

# Algoritma Emperor Penguin pada Efisiensi Pengiriman Produk UMKM dengan Konsep Pembagian Ongkos Kirim

Hendrawan Armanto<sup>a</sup>, C. Pickerling<sup>a</sup>, Kevin Susanto<sup>a</sup>

<sup>a</sup>Teknik Informatika, Institut Sains dan Teknologi Terpadu Surabaya  
E-mail: hendrawan@stts.edu, pickerling@stts.edu, ksksk0408@gmail.com

**Abstrak** — Di Era Pandemi saat ini, perkembangan ekonomi atau bisnis Usaha Mikro, Kecil, dan Menengah (UMKM) semakin menjamur di masyarakat. Perkembangan UMKM ini tidak hanya berhenti pada jumlah saja akan tetapi juga pada sistem pemesanan UMKM tersebut. Saat ini banyak UMKM yang menerapkan konsep *Pre-Order (PO)* di dalam pemesanannya dimana mayoritas UMKM tersebut merupakan UMKM yang memasarkan produknya melalui media sosial atau *chatting*. Munculnya sistem baru ini memunculkan juga beberapa permasalahan bagi UMKM, salah satu diantaranya adalah besarnya biaya pengiriman apabila sebuah produk dikirim secara khusus ke seorang pembeli. Hal ini mungkin tidak terlihat memberatkan bagi UMKM akan tetapi permasalahan ini memberatkan pelanggan dan semakin mahal biaya yang harus dikeluarkan pelanggan maka akan semakin kecil kemungkinan pelanggan membeli produk di UMKM tersebut. Penelitian ini berfokus pada dua hal, yang pertama adalah menyediakan *marketplace* bagi UMKM *Pre-Order* yang dapat mengatasi permasalahan biaya pengiriman dan kedua adalah dampak Algoritma Emperor Penguin pada pencarian rute pengiriman. *Marketplace* UMKM *Pre-Order* yang dikembangkan akan menerapkan teknologi *flutter* dan berbasis *mobile apps*. Hal ini dikarenakan semakin besarnya pengaruh *mobile apps* di kalangan masyarakat dibandingkan *web apps* atau *desktop apps*. *Mobile Apps* akan secara otomatis melakukan pencarian rute pengiriman untuk seluruh *pre-order* yang terjadi pada hari tertentu dan melakukan *assign* kepada *driver* yang bertugas dimana rute yang dicari bukan hanya berdasarkan jarak melainkan juga berdasarkan biaya pengiriman yang dikeluarkan oleh keseluruhan pembeli pada 1x pengiriman. Algoritma Emperor Penguin akan berusaha mencari rute pengiriman yang terdekat dan biaya yang dikeluarkan pembeli yang terkecil. Setelah melalui berbagai uji coba, dapat disimpulkan bahwa 94.7% UMKM yang didukung oleh penelitian ini puas dan merasa terbantu, 97.4% pelanggan merasa terbantu dengan sistem pembagian ongkos kirim, dan Algoritma Emperor Penguin bekerja dengan baik dan dapat menghasilkan rute optimal dengan ongkos terkecil.

**Kata Kunci** — UMKM, Flutter, Emperor Penguin Optimizer, Pencarian Rute

Naskah Masuk : 19 November 2021  
Naskah Direvisi : 02 Desember 2021  
Naskah Diterima : 02 Desember 2021



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

## I. PENDAHULUAN

Pada era industri 4.0, semua kebutuhan manusia semakin dimudahkan oleh internet. Melalui internet manusia dapat berkomunikasi satu dengan yang lain. Bersamaan dengan meningkatkan penggunaan internet maka aplikasi berbasis internet pun semakin meningkat tidak terkecuali aplikasi *mobile*. Saat ini teknologi *mobile* yang paling diminati dan berkembang adalah teknologi *mobile apps* yang dapat multi platform (minimal mencakup android dan iOS). Hal ini menyebabkan terjadinya pergeseran pengembangan aplikasi yang dulunya berfokus pada *windows apps*, kemudian digantikan oleh *web apps*, dan saat ini mulai digantikan oleh *mobile apps*. Tentu saja pergeseran ini memberikan dampak positif, saat ini user dapat mengakses internet dan aplikasi dimana pun mereka berada sehingga tidak hanya mempermudah user tetapi juga memberikan ide-ide baru bagi pelaku bisnis.

