

**Algoritma *artificial bee colony* pada *capacitated vehicle routing problem* (CVRP) dan implementasi programnya**

**Dewi Setyowati<sup>1</sup>, Purwanto<sup>2</sup>, dan Mahmuddin Yunus<sup>3</sup>**

**JURUSAN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS NEGERI MALANG**

E-mail: setyowatidewi15@gmail.com

**ABSTRAK** : *Capacitated Vehicle Routing Problem* (CVRP) adalah masalah pencarian sejumlah rute kendaraan berjarak minimum yang berawal dan berakhir di depot yang sama untuk memenuhi permintaan sejumlah *customer* yang hanya dikunjungi tepat satu kali dan tidak melebihi kapasitas kendaraan dengan perluasan setiap kendaraan memiliki kapasitas yang sama. Algoritma *Artificial Bee Colony* merupakan algoritma yang didasarkan pada kecerdasan kelompok dari perilaku lebah dalam mencari sumber makanan. Algoritma *Artificial Bee Colony* pada CVRP terdiri dari 7 langkah, yaitu inisialisasi parameter, membangkitkan solusi awal, tahap lebah pekerja, pemilihan solusi, tahap lebah penjaga, ganti solusi, dan tahap lebah pengintai. Dalam menyelesaikan permasalahan CVRP dengan menggunakan algoritma *Artificial Bee Colony* dibutuhkan proses yang panjang. Oleh sebab itu, untuk mempermudah pencarian rute, diimplementasikan ke dalam program dengan menggunakan bahasa pemrograman *Borland Delphi 7*.

**Kata kunci** : Algoritma *Artificial Bee Colony*, *Capacitated Vehicle Routing Problem* (CVRP).

**ABSTRACT** : *Capacitated Vehicle Routing Problem* (CVRP) is a problem to find some routes for vehicles which start and finish in depot to fulfill customer's demand which only be visited exactly once and cannot exceed the vehicle capacity with expansion of each vehicle which has same capacity. *Artificial Bee Colony* algorithm is an algorithm based on the intelligent foraging behavior of honey bee swarm *Artificial Bee Colony* algorithm on CVRP contains with 7 steps which are parameter initialization, raising initial solution, employed bee phase, solution selection, onlooker bee phase, solution changes and scout bee phase. In solving the CVRP problem using *Artificial Bee Colony* algorithm it needs a long process. Therefore, to simplify the search of route, it is implemented into a computer program using *Borland Delphi 7*.

**Keywords** : *Artificial Bee Colony* algorithm, *Capacitated Vehicle Routing Problem* (CVRP).

- 
1. Dewi Setyowati adalah mahasiswa jurusan Matematika FMIPA Universitas Negeri Malang
  2. Purwanto adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang
  3. Mahmuddin Yunus adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang

Teori graph merupakan salah satu cabang ilmu matematika yang dapat diaplikasikan dalam kehidupan sehari-hari. Pengaplikasian tersebut dapat dilakukan dengan memodelkan suatu masalah ke dalam model graph yang sesuai dan kemudian mencari penyelesaian yang tepat untuk masalah tersebut. Salah satu permasalahan yang dapat diselesaikan dalam bentuk graph adalah masalah pendistribusian barang dari depot ke *customer*. Konsep yang dapat diterapkan dalam menyelesaikan masalah pendistribusian adalah *Vehicle Routing Problem* (VRP). VRP merupakan masalah pencarian sejumlah rute kendaraan berjarak minimum yang berawal dan berakhir di depot yang sama untuk memenuhi permintaan sejumlah *customer* yang hanya dikunjungi tepat satu kali dan tidak melebihi kapasitas kendaraan. *Capacitated Vehicle Routing Problem* (CVRP) adalah salah satu varian dari VRP dengan perluasan setiap kendaraan memiliki kapasitas yang sama.

Algoritma *Artificial Bee Colony* merupakan algoritma berbasis populasi dimana posisi sumber makanan merupakan solusi yang mungkin untuk masalah optimasi dan jumlah nektar dari sumber makanan sesuai dengan kualitas (fitness) dari solusi terkait <sup>[1]</sup>. Pada algoritma *Artificial Bee Colony* (ABC), lebah dibagi menjadi 3 tipe, yaitu lebah pekerja (*employed bee*), lebah penjaga (*onlooker bee*), dan lebah pengintai (*scout bee*) <sup>[2]</sup>. Setiap tipe lebah memiliki peran atau tugas yang berbeda dalam pencarian makanan. Lebah pekerja bertugas untuk mencari sumber makanan dan menginformasikan tentang letak sumber makanan kepada lebah penjaga <sup>[3]</sup>. Lebah penjaga memilih sumber makanan yang diperoleh untuk dieksplorasi lebih lanjut. Lebah pengintai bertugas untuk menemukan sumber makanan. Cara kerja algoritma *Artificial Bee Colony* yang selalu menghasilkan solusi-solusi baru pada setiap tahap lebah dapat memberikan beberapa pilihan rute. Solusi dengan total jarak tempuh minimum akan terpilih sebagai solusi terbaik.

