengan rute yang berawal dari Pal 9 Muarabungo (Gudang), melewati Lubuk Beringin, Limbur, Pulau Kerapak, Rantau Ikil, Mangun Jayo, Tanjung Menanti, Sungai Binjai, Pasar SPA Unit 1, Tirta Mulya, Senamat, dan kembali ke Pal 9 Muarabungo (Gudang). Perbandingan antara rute yang biasa digunakan dengan rute optimal yang didapat menggunakan algoritma *Artificial Bee Colony* dapat dilihat pada Gambar 3(a) dan 3(b).



(a)

(b)

Gambar 3. (a) Rute Awal (b) Rute Optimal

Gambar 3(a) merupakan rute yang biasa digunakan oleh PT BDMU dengan total jarak 378 km, sedangkan pada Gambar 3(b) merupakan rute minimum yang didapat menggunakan algoritma *Artificial Bee Colony* yang lebih pendek 48 km dibandingkan rute biasa yang digunakan.

4. KESIMPULAN

Dari hasil penelitian dapat disimpulkan bahwa algoritma *Artificial Bee Colony* dapat diterapkan dalam menyelesaikan permasalahan optimasi rute terpendek jalur distribusi pupuk pada PT BDMU dan menghasilkan rute terpendek yang berawal dari Gudang, melewati Lubuk Beringin, Limbur, Pulau Kerakap, Rantau Ikil, Mangun Jayo, Tanjung Menanti, Sungai Binjai, Pasar SPA Unit 1, Tirta Mulya, Senamat, dan kembali ke Gudang, dengan total jarak 330 km. Rute optimal ini lebih pendek 48 km dibandingkan rute biasa yang digunakan oleh PT BDMU yang mencakup 378 km.

REFERENSI

- [1] Auliasari, K., Kertaningtyas, M., dan Basuki, D. W. (2018). *Optimalisasi Rute Distribusi Produk Menggunakan Metode Traveling Salesman Problem*. Jurnal Sains, Teknologi dan Industri, 16(1), 15-23.
- [2] Rizki, A. M., Mahmudy, W. F., dan Yuliastuti, G. E. (2017). *Optimasi Multi Travelling Salesman Problem (M-TSP) untuk Distribusi Produk pada Home Industri Tekstil dengan Algoritma Genetika*. Kumpulan Jurnal Ilmu Komputer (KLIK), 04(02), 125-135.
- [3] Utomo, R. G., Maylawati, D. S., dan Alam, C. N. (2018). *Implementasi Algoritma Cheapest Insertion Heuristic (CIH) dalam Penyelesaian Travelling Salesman Problem (TSP)*. JOIN (Jurnal Online Informatika), 3(1), 61-67.
- [4] Adventia, Agnes. (2018). *Penyelesaian Masalah Pendistribusian Barang Menggunakan Algoritma Bee Colony Optimization*. Eureka Matika, Vol. 6, No. 2, 64-72.
- [5] Karaboga, D., & Basturk, B. (2007). *A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm*. Journal of Global Optimization, 39(3), 459-471.
- [6] Furqan, Muhammad dkk. (2022). *Penerapan Algoritma Artificial Bee Colony Untuk Optimasi Rute Jalur Terpendek Distribusi Pipa Air Bersih*. Journal of Machine Learning and Artificial Intelligence. Vol. 1, No. 2, 125-132.
- [7] Chaerunnissa, Hilda., Respitawulan (2021). *Penyelesaian Travelling Salesman Problem dengan Menggunakan Algoritma Artificial Bee Colony*. Jurnal Riset Matematika, Vol. 1, No.1, 37-45.
- [8] Sugiyono. (2013). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta
- [9] Robandi, Imam. (2019). *Artificial Intelligence: Mengupas rekayasa kecerdasan tiruan*. Yogyakarta: ANDI.
- [10] Jamaluddin, Siti Hafawati, dkk. (2022). *Solving the Travelling Salesman Problem by Using Artificial Bee Colony Algorithm*. Journal of Computing Research and Innovation (JCRINN). Vol 7 No 2, 121-131.
- [11] Gutin, G. dan Punnen, A., (2004). *The Travelling Salesman Problem and Its Variations*. Dordrecht: Kluwe Academic Publishers.
- [12] A Akay, B., & Karaboga, D. (2012). *A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization*. Information Sciences (192), 120-142
- [13] Daniel B, Paillin and Johan M Tupan. 2018. *Pemecahan Travelling Salesman Problem Menggunakan Teknik Branch and Bound Dan Cheapest Insertion Heuristic (Studi Kasus : PT Paris Jaya Mandiri - Ambon)*.
- [14] Gutin, G. dan Punnen, A., (2004). *The Travelling Salesman Problem and Its Variations*. Dordrecht: Kluwe Academic Publishers.
- [15] Rahmalia., D., & Herlambang, T. (2017). *Optimasi Masalah Transportasi Distribusi Semen Menggunakan Algoritma Artificial Bee Colony*. Multitek Indonesia Jurnal Ilmiah, 11, 105-113.