Di era pandemi ini, salah satu bisnis yang sangat berkembang adalah Usaha Mikro, Kecil, dan Menengah (UMKM) terutama UMKM yang berfokus pada makanan dan minuman. Selama ini pelaku bisnis harus menyediakan tempat untuk berjualan dan menyediakan produk yang dijual terlebih dahulu akan tetapi dengan perkembangan *mobile apps* yang sedemikian pesat, UMKM jaman sekarang bergerak melalui media sosial dan menggunakan sistem *pre-order*. Sehingga dana awal yang dibutuhkan oleh sebuah UMKM tidak terlalu besar akan tetapi apabila dimanage dengan baik akan mendatangkan keuntungan yang besar. Hal ini baik tetapi memiliki beberapa kelemahan, dimana salah satu kelemahannya adalah pada segi promosi dan biaya pengiriman. Dari segi promosi, UMKM yang baru tidak memiliki cara untuk mempromosikan UMKM nya selain melalui viral marketing atau fitur promosi yang disediakan oleh media sosial. Sedangkan dari segi biaya pengiriman (terutama bagi UMKM yang produknya mayoritas tidak mahal dan konsumen yang membeli bertempat tinggal relative jauh dari tempat UMKM), biaya pengiriman yang mahal akan menyebabkan konsumen mempertimbangkan ulang pembelian produk di UMKM tersebut. Apalagi disaat UMKM tersebut adalah UMKM baru yang sedang mencari nama maka biaya pengiriman akan menyebabkan konsumen tidak jadi beli.

Melihat permasalahan yang dihadapi oleh UMKM tersebut maka penelitian ini difokuskan untuk memberikan

solusi terkait kedua permasalahan tersebut. Diharapkan melalui penelitian ini order UMKM yang bergabung didalamnya dapat ditingkatkan dimana peningkatan order tersebut dikarenakan *marketplace* yang dihasilkan penelitian ini dapat menjadi sarana promosi UMKM baru dan sistem pembagian ongkos kirim yang ditawarkan di penelitian ini akan mengurangi biaya kirim yang dibayarkan oleh konsumen sehingga memperbesar keinginan konsumen untuk mencoba produk baru atau membeli produk.

II. PENELITIAN TERDAHULU

Penelitian terakhir yang dibahas di section ini berkaitan dengan penelitian di bidang pencarian rute saja dan bukan di bagian peningkatan pesanan UMKM. Walaupun penelitian ini juga bertujuan untuk meningkatkan pemesanan UMKM akan tetapi fokus permasalahan utama dari penelitian ini tetap terletak pada permasalahan pengiriman dan biaya pengiriman dari produk UMKM.

Penelitian terkait pencarian rute atau yang lebih dikenal dengan Travelling Salesman Problem telah dimulai sejak tahun 80an. Walaupun penelitian ini telah lama dilakukan kana tetapi sistem pencarian rute terus berkembang seiring dengan berkembangnya jaman. Berbagai jenis Travelling Salesman Problem ditemukan dan diteliti untuk dapat diselesaikan dengan cepat, optimal, dan efisien. Metode paling pimitif akan tetapi masih paling baik untuk data kecil adalah keluarga backtracking tetapi metode ini membutuhkan waktu yang sangat lama ketika berhadapan dengan data besar. Hal tersebut dikarenakan backtracking dan keluarganya berusaha mencoba semua kemungkinan jalan sehingga dapat dikatakan mustahil apabila digunakan untuk data besar.

