



**PENERAPAN *ARTIFICIAL FISH SWARM ALGORITHM*  
(AFSA) PADA *MULTIPLE TRAVELLING SALESMAN  
PROBLEM* (m-TSP)  
(Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab.  
Banyuwangi)**

**SKRIPSI**

Oleh  
**Floencia Wahyu Ganda Fismaya**  
**NIM 151810101055**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2019**



**PENERAPAN *ARTIFICIAL FISH SWARM ALGORITHM*  
(AFSA) PADA *MULTIPLE TRAVELLING SALESMAN  
PROBLEM* (m-TSP)  
(Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab.  
Banyuwangi)**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan studi pada Program Studi Matematika (S-1)  
dan mencapai gelar Sarjana Sains

Oleh  
**Florecia Wahyu Ganda Fismaya**  
**NIM 151810101055**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2019**

## **PERSEMBAHAN**

Dengan nama Allah SWT Yang Maha Pengasih dan Maha Penyayang, skripsi ini saya persembahkan kepada:

1. Orang tua saya Bapak Fajar Utomo dan Ibu Ismaunah yang selalu mendoakan serta memberikan dukungan, semangat, dan materi.
2. Keluarga Besar di Banyuwangi yang selalu menguatkan melalui doa maupun nasihat.
3. Seluruh jajaran guru TK. ANGKASA Lanud Balikpapan, SDN 007 Balikpapan Selatan, SMPN 5 Balikpapan, dan SMAN 4 Balikpapan yang telah memberikan banyak ilmu.
4. Sahabat “Songong Manjah” Bayu Nalendra Utama, Dwi Alfi Rohmatin, Ega Bandawa Winata, Hadi Sutrisno, Izdihar Salsabila, Mahrita Arifah, Nazar Amir, dan Vidiyanti Lestari yang selalu menemani dalam proses pengerjaan revisi.
5. Teman-teman seperjuangan SIGMA’15 yang selalu memberikan semangat selama perkuliahan.
6. UKM SPORA dan HIMATIKA “Geokompstat” yang telah memberikan pengalaman organisasi kepada saya.
7. Bid. PSM BPH UKM SPORA 2016, Sekbid Kaderisasi Pengurus HIMATIKA “Geokompstat” 2017, dan Bid. PSM BPH UKM SPORA 2018 sebagai tim terbaik dalam berorganisasi.
8. Semua pihak yang membantu saya dalam menyelesaikan tugas akhir.

## **MOTTO**

“Kebahagiaan bersama syukur. Sesungguhnya jika kamu bersyukur, pasti Aku akan menambah (nikmat) kepadamu, dan jika kamu mengingkari (nikmat-Ku), maka sesungguhnya azab-Ku sangat pedih”

(Terjemahan QS. Ibrahim : 7)

“Semua Kembali Padamu. Jika kamu berbuat baik (berarti) kamu berbuat baik bagi dirimu sendiri dan jika kamu berbuat jahat, maka (kejahatan) itu bagi dirimu sendiri ”

(Terjemahan QS. Al Israa : 7)

## PERNYATAAN

Saya yang bertanda tangan di bawah ini

Nama : Florencia Wahyu Ganda Fismaya

NIM : 151810101055

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penerapan *Artificial Fish Swarm Algorithm* (AFSA) Pada *Multiple Travelling Salesman Problem* (m-TSP) (Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab. Banyuwangi)" adalah benar-benar hasil karya ilmiah sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, April 2019

Yang menyatakan,

Florencia Wahyu Ganda Fismaya

NIM 151810101055

**SKRIPSI**

**PENERAPAN *ARTIFICIAL FISH SWARM ALGORITHM*  
(AFSA) PADA *MULTIPLE TRAVELLING SALESMAN*  
*PROBLEM* (m-TSP)  
(Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab.  
Banyuwangi)**

Oleh

**Floencia Wahyu Ganda Fismaya  
NIM 151810101055**

Pembimbing

Dosen Pembimbing Utama : Abduh Riski, S.Si., M.Si

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom.

## PENGESAHAN

Skripsi berjudul “Penerapan *Artificial Fish Swarm Algorithm* (AFSA) Pada *Multiple Travelling Salesman Problem* (m-TSP) (Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab. Banyuwangi)”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas  
Jember.

Tim Penguji:

Ketua,

Anggota I,

Abduh Riski, S.Si., M.Si  
NIP. 199004062015041001

Ahmad Kamsyakawuni, S.Si., M.Kom  
NIP. 197211291998021001

Anggota II,

Anggota III,

Kiswara Agung Santoso, S.Si., M.Kom  
NIP. 197209071998031003

Bagus Julianto, S.Si., M.Si  
NIP.198007022003121001

Mengesahkan  
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Jember

Drs. Sujito, Ph.D.  
NIP. 196102041987111001

## RINGKASAN

**Penerapan *Artificial Fish Swarm Algorithm* (AFSA) Pada *Multiple Travelling Salesman Problem* (m-TSP) (Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab. Banyuwangi);** Florencia Wahyu Ganda Fismaya, 151810101055; 2019; 35 halaman; Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jember.

Banyak orang mencari peluang dalam berdagang kebutuhan sehari-hari, karena meningkatnya sikap masyarakat yang konsumtif pada saat ini. Berjualan atau berdagang pada era industrial 4.0 seperti saat ini bisa dilakukan dengan cara membuka toko secara *online* (Hoedi, 2018). Oleh karena itu, berbelanja saat ini juga dapat dilakukan secara online, sehingga menuntut pemilik toko *online* mengirimkan pesanan yang tidak memungkinkan untuk transaksi *Cash On Delivery* (COD) dengan menggunakan jasa pengiriman paket.

*Travelling Salesman Problem* (TSP) merupakan masalah seorang kurir dalam mengantarkan barang untuk mencari rute perjalanan yang optimal. Sedangkan, jika suatu masalah yang melibatkan lebih dari satu kurir, maka permasalahan tersebut disebut dengan *Multiple Travelling Salesman Problem* (m-TSP). Permasalahan *Multiple Travelling Salesman Problem* (m-TSP) dapat dibagi menjadi dua, yaitu mTSP simetris dan m-TSP asimetris. *Artificial Fish Swarm Algorithm* (AFSA) merupakan salah satu jenis algoritma optimisasi yang termasuk ke dalam swarm intelligence algorithm. AFSA merupakan algoritma yang terinspirasi dari kecerdasan kolektif tingkah laku kumpulan ikan di alam seperti memangsa, bergerak bebas, berkumpul, dan mengikuti. Penelitian kali ini penulis tertarik untuk menerapkan *Artificial Fish Swarm Algorithm* (AFSA) untuk menyelesaikan masalah *Multiple Travelling Salesman Problem* (m-TSP) simetris dengan stasiun pemberangkatan tunggal untuk mendapatkan rute pengantaran barang dengan jarak minimum.

Data penelitian yang digunakan untuk menerapkan kasus m-TSP ini merupakan data primer berupa alamat-alamat konsumen pengguna jasa pengiriman



paket yang diambil dari PT. Titipan Kilat (TIKI) Cabang Banyuwangi Kota yang akan diolah menjadi matriks  $n \times n$  berisi jarak dari stasiun pemberangkatan menuju alamat setiap konsumen maupun antar alamat dari masing-masing konsumen.

Berdasarkan hasil yang diperoleh dari perapan AFSA pada m-TSP dengan percobaan pada perubahan parameter populasi ( $N$ ) dan *Try Number*, jika semakin besar populasi ikan buatan ( $N$ ), *Try Number*, dan iterasi maksimal yang digunakan, maka algoritma mampu menemukan rute yang total jarak tempuhnya semakin kecil dan meskipun Running Time untuk memproses data semakin besar. Hasil solusi rute untuk masing-masing kurir yang didapatkan yaitu:

Kurir 1 : 0-15-50-19-61-3-49-34-29-42-41-53-31-26-46-40-44-39-57-33-32-  
22-0

Kurir 2 : 0-23-5-14-18-54-11-7-4-60-55-21-51-12-9-59-17-16-8-2-13-0

Kurir 3 : 0-27-6-56-20-52-35-45-47-24-43-30-37-48-25-38-10-36-58-28-1-0

Dengan total jarak ( $Z$ ) dari ketiga kurir tersebut adalah 87,28 Km dan nilai iterasi terkecil mencapai minimum lokal pada iterasi 1169.

## PRAKATA

Puji Syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul “*Artificial Fish Swarm Algorithm (AFSA) Pada Multiple Travelling Salesman Problem (m-TSP) (Studi Kasus Pengiriman Barang PT. Titipan Kilat Cabang Kab. Banyuwangi)*”. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih kepada :

1. Abduh Riski, S.Si., M.Si. selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni, S.Si., M.Kom selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, serta perhatian dalam penulisan skripsi ini;
2. Kiswara Agung Santoso, S.Si., M.Kom. dan Bagus Julianto, S.Si., M.Si. selaku dosen penguji yang telah memberikan kritik dan saran demi perbaikan tugas akhir ini;
3. Seluruh dosen dan karyawan Fakultas MIPA Universitas Jember yang telah memberikan ilmu serta bimbingannya sehingga penulis dapat menyelesaikan tugas akhir ini;
4. Teman-teman SIGMA'15 Jurusan Matematika Universitas Jember yang tidak dapat saya sebutkan satu per satu;
5. Semua pihak yang telah memberikan sumbangan tenaga, semangat, dan pikiran yang tidak dapat disebutkan satu persatu oleh penulis.

Penulis juga menerima kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, April 2019

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERSEMBAHAN</b> .....	ii
<b>HALAMAN MOTTO</b> .....	iii
<b>HALAMAN PERNYATAAN</b> .....	iv
<b>HALAMAN PEMBIMBING</b> .....	v
<b>HALAMAN PENGESAHAN</b> .....	vi
<b>RINGKASAN</b> .....	vii
<b>PRAKATA</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xiii
<b>DAFTAR LAMPIRAN</b> .....	xiv
<b>BAB 1. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Batasan Masalah</b> .....	2
<b>1.4 Tujuan</b> .....	2
<b>1.5 Manfaat</b> .....	3
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	4
<b>2.1 Travelling Salesman Problem (TSP)</b> .....	4
<b>2.2 Artificial Fish Swarm Algorithm (AFSA)</b> .....	6
<b>BAB 3. METODE PENELITIAN</b> .....	10
<b>3.1 Data Penelitian</b> .....	10
<b>3.2 Langkah Penelitian</b> .....	10
<b>BAB 4. HASIL DAN PEMBAHASAN</b> .....	14
<b>4.1 Hasil Penelitian</b> .....	14
4.1.1 Langkah Penelitian .....	14

4.1.2 Hasil Program .....	25
4.1.3 Hasil Percobaan .....	28
<b>4.2 Pembahasan</b> .....	<b>31</b>
<b>BAB 5. PENUTUP</b> .....	<b>34</b>
<b>5.1 Kesimpulan</b> .....	<b>34</b>
<b>5.2 Saran</b> .....	<b>34</b>
<b>DAFTAR PUSTAKA</b> .....	<b>35</b>
<b>LAMPIRAN</b> .....	<b>37</b>

## DAFTAR TABEL

	Halaman
Tabel 4.1 Jarak antar stasiun dan konsumen .....	14
Tabel 4.2 Hasil percobaan populasi .....	29
Tabel 4.3 Hasil percobaan <i>Try Number</i> .....	30
Tabel 4.4 Hasil percobaan pada iterasi 2000 .....	30
Tabel 4.5 Hasil percobaan pada iterasi 5000 .....	31

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh solusi penyelesaian m-TSP .....	5
Gambar 2.2 Beberapa ikan buatan dan lingkungannya.....	7
Gambar 2.3 Pseudocode Artificial Fish Swarm Algorithm (AFSA) .....	9
Gambar 3.1 <i>Flow chart</i> AFSA bagian 1 .....	11
Gambar 3.2 <i>Flow chart</i> AFSA bagian 2 .....	12
Gambar 3.3 Skema langkah-langkah penelitian.....	13
Gambar 4.1 Tampilan awal program .....	26
Gambar 4.2 Tampilan program saat proses data.....	27
Gambar 4.3 Tampilan saat proses selesai .....	28
Gambar 4.4 Tampilan program dengan visualisasi rute.....	28

## DAFTAR LAMPIRAN

	Halaman
1. Data Titik Koordinat .....	37
2. Data antar Titik .....	39
3. Skrip Utama AFSA pada Matlab .....	48

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Meningkatnya sikap masyarakat yang konsumtif pada saat ini, menyebabkan banyak orang mencari peluang dalam berdagang kebutuhan sehari-hari. Berjualan atau berdagang pada era industrial 4.0 seperti saat ini bisa dilakukan dengan cara membuka toko secara *online* (Hoedi, 2018). Oleh karena itu, berbelanja saat ini juga dapat dilakukan secara *online*, sehingga menuntut pemilik toko *online* mengirimkan pesanan yang tidak memungkinkan untuk transaksi *Cash On Delivery* (COD) dengan menggunakan jasa pengiriman paket. Perusahaan penyedia jasa pengiriman paket dituntut untuk mengantarkan paket yang datang setiap harinya ke alamat tujuan masing-masing. Hal tersebut menyebabkan dibutuhkan kurir dalam suatu perusahaan penyedia jasa pengiriman paket untuk mengantarkan paket tersebut, karena tidak semua paket memiliki alamat tujuan yang sama.