Berdasarkan uraian diatas, akan dibahas mengenai Algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP).

## HASIL YANG DIHARAPKAN

1. Mengetahui langkah-langkah algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP)
2. Menerapkan algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP) dalam menyelesaikan permasalahan pendistribusian elpiji 3 kg di CV. Sumber Pelita Gas
3. Mengimplementasikan algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP) dengan menggunakan bahasa pemrograman *Borland Delphi 7*

## HASIL DAN PEMBAHASAN

- ***Capacitated Vehicle Routing Problem* (CVRP)**

*Capacitated Vehicle Routing Problem* (CVRP) merupakan salah satu varian VRP dengan perluasan setiap kendaraan memiliki kapasitas yang sama. Tujuan dari permasalahan ini adalah meminimumkan jarak tempuh tiap kendaraan dengan memperhitungkan jumlah permintaan dalam setiap rute tidak melebihi kapasitas kendaraan.

Formulasi matematika untuk CVRP adalah sebagai berikut <sup>[4]</sup> :

a) Fungsi tujuan: Untuk meminimalkan total jarak tempuh

$$\min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ij}^k \quad (1)$$

b) Untuk memastikan bahwa setiap *customer* hanya dikunjungi tepat satu kali

$$\sum_{k \in K} \sum_{i \in N} X_{ij}^k = 1, \quad \forall j \in N/\{0\} \quad (2)$$

$$\sum_{k \in K} \sum_{j \in N} X_{ij}^k = 1, \quad \forall i \in N/\{0\} \quad (3)$$

c) Untuk menjamin bahwa setiap kendaraan harus meninggalkan *customer* yang telah dikunjungi

$$\sum_{i \in N} X_{ih}^k - \sum_{j \in N} X_{hj}^k = 0, \quad \forall h \in \frac{N}{\{0\}}, \forall k \in K \quad (4)$$

d) Untuk menjamin bahwa total permintaan dari setiap *customer* dalam satu rute tidak boleh melebihi kapasitas kendaraan

$$\sum_{j \in N} q_j \left( \sum_{i \in N} X_{ij}^k \right) \leq Q_k, \quad \forall k \in K \quad (5)$$

e) Untuk memastikan bahwa setiap kendaraan digunakan tidak lebih dari satu kali

$$\sum_{j \in N/\{0\}} X_{0j}^k \leq 1, \quad \forall k \in K \quad (6)$$

$$\sum_{i \in N/\{0\}} X_{i0}^k \leq 1, \quad \forall k \in K \quad (7)$$

f) Untuk menjamin variable keputusan  $X_{ij}^k$  merupakan integer biner

$$X_{ij}^k \in \{0,1\}, \quad \forall i, j \in N, k \in K \quad (8)$$

Keterangan:

- $k$  : kendaraan yang digunakan
- $q_j$  : permintaan *customer*  $j$  dengan  $j \in \{1,2, \dots, n\}$
- $Q_k$  : kapasitas kendaraan  $k$  dengan  $k \in \{1,2, \dots, n\}$
- $c_{ij}$  : jarak tempuh antara titik  $i$  dan  $j$
- $X_{ij}^k$  : integer biner, bernilai 1 jika kendaraan  $k$  dijalankan dari titik  $i$  ke titik  $j$ , dan bernilai 0 untuk kondisi lainnya

- **Algoritma Artificial Bee Colony pada Capacitated Vehicle Routing Problem (CVRP)**

Algoritma Artificial Bee Colony pada CVRP terdiri dari 7 langkah, yaitu inisialisasi parameter, membangkitkan solusi awal, tahap lebah pekerja, pemilihan solusi, tahap lebah penjaga, ganti solusi, dan tahap lebah pengintai.

### 1. Inisialisasi parameter

Parameter yang digunakan adalah banyak lebah dan banyak iterasi. Banyak lebah merupakan parameter untuk menentukan banyaknya solusi. Solusi tersebut terdiri dari beberapa rute yang digunakan untuk pendistribusian dari depot ke semua *customer*.

### 2. Membangkitkan solusi awal

Membangkitkan solusi awal adalah membangkitkan sejumlah sumber makanan. Banyak solusi awal yang dibangkitkan bergantung pada banyaknya lebah.

### 3. Tahap Lebah Pekerja

#### (i) *Swap Mutation I*

*Swap mutation I* merupakan proses pertukaran satu *customer* pada setiap dua rute berbeda yang terpilih secara acak di dalam satu solusi.