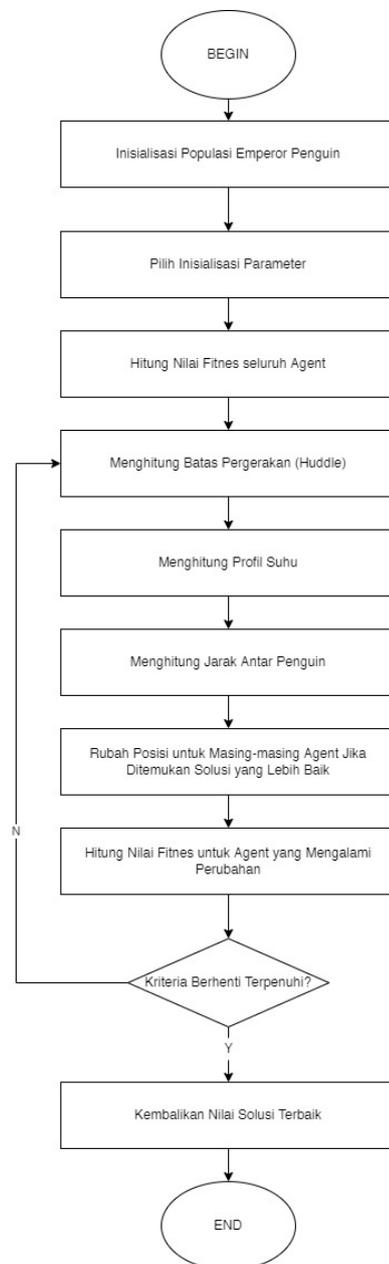
Di tahun yang hampir bersamaan, Algoritma Genetik diperkenalkan dimana algoritma ini merupakan cikal bakal algoritma evolutionary yang saat ini semakin lama semakin berkembang. Dapat dikatakan hampir setiap tahun muncul minimal satu algoritma evolutionary baru yang masing-masing algoritmanya meniru tingkah laku makhluk hidup atau gejala alam yang berbeda-beda dan menyatakan kalau dapat memberikan solusi paling optimal untuk kasus-kasus kombinasi [1], [2], [3]. Hal ini dibuktikan dengan penelitian-penelitian menggunakan algoritma evolutionary untuk menyelesaikan berbagai permasalahan yang ada di luar sana. Mulai dari permasalahan terkait optimasi parameter [4], penjadwalan praktikum [5], penentuan dosen pengampu mata kuliah [6], bahkan salah satu permasalahannya adalah traveling salesman problem.

Penelitian terkait traveling salesman problem atau pencarian rute terus berkembang bergantung kepada permasalahan rute yang sedang diteliti dan algoritma evolutionary yang digunakan. Beda kasus beda algoritma jelas akan menghasilkan beda solusi dan hasil. Mayoritas penelitian terkait traveling salesman problem umumnya dibedakan menjadi dua yaitu traveling salesman problem (TSP) secara umum dan multiple traveling salesman problem (MTSP) yang kemudian dikembangkan menjadi colored traveling salesman problem (CTSP). Pada penelitian untuk permasalahan TSP umum telah dilakukan ujicoba dengan beberapa algoritma evolutionary diantaranya adalah

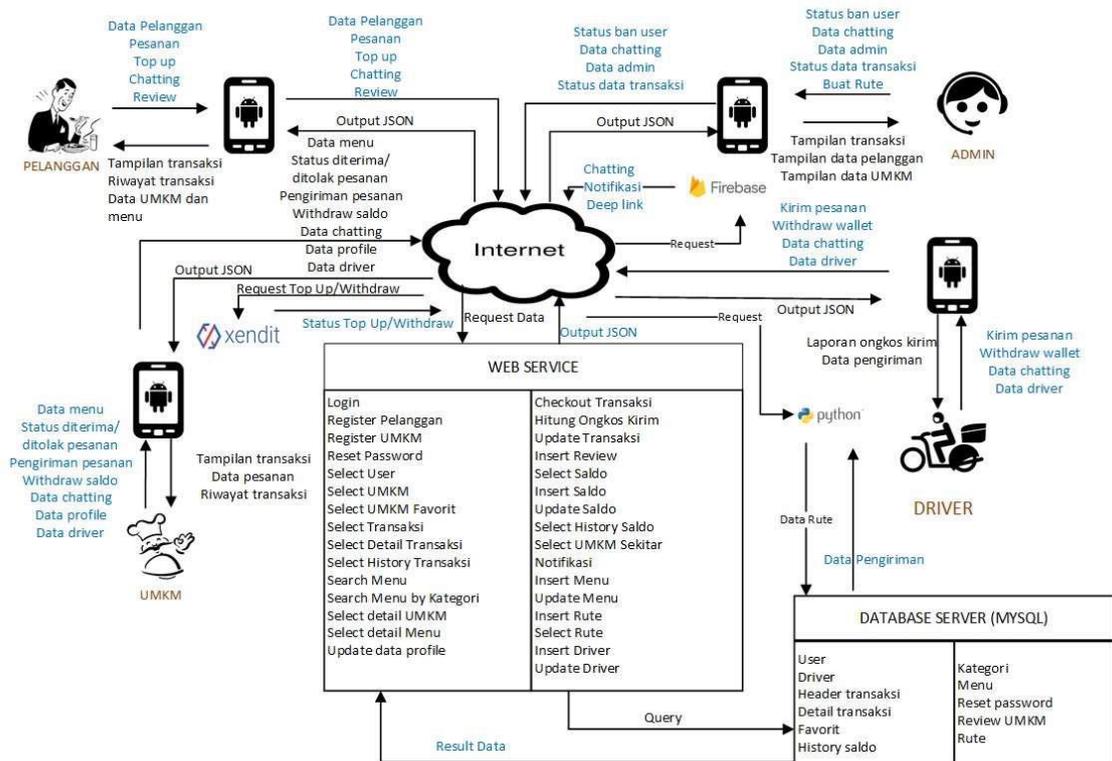
Algoritma Genetik yang dilakukan perubahan pada operator crossovernya [7], Particle Swarm Optimization (PSO) dan perkembangannya yang disebut sebagai MPSO [8], Algoritma Ant Colony Optimization (ACO) dan perkembangannya yang disebut sebagai ACS [9], bahkan terdapat juga penelitian yang berusaha membandingkan beberapa algoritma (termasuk algoritma evolutionary) untuk mencari solusi paling optimal dari TSP [10]. Sedangkan pada CTSP telah dilakukan penelitian menggunakan algoritma genetik dan pengembangannya yang menggabungkan Algoritma Genetik dengan Hill Climbing dan Algoritma Greedy [11].