*Travelling Salesman Problem* (TSP) merupakan masalah seorang sales atau kurir dalam mengantarkan barang untuk mencari rute perjalanan yang optimal. Sedangkan, jika suatu masalah yang melibatkan lebih dari satu sales atau kurir, maka permasalahan tersebut disebut dengan *Multiple Travelling Salesman Problem* (m-TSP). Permasalahan *Multiple Travelling Salesman Problem* (m-TSP) dapat dibagi menjadi dua, yaitu m-TSP simetris dan m-TSP asimetris, dimana m-TSP simetris, biaya atau jarak dari kota 1 ke kota 2 adalah sama dengan biaya atau jarak dari kota 2 ke kota 1. Sedangkan pada m-TSP asimetris, biaya atau jarak dari kota 1 ke kota 2 tidak sama dengan biaya atau jarak dari kota 2 ke kota 1. Masalah m-TSP ini dapat diselesaikan dengan teknik pendekatan eksak dan teknik pendekatan heuristik. Land dan Doig pada tahun 1960 mengembangkan teknik eksak yaitu algoritma *branch and bound* (Bondal, 2008). Sedangkan teknik pendekatan heuristik meliputi *priority dispatch rules*, *bottleneck based heuristic*, *opportunistic scheduling*, dan *swarm intelligence algorithm* (Sempena, 2009). Beberapa metode telah digunakan untuk menyelesaikan m-TSP, seperti pada penelitian sebelumnya telah membahas permasalahan m-TSP dengan



pendekatan eksak melalui pemrograman integer dengan metode *Branch and Bound* (Eka, 2009) dan *Simulated Annealing* (Eka, 2012).

*Artificial Fish Swarm Algorithm* (AFSA) merupakan salah satu jenis algoritma optimisasi yang termasuk ke dalam *swarm intelligence algorithm*. Terdapat beberapa penelitian yang telah menggunakan algoritma ini sebagai metode untuk menyelesaikan beberapa permasalahan, seperti Penjadwalan *Job Shop* (Dyah, 2014), *fuzzy clustering*, *Resource Leveling*, optimisasi parameter *PID controller*, *spread spectrum code estimation*, *data mining*, optimisasi *DNA encoding sequence*, dsb. AFSA merupakan algoritma yang terinspirasi dari kecerdasan kolektif tingkah laku kumpulan ikan di alam seperti memangsa, berkumpul, dan mengikuti (Yazdani, et al, 2010). Penelitian kali ini penulis tertarik untuk menerapkan *Artificial Fish Swarm Algorithm* (AFSA) untuk menyelesaikan masalah *Multiple Travelling Salesman Problem* (m-TSP) dan mendapatkan rute pengantaran barang dengan jarak minimum.

## **1.2 Rumusan Masalah**

Adapun permasalahan yang akan dibahas pada penelitian kali ini adalah bagaimana penerapan *Artificial Fish Swarm Algorithm* (AFSA) pada masalah *Multiple Travelling Salesman Problem* (m-TSP) untuk mendapatkan rute pengantaran barang (dokumen atau paket) yang optimal dari beberapa kurir PT. Titipan Kilat (TIKI) Cabang Kab. Banyuwangi?

## **1.3 Batasan Masalah**

Adapun pada penelitian ini dibatasi oleh permasalahan m-TSP simetris dimana jarak dari titik 1 ke titik 2 sama dengan jarak dari titik 2 ke titik 1 dengan stasiun pemberangkatan tunggal untuk pengantaran barang di wilayah Banyuwangi Kota.

## **1.4 Tujuan**

Tujuan yang ingin didapatkan dari penulisan penelitian ini adalah menerapkan *Artificial Fish Swarm Algorithm* (AFSA) pada *Multiple Travelling Salesman Problem*

(m-TSP) untuk mendapatkan rute pengantaran barang (dokumen atau paket) dengan jarak pendek dari beberapa kurir PT. Titipan Kilat (TIKI) Cabang Kab. Banyuwangi untuk wilayah Banyuwangi Kota.

### **1.5 Manfaat**

Manfaat dari penelitian ini diharapkan dapat memberikan pengetahuan kepada penulis dan pembaca, mengenai alternatif metode baru dalam penyelesaian *Multiple Travelling Salsesman Problem* (m-TSP) menggunakan algoritma AFSA.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 *Travelling Salesman Problem (TSP)*

*Travelling Salesman Problem (TSP)* merupakan permasalahan perjalanan seorang *salesman* yang harus mengunjungi  $n$  tempat tujuan pengiriman dalam satu kali kunjungan, kemudian kembali lagi ke tempat asal (Suprayogi dan Mahmudy, 2015). Secara umum, pada saat ini penyedia jasa pengantaran barang atau dokumen yang besar tidak hanya memiliki satu *salesman* atau kurir untuk mengantarkan barang konsumen. Oleh karena itu, terdapat sebuah masalah yang lebih kompleks yang melibatkan lebih dari satu *salesman* atau kurir, yaitu *Multiple Travelling Salesman Problem (m-TSP)* yang digunakan pada penelitian kali ini.

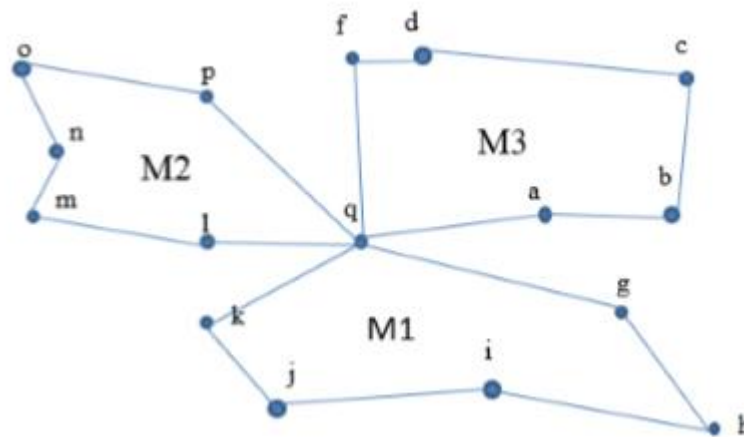
*Multiple Travelling Salesman Problem (m-TSP)* merupakan sebuah masalah optimasi yang merupakan perluasan dari permasalahan TSP yang memiliki perbedaan dari jumlah *salesman* yang melakukan kunjungan. m-TSP juga merupakan aturan optimasi dalam penyusunan rute dari  $m$  *salesman* dan  $n$  kota (Zhou and Li, 2010), dimana setiap  $m$  *salesman* mengunjungi  $n$  kota dan setiap *salesman* memulai dan berakhir di titik yang sama. Dalam kasus ini, setiap kota harus dikunjungi setidaknya satu kali oleh satu *salesman*  $s_i$ , dengan  $i$  bernilai 1 sampai  $m$ .

M-TSP ini memiliki dua jenis permasalahan, seperti permasalahan stasiun tunggal dan *multiple* stasiun. Pada permasalahan stasiun tunggal, seluruh *salesman* memulai dan mengakhiri perjalanan pada titik yang sama. Sedangkan pada permasalahan *multiple* stasiun, *salesman* tidak perlu berhenti pada titik yang sama, tetapi dapat berhenti di titik manapun dengan pembatasan pada pemberhentian perjalanan, jumlah dari *salesman* disemua titik harus sama dengan jumlah pada saat pemberangkatan seperti yang sudah dijelaskan oleh Levin dan Yovel (2012) dan Yadlapalli (2010).

M-TSP juga dapat diklasifikasikan menjadi simetrik dan asimetrik. M-TSP dikatakan simetrik jika *cost* atau jarak perjalanan dari titik  $n_i$  ke titik  $n_j$  sama dengan *cost* atau jarak perjalanan dari titik  $n_j$  ke titik  $n_i$ . Jalurnya memiliki dua fungsi, ini bisa digambarkan dengan graf tidak terarah. Sedangkan m-TSP dikatakan

asimetrik, jika *cost* atau jarak perjalanan dari titik  $n_i$  ke  $n_j$  berbeda atau tidak sama dengan *cost* atau jarak perjalanan dari titik  $n_j$  ke  $n_i$  dan bisa digambarkan dengan graf terarah.

Misal suatu perusahaan mempunyai 3 *saleman* dan ada 16 kota yang akan di lewati, maka akan didapatkan gambar solusi penyelesaian seperti di bawah ini. M1, M2, dan M3 merupakan *salesman* yang mengunjungi masing-masing 5 kota. Kota  $q$  merupakan pusat dari pengiriman dan yang lain merupakan kota-kota yang dikunjungi sales.



Gambar 2.1 Contoh Solusi Penyelesaian M-TSP

(Sumber : Hanafi, *dkk*, 2018)

Menurut Setiyono (2002) secara matematis, persoalan m-TSP dapat diformulasikan sebagai berikut:

$$Z = \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1)$$

dengan kendala

$$\sum_{i=1}^n x_{ij} = 1, \text{ untuk } j = 1, 2, 3, \dots, n - 1 \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \text{ untuk } i = 1, 2, 3, \dots, n - 1 \quad (2.3)$$

$$\sum_{i=1}^n x_{i1} = m \quad (2.4)$$

$$\sum_{j=1}^n x_{1j} = m \quad (2.5)$$

dengan keterangan

$$Z = \text{total nilai minimum}$$

$$\begin{aligned}
 x_{ij} &= 1, \text{ jika ada perjalanan } \textit{salesman} \text{ dari titik } i \text{ menuju titik } j \\
 x_{ij} &= 0, \text{ jika tidak ada perjalanan } \textit{salesman} \text{ dari titik } i \text{ menuju titik } j \\
 c_{ij} &= \text{jarak antara titik } i \text{ menuju titik } j \text{ dengan } i \text{ dan } j = 1, 2, 3, \dots
 \end{aligned}$$

Persamaan (2.1) merupakan fungsi objektif yang akan dicari. Persamaan (2.2) dan (2.3) menjamin dimana setiap titik hanya dikunjungi sekali, sedangkan persamaan (2.4) dan (2.5) menjamin jika sejumlah  $m$  kurir melakukan perjalanan.

