

IMPLEMENTASI ALGORITMA BEE COLONY UNTUK OPTIMASI RUTE DISTRIBUSI CARICA NIDA FOOD WONOSOBO

¹Muhammad Darwis Arifin, ²Arif Dwi Laksito

^{1,2} Prodi Informatika, Fakultas Ilmu Komputer, Universitas Amikom,
Jl Ringroad Utara, Condong catur, Depok, Sleman 55283

Email: muhammad8110@students.amikom.ac.id, arif.laksito@amikom.ac.id

(Diterima: 19 Maret 2019, direvisi: 11 Mei 2019, disetujui: 25 Mei 2019)

ABSTRAK

“Nida Food” merupakan UMKM produksi carica, memiliki permasalahan dalam menentukan rute minimum untuk mendistribusikan produksi carica ke semua konsumen. Penyelesaian permasalahan ini secara manual dapat menghabiskan banyak waktu untuk menentukan solusi rute terdekat. Masalah Travelling Salesman Problem (TSP) dapat dikaitkan dimana seseorang akan mengunjungi ke sejumlah kota, dimana rangkaian kota-kota yang dikunjungi harus tepat satu kali dilewati dan pada akhirnya kembali lagi ke kota awal. Tujuan dari masalah TSP ini adalah untuk mencari rute atau jarak terpendek. Algoritma Artificial Bee Colony (ABC) merupakan salah satu algoritma yang dapat menyelesaikan permasalahan TSP. Penelitian ini menghasilkan aplikasi berbasis web untuk pencarian rute distribusi terdekat menggunakan algoritma Artificial Bee Colony, sehingga dapat memberikan solusi kepada salesman.

Kata Kunci: Travelling Salesman Problem, Artificial Bee Colony, Optimasi rute

1 PENDAHULUAN

Distribusi barang adalah suatu kegiatan yang dilakukan oleh distributor untuk mengirimkan barang ke beberapa konsumen. Meminimalkan biaya adalah salah satu faktor yang perlu dipertimbangkan dalam proses distribusi tersebut. Oleh karena itu pemilihan rute terpendek diperlukan dalam meminimalkan jalur pendistribusian. Masalah pendistribusian barang dapat dipandang sebagai masalah *Travelling Salesman Problem (TSP)*. TSP adalah salah satu bentuk masalah optimisasi yang bertujuan untuk menemukan rute perjalanan mulai dari lokasi awal, mengunjungi beberapa lokasi yang telah ditentukan, kemudian kembali lagi ke lokasi asal sedemikian rupa sehingga jarak total yang ditempuh minimum dan setiap lokasi dikunjungi tepat satu kali [1].

Nida Food merupakan UMKM produksi dan distributor carica dimana memiliki banyak konsumen di berbagai toko di Wonosobo. Nida Food memiliki permasalahan dalam menentukan rute minimum untuk mengantarkan produksi carica ke semua konsumen tepat satu kali dan kembali ke tempat asal. Penyelesaian permasalahan ini secara manual dapat menghabiskan banyak waktu untuk menentukan solusi rute terdekat.

Berdasarkan permasalahan tersebut, diperlukan metode yang dapat menghitung rute paling optimal dalam melakukan distribusi barang. Dalam penelitian ini akan digunakan metode *Artificial Bee Colony (ABC)* untuk mensolusikan masalah di Nida Food. Algoritma ini terinspirasi dari perilaku sekumpulan lebah berkelompok untuk mencari sumber makanan. Setelah mereka menemukan sumber makanan kemudian mereka akan kembali ke sarang dan melakukan tarian lebah (*waggle dance*), dengan menggunakan *waggle dance* semua koloni saling berkomunikasi tentang sumber makanan yang mereka temukan, sehingga lebah-lebah yang lain akan mengetahui letak dari sumber makanan yang paling dekat dari sarang [2].

Google Map API adalah salah satu layanan yang disediakan oleh Google untuk para pengembang yang akan mengembangkan aplikasi secara mandiri. Google telah memberikan dukungan yang baik untuk para pengembang dengan menyediakan dokumentasi yang lengkap [3]. Pada penelitian ini akan dibangun suatu program yang dapat mempercepat proses pencarian rute distribusi Nida Food menggunakan metode *Artificial Bee Colony* yang diimplementasikan menggunakan *Google Map* untuk menampilkan jalur distribusinya.

2 TINJAUAN PUSTAKA

Pada beberapa penelitian sebelumnya telah dilakukan kajian atau implementasi metode *Artificial Bee Colony*. Seperti yang dilakukan oleh Eno Rahmandha dengan membangun aplikasi pencarian lokasi fasilitas pelayanan umum terdekat menggunakan *Artificial Bee Colony*. Dimana penelitian tersebut menerapkan ABC untuk pencarian lokasi, pengujian dengan mengukur waktu komputasi dan jarak dari beberapa kombinasi parameter yg digunakan. Hasilnya adalah relevansi algoritma *Artificial Bee Colony* linier dengan jumlah lebah yang dipekerjakan dan iterasi yang ditetapkan, sedangkan parameter limit maksimum tidak begitu mempengaruhi [2]. Selain itu, Danuri juga melakukan penelitian dengan menerapkan *Artificial Bee Colony* untuk penentuan rute terpendek pada obyek wisata di Daerah Istimewa Yogyakarta. Pada penelitian ini *Artificial Bee Colony* dapat digunakan untuk penentuan rute terpendek dan banyaknya jumlah lebah yang dilepas akan mempengaruhi dalam menemukan rute-rute yang ditelusuri. Semakin banyak jumlah lebah dalam *colony* akan semakin besar pula peluang ditemukannya rute terbaik [4]. Penyelesaian masalah TSP menggunakan algoritma *Artificial Bee Colony* telah diteliti juga oleh Faisal Amri. Penelitian tersebut menggunakan data yang diambil dari TSPLIB dan menghasilkan kesimpulan bahwa semakin besar jumlah kota yang diproses maka tingkat kesalahan juga semakin meningkat [5].