#### (ii) Seleksi Turnamen

Seleksi turnamen merupakan proses membandingkan dan menyeleksi sumber makanan yang lebih baik, antara sebelum *swap mutation I* dengan sesudah dilakukan *swap mutation I*. Dua solusi dibandingkan dan diterapkan kriteria berikut:

- Setiap solusi yang memenuhi kendala kapasitas lebih dipilih daripada solusi yang melebihi kendala kapasitas (*infeasible*).
- Diantara dua solusi yang memenuhi kendala kapasitas, pilih solusi yang memiliki total jarak tempuh yang lebih minimum.

(iii) Jika solusi yang baru memiliki total jarak tempuh lebih minimum, maka akan dilakukan perbaikan solusi dengan proses *Swap Mutation II*, yaitu proses pertukaran semua *customer* dalam setiap rute. Jika pertukaran tersebut menghasilkan jarak tempuh paling minimum, rute disusun ulang. Jika tidak, maka rute tidak berubah dan proses selanjutnya adalah menghitung fungsi tujuan, yaitu menghitung total jarak tempuh dari setiap solusi lebah pekerja.

### 4. Pemilihan solusi

Pemilihan solusi bertujuan untuk mengeksplor lebih lanjut beberapa sumber makanan terpilih dari solusi lebah pekerja untuk mendapatkan total jarak tempuh yang lebih minimum. Hitung nilai *fitness* dan nilai *probability* setiap sumber makanan dengan menggunakan rumus berikut:

$$fitness(x_i) = \frac{1}{total\ jarak\ tempuh} \quad (9)$$

$$p_{x_i} = \frac{fitness_{x_i}}{\sum_{i=1}^{\tau} fitness_{x_i}} \quad (10)$$

Keterangan:

- $fitness_{x_i}$  = nilai *fitness*  $x_i$  (solusi ke- $i$ )
- $p_{x_i}$  = nilai *probability*  $x_i$  (solusi ke- $i$ )
- $\sum_{i=1}^{\tau} fitness_{x_i}$  = jumlah dari nilai *fitness*  $x_i$  sampai  $x_{\tau}$

Kemudian bangkitkan bilangan *real* (0,1) secara *random* untuk menentukan sumber makanan yang akan dipilih oleh lebah penjaga untuk ditelusuri lebih lanjut.

## 5. Tahap Lebah Penjaga

### (i) *Insertion Mutation*

*Insertion mutation* merupakan proses penyisipan satu *customer* pada suatu rute ke rute berbeda yang terpilih di dalam satu solusi.

### (ii) Seleksi Turnamen

Seleksi turnamen merupakan proses membandingkan dan menyeleksi sumber makanan yang lebih baik, antara sebelum *insertion mutation* dengan sesudah dilakukan *insertion mutation*. Dua solusi dibandingkan dan diterapkan kriteria seperti pada tahap lebah pekerja

### (iii) Jika solusi yang baru memiliki total jarak tempuh lebih minimum, maka akan dilakukan perbaikan solusi dengan proses *Swap Mutation II* dan diterapkan aturan seperti pada tahap lebah pekerja.

## 6. Ganti Solusi

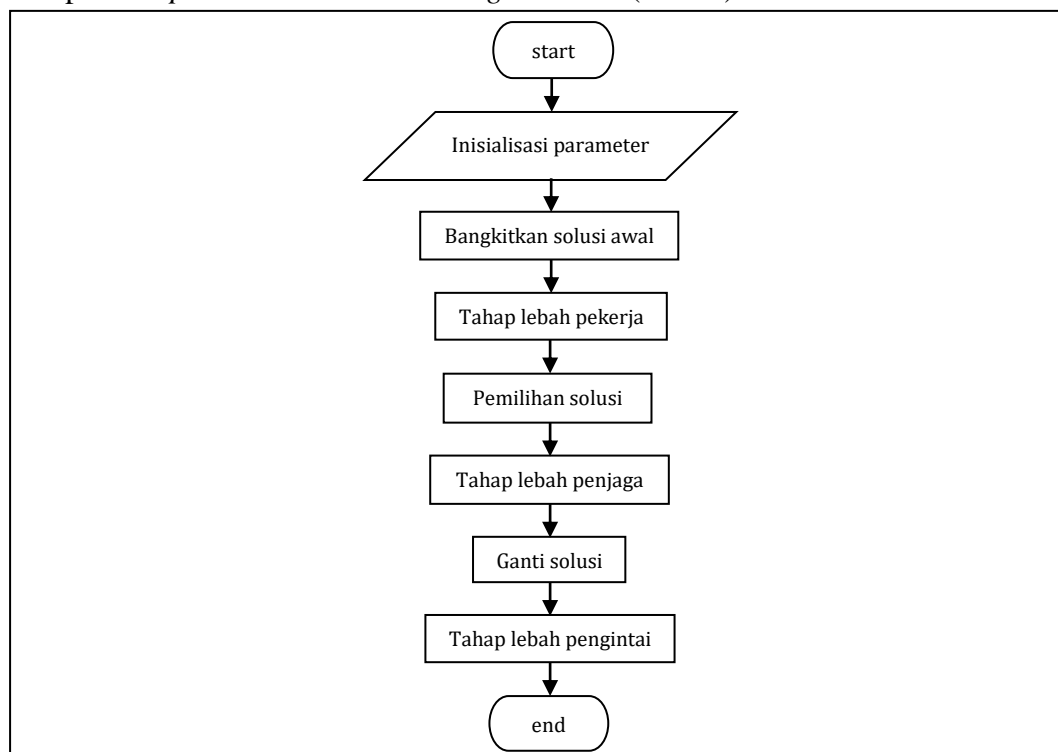
Bandungkan total jarak tempuh solusi terpilih dengan indeks yang sama kemudian pilih total jarak tempuh yang paling minimal. Selanjutnya ganti solusi lebah pekerja dengan solusi lebah penjaga yang bersesuaian.