## 2.2 Artificial Fish Swarm Algorithm (AFSA)

*Artificial Fish Swarm Algorithm* (AFSA) merupakan salah satu algoritma metaheuristik yang berbasis *swarm intelligence* dan mengembangkan teknik optimasi yang diusulkan oleh Li Xiao-lei pada tahun 2002 (Li dkk, 2002). Konsep AFSA pada umumnya adalah meniru perilaku, seperti memangsa, bergerombol atau berkoloni, berburu, dan lain-lain dari kelompok atau kumpulan ikan di alam. Kecerdasan dalam beradaptasi dari segerombol ikan dikombinasikan ke dalam *Artificial Intelligence* sebagai salah satu teknik untuk menemukan solusi dalam berbagai masalah optimasi.

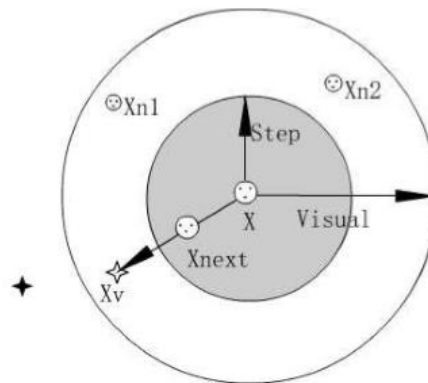
Dalam kehidupan bawah air, ikan dapat menemukan area yang memiliki makanan lebih, dimana untuk mendapatkan makanan tersebut dapat dilakukan oleh seekor maupun sekelompok ikan. Tingkah laku seekor maupun pada sekelompok ikan pada saat makan memiliki pengaruh penting dalam penyederhanaan proses pencarian makanan. Pengaruh penyederhanaan ini terjadi karena pada sebuah tempat yang terdapat makanan tersebut sebelumnya telah dicatat dan dijelaskan oleh beberapa individu yang pertama kali mengetahui makanan ini. Pada saat seekor ikan muncul untuk makan, maka ada daya tarik bagi ikan-ikan lain, kemudian dengan cepat seluruh kelompok ikan berkumpul dan mulai untuk makan. Pada dasarnya koloni atau populasi dinyatakan dengan  $N$  ikan buatan. Posisi ikan buatan dapat dinyatakan sebagai berikut :

$$X = (X_1, \dots, X_n) \text{ untuk } i = 1, \dots, n \quad (2.6)$$

dimana  $i$  menyatakan posisi dari populasi ikan. *Visual* sama dengan area penglihatan dari ikan buatan dan  $X_v$  menyatakan posisi ikan dalam area *visual* dimana ikan buatan bergerak. Jika  $X_v$  memiliki nilai lebih baik dari posisi ikan

buatan sebelumnya, maka ikan tersebut berpindah sebanyak *step* mengikuti  $X_v$  yang menyebabkan perubahan posisi ikan buatan dari  $X$  ke  $X_{next}$ , tetapi jika posisi ikan buatan  $i$  lebih baik dari  $X_v$ , maka tetap melanjutkan mencari makanan di area *visual* tadi. *Step* sama dengan panjang maksimum dari pergerakan ikan. Jarak antara dua ikan buatan pada posisi  $X_i$  dan  $X_j$  adalah  $Dis_{ij} = \|X_i - X_j\|$  (Euclidean Distance).

Model ikan buatan adalah beberapa bagian fungsi dan variabel atau parameter. Variabel atau parameternya meliputi  $X$  (posisi ikan buatan), *step* (panjang langkah maksimum), *visual* (area jarak pandang) merupakan bilangan bulat positif, *try\_number* (maksimum pengulangan perhitungan), *bulletin*, dan *crowd factor*  $\delta$  ( $0 < \delta < 1$ ). Fungsi juga terdiri dari perilaku memangsa, perilaku berkoloni, dan perilaku mengikuti. Setiap *step* dari proses optimasi, ikan buatan mencari tempat dengan nilai *fitness* terbaik, dalam masalah mencari tempat yang disebutkan oleh tiga perilaku dasar dari prosedur algoritma ini (Li dkk, 2002).



Gambar 2.2 Beberapa ikan buatan dan lingkungannya

(Sumber : M.Jian dkk, 2007)

Perilaku-perilaku dasar dari *artificial fish* atau ikan buatan antara lain:

#### 1. Perilaku Memangsa

Di alam, semua ikan konstan mencari makan atau tempat dengan makanan lebih. Jika  $X_i$  merupakan posisi ikan buatan  $i$ , kita memilih posisi  $X_j$  dalam *visual* dari ikan buatan  $i$  secara acak.  $f(X)$  merupakan konsistensi makan di posisi  $X$  atau juga nilai *fitness*. Posisi  $X_j$  dapat dihitung dengan persamaan (2.7).

$$X_{j,d}(t+1) = X_{i,d}(t) + \text{visual} \times \text{Rand}_d(-1,1) \quad (2.7)$$

Setelah itu, kepadatan makanan di  $X_i$  dibandingkan dengan posisi awal, jika  $f(X_i) \geq f(X_j)$ , ikan buatan bergerak sebanyak  $step$  dari posisi terakhir ke  $X_j$ , dijelaskan oleh persamaan (2.8).

$$\vec{X}_i(t+1) = \vec{X}_i(t) + Rand(0,1) \times step \times \frac{\vec{X}_j - \vec{X}_i(t)}{Dis_{i,j}} \quad (2.8)$$

## 2. Perilaku Berkumpul atau berkoloni

Salah satu dari anggota sekelompok ikan, selalu mencoba untuk bergerak mengelelingi anggota ikan lainnya. Hal ini dikarenakan kelompok ikan tidak menyebar dan menjaga satu sama lain dari bahaya. Pada AFSA, untuk menjaga keseluruhan kelompok, di setiap iterasi nya, ikan buatan mencoba untuk bergerak ke posisi tengah. Posisi tengah dari kelompok, seperti persamaan (2.9).

$$X_{Center,d} = \frac{1}{N} \sum_{i=1}^n X_{i,d} \quad (2.9)$$

Seperti persamaan diatas, anggota  $d$  dari vektor  $X_{Center}$  adalah rata-rata dari seluruh anggota ikan kelompok  $d$ . Misal  $N$  banyak ikan buatan,  $nc$  merupakan banyak tetangga dari ikan buatan di area *visual* dari  $X_{center}$  yang dihitung menggunakan *Euclidean Distance* dari masing-masing ikan buatan. Jika  $f(X_{center}) \leq f(X_i)$  dan nilai  $\delta > (nc/N)$ , posisi tengah memiliki konsistensi makanan yang lebih baik dibandingkan posisi saat itu dan kepadatan populasi di tetangga tidak banyak, jadi ikan buatan  $i$  bergerak ke posisi tengah dengan persamaan (2.10).

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \frac{\vec{X}_{center} - \vec{X}_i(t)}{Dis_{i,center}} \times step \times Rand(0,1) \quad (2.10)$$

Jika  $nc = 0$  atau kondisi saat bergerak ke posisi tengah tidak memuaskan, perilaku memangsa sama dengan ikan buatan  $i$ .

## 3. Perilaku Mengikuti

Saat proses dari kelompok ikan bergerak, ketika seekor atau beberapa ekor ikan menemukan makanan, ikan-ikan tetangga akan ikut untuk mengambil makanan. Jika  $X_i$  posisi dari ikan buatan  $i$ , ikan tersebut akan memeriksa tetangga  $X_{best}$ , jika  $nn$  adalah jumlah tetangga dari ikan buatan di area *visual* yang dihitung menggunakan *Euclidean Distance* dari ikan buatan *best* atau terbaik, dan jika

$f(X_{best}) \leq f(X_i)$  dan nilai  $\delta > (nn/N)$ , maka posisi  $X_{best}$  memiliki konsistensi makanan yang lebih baik dibandingkan posisi dari ikan buatan  $i$  dan kepadatan populasi di tetangga nya tidak banyak, oleh karena itu, ikan buatan  $i$  bergerak satu *step* dengan persamaan (2.11).

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \frac{\vec{X}_{best} - \vec{X}_i(t)}{Dis_{i,best}} \times step \times Rand(0,1) \quad (2.11)$$

Jika ikan buatan  $i$  tidak memiliki tetangga atau tidak ada tetangga yang mengikutinya, perilaku memangsa sama dengan ikan buatan  $i$ .

---

**Algorithm 1: Standard AFSA**


---

```

begin
  for each AF  $i$  do
    | initialize  $x_i$ 
  end
   $bulletin = \arg \min f(x_i)$  repeat
    for each AF  $i$  do
      | Perform SwarmBehaviour on  $x_i(t)$  and compute  $x_{i,Swarm}$ 
      | Perform FollowBehaviour on  $x_i(t)$  and compute  $x_{i,Follow}$ 
      | if  $f(x_{i,Swarm}) > f(x_{i,Follow})$  then
      |   |  $x_i(t+1) = x_{i,Follow}$ 
      | else
      |   |  $x_i(t+1) = x_{i,Swarm}$ 
      | end
    end
    if  $f(x_{Best-AF}) \leq f(bulletin)$  then
      |  $bulletin = x_{Best-AF}$ 
    end
  until stopping criterion is met
end
```

---

Gambar 2.3 Pseudocode Artificial Fish Swarm Algorithm (AFSA)

(Sumber : Yazdani dkk, 2010)



## BAB 3. METODE PENELITIAN

### 3.1 Data Penelitian

Data penelitian yang digunakan untuk menerapkan kasus m-TSP ini merupakan data primer berupa alamat-alamat konsumen pengguna jasa pengiriman paket yang diambil dari PT. Titipan Kilat (TIKI) Kantor Cabang Banyuwangi Kota yang akan diolah menjadi matriks  $n \times n$  berisi jarak antar masing-masing stasiun pemberangkatan dan alamat konsumen.

### 3.2 Langkah Penelitian

Penelitian mengenai penerapan *Artificial Fish Swarm Algorithm* (AFSA) untuk menyelesaikan permasalahan *Multiple Travelling Salesman Problem* (m-TSP) akan dilakukan dengan langkah-langkah sebagai berikut:

1. Studi Literatur.

Pengumpulan, membaca, serta memahami materi dan referensi yang berkaitan dengan permasalahan *Multiple Travelling Salesman Problem* (m-TSP), *Artificial Fish Swarm Algorithm* (AFSA), dan materi pendukung lainnya.

2. Pengambilan dan Pengolahan Data.

Pengambilan data dilakukan dengan mengambil data alamat-alamat konsumen dari PT. Titipan Kilat (TIKI) Cabang Kab. Banyuwangi dan mengolah data tersebut sesuai dengan permasalahan m-TSP. Data yang digunakan berupa jarak antar titik dan titik koordinat dari masing-masing stasiun pemberangkatan maupun alamat-alamat konsumen.