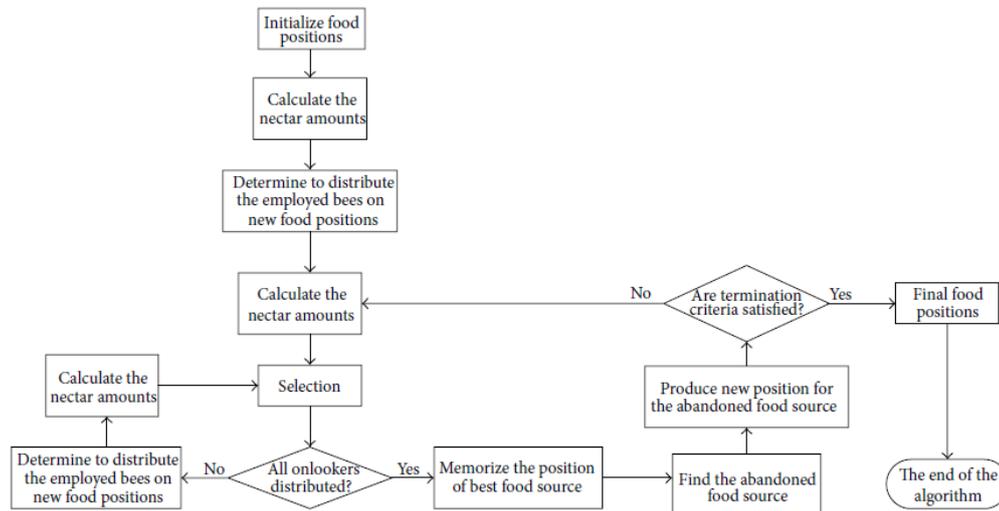
2.1 *Travelling Salesman Problem (TSP)*

Traveling Salesman Problem (TSP) adalah masalah yang solusinya telah banyak memunculkan matematikawan selama bertahun-tahun. Sampai dengan saat ini belum ada solusi yang memuaskan para ahli matematika tersebut. Matematika yang terkait dengan TSP ini dikembangkan sekitar tahun 1800-an oleh dua orang ahli matematika, Sir William Rowan Hamilton, berkebangsaan Irlandia dan Thomas Penyngton Kirkman, berkebangsaan Inggris. Hamilton adalah pencipta permainan *Icosian* pada tahun 1857, dimana pemain harus menyelesaikan perjalanan dari 20 titik hanya menggunakan jalur-jalur tertentu. Namun, bentuk umum TSP pertama kali dipelajari oleh Karl Menger di Wina dan Harvard pada akhir 1920-an hingga awal 1930-an [6].

2.2 *Artificial Bee Colony (ABC)*

Swarm intelligence telah didefinisikan oleh Bonabeau sebagai suatu percobaan untuk merancang algoritma atau mendistribusikan alat pemecahan masalah yang terinspirasi oleh tingkah laku kolektif dari suatu koloni sosial serangga dan hewan lainnya [7]. Pengertian *swarm* secara umum merujuk pada suatu koleksi pengendalian dari agen yang saling berinteraksi maupun individual. Salah satu algoritma yang merupakan *swarm intelligence* adalah *Artificial Bee Colony (ABC) Algorithm*.

Artificial Bee Colony (ABC) Algorithm adalah salah satu algoritma berdasarkan suatu pendekatan *population-based metaheuristic* yang diusulkan oleh Karaboga dan Basturk [8]. Cara kerja algoritma ini seperti pada kawanan lebah dalam koloni yang bekerjasama untuk mencari sumber makanan. Pada model algoritma ini membagi lebah menjadi 3 kelompok, yaitu: *employed bees*, *onlookers*, dan *scouts*. *Employed bee* disini merupakan lebah yang berhubungan dengan sumber makanan tertentu, *onlooker bee* bertugas memilih sumber makanan setelah menyaksikan tarian lebah dalam sarang, dan *scout bee* bertugas mencari sumber makanan secara acak. *Onlooker* dan *scout bee* merupakan *unemployed bee*. Proses awal dimulai dengan *scout bee* menemukan lokasi dari semua sumber makanan, selanjutnya *employed bee* secara probabilitas memperoleh beberapa modifikasi pada posisi dalam memori untuk menargetkan sumber makanan baru dan menemukan jumlah nektar atau nilai *fitness* dari sumber baru. Kemudian, *scout bee* melakukan evaluasi informasi yang telah diambil dari semua *employed bee* dan memilih sumber makanan akhir dengan nilai probabilitas tertinggi terkait dengan jumlah nektar tersebut. Jika nilai *fitness* yang baru lebih baik dari sebelumnya, lebah tersebut akan melupakan informasi lama dan menghafal posisi baru. Hal ini disebut sebagai *greedy selection*. Kemudian *employed bee* dengan sumber makanan yang telah habis akan menjadi *scout bee* untuk kembali mencari sumber makanan selanjutnya sekali lagi [5]. Pada Gambar 1 dibawah ini menunjukkan *flowchart* dari algoritma *Artificial Bee Colony*.