III. EMPEROR PENGUIN OPTIMIZER [12]

Sama halnya dengan algoritma evolutionary lainnya, Algoritma Emperor Penguin Optimizer juga meniru perilaku dari makhluk hidup yang ada di sekitar kita, dimana untuk algoritma ini yang ditiru adalah perilaku kerumunan penguin kaisar.



Gambar. 1. Flow Chart Algoritma Emperor Penguin Optimizer



Gambar. 2. Arsitektur Sistem Marketplacve UMKM

Penguin kaisar umumnya suka berkerumun dan bergerak bersama-sama, akan tetapi berbeda dengan binatang lain yang bergerak hanya dengan tujuan mencapai sebuah titik sasaran, penguin kaisar berkerumun juga dengan tujuan menghangatkan badan mereka. Dengan bergerak secara berkerumun, antar penguin akan saling menghangatkan dan dapat bergerak menuju target sasaran mereka dengan cepat (dalam hal ini target sasaran kita adalah solusi paling optimal). Gambar 1 merupakan flowchart dari algoritma Emperor Penguin Optimizer dimana diawali dengan inisialisasi populasi emperor penguin (sebuah penguin / agent melambangkan sebuah calon solusi) dan diakhiri dengan mengembalikan penguin terbaik (calon solusi paling optimal dan dapat digunakan sebagai solusi permasalahan yang sedang diteliti)

IV. SISTEM MARKETPLACE UMKM

Pada section ini, akan diberikan gambaran singkat terkait sistem marketplace yang dikembangkan di penelitian ini mulai dari desain arsitektur hingga desain interface dari aplikasi tersebut. Walaupun sistem ini merupakan penunjang dari fokus penelitian tetapi tetap dirasa penting untuk mengembangkan sebuah sistem yang baik dan terbukti berguna untuk masyarakat.

A. Desain Arsitektur

Gambar 2 merupakan desain arsitektur yang ditawarkan di penelitian ini. Melalui gambar tersebut dapat dilihat bahwa terdapat 4 aktor di sistem ini yaitu dari sisi UMKM, konsumen atau pelanggan, pengirim (driver) dan admin marketplace. Pengembangan sistem secara keseluruhan memanfaatkan flutter framework [13]–[16] yang diyakini merupakan salah satu framework multi platform terbaik saat

ini dan diharapkan dapat bertahan lama kedepannya sehingga sistem ini dapat dipakai tidak hanya satu atau dua tahun melainkan beberapa tahun kedepan.