## 7. Tahap Lebah Pengintai

Lebah pengintai memilih solusi yang memiliki total jarak tempuh paling minimal dan solusi inilah yang akan digunakan untuk rute pendistribusian dari depot ke semua *customer*.

Proses kembali menuju tahap lebah pekerja dan berulang sampai kriteria pembatas terpenuhi, dimana kriteria pembatas yang dimaksud adalah banyaknya iterasi.

Berikut ini adalah *Block Diagram* dari algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem (CVRP)*.



**Gambar 1. Block Diagram**

- **Contoh Penerapan**

CV. Sumber Pelita Gas akan melakukan pengiriman elpiji 3 kg kepada 8 *customer* yang berada di beberapa wilayah Kabupaten Blitar dan sekitarnya. Dalam kegiatan pendistribusian, agen ini menggunakan *pick-up* berkapasitas 75 tabung elpiji 3 kg. Selanjutnya akan dicari penyelesaian berupa rute yang optimal sehingga agen tersebut dapat melayani semua *customer*. Adapun daftar permintaan tiap *customer* serta jarak antara depot dengan *customer* dan antar *customer* adalah sebagai berikut :

**Tabel 1. Permintaan *customer***

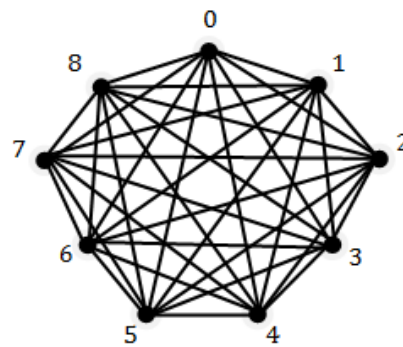
Customer	1	2	3	4	5	6	7	8
Demand	35	22	28	14	26	18	33	21

**Tabel 2. Jarak (km) Depot ke Customer dan Antar Customer**

$c_{ij}$	0	1	2	3	4	5	6	7	8
0	0	14	8	19	21	30	9	13	28
1	14	0	40	18	25	12	28	31	14
2	8	40	0	5	15	39	12	4	8
3	19	18	5	0	10	22	51	36	7
4	21	25	15	10	0	6	15	41	60
5	30	12	39	22	6	0	28	33	2
6	9	28	12	51	15	28	0	8	17
7	13	31	4	36	41	33	8	0	29
8	28	14	8	7	60	2	17	29	0

**Keterangan:**

Untuk kolom pertama dan baris pertama, 0 menyatakan depot dan 1,2,...8 merupakan *customer*. Jarak dari 0 (depot) ke *customer* 1 adalah 14 km, jarak dari depot ke *customer* 2 adalah 8 km, dan seterusnya.



**Gambar 2. Model Graph  $K_9$**

Pada Gambar 2, depot dan *customer-customer* diwakili oleh titik dan jalan yang dilalui dari depot menuju *customer* atau dari suatu *customer* ke *customer* lainnya diwakili oleh sisi.

Berikut ini adalah penyelesaian permasalahan distribusi tersebut dengan menggunakan algoritma *Artificial Bee Colony*.

- Banyak lebah : 5
- Banyak iterasi : 1

**Tabel 3. Solusi awal**

$x_i$	Solusi awal
1	0 - 1 - 7 - 0 - 6 - 8 - 3 - 0 - 2 - 5 - 4 - 0
2	0 - 7 - 4 - 2 - 0 - 1 - 3 - 0 - 6 - 8 - 5 - 0
3	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
4	0 - 2 - 4 - 6 - 8 - 0 - 3 - 7 - 0 - 5 - 1 - 0
5	0 - 3 - 4 - 5 - 0 - 6 - 7 - 8 - 0 - 1 - 2 - 0

Solusi awal tersebut dilanjutkan dengan tahap lebah pekerja yang terdiri dari 3 proses, yaitu *swap mutation I*, seleksi turnamen, dan *swap mutation II*. Berikut adalah hasil dari setiap prosesnya.