3. Penerapan *Artificial Fish Swarm Algorithm* (AFSA).

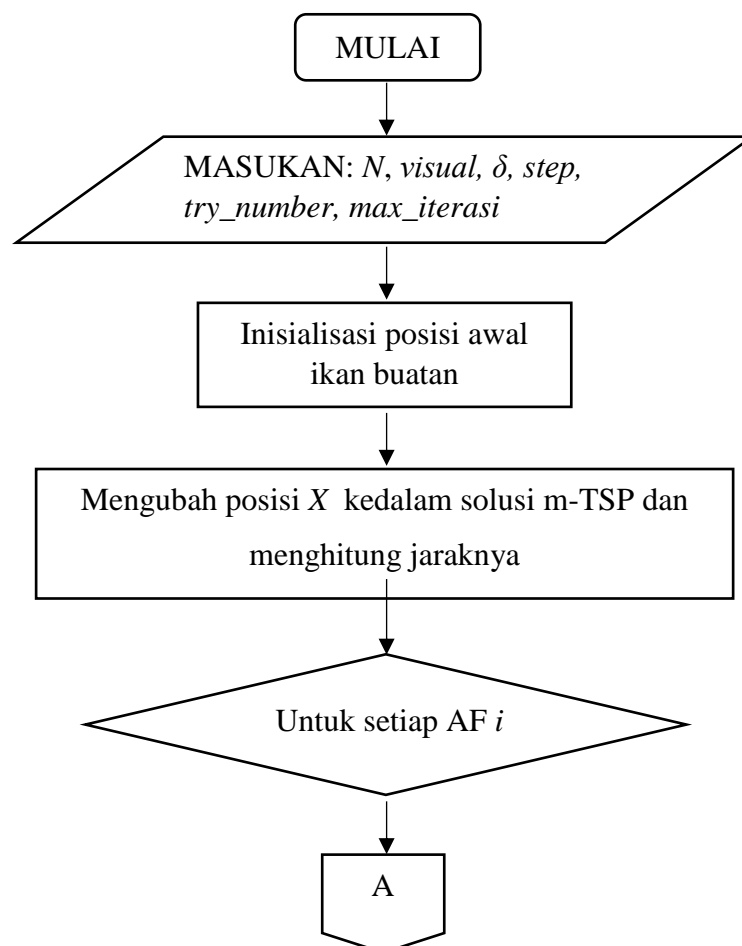
Menerapkan *Artificial Fish Swarm Algorithm* (AFSA) pada data yang telah diidentifikasi dengan langkah-langkah berikut:

a. Inisialisasi posisi populasi awal ikan buatan

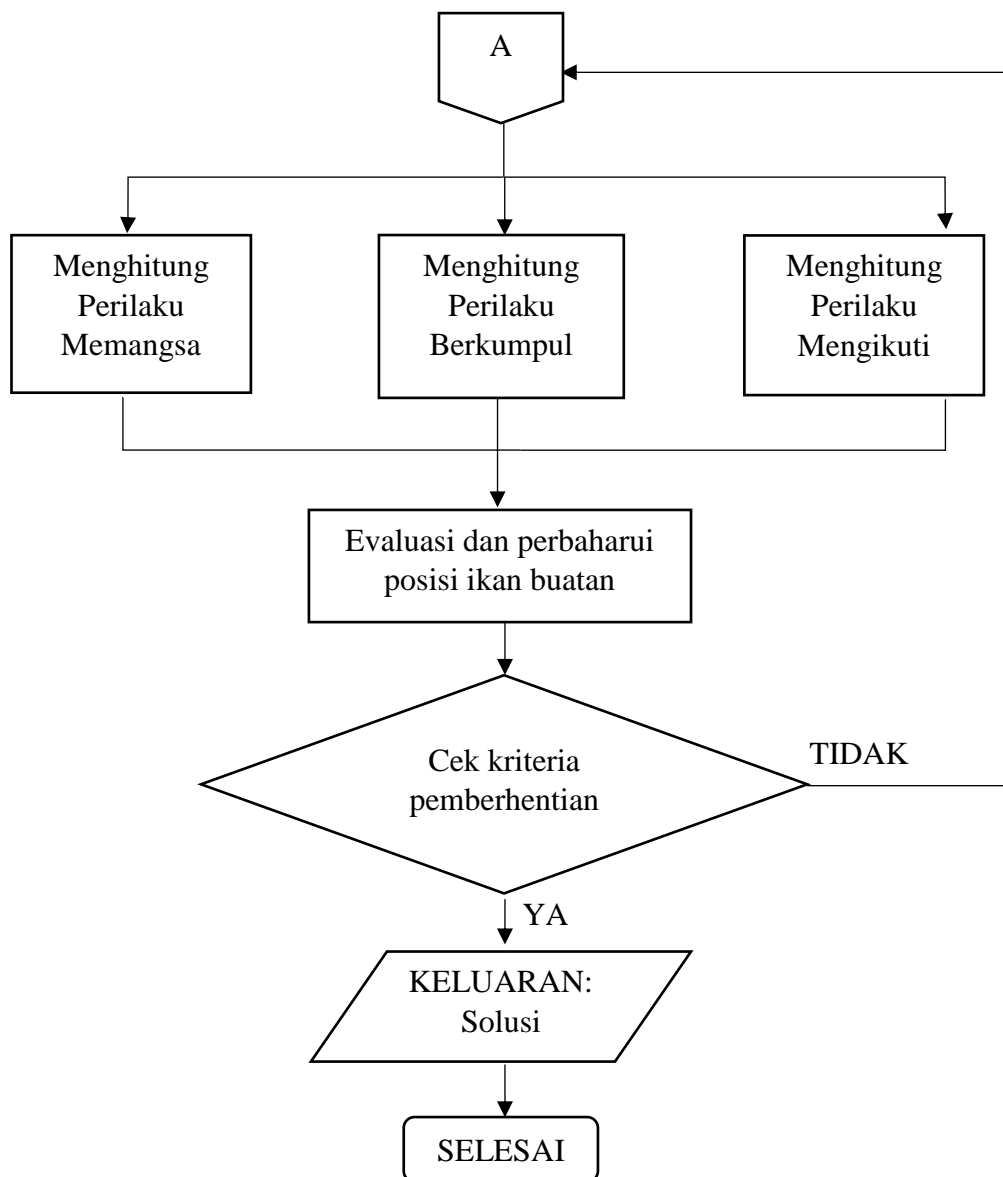
$$\left\{ \begin{array}{l} X \text{ dipilih secara acak} \\ Y \text{ mengurutkan } X \text{ dari nilai terkecil} \end{array} \right.$$

b. Mengubah posisi  $X$  kedalam solusi m-TSP dan menghitung jaraknya

- c. Menentukan, dan menyimpan solusi awal terbaik
- d. Menghitung perilaku dasar dari masing-masing  $X_i$ 
  - 1) Perilaku Memangsa
  - 2) Perilaku Berkumpul atau Berkoloni
  - 3) Perilaku Mengikuti
- e. Pengecekan dan memperbarui posisi terbaik dari masing-masing  $X$  di masing-masing perilaku.
- f. Cek kriteria pemberhentian.  
Jika iterasi sudah = max iterasi, maka iterasi dihentikan.
- g. Penentuan solusi akhir



Gambar 3.1 *Flow chart* AFSA bagian 1



Gambar 3.2 *Flow chart* AFSA bagian 2

4. Pembuatan program.

Pada langkah ini, dilakukan pembuatan program penerapan AFSA untuk menyelesaikan permasalahan m-TSP menggunakan *software* MATLAB.

5. Simulasi Program

Melakukan simulasi pada program yang telah dibuat dengan beberapa parameter yang ditentukan sebagai input.

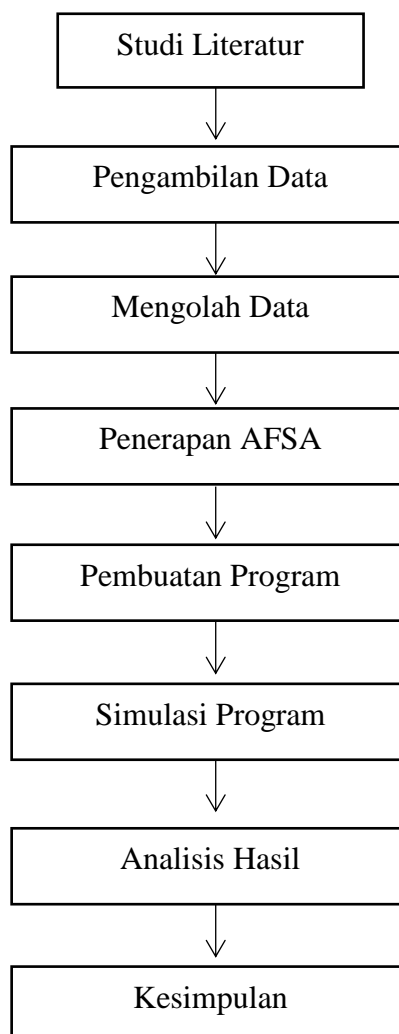
#### 6. Analisis hasil.

Dari langkah simulasi program yang dilakukan sebelumnya, dilakukan analisis terhadap hasil pengujian yang telah dilakukan, kemudian dilihat efektifitas dari proses mendapatkan hasil.

#### 7. Kesimpulan

Langkah terakhir yaitu menarik kesimpulan dari hasil perhitungan manual maupun simulasi program yang dibahas. Sehingga diperoleh solusi akhir yang diberikan.

Berikut merupakan skema untuk menyelesaikan permasalahan *Multiple Travelling Salesman Problem* (m-TSP) dengan menggunakan *Artificial Fish Swarm Algorithm* (AFSA).



Gambar 3.3 Skema Langkah-langkah Penelitian

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa:

- a. Hasil percobaan pada kombinasi parameter populasi ( $N$ ) dan *Try Number*, jika semakin besar nilai parameter ikan buatan ( $N$ ), *Try Number*, dan jumlah iterasi yang digunakan, maka AFSA mampu memberikan hasil rute dengan total jarak tempuhnya semakin kecil dan.

- b. Solusi rute untuk masing-masing kurir yaitu

Kurir 1 : 0-15-50-19-61-3-49-34-29-42-41-53-31-26-46-40-44-39-57-33-32-  
22-0

Kurir 2 : 0-23-5-14-18-54-11-7-4-60-55-21-51-12-9-59-17-16-8-2-13-0

Kurir 3 : 0-27-6-56-20-52-35-45-47-24-43-30-37-48-25-38-10-36-58-28-1-0

Dengan total jarak ( $Z$ ) dari ketiga kurir tersebut adalah 87,28 Km dan nilai iterasi terkecil mencapai minimum lokal pada iterasi 1169.

### 5.2 Saran

Penelitian selanjutnya diharapkan dapat menyelesaikan permasalahan *Multiple Travelling Salesma Problem* (m-TSP) menggunakan algoritma lain yang lebih efisien dan menerapkan, membandingkan maupun memodifikasi *Artificial Fish Swarm Algorithm* (AFSA) pada permasalahan lainnya. Serta diharapkan peneliti mampu menganalisis dan menentukan parameter-parameter yang digunakan agar mendapatkan hasil yang optimal.

## DAFTAR PUSTAKA

- Andras Kiraly, J. Abonyi. 2011. Optimization of Multiple Traveling Salesmen Problem by a Novel Representation based Genetic Algorithm. International Symposium of Hungarian Researchers on Computational Intelligence and Informatics. 366,241-269.
- Bondal, A. A. 2008. Artificial Immune Systems Applied to Job Shop Scheduling. Fakultas Mesin dan Teknologi College Russ. Universitas Ohio:Ohio.
- Dewi Eka P. 2009. Optimasi Rute *Multiple Travelling Salesman Problem* Melalui Pemrograman Integer dengan Metode *Branch And Bound*. Jember. Universitas Jember
- Hanafi A., Wihandika R.C., dan Adikara P.P., 2018. Penyelesaian Multiple Travelling Salesman Problem (M-TSP) Dengan Menggunakan Algoritme Genetika: Studi Kasus Pendistribusian Barang Di Kantor Pos Lumajang. Malang.
- Levin, A. dan U. Yovel, 2012. Local search algorithms for multiple-depot vehicle routing and for multiple traveling salesman problems with proved performance guarantees. J. Combinat. Optim. DOI 10.1007/s10878-012-9580-x
- Li, X.L, Z.J., Shao, Z.J., dan Qian, J.X. 2002. An optimizing method based on autonomous animates: fish-swarm algorithm. Systems Engineering-theory & Practice, 11, pp. 32-38..
- Mayasari, E. 2012. Penyelesaian *Multiple Travelling Salesman Problem* Dengan *Simulated Annealing*. Jember. Universitas Jember
- Jian, M, Y.wang, F.Rubio, dan D.Yuan, 2007. Spread Spectrum code estimation by artificial fish swam algorithm block-coding and antenna selection, " IEEE international symposium on intelligent signal processing (WISP).
- Pythaloka D., Wibowo A.T., dan Suliyo M.D., 2014. Penjadwalan Job Shop Menggunakan Artificial Fish Swarm Algorithm (AFSA). Bandung. Telkom University
- Prasetyo, H dan Sutopo, W. 2018. Industri 4.0: Telaah Klasifikasi Aspek Dan Arah Perkembangan Riset. Surakarta. Universitas Sebelas Maret
- Sempena, Samsu. 2009. "Algoritma Genetik Hibrida dalam Penyelesaian Job-shop Scheduling Problem". Bandung, Indonesia: Institut Teknologi Bandung.