Dari keseluruhan fitur yang ditawarkan oleh sistem ini, yang menjadi fokus utama penelitian ini terletak pada alur berikut ini:

1. UMKM membuka *Pre-Order* dan menentukan batas tanggal pembelian terakhir
2. Pelanggan membeli pesanan (*Pre-Order*) untuk tanggal tertentu
3. UMKM menentukan apakah pesanan diterima atau tidak, jika ditolak maka uang akan dikembalikan
4. UMKM menentukan list *driver* yang digunakan
5. Pada waktu penutupan *Pre-Order*, secara otomatis sistem akan melakukan pencarian rute terbaik dan memberikan list pengiriman untuk seluruh *driver* yang ditugaskan.
6. *Driver* mengirimkan pesanan sesuai list pengiriman

B. Desain Interface

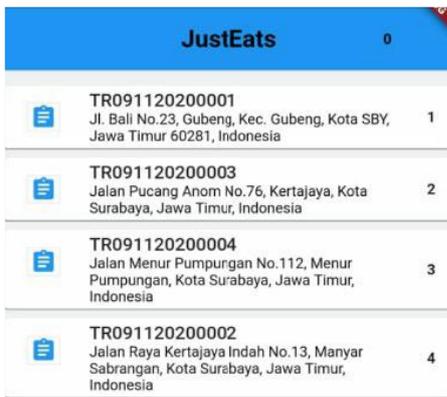
Berikut adalah beberapa desain interface yang dikembangkan pada sistem marketplace UMKM ini. Tidak seluruh desain interface ditampilkan, pada penelitian ini hanya ditampilkan desain interface yang berhubungan dengan fokus utama penelitian.

Gambar 3 merupakan gambar desain interface untuk penambahan drier pada pembukaan *Pre-Order* yang akan tutup dan diproses oleh sistem rute pengirimannya. Tanpa penentuan *driver* terlebih dahulu maka sistem tidak dapat melakukan proses secara otomatis. Penambahan *driver* bertujuan untuk mengetahui berapa jumlah *driver* yang akan diminta melakukan pengiriman pesanan.



Gambar. 3. Interface Penambahan Driver

Gambar 4 merupakan interface list pengiriman untuk *driver* yang telah ditentukan secara otomatis oleh sistem pencarian rute otomatis. Interface ini tidak saja menampilkan berupa urutan pengiriman akan tetapi juga berupa peta dimana peta digambarkan menggunakan google map API.



Gambar. 4. Interface List Pengiriman Driver

### V. PENCARIAN RUTE DENGAN ALGORITMA EMPEROR PENGUIN OPTIMIZER

Pada section ini akan dijabarkan penerapan algoritma emperor penguin optimizer pada pencarian rute mulai dari ilustrasi permasalahan yang dihadapi, representasi permasalahan menjadi agent atau individu penguin, perumusan nilai fitness, hingga contoh hasil akhir yang dihasilkan oleh algoritma.

#### A. Ilustrasi Permasalahan yang Dihadapi

Data yang digunakan pada pelatihan ini terdiri dari periode pesanan, nama pelanggan, pesanan, alamat UMKM (asal pengiriman), alamat pelanggan, jumlah *driver* yang akan mengirimkan pesanan, jarak antara alamat UMKM ke alamat pelanggan, jarak antara alamat pelanggan satu ke alamat pelanggan lainnya (penentuan jarak menggunakan bantuan Google API), dan harga per KM (per KM menggunakan fix rate yang ditentukan oleh UMKM).

Untuk setiap periode pesanan yang ditutup akan dilakukan proses pencarian rute dan mengembalikan list pengiriman untuk masing-masing *driver* yang telah ditentukan sebelumnya. Di dalam pencarian rute, nama

pelanggan dan pesanan tidak menjadi tolak ukur pencarian. Pencarian hanya menggunakan tolak ukur alamat dan harga per KM yang telah ditentukan sebelum pencarian rute dilakukan.

#### B. Representasi Individu Penguin

Sebuah individu penguin di dalam populasi merupakan sebuah calon solusi untuk permasalahan pencarian rute. Akan tetapi permasalahan tersebut harus direpresentasikan terlebih dahulu sebelum dapat diproses oleh Algoritma Emperor Penguin Optimizer. Dikarenakan permasalahan ini membutuhkan  $N$  *Driver* maka representasi yang diajukan di penelitian ini adalah sebagai berikut:

1. Panjang sel pada individu adalah

$$\text{Panjang Sel} = \text{Jumlah Lokasi Pengiriman} * \text{Jumlah Driver } (N)$$

2. Isi dari sel dalam sebuah individu adalah bilangan desimal yang melambangkan urutan dari lokasi dan *driver* yang ditugaskan

Berikut adalah contoh representasi yang diajukan pada penelitian ini untuk kasus 4 lokasi dan 2 driver saja.