**Tabel 4. Hasil Swap Mutation I**

$x_i'$	Swap Mutation I
1	0 - 1 - 3 - 0 - 6 - 8 - 7 - 0 - 2 - 5 - 4 - 0
2	0 - 7 - 4 - 2 - 0 - 5 - 3 - 0 - 6 - 8 - 1 - 0
3	0 - 3 - 7 - 2 - 0 - 1 - 8 - 0 - 4 - 6 - 5 - 0
4	0 - 2 - 4 - 3 - 8 - 0 - 6 - 7 - 0 - 5 - 1 - 0
5	0 - 3 - 4 - 5 - 0 - 6 - 7 - 2 - 0 - 1 - 8 - 0

**Tabel 5. Hasil Seleksi Turnamen pada Tahap Lebah Pekerja**

$x_i$	Solusi Seleksi Turnamen
1	0 - 1 - 7 - 0 - 6 - 8 - 3 - 0 - 2 - 5 - 4 - 0
2	0 - 7 - 4 - 2 - 0 - 1 - 3 - 0 - 6 - 8 - 5 - 0
3	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
4	0 - 2 - 4 - 6 - 8 - 0 - 3 - 7 - 0 - 5 - 1 - 0
5	0 - 3 - 4 - 5 - 0 - 6 - 7 - 2 - 0 - 1 - 8 - 0

**Tabel 6. Solusi Lebah Pekerja**

$x_i$	Solusi Swap Mutation II (Solusi Lebah Pekerja)
1	0 - 1 - 7 - 0 - 6 - 8 - 3 - 0 - 2 - 5 - 4 - 0
2	0 - 7 - 4 - 2 - 0 - 1 - 3 - 0 - 6 - 8 - 5 - 0
3	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
4	0 - 2 - 4 - 6 - 8 - 0 - 3 - 7 - 0 - 5 - 1 - 0
5	0 - 3 - 4 - 5 - 0 - 6 - 7 - 2 - 0 - 1 - 8 - 0

Kemudian hitung fungsi tujuan untuk setiap solusi lebah pekerja.

**Tabel 7. Fungsi Tujuan Solusi Lebah Pekerja**

$x_i$	Rute 1	Jarak 1	Rute 2	Jarak 2	Rute 3	Jarak 3	Total Jarak (km)
1	[0,1,7,0]	58	[0,6,8,3,0]	62	[0,2,5,4,0]	74	184
2	[0,7,4,2,0]	77	[0,1,3,0]	51	[0,6,8,5,0]	58	186
3	[0,3,7,4,0]	117	[0,1,8,0]	56	[0,2,6,5,0]	78	251
4	[0,2,4,6,8,0]	83	[0,3,7,0]	68	[0,5,1,0]	56	207
5	[0,3,4,5,0]	65	[0,6,7,2,0]	29	[0,1,8,0]	56	150

Langkah selanjutnya adalah pemilihan solusi dengan menghitung nilai *fitness*, nilai *probability*, dan menetapkan nilai *probability* kumulatif untuk setiap sumber makanan. Kemudian bangkitkan bilangan *random* sebanyak sumber makanan dan sesuaikan dengan nilai *probability* kumulatif sehingga diperoleh solusi yang terpilih untuk dieksplor lebih lanjut.

**Tabel 8. Nilai *fitness***

$x_i$	Total Jarak	Nilai <i>Fitness</i>
1	184	0.00544
2	186	0.00538
3	251	0.00398
4	207	0.00483
5	178	0.00562

**Tabel 9. Nilai *Probability***

$x_i$	Nilai <i>Probability</i>
1	0.21545
2	0.21307
3	0.15762
4	0.19129
5	0.22257

**Tabel 10. Nilai *Probability* Kumulatif**

$x_i$	Nilai <i>Probability</i> Kumulatif
1	0 – 0.21545
2	0.21546 – 0.42852
3	0.42853 – 0.58614
4	0.58615 – 0.77743
5	0.77744 – 1

**Tabel 11. Bilangan *random* dan Sumber Makanan yang bersesuaian**

Bilangan <i>random</i>	Sumber Makanan Terpilih
0.59717	$x_4$
0.43829	$x_3$
0.49683	$x_3$
0.63135	$x_4$
0.52741	$x_3$