Gambar 1 Flowchart algoritma Artificial Bee Colony [6]

Adapun tahapan-tahapan dalam algoritma *Artificial Bee Colony* untuk penyelesaian masalah *Travelling Salesman Problem* adalah sebagai berikut [9]:

1. Pembentukan rute

Di dalam *Travelling Salesman Problem* untuk pembentukan rute digunakan metode *Nearest Neighbor*. Rute-rute yang terbentuk dijadikan sebagai sumber-sumber makanan sebagai acuan untuk penyelesaian *Travelling Salesman Problem* menggunakan *Artificial Bee Colony*.

2. Inisialisasi

Rute yang telah terbentuk pada tahapan pertama, akan diseleksi oleh para lebah untuk menentukan rute mana yang terbaik dan merupakan solusi dari permasalahan. Pada tahap ini akan diberikan nilai percobaan dari setiap kemungkinan solusi yaitu 0. Proses inisialisasi dari kemungkinan solusi (sumber makanan) dilakukan secara acak dengan menggunakan persamaan berikut :

$$X_{ij} = X_{jmin} + rand(0,1) \cdot (X_{jmax} - X_{jmin}) \quad (1)$$

Dimana:

- X_{ij} = inisialisasi kemungkinan solusi ke- i dengan parameter ke- j
- X_{jmin} = nilai kemungkinan solusi terkecil berdasarkan parameter j
- X_{jmax} = nilai kemungkinan solusi terbesar berdasarkan parameter j
- $rand(0,1)$ = nilai acak antara 0 sampai 1
- $i = 1.. SN$, SN adalah jumlah kemungkinan solusi (sumber makanan)
- $j = 1.. D$, D adalah jumlah parameter yang digunakan

3. Employed bee phase

Selanjutnya adalah tahap dimana para lebah memperluas nilai dari setiap kemungkinan solusi yang ada dengan menggunakan persamaan dibawah ini:

$$V_{ij} = X_{ij} + \phi_{ij} \cdot (x_{kj} - x_{ij}) \quad (2)$$

Dimana:

- V_{ij} = nilai perluasan kemungkinan solusi ke- i dengan parameter j
- X_{ij} = nilai kemungkinan solusi ke- i dengan parameter j
- $i = 1.. SN$, SN adalah jumlah kemungkinan solusi (sumber makanan)
- $j = 1.. D$, D adalah jumlah parameter yang digunakan
- $k = 1.. SN$, SN adalah jumlah parameter yang digunakan
- ϕ_{ij} = bilangan real acak antara [-1, 1]

Setelah setiap kemungkinan solusi diperluas, akan diaplikasikan *greedy selection* antara nilai kemungkinan solusi x_i dengan nilai baru hasil perluasan yaitu v_i . Jika nilai v_i lebih kecil dari nilai x_i maka nilai v_i tersebut akan dianggap sama dengan nilai x_i dan nilai percobaan tetap bernilai 0. Jika

tidak, nilai x_i yang disimpan dan nilai percobaan ke- i ditambah dengan 1. Proses ini berulang sampai jumlah perluasan sama dengan jumlah sumber makanan.

4. Tahap evaluasi populasi

Dari setiap kemungkinan solusi yang diperluas, akan dibandingkan dengan nilai awal inisialisasi. Selanjutnya dilakukan penghitungan kualitas dari masing-masing kemungkinan solusi menggunakan fungsi *fitness* sebagai berikut:

$$fitness(x_i) = \begin{cases} \frac{1}{(1+f(x_i))}, f(x_i) \geq 0 \\ 1 + |f(x_i)|, f(x_i) < 0 \end{cases} \quad (3)$$

Dimana:

- $i = 1.. SN$, SN adalah jumlah kemungkinan solusi (sumber makanan)

5. Onlooker bee phase

Setelah tahap lebah pekerja (*employed bee*) terpenuhi semua perluasan untuk setiap kemungkinan solusi dan telah dihitung masing-masing nilai probabilitasnya, informasi dari lebah pekerja akan diteruskan ke lebah penjaga (*onlooker bee*). Lebah penjaga akan menghitung nilai probabilitas berdasar informasi tersebut. Dari setiap kemungkinan solusi yang ada akan dihitung menggunakan persamaan berikut:

$$P_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (4)$$

Dimana:

- $fitness_i$ = nilai *fitness* solusi ke- i
- $\sum_{i=1}^{SN} fitness_i$ = jumlah dari nilai *fitness* ke- i sampai SN

Setelah nilai probabilitas dari setiap kemungkinan solusi dihitung, selanjutnya lebah penjaga memilih kemungkinan solusi selanjutnya yang akan ditelusuri oleh lebah pengintai (*scout bee*) dengan menggunakan metode *roulette-wheel*.

6. Scout bee phase

Dalam mengaplikasikan metode *roulette-wheel*, awalnya dipilih bilangan real secara acak antara [0,1] untuk setiap kemungkinan solusi. Jika nilai P_i lebih besar dari bilangan acak yang ditentukan, maka lebah penjaga akan menugaskan lebah pengintai untuk memperluas kembali kemungkinan solusi yang terpilih sesuai dengan tahap lebah pekerja sebelumnya. Setelah kemungkinan solusi terpilih diperluas, akan diaplikasikan *greedy selection* antara nilai kemungkinan solusi x_i dengan nilai baru hasil perluasan yaitu v_i . Jika nilai v_i lebih kecil dari nilai x_i maka nilai v_i tersebut akan dianggap sama dengan nilai x_i . Jika tidak, nilai v_i yang disimpan dan nilai percobaan ke- i ditambah dengan 1. Proses ini berulang sampai jumlah perluasan sesuai dengan kemungkinan solusi. Setelah semua kemungkinan solusi memiliki nilai percobaan, selanjutnya dipilih kemungkinan solusi dengan nilai percobaan maksimum dan dijadikan pilihan solusi terbaik. Proses kembali menuju tahap lebah pekerja dan berulang sampai kriteria pembatas terpenuhi, kriteria pembatas adalah jumlah lebah dalam *colony*.