Representasi:

0.70	0.52	0.17	0.38	0.13	0.71	0.37	0.79
------	------	------	------	------	------	------	------

Untuk memahami arti dari representasi tersebut maka berikut adalah langkah membalik dari representasi menjadi list pengiriman *driver*.

1. Setiap sel diberi nomor urut berdasarkan nilai selnya
2. Untuk  $M$  jumlah lokasi dan  $N$  jumlah *driver* maka  $M$  urutan pertama untuk *driver* 1,  $M$  urutan kedua untuk *driver* 2, dan  $M$  urutan ke  $N$  untuk *driver* ke  $N$
3. Dilakukan penentuan list pengiriman urut dari kiri ke kanan

Berikut adalah contoh perubahan representasi menjadi list pengiriman *driver*. Contoh hanya untuk kasus 4 lokasi dan 2 *driver* saja.

Berdasarkan contoh representasi diatas maka ditentukan nomor urut terlebih dahulu:

0.70	0.52	0.17	0.38	0.13	0.71	0.37	0.79
6	5	2	4	1	7	3	8

Berdasarkan nomor urut ditentukan untuk pengiriman *driver* ke berupa:

0.70	0.52	0.17	0.38	0.13	0.71	0.37	0.79
6	5	2	4	1	7	3	8
B2	A2	B1	D1	A1	C2	C1	D2

dimana huruf A hingga D melambangkan lokasi ke  $i$  (terdapat  $M$  lokasi) dan angka 1 dan 2 melambangkan *driver* ke  $j$  (terdapat  $N$  *driver*)

Dilakukan penentuan list pengiriman *driver* dari urutan kiri ke kanan dimana 1 melambangkan pesanan yang dikirimkan *driver* 1 dan 2 melambangkan pesanan yang dikirimkan oleh *driver* 2 (Apabila terdapat  $N$  *driver* maka angka yang

muncul maksimum N):

0.70	0.52	0.17	0.38	0.13	0.71	0.37	0.79
6	5	2	4	1	7	3	8
B2	A2	B1	D1	A1	C2	C1	D2
2	2	-	1	-	2	-	-

List urutan pengiriman *driver* yang diperoleh berdasarkan contoh representasi diatas adalah:

*Driver 1:*

- Mengirimkan ke Lokasi Kedua (B)
- Mengirimkan ke Lokasi Pertama (A)
- Mengirimkan ke Lokasi Ketiga (C)

*Driver 2:*

- Hanya mengirimkan ke Lokasi Keempat (D)

*C. Perhitungan Nilai Fitness*

Nilai fitness pada algoritma evolutionary adalah hal penting yang menjadi dasar arah pergerakan perilaku individu, dalam hal ini arah pergerakan penguin. Untuk penelitian ini, kami mengajukan dan mencoba 2 cara perhitungan nilai fitness, antara lain:

1. Total jarak antar lokasi pengiriman untuk seluruh *driver*  
 Pada perhitungan fitness ini, kami berasumsi bahwa jarak sudah mewakili biaya, apabila jarak semakin besar maka secara otomatis biaya juga akan semakin mahal. Maka dengan meminimalkan jarak maka secara otomatis akan meminimalkan biaya.
2. Total ongkos kirim yang harus dibayarkan bersama oleh seluruh pelanggan dalam satu *Pre-Order*.  
 Pada perhitungan fitness ini, kami berasumsi bahwa jarak tidak mewakili ongkos kirim yang dibayarkan mengingat perhitungan ongkos kirim tidak murni berdasarkan jarak akan tetapi juga ada biaya tambahan. Berikut adalah equation untuk perhitungan ongkos kirim yang kami lakukan di penelitian ini:

$$\begin{aligned}
 & \text{Total Ongkos Kirim} \\
 & = \text{Jarak} * 1.500 \\
 & + \text{Biaya Tambahan}
 \end{aligned}$$

Nilai Biaya Tambahan berdasarkan jarak, apabila jarak antar pengiriman tidak lebih dari 5 km maka tidak dikenakan biaya tambahan. Akan tetapi jika lebih dari 5 km dikenakan biaya tambahan sebesar X.