**Tabel 12. Solusi Terpilih**

$x_i$	Solusi Terpilih
1	0 – 2 – 4 – 6 – 8 – 0 – 3 – 7 – 0 – 5 – 1 – 0
2	0 – 3 – 7 – 4 – 0 – 1 – 8 – 0 – 2 – 6 – 5 – 0
3	0 – 3 – 7 – 4 – 0 – 1 – 8 – 0 – 2 – 6 – 5 – 0
4	0 – 2 – 4 – 6 – 8 – 0 – 3 – 7 – 0 – 5 – 1 – 0
5	0 – 3 – 7 – 4 – 0 – 1 – 8 – 0 – 2 – 6 – 5 – 0

Solusi terpilih dilanjutkan dengan tahap lebah penjaga yang terdiri dari 3 proses, yaitu *insertion mutation*, seleksi turnamen, dan *swap mutation II*. Kemudian hitung fungsi tujuan untuk setiap solusi lebah penjaga. Berikut adalah hasil dari setiap prosesnya.

**Tabel 13. Hasil *Insertion Mutation***

$x_i'$	<i>Insertion Mutation</i>
1	0 – 2 – 6 – 8 – 0 – 3 – 4 – 7 – 0 – 5 – 1 – 0
2	0 – 3 – 7 – 4 – 0 – 2 – 1 – 8 – 0 – 6 – 5 – 0
3	0 – 8 – 3 – 7 – 4 – 0 – 1 – 0 – 2 – 6 – 5 – 0
4	0 – 2 – 4 – 8 – 0 – 3 – 7 – 0 – 5 – 6 – 1 – 0
5	0 – 3 – 7 – 0 – 1 – 8 – 4 – 0 – 2 – 6 – 5 – 0

**Tabel 14. Hasil Seleksi Turnamen pada Tahap Lebah Penjaga**

$x_i$	Solusi Seleksi Turnamen
1	0 - 2 - 6 - 8 - 0 - 3 - 4 - 7 - 0 - 5 - 1 - 0
2	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
3	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
4	0 - 2 - 4 - 6 - 8 - 0 - 3 - 7 - 0 - 5 - 1 - 0
5	0 - 3 - 7 - 0 - 1 - 8 - 4 - 0 - 2 - 6 - 5 - 0

**Tabel 15. Solusi Lebah Penjaga**

$x_i$	Solusi Lebah Penjaga setelah <i>Swap Mutation II</i>
1	0 - 2 - 8 - 6 - 0 - 4 - 3 - 7 - 0 - 5 - 1 - 0
2	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
3	0 - 3 - 7 - 4 - 0 - 1 - 8 - 0 - 2 - 6 - 5 - 0
4	0 - 2 - 4 - 6 - 8 - 0 - 3 - 7 - 0 - 5 - 1 - 0
5	0 - 3 - 7 - 0 - 8 - 1 - 4 - 0 - 2 - 6 - 5 - 0

**Tabel 16. Fungsi Tujuan Solusi Lebah Penjaga**

$x_i$	Rute 1	Jarak 1	Rute 2	Jarak 2	Rute 3	Jarak 3	Total Jarak (km)
1	[0,2,8,6,0]	42	[0,4,3,7,0]	80	[0,5,1,0]	56	178
2	[0,3,7,4,0]	117	[0,1,8,0]	56	[0,2,6,5,0]	78	251
3	[0,3,7,4,0]	117	[0,1,8,0]	56	[0,2,6,5,0]	78	251
4	[0,2,4,6,8,0]	83	[0,3,7,0]	68	[0,5,1,0]	56	207
5	[0,3,7,0]	68	[0,8,1,4,0]	88	[0,2,6,5,0]	78	234

Langkah selanjutnya adalah ganti solusi yang diawali dengan membandingkan total jarak tempuh sumber makanan terpilih dengan indeks yang sama dan pilih total jarak tempuh yang minimal. Kemudian ganti solusi lebah pekerja dengan solusi lebah penjaga yang bersesuaian

**Tabel 17. Sumber Makanan Terpilih**

Sumber Makanan Terpilih	Total Jarak Tempuh (km)
$x_4$	178
$x_3$	251
$x_3$	251
$x_4$	207
$x_3$	234