2.3 Google Map

Untuk menampilkan daftar toko dan jalur rute yang terpilih akan digunakan layanan peta dari Google. *Google Maps Javascript API* memungkinkan kita untuk menyesuaikan peta dengan konten yang akan kita gunakan. Pada layanan ini Google menyediakan empat tipe map, yaitu: *roadmap*, *satellite*, *hybrid* dan *terrain*, dimana kita dapat memodifikasi menggunakan *layer* dan *style*, *controls* dan *events*, serta beberapa *library* tambahan [10]. Untuk menggunakan layanan dari Google ini, kita harus mendapatkan *API Key*, sehingga kita dapat menempatkan fitur peta google pada aplikasi mobile atau website. *API Key* digunakan untuk merekam permintaan kita ke server Google dimana nantinya akan berkaitan dengan biaya.

Dalam mengimplementasikan layanan ini, kita bisa menggunakan bahasa *javascript* dengan memanggil *library* secara online menggunakan kode berikut:

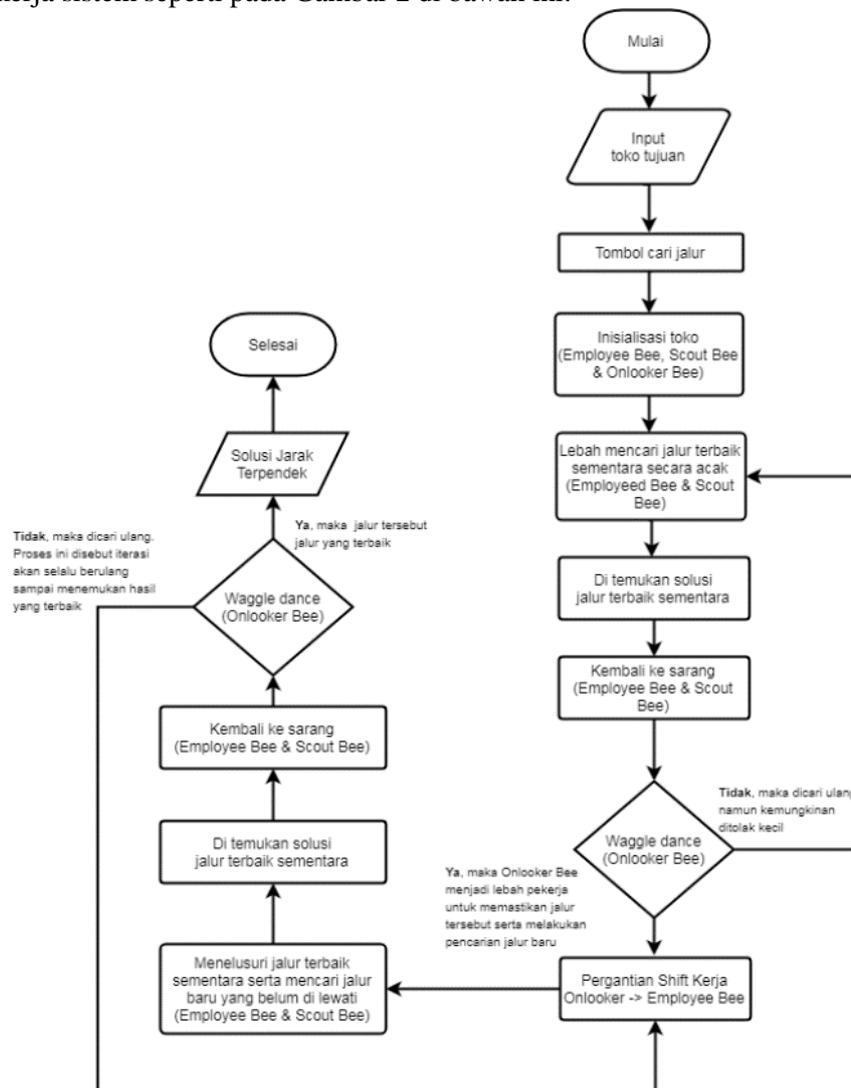
```
<script async defer src="https://maps.googleapis.com/maps/api/js?key=[YOUR_API_KEY]&callback=initMap"></script>
```

2.4 PHP dan MySQL

Program yang akan dibangun untuk pencarian rute distribusi Nida Food ini berbasis web menggunakan bahasa pemrograman PHP dan database Mysql. PHP (*HyperText Preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat situs web dinamis. PHP sering juga digunakan untuk membangun sebuah CMS. PHP didesain untuk pengembangan web. PHP adalah salah satu pemrograman web pada server yang banyak digunakan dan bersifat *open source*. MySQL merupakan sistem manajemen database yang bersifat *open source* dan populer digunakan. MySQL sudah mendukung beberapa fitur seperti *multithreaded*, *multi-user* dan *SQL Database Management System (DBMS)*. Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan [11].

3 METODE PENELITIAN

Penerapan Algoritma *Artificial Bee Colony* pada aplikasi berbasis web untuk menentukan jalur distribusi terpendek dari Nida Food menuju beberapa lokasi di Wonosobo digambarkan dalam diagram alur kerja sistem seperti pada Gambar 2 di bawah ini:



Gambar 2. Flowchart proses pemilihan jalur distribusi

Tahapan-tahapan proses pencarian jalur terdekat dengan algoritma *Artificial Bee Colony*, sebagai berikut :

1. Menentukan lokasi toko

Proses pertama untuk mencari jalur pada aplikasi ini adalah dengan menentukan lokasi toko mana saja yang akan di kunjungi, toko yang akan dikunjungi otomatis akan muncul di dalam google maps dalam bentuk marker map berwarna merah untuk menunjukkan lokasi toko tersebut.