- Setiyono, Budi. 2002. Pembuatan Perangkat Lunak Penyelesaian Multi Travelling Salesman Problem (m-TSP). KAPPA, 3(2): 55-65
- Suprayogi, D. dan Mahmudy, W. 2015. Penerapan Algoritme Genetika Travelling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry. Jurnal Buana Informatika, Volume 6, pp. 121-130.
- Yadlapalli, S., S. Rathinam dan S. Darbha, 2010. 3-Approximation algorithm for a two depot, heterogeneous traveling salesman problem. Optim. Lett., 6: 141-152. DOI: 10.1007/s11590-010-0256-0
- Yazdani, Danial, Toosi, Adel Nadjaran dan Meybodi, Mohammad Reza. 2010. "Fuzzy Adaptive Artificial Fish Swarm Algorithm". Qazvin, Mashhad, Tehran, Iran: IAU of Qazvin, IAU of Mashhad and Amirkabir University of Technology
- Zenger, Gao, X.Z., and Wu, Ying. 2011. "*Knowledge-based artificial fish-swarm algorithm*". Milano (Italy): The International Federation of Automatic Control
- Zhou, W. and Li, Y., 2010. An Improved Genetic Algorithm for Multiple Traveling Salesman Problem. pp.493– 495.

## LAMPIRAN

### LAMPIRAN 1. Data Titik Koordinat

Node	Latitude	Longitude
0	-8,233952	114,363851
1	-8,225417	114,357355
2	-8,227617	114,35107
3	-8,237513	114,35557
4	-8,231628	114,34930
5	-8,216547	114,35445
6	-8,235587	114,35899
7	-8,228592	114,34985
8	-8,233071	114,34692
9	-8,229898	114,34938
10	-8,224740	114,35704
11	-8,223792	114,35114
12	-8,226500	114,34928
13	-8,232460	114,35590
14	-8,219845	114,35276
15	-8,222290	114,35489
16	-8,234345	114,35435
17	-8,226657	114,35282
18	-8,226671	114,35284

Node	Latitude	Longitude
19	-8,211162	114,35482
20	-8,216634	114,36541
21	-8,232135	114,35415
22	-8,229144	114,36497
23	-8,224349	114,35209
24	-8,215876	114,36798
25	-8,215754	114,36983
26	-8,223243	114,36645
27	-8,237801	114,36117
28	-8,218535	114,36185
29	-8,216771	114,37437
30	-8,215858	114,37104
31	-8,222272	114,36528
32	-8,221850	114,36533
33	-8,212894	114,37304
34	-8,213295	114,37073
35	-8,216557	114,36985
36	-8,215058	114,35537
37	-8,213711	114,37438



<b>Node</b>	<b>Latitude</b>	<b>Longitude</b>
38	-8,218269	114,36967
39	-8,215149	114,35663
40	-8,225327	114,36654
41	-8,218871	114,36542
42	-8,219409	114,36891
43	-8,210540	114,37408
44	-8,214843	114,36531
45	-8,217788	114,36155
46	-8,218746	114,36915
47	-8,221507	114,37218
48	-8,221612	114,36806
49	-8,237552	114,35558

<b>Node</b>	<b>Latitude</b>	<b>Longitude</b>
50	-8,212001	114,34955
51	-8,219458	114,35008
52	-8,213951	114,37362
53	-8,222874	114,35843
54	-8,226043	114,34998
55	-8,229898	114,34938
56	-8,222361	114,36795
57	-8,214376	114,37253
58	-8,212710	114,35580
59	-8,224545	114,34580
60	-8,225240	114,35733
61	-8,224792	114,35801

**LAMPIRAN 2. Data antar Titik**

<b>Node</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>
<b>0</b>	0,0	2,3	4,5	2,3	4,7	4,0	0,8	4,5	2,4	4,3	2,4	3,6	2,8	1,8	3,6	2,8	1,4	0,9	2,1	5,5	2,9	1,7	0,8
<b>1</b>	2,3	0,0	2,1	2,4	2,6	1,1	3,5	2,5	2,9	2,3	0,9	1,5	2,9	1,2	1,5	0,6	2,0	2,4	1,5	2,7	2,5	1,5	2,6
<b>2</b>	4,5	2,1	0,0	2,3	0,9	3,2	2,7	0,6	1,1	0,5	2,1	3,5	2,8	1,8	3,6	1,2	1,7	2,0	0,5	4,7	4,8	1,5	4,5
<b>3</b>	2,3	2,4	2,3	0,0	0,4	1,7	4,2	0,8	0,5	0,6	2,2	0,8	0,5	1,2	1,3	2,3	1,0	3,5	1,2	3,2	3,0	1,0	4,9
<b>4</b>	4,7	2,6	0,9	0,4	0,0	2,1	1,9	0,6	0,4	0,4	2,6	1,7	0,8	1,0	1,7	2,7	0,7	1,2	1,3	3,6	3,5	0,7	4,1
<b>5</b>	4,0	1,1	3,2	1,7	2,1	0,0	4,2	2,3	2,8	2,1	1,5	1,4	4,5	2,8	0,4	0,9	3,3	3,6	2,4	1,5	1,4	3,1	3,2
<b>6</b>	0,8	3,5	2,7	4,2	1,9	4,2	0,0	3,7	1,6	3,6	2,3	2,8	2,0	0,8	2,8	2,7	0,7	0,2	1,9	4,8	3,3	1,0	2,6
<b>7</b>	4,5	2,5	0,6	0,8	0,6	2,3	3,7	0,0	0,9	0,2	2,7	0,6	0,6	1,4	4,7	3,2	1,3	1,7	0,8	5,3	4,5	1,2	4,2
<b>8</b>	2,4	2,9	1,1	0,5	0,4	2,8	1,6	0,9	0,0	0,7	2,8	1,9	1,0	1,4	1,9	2,9	1,2	1,5	1,9	3,8	3,7	1,0	4,0
<b>9</b>	4,3	2,3	0,5	0,6	0,4	2,1	3,6	0,2	0,7	0,0	2,5	0,8	0,7	1,2	4,5	3,0	1,0	1,5	1,0	5,1	4,4	1,0	4,0
<b>10</b>	2,4	0,9	2,1	2,2	2,6	1,5	2,3	2,7	2,8	2,5	0,0	1,0	3,0	1,3	1,0	0,5	2,1	2,5	0,7	2,6	2,4	1,6	2,7
<b>11</b>	3,6	1,5	3,5	0,8	1,7	1,4	2,8	0,6	1,9	0,8	1,0	0,0	0,5	2,1	0,5	0,6	1,9	2,3	0,5	5,9	5,1	1,7	4,7
<b>12</b>	2,8	2,9	2,8	0,5	0,8	4,5	2,0	0,6	1,0	0,7	3,0	0,5	0,0	3,0	1,0	1,0	2,8	3,2	0,8	2,9	2,7	2,6	4,5
<b>13</b>	1,8	1,2	1,8	1,2	1,0	2,8	0,8	1,4	1,4	1,2	1,3	2,1	3,0	0,0	2,5	1,7	0,5	0,7	1,0	3,1	2,7	0,3	2,9
<b>14</b>	3,6	1,5	3,6	1,3	1,7	0,4	2,8	4,7	1,9	4,5	1,0	0,5	1,0	2,5	0,0	0,6	1,9	2,3	0,5	5,1	5,1	1,7	4,7
<b>15</b>	2,8	0,6	1,2	2,3	2,7	0,9	2,7	3,2	2,9	3,0	0,5	0,6	1,0	1,7	0,6	0,0	2,5	2,9	0,8	1,8	2,4	2,4	3,1
<b>16</b>	1,4	2,0	1,7	1,0	0,7	3,3	0,7	1,3	1,2	1,0	2,1	1,9	2,8	0,5	1,9	2,5	0,0	0,6	1,3	3,4	4,0	0,4	3,6
<b>17</b>	0,9	2,4	2,0	3,5	1,2	3,6	0,2	1,7	1,5	1,5	2,5	2,3	3,2	0,7	2,3	2,9	0,6	0,0	1,8	3,9	3,0	0,9	2,6
<b>18</b>	2,1	1,5	0,5	1,2	1,3	2,4	1,9	0,8	1,9	1,0	0,7	0,5	0,8	1,0	0,5	0,8	1,3	1,8	0,0	3,4	3,0	0,9	3,2
<b>19</b>	5,5	2,7	4,7	3,2	3,6	1,5	4,8	5,3	3,8	5,1	2,6	5,9	2,9	3,1	5,1	1,8	3,4	3,9	3,4	0,0	1,8	3,4	3,6
<b>20</b>	2,9	2,5	4,8	3,0	3,5	1,4	3,3	4,5	3,7	4,4	2,4	5,1	2,7	2,7	5,1	2,4	4,0	3,0	3,0	1,8	0,0	4,0	1,8
<b>21</b>	1,7	1,5	1,5	1,0	0,7	3,1	1,0	1,2	1,0	1,0	1,6	1,7	2,6	0,3	1,7	2,4	0,4	0,9	0,9	3,4	4,0	0,0	3,4