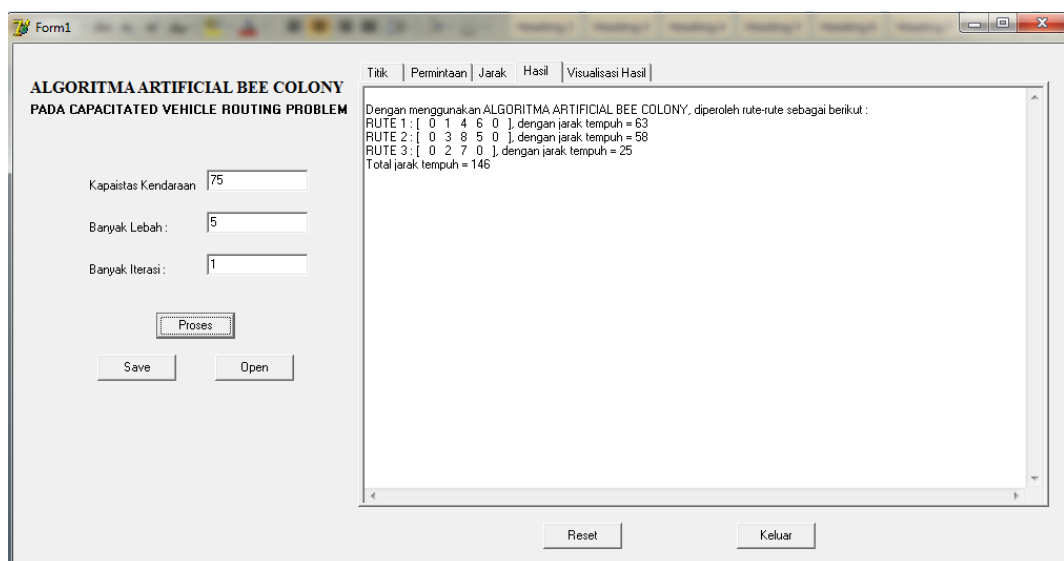
**Tabel 18. Solusi**

$x_i$	Solusi	Jarak Tempuh (km)
1	0 - 1 - 7 - 0 - 6 - 8 - 3 - 0 - 2 - 5 - 4 - 0	184
2	0 - 7 - 4 - 2 - 0 - 1 - 3 - 0 - 6 - 8 - 5 - 0	186
3	0 - 3 - 7 - 0 - 8 - 1 - 4 - 0 - 2 - 6 - 5 - 0	234
4	0 - 2 - 8 - 6 - 0 - 4 - 3 - 7 - 0 - 5 - 1 - 0	178
5	0 - 3 - 4 - 5 - 0 - 6 - 7 - 2 - 0 - 1 - 8 - 0	150

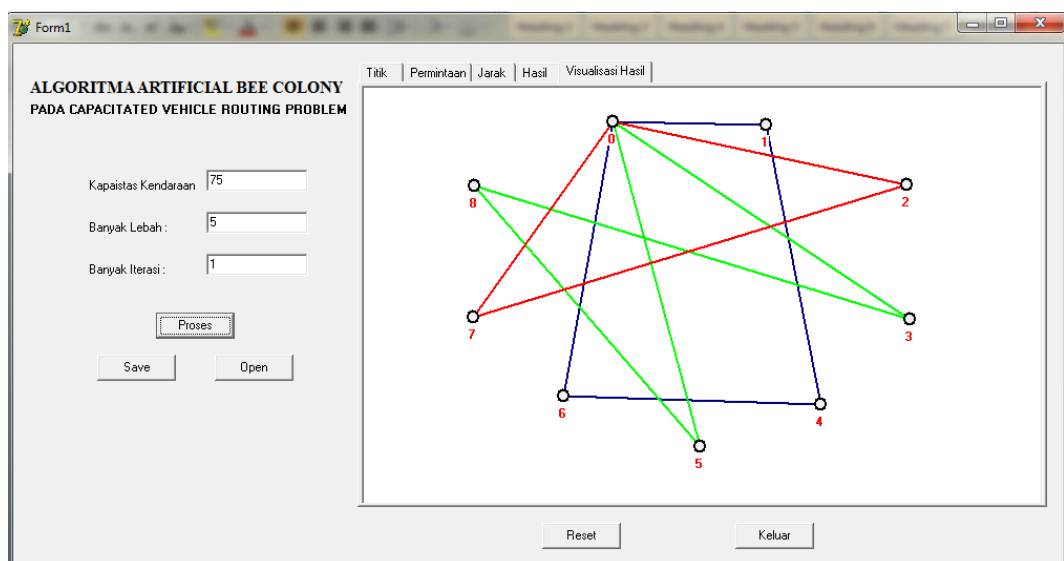
Berdasarkan hasil penyelesaiannya, solusi yang dipilih adalah  $x_5$  dengan rute 0 - 3 - 4 - 5 - 0 - 6 - 7 - 2 - 0 - 1 - 8 - 0 yang memiliki total jarak tempuh 150 km. Solusi tersebut dapat digunakan oleh CV. Sumber Pelita Gas untuk rute pendistribusian ke 8 *customer* tersebut.