2. Mulai proses pencarian

Pada proses pencarian dilakukan oleh lebah aktif (*employed bee*) dan lebah pencari (*scout bee*), mereka bekerja sama untuk mencari solusi jalur terbaik sementara. Proses penentuan nilai terbaik untuk masalah ini adalah nilai jarak antar lokasi toko, nilai jarak dapat di hitung melalui rumus pencarian jarak dengan menggunakan nilai *latitude* dan *longitude* pada google maps dengan rumus:

$$d_{ij} = (69 \cdot \sqrt{(lon_i - lon_j)^2 + (lat_i - lat_j)^2}) \cdot 1,60934 \quad (5)$$

Dimana:

- d_{ij} = jarak antara 2 titik koordinat i dan j (dalam km)
- lon_i = titik longitude i
- lon_j = titik longitude j
- lat_i = titik longitude i
- lat_j = titik longitude j

3. *Waggle dance*

Setelah perhitungan jalur terbaik sementara selesai proses selanjutnya adalah kembali ke sarang untuk memberi informasi kepada lebah nonaktif dengan tarian *waggle*, jika lebah nonaktif tertarik dengan informasi dari lebah penari maka selanjutnya adalah proses pergantian kerja pada proses iterasi, lebah pekerja dan lebah pencari berubah menjadi lebah nonaktif, sedangkan lebah nonaktif berubah menjadi lebah aktif.

4. Tahap Iterasi

Pada tahap iterasi, dilakukan proses pencarian jalur seperti informasi yang disampaikan oleh lebah penari, jika ditemukan jalur terbaik baru maka jalur tersebut akan menjadi prioritas atau menjadi informasi jalur baru, proses ini akan terus berlanjut hingga ditemukan nilai jalur terbaik yaitu jarak terdekat paling optimal antar toko.

4 HASIL DAN PEMBAHASAN

Pada penelitian ini, data lokasi yang akan ditelusuri diambil dari input pengguna berdasarkan *latitude* dan *longitude* dari peta. Data tersebut disimpan dalam table di database MySQL yang diimplementasikan menggunakan bahasa query seperti Gambar 3 di bawah ini:

```
CREATE TABLE IF NOT EXISTS `tb_daftar_toko` (
  `id_toko` int(11) NOT NULL,
  `nama` varchar(60) NOT NULL,
  `alamat` varchar(70) NOT NULL,
  `lat` float(10,6) NOT NULL,
  `lng` float(10,6) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=49 DEFAULT CHARSET=latin1;
```

Gambar 3. Query create table

Selanjutnya untuk proses distribusi dilakukan dengan memilih toko yang akan dikunjungi melalui aplikasi web kemudian dilakukan perhitungan jarak dari tiap-tiap lokasi toko. Berikut Tabel 1 berikut ini contoh untuk distribusi dari Nida Food ke tiga toko yaitu: Toko Sadina, Toko Bu Lukito dan Toko Tambi.

Tabel 1. Titik koordinat toko

| No. | Nama Toko | Latitude | Longitude |
|-----|----------------|-----------|------------|
| 1. | Nida Food | -7.334210 | 109.916000 |
| 2. | Toko Sadina | -7.387544 | 109.962189 |
| 3. | Toko Bu Lukito | -7.355452 | 109.902679 |
| 4. | Toko Tambi | -7.378833 | 109.897552 |

Tahap berikutnya dihitung jarak untuk tiap-tiap toko menggunakan persamaan nomer (5). Perhitungan jarak dari Nida Food ke Toko Sadina adalah sebagai berikut:

$$d = (69. \sqrt{(109,916000 - 109,962189)^2 + (-7,334210 + 7,387544)^2}).1,60934$$

$$d = (69. \sqrt{(0,00213342372)^2 + (0,00284451555)^2}).1,60934$$

$$4,86826138005 \times 1,60934 = 7,83468776937 \text{ km}$$

Dari hasil perhitungan jarak masing-masing toko dengan persamaan diatas, dapat diambil kesimpulan jarak seperti pada Tabel 2 di bawah ini.

Tabel 2. Jarak masing-masing toko

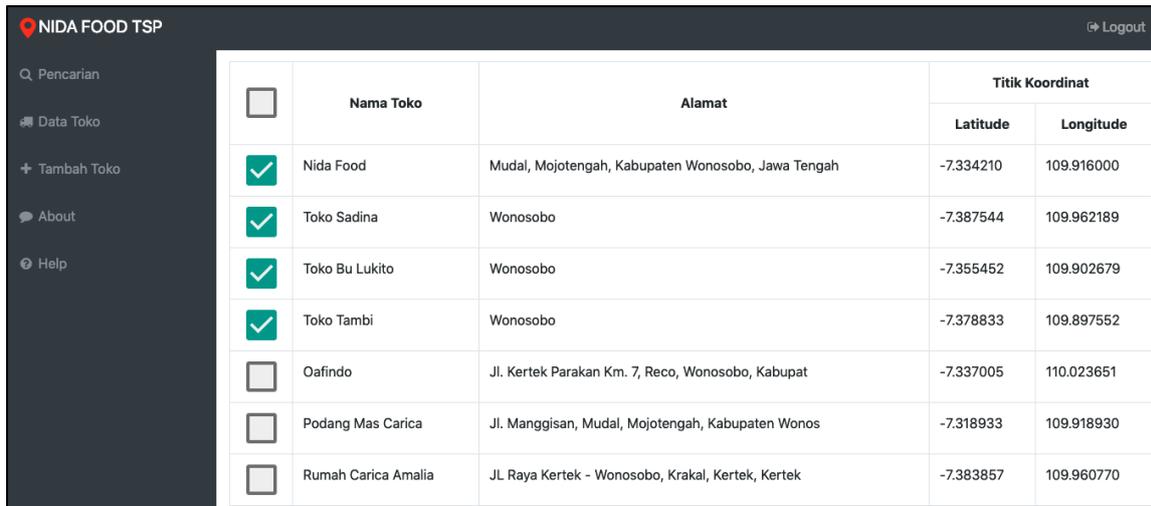
| | Nida Food | Bu Lukito | Tambi | Sadina |
|-----------|-----------|-----------|-------|--------|
| Nida Food | 0 | 2,78 | 5,36 | 7,83 |
| Bu Lukito | 2,78 | 0 | 2,66 | 7,50 |
| Tambi | 5,36 | 2,66 | 0 | 7,24 |
| Sadina | 7,83 | 7,50 | 7,24 | 0 |