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
22	0,8	2,6	4,5	4,9	4,1	3,2	2,6	4,2	4,0	4,0	2,7	4,7	4,5	2,9	4,7	3,1	3,6	2,6	3,2	3,6	1,8	3,4	0,0
23	2,4	1,6	0,5	1,0	1,8	1,4	2,2	0,8	2,0	0,9	1,0	0,2	0,7	1,3	0,2	0,7	1,6	2,1	0,3	1,7	2,3	1,2	3,1
24	2,9	2,8	4,9	3,3	3,7	1,7	3,3	4,6	3,9	4,4	2,7	5,1	3,0	2,7	5,1	2,7	4,3	3,0	3,1	2,0	0,3	4,3	2,7
25	2,7	2,7	4,7	3,5	4,0	1,9	3,1	4,3	4,2	4,2	2,8	4,9	3,2	3,1	4,9	2,9	4,5	2,8	3,4	2,3	0,5	3,4	2,4
26	1,8	1,8	3,8	4,1	2,9	2,6	2,2	3,5	3,3	3,3	1,9	4,0	3,8	2,2	4,0	2,3	2,7	1,9	2,5	2,9	1,1	2,5	0,7
27	0,5	2,6	3,3	5,1	2,8	4,1	0,6	3,1	2,0	2,0	2,7	2,7	3,7	1,2	2,7	3,4	1,4	0,6	2,2	4,4	3,2	1,4	1,3
28	2,7	1,8	3,0	4,0	3,1	2,3	3,0	3,6	3,5	4,5	1,9	4,2	3,6	2,2	4,2	2,4	2,9	2,7	2,5	2,7	0,6	2,5	2,7
29	2,4	3,1	5,1	4,1	4,5	2,4	3,5	4,8	4,6	4,6	3,2	5,3	3,7	3,5	5,3	3,5	4,0	3,2	3,9	2,8	1,0	3,8	2,3
30	2,5	2,7	4,9	3,9	4,3	1,9	3,4	4,6	4,4	4,4	3,0	5,1	3,5	3,3	5,1	3,3	3,6	2,8	3,4	2,3	0,8	3,6	2,5
31	2,1	1,5	4,0	3,9	3,2	2,8	2,5	3,7	3,6	3,5	1,6	4,3	3,5	1,9	4,3	2,0	3,0	2,2	2,2	3,1	1,0	2,9	0,9
32	2,1	1,6	4,1	4,1	3,3	2,7	2,6	3,8	3,6	3,6	1,7	4,3	3,6	1,9	4,3	2,1	3,0	2,2	2,3	3,1	1,0	3,0	1,0
33	3,2	3,2	5,2	4,0	4,4	2,5	3,6	4,8	4,7	4,7	3,3	5,4	3,8	3,6	5,4	3,7	4,1	3,3	3,9	2,8	1,1	3,9	2,9
34	3,6	3,4	5,5	3,7	4,4	2,3	4,0	5,2	4,6	5,1	3,4	5,8	3,6	3,4	5,8	3,4	4,9	3,7	3,7	2,7	0,9	4,9	3,3
35	2,6	2,5	4,5	1,9	3,7	2,0	3,0	4,2	4,0	4,0	2,6	4,8	3,3	3,0	4,8	3,1	3,5	2,7	3,3	2,4	0,7	3,4	2,3
36	4,2	1,4	3,4	4,1	2,3	0,2	3,5	4,0	2,5	3,8	1,3	4,6	1,6	2,6	4,6	1,1	2,9	3,4	2,9	0,7	1,1	2,9	4,4
37	2,9	3,2	5,1	3,9	4,3	2,4	3,6	4,8	4,7	4,6	3,3	5,4	3,8	3,6	5,4	3,7	4,1	3,3	3,9	2,8	1,1	3,8	2,8
38	2,4	2,4	4,3	4,0	3,5	2,2	2,8	4,0	3,9	3,8	2,5	4,6	3,5	2,8	4,6	2,9	3,3	2,5	3,1	2,6	0,7	3,0	2,1
39	4,4	1,6	4,0	1,9	2,5	2,9	2,4	3,6	3,5	4,0	1,5	4,7	1,7	2,7	4,7	1,5	3,0	3,6	3,1	0,8	1,1	3,1	4,6
40	1,5	1,5	3,4	3,8	2,7	2,7	1,9	3,2	3,0	3,0	1,6	3,7	3,5	1,9	3,7	2,0	2,4	1,6	2,3	3,1	1,4	2,4	0,5
41	2,5	1,9	4,5	3,5	3,7	1,8	3,0	4,2	4,1	4,0	1,9	4,7	3,2	2,2	4,7	2,4	3,4	2,7	2,6	2,2	0,5	2,5	2,3
42	2,4	2,4	4,3	3,9	3,5	2,2	2,7	3,9	3,8	3,7	2,3	4,4	3,5	2,6	4,4	2,7	3,1	2,3	3,0	2,6	0,8	3,1	2,0
43	4,1	4,1	6,2	4,6	3,4	2,9	4,6	5,9	5,2	5,7	4,0	6,4	4,3	4,0	6,4	4,0	5,6	4,3	4,4	2,7	1,6	5,6	4,0
44	3,2	2,6	5,1	3,2	5,0	1,5	3,6	4,8	3,8	4,6	2,5	5,4	2,8	3,0	5,4	2,6	4,1	3,3	3,3	1,9	0,3	4,1	2,9

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
<b>45</b>	2,4	0,9	2,3	3,4	3,6	2,6	2,3	2,9	2,7	2,7	0,9	3,4	3,1	1,5	3,4	1,6	2,1	2,0	1,8	3,0	1,0	1,7	1,1
<b>46</b>	2,3	2,2	4,2	3,9	3,4	2,2	2,7	3,9	3,7	3,7	2,3	4,5	3,6	2,7	4,5	2,8	3,2	2,4	3,0	2,6	0,8	2,9	2,0
<b>47</b>	1,8	2,5	4,5	4,2	3,7	2,5	3,0	4,2	4,0	4,0	2,6	4,8	3,9	3,0	4,8	3,1	3,4	2,7	3,3	2,9	1,2	3,4	1,8
<b>48</b>	2,0	2,0	3,9	4,3	3,1	2,3	2,4	3,6	3,5	3,4	2,0	4,2	3,9	2,4	4,2	2,5	2,9	2,1	2,7	2,7	0,9	2,6	1,7
<b>49</b>	2,2	2,3	2,2	1,0	0,7	3,7	1,4	1,8	1,6	1,6	2,6	2,4	3,3	0,8	2,4	3,0	0,5	0,6	1,9	4,0	4,6	0,9	1,7
<b>50</b>	4,9	2,1	4,1	2,6	3,0	0,9	4,1	4,6	3,1	4,4	1,9	5,2	2,2	3,2	5,2	1,9	3,5	4,0	3,5	0,9	1,9	3,5	5,0
<b>51</b>	2,8	2,9	2,8	0,5	0,8	4,5	3,0	0,6	1,0	0,7	3,0	0,7	1,1	2,7	0,7	0,9	2,4	2,9	3,3	1,9	2,5	2,4	4,0
<b>52</b>	3,1	3,1	5,0	4,0	4,2	2,3	3,5	4,7	4,5	4,5	3,1	5,3	3,7	3,5	5,3	3,6	4,0	3,2	3,8	2,7	1,0	3,7	2,8
<b>53</b>	2,4	0,9	2,3	3,4	2,4	2,6	2,3	2,9	2,7	2,7	0,9	3,4	3,1	1,5	3,4	1,6	2,1	2,0	1,8	3,0	1,0	1,7	1,1
<b>54</b>	4,7	2,6	0,9	0,4	0,9	4,5	2,0	0,4	1,1	0,6	1,3	0,3	0,2	1,7	0,3	0,9	1,5	2,0	0,7	4,8	4,1	1,4	3,0
<b>55</b>	4,3	2,3	0,6	0,6	0,4	2,1	3,5	0,2	0,7	0,2	2,2	0,8	0,7	1,2	0,8	1,4	1,0	3,5	1,0	2,4	3,0	1,0	4,5
<b>56</b>	1,9	1,9	3,9	4,2	3,0	2,4	2,3	3,5	3,4	3,4	2,0	4,1	3,9	1,2	4,1	2,4	2,8	2,0	2,6	2,8	1,0	2,6	1,6
<b>57</b>	3,0	2,9	3,6	3,9	4,1	2,2	3,4	4,6	4,4	4,4	3,0	5,2	3,6	2,3	5,2	3,5	3,8	3,1	3,7	2,9	0,9	3,6	2,7
<b>58</b>	6,9	4,0	4,2	4,6	5,0	0,5	6,1	6,6	5,2	6,5	3,9	7,2	4,2	3,4	7,2	4,0	5,5	6,0	5,6	0,3	3,2	5,5	7,1
<b>59</b>	4,4	2,4	1,8	2,1	2,6	2,2	3,7	2,0	2,8	2,1	2,3	1,4	1,8	3,4	1,4	1,5	3,1	3,6	4,0	2,5	3,1	3,1	4,7
<b>60</b>	2,3	0,3	2,0	2,3	2,7	1,6	2,2	2,7	2,9	2,5	0,1	1,1	2,0	1,2	1,1	0,5	1,9	1,9	0,7	1,9	2,5	1,5	3,2
<b>61</b>	6,2	3,4	3,6	3,9	4,3	2,2	5,4	6,0	4,5	5,8	3,3	6,6	3,6	4,0	2,6	3,3	4,9	5,4	4,9	2,0	3,2	4,9	6,4

Node	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
1	2,4	2,9	2,7	1,8	0,5	2,7	2,4	2,5	2,1	2,1	3,2	3,6	2,6	4,2	2,9	2,4	4,4	1,5	2,5	2,4	4,1	3,2	2,4
2	1,6	2,8	2,7	1,8	2,6	1,8	3,1	2,7	1,5	1,6	3,2	3,4	2,5	1,4	3,2	2,4	1,6	1,5	1,9	2,4	4,1	2,6	0,9
3	0,5	4,9	4,7	3,8	3,3	3,0	5,1	4,9	4,0	4,1	5,2	5,5	4,5	3,4	5,1	4,3	4,0	3,4	4,5	4,3	6,2	5,1	2,3
4	1,0	3,3	3,5	4,1	5,1	4,0	4,1	3,9	3,9	4,1	4,0	3,7	1,9	4,1	3,9	4,0	1,9	3,8	3,5	3,9	4,6	3,2	3,4
5	1,8	3,7	4,0	2,9	2,8	3,1	4,5	4,3	3,2	3,3	4,4	4,4	3,7	2,3	4,3	3,5	2,5	2,7	3,7	3,5	3,4	5,0	3,6
6	1,4	1,7	1,9	2,6	4,1	2,3	2,4	1,9	2,8	2,7	2,5	2,3	2,0	0,2	2,4	2,2	2,9	2,7	1,8	2,2	2,9	1,5	2,6
7	2,2	3,3	3,1	2,2	0,6	3,0	3,5	3,4	2,5	2,6	3,6	4,0	3,0	3,5	3,6	2,8	2,4	1,9	3,0	2,7	4,6	3,6	2,3
8	0,8	4,6	4,3	3,5	3,1	3,6	4,8	4,6	3,7	3,8	4,8	5,2	4,2	4,0	4,8	4,0	3,6	3,2	4,2	3,9	5,9	4,8	2,9
9	2,0	3,9	4,2	3,3	2,0	3,5	4,6	4,4	3,6	3,6	4,7	4,6	4,0	2,5	4,7	3,9	3,5	3,0	4,1	3,8	5,2	3,8	2,7
10	0,9	4,4	4,2	3,3	2,0	4,5	4,6	4,4	3,5	3,6	4,7	5,1	4,0	3,8	4,6	3,8	4,0	3,0	4,0	3,7	5,7	4,6	2,7
11	1,0	2,7	2,8	1,9	2,7	1,9	3,2	3,0	1,6	1,7	3,3	3,4	2,6	1,3	3,3	2,5	1,5	1,6	1,9	2,3	4,0	2,5	0,9
12	0,2	5,1	4,9	4,0	2,7	4,2	5,3	5,1	4,3	4,3	5,4	5,8	4,8	4,6	5,4	4,6	4,7	3,7	4,7	4,4	6,4	5,4	3,4
13	0,7	3,0	3,2	3,8	3,7	3,6	3,7	3,5	3,5	3,6	3,8	3,6	3,3	1,6	3,8	3,5	1,7	3,5	3,2	3,5	4,3	2,8	3,1
14	1,3	2,7	3,1	2,2	1,2	2,2	3,5	3,3	1,9	1,9	3,6	3,4	3,0	2,6	3,6	2,8	2,7	1,9	2,2	2,6	4,0	3,0	1,5
15	0,2	5,1	4,9	4,0	2,7	4,2	5,3	5,1	4,3	4,3	5,4	5,8	4,8	4,6	5,4	4,6	4,7	3,7	4,7	4,4	6,4	5,4	3,4
16	0,7	2,7	2,9	2,3	3,4	2,4	3,5	3,3	2,0	2,1	3,7	3,4	3,1	1,1	3,7	2,9	1,5	2,0	2,4	2,7	4,0	2,6	1,6
17	1,6	4,3	4,5	2,7	1,4	2,9	4,0	3,6	3,0	3,0	4,1	4,9	3,5	2,9	4,1	3,3	3,0	2,4	3,4	3,1	5,6	4,1	2,1
18	2,1	3,0	2,8	1,9	0,6	2,7	3,2	2,8	2,2	2,2	3,3	3,7	2,7	3,4	3,3	2,5	3,6	1,6	2,7	2,3	4,3	3,3	2,0
19	0,3	3,1	3,4	2,5	2,2	2,5	3,9	3,4	2,2	2,3	3,9	3,7	3,3	2,9	3,9	3,1	3,1	2,3	2,6	3,0	4,4	3,3	1,8
20	1,7	2,0	2,3	2,9	4,4	2,7	2,8	2,3	3,1	3,1	2,8	2,7	2,4	0,7	2,8	2,6	0,8	3,1	2,2	2,6	2,7	1,9	3,0
21	2,3	0,3	0,5	1,1	3,2	0,6	1,0	0,8	1,0	1,0	1,1	0,9	0,7	1,1	1,1	0,7	1,1	1,4	0,5	0,8	1,6	0,3	1,0
23	1,2	4,3	3,4	2,5	1,4	2,5	3,8	3,6	2,9	3,0	3,9	4,9	3,4	2,9	3,8	3,0	3,1	2,4	2,5	3,1	5,6	4,1	1,7