Algoritma *Artificial Bee Colony* diimplementasikan ke dalam program komputer menggunakan bahasa pemrograman *Borland Delphi 7*. Untuk menjalankannya, diawali dengan menginputkan data seperti banyaknya titik dengan titik 0 sebagai depot dan titik lainnya sebagai *customer*, jarak antar titik, permintaan tiap *customer*, kapasitas kendaraan, dan parameter algoritma *Artificial Bee Colony*. Data tersebut diproses sesuai dengan langkah-langkah algoritma *Artificial Bee Colony*, yaitu inisialisasi parameter, membangkitkan solusi awal, tahap lebah pekerja, pemilihan solusi, tahap lebah penjaga, ganti solusi, dan tahap lebah pengintai. Output dari proses tersebut adalah urutan rute, jarak tempuh masing-masing rute, dan total jarak tempuh semua rute berupa teks tulisan serta urutan rute berupa visualisasi gambar.

Berikut ini adalah penyelesaian permasalahan distribusi tersebut dengan menggunakan program.



Gambar 3. Tampilan hasil perhitungan program



Gambar 4. Tampilan visualisasi hasil

Berdasarkan Gambar 3, diperoleh hasil berupa solusi terbaik dengan rute sebagai berikut:

1. Rute 1 : 0 – 1 – 4 – 6 – 0 dengan jarak tempuh 63 km.
2. Rute 2 : 0 – 3 – 8 – 5 – 0 dengan jarak tempuh 58 km.
3. Rute 3 : 0 – 2 – 7 – 0 dengan jarak tempuh 25 km.

Sehingga diperoleh total jarak tempuh 146 km. Sedangkan Gambar 4 adalah hasil gambar rute yang diperoleh pada hasil perhitungan Gambar 3.

Program yang merupakan implementasi dari algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP) ini telah diuji coba dengan menggunakan 9, 25, 50 dan 110 titik.

## **PENUTUP**

### **Kesimpulan**

Langkah-langkah algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP) diawali dengan langkah insialisasi parameter, yaitu banyak lebah dan banyak iterasi. Setelah itu masuk ke langkah selanjutnya yaitu membangkitkan solusi awal sebanyak lebah dengan menggunakan permutasi *Josephus* Kemudian dilanjutkan ke tahap lebah pekerja, pemilihan solusi, tahap lebah penjaga, ganti solusi, dan tahap lebah pengintai.

Dengan menerapkan algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP) dalam menyelesaikan permasalahan pendistribusian elpiji 3 kg di CV. Sumber Pelita Gas diperoleh solusi terbaik yang terdiri dari 3 rute dengan total jarak tempuh 150 km.

Algoritma *Artificial Bee Colony* pada *Capacitated Vehicle Routing Problem* (CVRP) dapat diimplementasikan dengan menggunakan bahasa pemrograman *Borland Delphi 7*. Langkah awal penggunaan program ini yaitu menginputkan data seperti menggambar titik pada tabsheet titik dengan titik 0 sebagai depot dan titik lainnya sebagai *customer*, jarak antar titik, permintaan tiap *customer*, dan kapasitas kendaraan. Setelah itu langkah algoritma *Artificial Bee Colony* dimulai dengan inisialisasi parameter, yaitu menginputkan banyak lebah dan banyak iterasi. Selanjutnya data tersebut akan diproses untuk membangkitkan solusi awal yang kemudian dilanjutkan dengan tahap lebah pekerja, pemilihan solusi, tahap lebah penjaga, ganti solusi, dan tahap lebah pengintai. Output yang ditampilkan pada program adalah solusi terbaik berupa teks tulisan yang berisi urutan rute dengan total jarak masing-masing rute dan total jarak solusi tersebut serta urutan rute berupa visualisasi gambar.

### **Saran**

Dalam penelitian ini tidak menjelaskan perbandingan algoritma *Artificial Bee Colony* dengan algoritma lainnya dalam menyelesaikan permasalahan CVRP. Oleh karena itu, diharapkan untuk menganalisa algoritma yang lain sehingga dapat menjadi pembanding atau bahkan mendapatkan alternatif rute dengan total jarak tempuh yang lebih minimal.

**DAFTAR PUSTAKA**

- [1] Szeto, W.Y., Wu, Yongzhong & Ho, Sin C. 2011. An Artificial Bee Colony Algorithm For The Capacitated Vehicle Routing Problem. *European Journal of Operational Research* 215 (2011) 126-135.
- [2] Pratama, Rendra F. 2012. *Penyelesaian Travelling Salesman Problem (TSP) dengan Menggunakan Artificial Bee Colony*. Skripsi tidak diterbitkan. Malang: Universitas Negeri Malang.
- [3] Karaboga, D. & Basturk, B. 2008. On The Performance Of Artificial Bee Colony (ABC) Algorithm. *Applied Soft Computing* 8 (2008) 687-697.
- [4] Brajevic, Ivona. 2011. Artificial Bee Colony Algorithm for The Capacitated Vehicle Routing Problem. *Proceedings of the European Computing Conference*.