Dengan menggunakan algoritma *Artificial Bee Colony* menggunakan parameter berikut ini diperoleh hasil perhitungan seperti pada Tabel 3. Parameter yang digunakan: (1) Total lebah= 10, (2) Total lebah aktif= 7, (3) Total lebah nonaktif= 1, (4) Total lebah pencari= 2, (5) Jumlah maksimal perjalanan= 10, (6) Jumlah iterasi= 500

Tabel 3. Proses pencarian rute

| Proses | Jalur | Nilai |
|----------|---|----------|
| Lebah 1 | Nida Food - Toko Tambi - Toko Sadina - Toko Bu Lukito - Nida Food | 20,88 km |
| Lebah 2 | Nida Food - Toko Sadina - Toko Bu Lukito - Toko Tambi - Nida Food | 23,35 km |
| Lebah 3 | Nida Food - Toko Sadina - Toko Bu Lukito - Toko Tambi - Nida Food | 23,35 km |
| Lebah 4 | Nida Food - Toko Bu Lukito - Toko Sadina - Toko Tambi - Nida Food | 20,88 km |
| Lebah 5 | Nida Food - Toko Tambi - Toko Sadina - Toko Bu Lukito - Nida Food | 20,88 km |
| Lebah 6 | Nida Food - Toko Sadina - Toko Tambi - Toko Bu Lukito - Nida Food | 20,51 km |
| Lebah 7 | Nida Food - Toko Sadina - Toko Tambi - Toko Bu Lukito - Nida Food | 20,51 km |
| Lebah 8 | Nida Food - Toko Tambi - Toko Sadina - Toko Bu Lukito - Nida Food | 20,88 km |
| Lebah 9 | Nida Food - Toko Bu Lukito - Toko Sadina - Toko Tambi - Nida Food | 20,88 km |
| Lebah 10 | Nida Food - Toko Bu Lukito - Toko Tambi - Toko Sadina - Nida Food | 20,51 km |

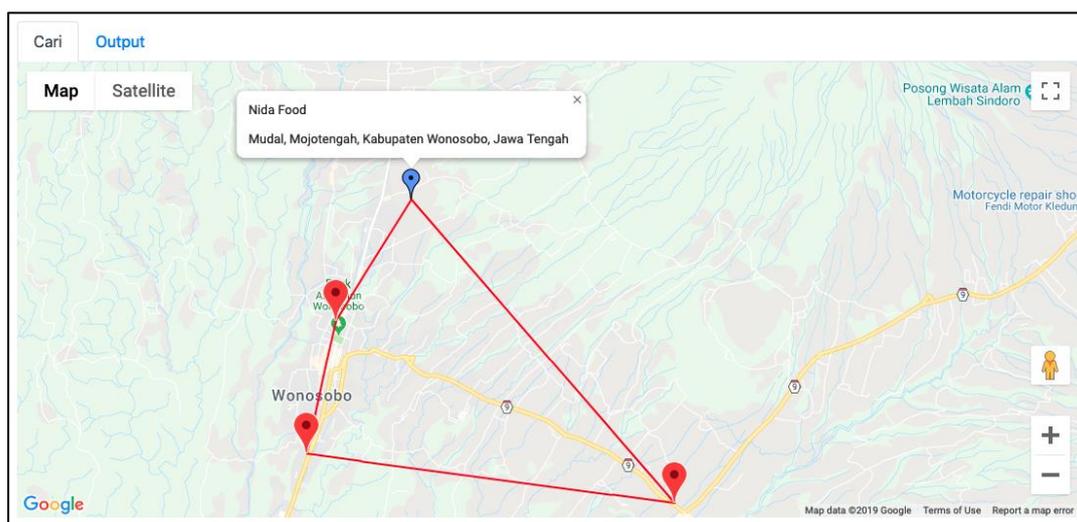
Dari hasil proses pencarian rute diatas dapat diambil solusi terbaik yaitu jarak tempuh paling minimum 20,51 dengan jalur Nida Food → Toko Bu Lukito → Toko Tambi → Toko Sadina → Nida Food atau Nida Food → Toko Sadina → Toko Tambi → Toko Bu Lukito → Nida Food. Implementasi pemilihan toko yang akan dituju dalam aplikasi web disajikan dengan tampilan seperti pada Gambar 4 di bawah ini.