<b>Node</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>	<b>61</b>
<b>1</b>	2,3	1,8	2,0	2,2	4,9	2,8	3,1	2,4	4,7	4,3	1,9	3,0	6,9	4,4	2,3	6,2
<b>2</b>	2,2	2,5	2,0	2,3	2,1	2,9	3,1	0,9	2,6	2,3	1,9	2,9	4,0	2,4	0,3	3,4
<b>3</b>	4,2	4,5	3,9	2,2	4,1	2,8	5,0	2,3	0,9	0,6	3,9	3,6	4,2	1,8	2,0	3,6
<b>4</b>	3,9	4,2	4,3	1,0	2,6	0,5	4,0	3,4	0,4	0,6	4,2	3,9	4,6	2,1	2,3	3,9
<b>5</b>	3,4	3,7	3,1	0,7	3,0	0,8	4,2	2,4	0,9	0,4	3,0	4,1	5,0	2,6	2,7	4,3
<b>6</b>	2,2	2,5	2,3	3,7	0,9	4,5	2,3	2,6	4,5	2,1	2,4	2,2	0,5	2,2	1,6	2,2
<b>7</b>	2,7	3,0	2,4	1,4	4,1	3,0	3,5	2,3	2,0	3,5	2,3	3,4	6,1	3,7	2,2	5,4
<b>8</b>	3,9	4,2	3,6	1,8	4,6	0,6	4,7	2,9	0,4	0,2	3,5	4,6	6,6	2,0	2,7	6,0
<b>9</b>	3,7	4,0	3,5	1,6	3,1	1,0	4,5	2,7	1,1	0,7	3,4	4,4	5,2	2,8	2,9	4,5
<b>10</b>	3,7	4,0	3,4	1,6	4,4	0,7	4,5	2,7	0,6	0,2	3,4	4,4	6,5	2,1	2,5	5,8
<b>11</b>	2,3	2,6	2,0	2,6	1,9	3,0	3,1	0,9	1,3	2,2	2,0	3,0	3,9	2,3	0,1	3,3
<b>12</b>	4,5	4,8	4,2	2,4	5,2	0,7	5,3	3,4	0,3	0,8	4,1	5,2	7,2	1,4	1,1	6,6
<b>13</b>	3,6	3,9	3,9	3,3	2,2	1,1	3,7	3,1	0,2	0,7	3,9	3,6	4,2	1,8	2,0	3,6
<b>14</b>	2,7	3,0	2,4	0,8	3,2	2,7	3,5	1,5	1,7	1,2	1,2	2,3	3,4	3,4	1,2	4,0

Node	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
24	3,4	0,0	0,2	1,0	3,2	0,9	0,7	0,6	0,9	0,9	0,8	0,7	0,4	1,4	0,8	0,4	1,4	1,4	0,5	0,5	1,5	0,6	1,2
25	3,8	0,2	0,0	0,9	3,0	1,1	0,5	0,3	1,1	1,2	0,6	0,4	0,1	1,6	0,6	0,4	1,6	1,2	0,8	0,5	0,9	0,8	1,4
26	2,8	1,0	0,9	0,0	2,0	1,2	1,3	1,1	0,3	0,4	1,4	1,8	0,8	2,2	1,4	0,6	2,2	0,3	0,8	0,5	2,4	1,4	0,8
27	2,9	3,2	3,0	2,0	0,0	2,9	3,0	3,0	2,3	2,4	3,5	3,9	2,8	3,9	3,4	2,6	4,2	1,8	2,8	2,5	4,5	3,4	2,1
28	2,8	0,9	1,1	1,2	2,9	0,0	1,7	1,5	0,9	0,9	1,9	1,6	1,2	1,4	1,9	1,1	1,4	1,5	0,5	0,9	2,3	0,8	0,8
29	4,2	0,7	0,5	1,3	3,0	1,7	0,0	0,5	1,6	1,6	0,6	0,8	0,6	2,1	0,5	0,7	2,1	1,6	1,2	0,9	1,1	1,3	2,0
30	4,0	0,6	0,3	1,1	3,0	1,5	0,5	0,0	1,4	1,4	0,5	0,7	0,4	2,0	0,5	0,6	1,9	1,4	1,1	0,7	1,0	1,1	1,8
31	2,5	0,9	1,1	0,3	2,3	0,9	1,6	1,4	0,0	0,5	1,6	2,0	1,0	1,8	1,6	0,8	1,8	0,5	0,6	0,7	2,7	1,6	0,5
32	2,6	0,9	1,2	0,4	2,4	0,9	1,6	1,4	0,5	0,0	1,7	2,1	1,0	1,8	1,7	0,8	1,7	0,5	0,5	0,7	2,7	1,2	0,5
33	4,3	0,8	0,6	1,4	3,5	1,9	0,6	0,5	1,6	1,7	0,0	0,3	0,7	2,2	0,3	0,8	2,1	1,7	1,3	1,0	0,6	1,3	2,0
34	4,0	0,7	0,4	1,8	3,9	1,6	0,8	0,7	2,0	2,1	0,3	0,0	0,5	2,1	0,5	0,7	2,0	1,5	1,2	0,8	0,6	1,1	1,9
35	3,6	0,4	0,1	0,8	2,8	1,2	0,6	0,4	1,0	1,0	0,7	0,5	0,0	1,8	0,6	0,2	1,7	1,0	0,8	0,4	1,7	0,9	1,4
36	3,2	1,4	1,6	2,2	3,9	1,4	2,1	2,0	1,8	1,8	2,2	2,1	1,8	0,0	2,2	2,0	0,2	2,5	1,6	1,9	2,7	1,3	2,4
37	4,2	0,8	0,6	1,4	3,4	1,9	0,5	0,5	1,6	1,7	0,3	0,5	0,6	2,2	0,0	0,8	3,1	2,6	2,4	2,1	0,6	2,3	3,0
38	3,4	0,4	0,4	0,6	2,6	1,1	0,7	0,6	0,8	0,8	0,8	0,7	0,2	2,0	0,8	0,0	0,8	0,9	0,6	0,0	1,9	1,1	1,3
39	3,4	1,4	1,6	2,2	4,2	1,4	2,1	1,9	1,8	1,7	2,1	2,0	1,7	0,2	3,1	0,8	0,0	0,5	0,9	0,8	2,6	1,5	0,6
40	2,6	1,4	1,2	0,3	1,8	1,5	1,6	1,4	0,5	0,5	1,7	1,5	1,0	2,5	2,6	0,9	0,5	0,0	1,0	0,9	2,7	1,6	0,6
41	2,9	0,5	0,8	0,8	2,8	0,5	1,2	1,1	0,6	0,5	1,3	1,2	0,8	1,6	2,4	0,6	0,9	1,0	0,0	0,7	1,8	0,8	0,8
42	3,3	0,5	0,5	0,5	2,5	0,9	0,9	0,7	0,7	0,7	1,0	0,8	0,4	1,9	2,1	0,0	0,8	0,9	0,7	0,0	1,9	0,9	1,3
43	4,7	1,5	0,9	2,4	4,5	2,3	1,1	1,0	2,7	2,7	0,6	0,6	1,7	2,7	0,6	1,9	2,6	2,7	1,8	1,9	0,0	2,5	3,3
44	3,6	0,6	0,8	1,4	3,4	0,8	1,3	1,1	1,6	1,2	1,3	1,1	0,9	1,3	2,3	1,1	1,5	1,6	0,8	0,9	2,5	0,0	1,1
45	2,1	1,2	1,4	0,8	2,1	0,8	2,0	1,8	0,5	0,5	2,0	1,9	1,4	2,4	3,0	1,3	0,6	0,6	0,8	1,3	3,3	1,1	0,0

Node	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
24	0,5	1,0	0,8	3,5	2,1	2,8	0,7	1,2	4,9	3,3	0,9	0,6	3,2	3,4	2,8	3,5
25	0,5	1,0	0,8	3,3	2,4	3,0	0,5	1,4	4,6	3,5	0,8	0,4	3,2	3,6	3,0	3,7
26	0,5	0,8	0,2	2,4	2,9	3,6	1,3	0,8	3,7	4,1	0,1	1,2	4,1	4,2	1,8	4,3
27	2,5	2,5	2,2	1,1	4,6	3,5	3,3	2,1	2,5	4,0	2,2	3,2	6,6	4,2	2,0	5,9
28	1,0	1,3	1,0	3,6	2,1	2,7	1,8	0,8	3,9	3,2	1,1	1,6	3,9	3,4	1,9	3,5
29	0,9	0,7	1,1	3,7	2,9	3,5	0,4	2,0	5,1	4,0	1,2	0,6	3,7	4,1	3,1	4,2
30	0,7	0,8	1,0	3,5	2,7	3,3	0,4	1,8	4,9	3,8	1,0	0,5	3,6	4,0	3,3	4,1
31	0,7	1,0	0,4	2,7	2,6	3,3	1,5	0,5	4,0	3,8	0,4	1,4	4,6	3,9	1,6	3,9
32	0,7	1,0	0,5	2,7	2,5	3,3	1,6	0,5	4,1	3,8	0,4	1,5	4,5	4,0	1,6	3,9
33	1,0	1,5	1,2	3,8	2,9	3,5	0,16	2,0	5,1	4,1	1,3	0,2	3,0	4,2	3,6	3,5
34	0,8	1,3	1,1	3,6	2,8	3,4	0,4	1,9	5,0	3,9	1,1	0,4	2,8	4,1	3,4	3,3
35	0,3	0,9	0,6	3,2	2,5	3,1	0,5	1,4	4,5	3,6	0,7	0,4	3,4	3,8	2,6	3,9
36	2,0	2,3	2,1	3,5	0,7	1,3	2,1	2,4	4,3	1,9	2,1	2,0	0,4	2,0	1,3	2,1
37	1,9	1,6	2,2	4,7	3,9	4,5	0,1	3,0	6,1	5,0	2,2	0,2	4,0	5,1	4,1	4,5
38	0,1	0,7	0,4	3,7	2,8	4,3	0,7	1,3	4,2	3,8	0,5	0,6	3,6	4,0	2,4	4,1
39	0,7	0,9	0,4	3,3	2,8	3,9	1,4	0,6	4,3	3,9	0,3	1,3	4,2	4,1	1,7	4,1
40	0,8	1,0	0,5	2,9	3,2	3,4	1,5	0,6	4,7	4,3	0,4	1,4	4,3	5,1	2,1	4,5
41	0,5	0,8	0,6	3,9	2,2	4,5	1,3	0,8	3,6	3,2	0,7	1,2	3,5	3,4	1,9	3,5
42	0,1	0,7	0,4	3,7	2,8	4,3	6,5	1,3	4,2	3,8	0,5	0,6	3,5	3,9	2,4	4,0
43	2,1	1,8	2,4	5,7	4,2	6,3	0,7	3,3	5,6	5,2	2,5	0,8	3,9	5,4	4,4	4,4
44	1,0	1,4	1,2	3,6	2,1	4,2	1,2	1,1	3,5	3,1	1,3	1,1	3,0	3,3	2,6	3,4
45	1,1	1,4	0,9	2,5	2,5	3,1	1,9	0,0	3,7	3,3	0,8	1,8	4,2	3,5	0,9	3,8