| | Nama Toko | Alamat | Titik Koordinat | |
|-------------------------------------|---------------------|--|-----------------|------------|
| | | | Latitude | Longitude |
| <input checked="" type="checkbox"/> | Nida Food | Mudal, Mojotengah, Kabupaten Wonosobo, Jawa Tengah | -7.334210 | 109.916000 |
| <input checked="" type="checkbox"/> | Toko Sadina | Wonosobo | -7.387544 | 109.962189 |
| <input checked="" type="checkbox"/> | Toko Bu Lukito | Wonosobo | -7.355452 | 109.902679 |
| <input checked="" type="checkbox"/> | Toko Tambi | Wonosobo | -7.378833 | 109.897552 |
| <input type="checkbox"/> | Oafindo | Jl. Kertek Parakan Km. 7, Reco, Wonosobo, Kabupat | -7.337005 | 110.023651 |
| <input type="checkbox"/> | Podang Mas Carica | Jl. Manggisan, Mudal, Mojotengah, Kabupaten Wonos | -7.318933 | 109.918930 |
| <input type="checkbox"/> | Rumah Carica Amalia | JL Raya Kertek - Wonosobo, Krakal, Kertek, Kertek | -7.383857 | 109.960770 |

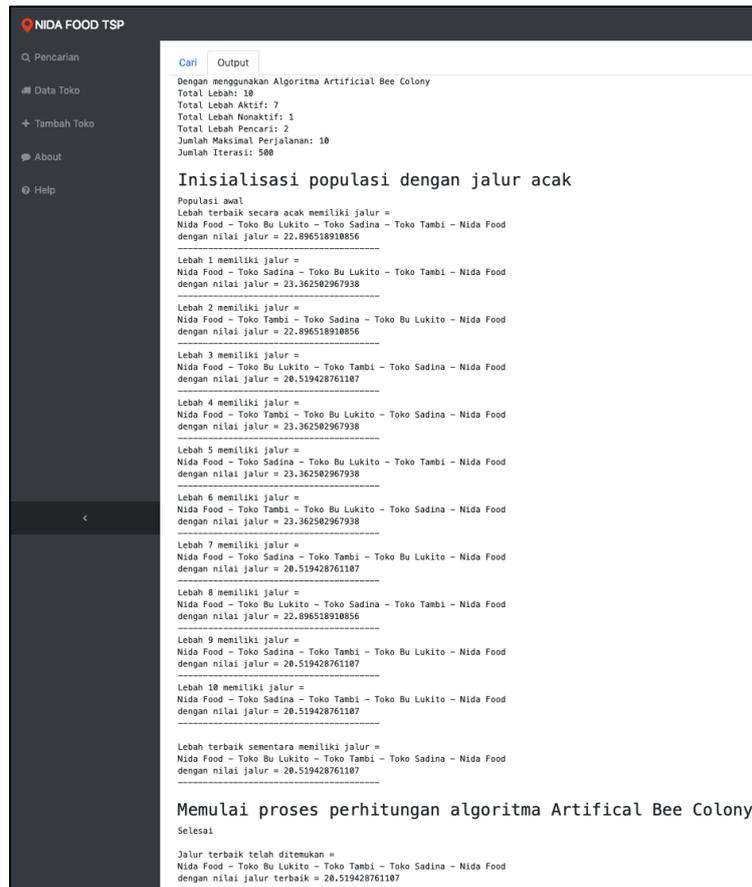
Gambar 4. Tampilan antarmuka pemilihan toko

Proses pencarian yang telah diimplementasikan pada aplikasi web, hasilnya adalah peta dengan jalur distribusi dimulai dari Nida Food menuju toko-toko tujuan distribusi dan kembali lagi ke Nida Food. Untuk market dalam peta Google, Nida Food dibedakan dengan warna Biru sedangkan toko-toko tujuan distribusi menggunakan warna merah. Berikut ini pada Gambar 5 ditampilkan jalur distribusi dari contoh sebelumnya menggunakan peta Google yang ditambahkan *layer polyline* untuk jalur distribusinya.



Gambar 5. Peta jalur distribusi

Output dari aplikasi web ini selain menampilkan jalur dalam peta juga ditampilkan proses perhitungan sesuai dengan algoritma *Artificial Bee Colony* dan jalur terbaik ditampilkan dalam text yang berurutan dari Nida Food menuju beberapa toko dan kembali lagi ke Nida Food. Hasil tersebut ditampilkan seperti pada Gambar 6 berikut:



Gambar 6. Proses perhitungan di aplikasi web

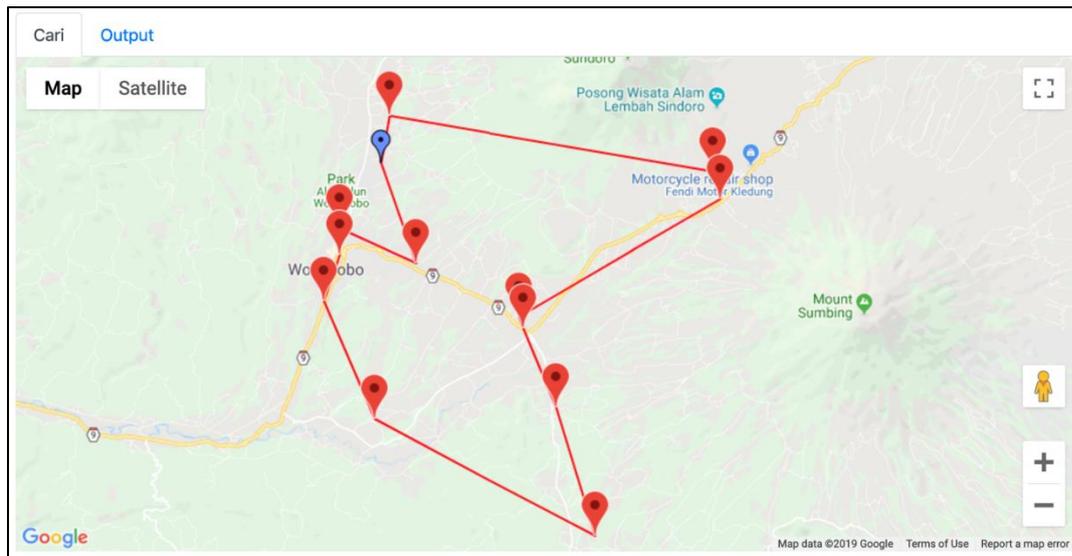
Untuk pengujian dari perhitungan ini akan digunakan 13 toko yang berada di wilayah Wonosobo, sesuai dengan toko-toko tujuan dari Nida Food kemudian dilakukan uji perubahan pada parameter jumlah lebah, jumlah perjalanan dan jumlah iterasi. Hasil dari pengujian beberapa parameter di aplikasi web terlihat pada Tabel 4 di bawah ini:

Tabel 4. Hasil pengujian aplikasi

| No | Jumlah lebah | Jumlah perjalanan | Jumlah iterasi | Jalur terbaik |
|----|--------------|-------------------|----------------|----------------|
| 1 | 10 | 10 | 500 | 55,00105548 km |
| 2 | 20 | 10 | 500 | 55,09218741 km |
| 3 | 30 | 10 | 500 | 54,89751532 km |
| 4 | 40 | 10 | 500 | 54.35695278 km |
| 5 | 50 | 10 | 500 | 54.35695278 km |
| 6 | 10 | 20 | 500 | 54.89751532 km |
| 7 | 10 | 30 | 500 | 54.35695278 km |
| 8 | 10 | 40 | 500 | 54.35695278 km |
| 9 | 10 | 50 | 500 | 54.89751532 km |
| 10 | 10 | 10 | 1000 | 54.89751532 km |
| 11 | 10 | 10 | 2000 | 54.35695278 km |
| 12 | 10 | 10 | 3000 | 54.35695278 km |
| 13 | 10 | 10 | 4000 | 54.35695278 km |

Dari hasil pengujian tersebut, menunjukkan bahwa pada kasus Nida Food untuk pencarian rute terbaik di 13 toko, dari ketiga parameter jumlah lebah, jumlah perjalanan dan jumlah iterasi tidak menunjukkan perbedaan hasil yang signifikan. Setelah dicoba beberapa kali proses pencarian

menunjukkan hasil jarak terbaik berkisar diantara 54,35 km sampai 55,09 km. Gambar 7 di bawah ini menunjukkan hasil pencarian jalur terbaik pada tampilan Google Map dari 13 kota yang dikunjungi oleh Nida Food dengan jarak 54,35 km.



Gambar 7. Jalur terbaik 13 toko di Wonosobo

5 KESIMPULAN

Pada penelitian ini aplikasi web pencarian jalur distribusi terbaik untuk Nida Food menggunakan algoritma *Artificial Bee Colony* telah berhasil dibangun dan diujicoba. Hasil dari aplikasi ini menunjukkan jarak terpendek yang digambarkan pada Google Map menggunakan *layer polyline* dan informasi urutan toko yang harus dikunjungi. Pada kasus Nida Food untuk pencarian rute terbaik di 13 toko, dari ketiga parameter yang diujikan, yaitu: jumlah lebah, jumlah perjalanan dan jumlah iterasi tidak menunjukkan perbedaan hasil yang signifikan. Pada penelitian selanjutnya perlu dipertimbangkan jarak tempuh yang sebenarnya sesuai jalur pada peta atau dengan memperhatikan *traffic* yang ada.

REFERENSI

- [1] A. Adventia, K. Novianingsih, and H. Serviana, "Penyelesaian Masalah Pendistribusian Barang Menggunakan Algoritma Bee Colony Optimization Solving the Distribution Problem Using Bee Colony Optimization Algorithm," pp. 64–72.
- [2] E. Rahmandha, R. Efendi, and D. Puspitaningrum, "Aplikasi Pencarian Lokasi Fasilitas Pelayanan Umum Terdekat Menggunakan Metode Artificial Bee Colony Di Kota Bengkulu Berbasis Webview Android," *J. Teknol. Inf.*, vol. 12, pp. 141–154, 2016.
- [3] M. R. Ahmad Fuad and M. Driberg, "Remote vehicle tracking system using GSM Modem and Google map," *Proc. - 2013 IEEE Conf. Sustain. Util. Dev. Eng. Technol. IEEE CSUDET 2013*, pp. 15–19, 2013.
- [4] W. P. Danuri, "Penerapan Bee Colony Optimization Algorithm untuk Penentuan Rute Terpendek (Studi Kasus : Objek Wisata Daerah Istimewa Yogyakarta)," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 7, no. 1, pp. 65–76, 2013.
- [5] F. Amri, E. B. Nababan, and M. F. Syahputra, "Artificial Bee Colony Algorithm untuk Menyelesaikan Travelling Salesman Problem," *J. Dunia Teknol. Inf.*, vol. 1, no. 1, pp. 8–13, 2012.
- [6] A. R. Saiyed, "The Traveling Salesman problem History of The TSP," *Indiana State Univ.*, pp. 1–15, 2012.
- [7] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press, Inc, 1999.

- [8] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.
- [9] R. Firman Pratama, Puwanto, and M. Yasin, “Penyelesaian Travelling Salesman Problem (Tsp) Dengan Menggunakan Artificial Bee Colony,” *Univ. Negeri Malang*, pp. 1–6, 2006.
- [10] G. Developers, “Maps Javascript API.” [Online]. Available: <https://developers.google.com/maps/documentation/javascript/tutorial>. [Accessed: 24-Nov-2018].
- [11] Madcoms, *Pemrograman PHP dan MySQL untuk Pemula*, 1st ed. Andi Offset, 2016.