Node	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	3,3	0,5	0,5	0,5	2,5	1,0	0,9	0,7	0,7	0,7	1,0	0,8	0,3	2,0	1,9	0,1	0,7	0,8	0,5	0,1	2,1	1,0	1,1
47	3,6	1,0	1,0	0,8	2,5	1,3	0,7	0,8	1,0	1,0	1,5	1,3	0,9	2,3	1,6	0,7	0,9	1,0	0,8	0,7	1,8	1,4	1,4
48	3,0	0,8	0,8	0,2	2,2	1,0	1,1	1,0	0,4	0,5	1,2	1,1	0,6	2,1	2,2	0,4	0,4	0,5	0,6	0,4	2,4	1,2	0,9
49	2,5	3,5	3,3	2,4	1,1	3,6	3,7	3,5	2,7	2,7	3,8	3,6	3,2	3,5	4,7	3,7	3,3	2,9	3,9	3,7	5,7	3,6	2,5
50	3,8	2,1	2,4	2,9	4,6	2,1	2,9	2,7	2,6	2,5	2,9	2,8	2,5	0,7	3,9	2,8	2,8	3,2	2,2	2,8	4,2	2,1	2,5
51	0,8	2,8	3,0	3,6	3,5	2,7	3,5	3,3	3,3	3,3	3,5	3,4	3,1	1,3	4,5	4,3	3,9	3,4	4,5	4,3	6,3	4,2	3,1
52	4,1	0,7	0,5	1,3	3,3	1,8	0,4	0,4	1,5	1,6	0,16	0,4	0,5	2,1	0,1	0,7	1,4	1,5	1,3	6,5	0,7	1,2	1,9
53	2,1	1,2	1,4	0,8	2,1	0,8	2,0	1,8	0,5	0,5	2,0	1,9	1,4	2,4	3,0	1,3	0,6	0,6	0,8	1,3	3,3	1,1	0,0
54	0,5	4,9	4,6	3,7	2,5	3,9	5,1	4,9	4,0	4,1	5,1	5,0	4,5	4,3	6,1	4,2	4,3	4,7	3,6	4,2	5,6	3,5	3,7
55	0,9	3,3	3,5	4,1	4,0	3,2	4,0	3,8	3,8	3,8	4,1	3,9	3,6	1,9	5,0	3,8	3,9	4,3	3,2	3,8	5,2	3,1	3,3
56	2,9	0,9	0,8	0,1	2,2	1,1	1,2	1,0	0,4	0,4	1,3	1,1	0,7	2,1	2,2	0,5	0,3	0,4	0,7	0,5	2,5	1,3	0,8
57	4,0	0,6	0,4	1,2	3,2	1,6	0,6	0,5	1,4	1,5	0,2	0,4	0,4	2,0	0,2	0,6	1,3	1,4	1,2	0,6	0,8	1,1	1,8
58	5,9	3,2	3,2	4,1	6,6	3,9	3,7	3,6	4,6	4,5	3,0	2,8	3,4	0,4	4,0	3,6	4,2	4,3	3,5	3,5	3,9	3,0	4,2
59	1,5	3,4	3,6	4,2	4,2	3,4	4,1	4,0	3,9	4,0	4,2	4,1	3,8	2,0	5,1	4,0	4,1	5,1	3,4	3,9	5,4	3,3	3,5
60	1,0	2,8	3,0	1,8	2,0	1,9	3,1	3,3	1,6	1,6	3,6	3,4	2,6	1,3	4,1	2,4	1,7	2,1	1,9	2,4	4,4	2,6	0,9
61	5,2	3,5	3,7	4,3	5,9	3,5	4,2	4,1	3,9	3,9	3,5	3,3	3,9	2,1	4,5	4,1	4,1	4,5	3,5	4,0	4,4	3,4	3,8

<b>Node</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>	<b>61</b>
<b>46</b>	0,0	0,6	0,3	3,6	2,7	4,2	0,8	1,1	4,1	3,7	0,4	0,7	3,7	3,9	2,3	4,0
<b>47</b>	0,6	0,0	0,6	3,9	3,0	4,5	1,3	1,4	4,4	4,1	0,7	1,2	4,2	4,2	2,5	4,3
<b>48</b>	0,3	0,6	0,0	3,3	2,8	3,9	1,1	0,9	4,2	3,8	0,1	1,0	3,9	4,0	2,0	4,1
<b>49</b>	3,6	3,9	3,3	0,0	4,1	3,0	3,7	2,4	2,1	3,6	2,5	3,5	6,1	3,7	2,4	5,5
<b>50</b>	2,7	3,0	2,8	4,1	0,0	1,9	2,8	3,1	4,9	2,5	2,8	2,7	1,0	2,2	1,9	1,3
<b>51</b>	4,2	4,5	3,9	3,0	1,9	0,0	3,4	2,8	1,0	1,4	3,6	3,3	4,0	1,0	1,7	2,7
<b>52</b>	0,8	1,3	1,1	3,7	2,8	3,4	0,0	2,9	5,9	4,8	2,1	0,1	4,2	5,0	4,0	4,7
<b>53</b>	1,1	1,4	0,9	2,4	3,1	2,8	2,9	0,0	3,2	3,3	0,8	1,8	4,5	3,5	0,9	3,8
<b>54</b>	4,1	4,4	4,2	2,1	4,9	1,0	5,9	3,2	0,0	0,6	3,7	3,4	4,1	1,7	1,8	3,4
<b>55</b>	3,7	4,1	3,8	3,6	2,5	1,4	4,8	3,3	0,6	0,0	3,4	4,4	7,8	2,1	3,2	5,8
<b>56</b>	0,4	0,7	0,1	2,5	2,8	3,6	2,1	0,8	3,7	3,4	0,0	1,1	4,0	4,1	1,9	4,2
<b>57</b>	0,7	1,2	1,0	3,5	2,7	3,3	0,1	1,8	3,4	4,4	1,1	0,0	4,3	5,1	4,1	4,8
<b>58</b>	3,7	4,2	3,9	6,1	1,0	4,0	4,2	4,5	4,1	7,8	4,0	4,3	0,0	2,2	1,6	2,3
<b>59</b>	3,9	4,2	4,0	3,7	2,2	1,0	5,0	3,5	1,7	2,1	4,1	5,1	2,2	0,0	2,4	2,9
<b>60</b>	2,3	2,5	2,0	2,4	1,9	1,7	4,0	0,9	1,8	3,2	1,9	4,1	1,6	2,4	0,0	3,3
<b>61</b>	4,0	4,3	4,1	5,5	1,3	2,7	4,7	3,8	3,4	5,8	4,2	4,8	2,3	2,9	3,3	0,0

### LAMPIRAN 3. Skrip Utama AFSA pada Matlab

```

function pushbutton1_Callback(hObject, eventdata,
handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata   reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)
clc;
[File, Path]=uigetfile({'*.xlsx;*.xls','Excel
Spreadsheet (*.xlsx, *xls)'}, 'Open');
if File~=0
    [Coor, txt]=xlsread(fullfile(Path,File),1);
    node=size(Coor,1)-1;

set(handles.uitable1,'data',Coor(:,2:3),'rowname',0:node);
    [Dist, txt]=xlsread(fullfile(Path,File),2);

set(handles.uitable2,'data',Dist(2:end,2:end),'rowname'
,0:node,'columnname',0:node);
    set(gcf,'Name',[ 'AFSA for mTSP - '
fullfile(Path,File)]);
    set(handles.uipanel1,'visible','on');
    set(handles.uipanel5,'visible','off');
    cla(handles.axes2,'reset');
    set(handles.axes2,'XTick',[],'YTick',[]);
    set(handles.pushbutton3,'string','Route');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata,
handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata   reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)
clc;
set(handles.uipanel1,'visible','on');
set(handles.uipanel5,'visible','off');
cla(handles.axes2,'reset');
set(handles.axes2,'XTick',[],'YTick',[]);
set(handles.pushbutton3,'string','Route');
cla(handles.axes1,'reset');

```

```

axes(handles.axes1);
xlim([0 1000]);
xlabel('Iteration (t)'); ylabel('TDistance (Z)');
set(handles.listbox1,'string','Solution','value',1,'UserData',[]);
set(handles.text12,'string','0 s');
pause(1);

N=str2num(get(handles.edit1,'string'));
Visual=str2num(get(handles.edit2,'string'));
Step=str2num(get(handles.edit3,'string'));
Delta=str2num(get(handles.edit4,'string'));
TryNumber=str2num(get(handles.edit5,'string'));
MaxIter=str2num(get(handles.edit6,'string'));
NumSales=str2num(get(handles.edit7,'string'));
DistData=get(handles.uitable2,'data');
tic;
axes(handles.axes1);
[Routef, TDistf,
TDistancef]=AFSAmTSP(N,Visual,Delta,Step,TryNumber,MaxIter,NumSales,DistData);
output='Solution';
for i=1:NumSales
    str=['Route ' num2str(i) ' : 0-'];
    for j=1:length(Routef{i})
        str=[str num2str(Routef{i}(j)) '-'];
    end
    str=[str '0'];
    output={char(output);str;['T.Distance : '
num2str(TDistf(i))]};
end
set(handles.listbox1,'string',char(output),'UserData',Routef);
set(handles.text12,'string',[num2str(toc) ' s']);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
clc;
str=get(handles.pushbutton3,'string');
if strcmp(str,'Route')
    Route=get(handles.listbox1,'UserData');

```

```

    if ~isempty(Route)
        NumSales=size(Route,2);
        Coordata=get(handles.uitable1,'data');
        DistData=get(handles.uitable2,'data');
        node=size(Coordata,1);
        Coord=[Coordata(:,2) Coordata(:,1)];

        cla(handles.axes2,'reset');
        axes(handles.axes2);
        color=['r';'g';'b';'c';'m';'y';'k'];
        for i=1:NumSales
            RouteMap=zeros(node);

RouteMap(1,Route{i}(1)+1)=DistData(1,Route{i}(1)+1);
            for j=1:length(Route{i})-1

RouteMap(Route{i}(j)+1,Route{i}(j+1)+1)=DistData(Route{
i}(j)+1,Route{i}(j+1)+1);
                end

RouteMap(Route{i}(end)+1,1)=DistData(Route{i}(end)+1,1)
;
                subplot(RouteMap,Coord,['-' color(mod(i-
1,7)+1)]);
                hold on;
                lgd{i}=['Route ' num2str(i)];
            end
            legend(lgd);
            inlat=max(Coord(:,2))-min(Coord(:,2));
            inlon=max(Coord(:,1))-min(Coord(:,1));
            for i=1:node

line(Coord(i,1),Coord(i,2),'Marker','s','MarkerEdgeColor'
,'k','MarkerFaceColor','r','MarkerSize',5);
                text(Coord(i,1),Coord(i,2),sprintf(['N'
num2str(i-1)]));
            end

line(Coord(1,1),Coord(1,2),'Marker','s','MarkerEdgeColor'
,'k','MarkerSize',15);
            set(handles.axes2,'Xlim',[min(Coord(:,1))-
inlon/50
max(Coord(:,1))+inlon/5],'XTick',[],'YTick',[]);
            hold off;

            set(handles.uipanel1,'visible','off');
            set(handles.uipanel5,'visible','on');

```

```

        set(handles.pushbutton3,'string','Data');
    end
elseif strcmp(str,'Data')
    set(handles.uipanel1,'visible','on');
    set(handles.uipanel5,'visible','off');
    set(handles.pushbutton3,'string','Route');
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
clc;
movegui(gcf,'center');
set(gcf,'Name','ASFA for mTSP');
set(handles.uipanel1,'visible','on');
set(handles.uitable1,'data',[],'rowname','');
set(handles.uitable2,'data',[],'rowname','','columnname',
'');
set(handles.uipanel5,'visible','off');
cla(handles.axes2,'reset');
set(handles.axes2,'XTick',[],'YTick',[]);
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.edit5,'string','');
set(handles.edit6,'string','');
set(handles.edit7,'string','');
set(handles.pushbutton3,'string','Route');
cla(handles.axes1,'reset');
axes(handles.axes1);
xlim([0 1000]);
xlabel('Iteration (t)'); ylabel('TDistance (Z)');
set(handles.listbox1,'string','Solution','value',1,'Use
rData',[]);
set(handles.text12,'string','0 s');

```