Pada akhirnya nilai ongkos kirim yang harus dibayarkan oleh masing-masing pelanggan adalah sebagai berikut:

$$\begin{aligned}
 & \text{Ongkos Kirim} \\
 & = \left[ \frac{\text{Jarak UMKM ke Pelanggan}}{\text{Total Jarak UMKM ke Seluruh Pelanggan}} \right] \\
 & * \text{Total Ongkos Kirim}
 \end{aligned}$$

Perhitungan ongkos kirim berdasarkan proporsi jarak UMKM ke pelanggan dilakukan dengan tujuan agar pelanggan yang dekat dengan UMKM tetap membayar lebih kecil dari pada pelanggan yang jauh dari UMKM. Jika tidak demikian dan hanya menggunakan rata-rata

maka dapat merugikan pelanggan yang dekat dengan UMKM.

VI. HASIL EKSPERIMEN DAN PENELITIAN

Pada penelitian ini, kami melakukan dua macam uji coba. Uji coba pertama adalah uji coba kelayakan sistem *marketplace* UMKM di kalangan UMKM dan pelanggan UMKM. Sedangkan uji coba kedua adalah uji coba terkait hasil pencarian rute yang dilakukan oleh Algoritma Emperor Penguin.

A. Uji coba Kelayakan Sistem Marketplace UMKM

Sekali lagi walaupun sistem ini bukan fokus utama penelitian ini tetapi semua sistem yang layak di mata pengguna dapat membuktikan fungsionalitas dan kehandalan dari fitur-fitur di dalamnya termasuk pencarian list pengiriman pesanan otomatis. Target uji coba kelayakan sistem ini adalah 38 pelanggan UMKM, 19 UMKM, dan 9 *Driver* yang bertugas mengirimkan pesanan sesuai dengan list pengiriman otomatis.

TABEL I  
HASIL KUISIONER SECARA UMUM

Topik	A	B	C	D	E
Tampilan Fitur pada Aplikasi	70.2%	26.3%	3.5%	0%	0%
Kecepatan Pemakaian Aplikasi	66.7%	28.1%	5.3%	0%	0%
Kemudahan Informasi	71.9%	24.6%	3.5%	0%	0%
Kemudahan Pemakaian	71.9%	22.8%	5.3%	0%	0%
Penilaian Secara Menyeluruh	73.7%	22.8%	3.5%	0%	0%

- A = Sangat Baik
- B = Baik
- C = Cukup
- D = Kurang
- E = Sangat Kurang

Tabel I merupakan rangkuman dari pertanyaan-pertanyaan terkait topik umum. Melalui tabel ini dapat dilihat dan disimpulkan bahwa diatas 88% (minimal 66% sangat baik dan minimal 22% baik) pengguna merasa puas dengan sistem yang dikembangkan. Walaupun tidak disangkal bahwa masih ada 3-5% pengguna yang merasa sistem ini cukup dan masih dapat ditingkatkan lagi.

Apabila tabel I dikhususkan untuk pertanyaan-pertanyaan umum, tabel II merupakan rangkuman dari pertanyaan-pertanyaan khusus untuk UMKM dan bukan untuk pelanggan. Hal ini kami rasa penting mengingat tujuan kami adalah mendukung UMKM di dalam menjalankan usaha masing-masing. Melalui tabel II tersebut dapat disimpulkan bahwa diatas 84.3% (minimal 63.2% sangat baik dan minimal 21.1% baik) UMKM merasa puas dengan sistem yang dikembangkan, bahkan apabila difokuskan ke topik peningkatan omzet penjualan, 94.7% menyatakan bahwa omzet penjualan mereka mengalami peningkatan dibandingkan hanya menggunakan sosial media saja. Meskipun hasil kuisiner menandakan hal baik tetapi masih ada 5-10% UMKM yang merasa cukup dengan sistem ini.

TABEL II  
HASIL KUISIONER KHUSUS UMKM

Topik	A	B	C	D	E
Kemudahan Manajemen Produk	73.7%	21.1%	5.3%	0%	0%
Kemudahan Manajemen Pesanan	63.2%	26.3%	10.5%	0%	0%
Kemudahan Proses Komunikasi dengan Pelanggan	63.2%	28.1%	8.7%	0%	0%
Fungsionalitas Fitur Pencarian List Pengiriman	68.4%	21.1%	10.5%	0%	0%
Peningkatan Omzet Penjualan Produk	68.4%	26.3%	5.3%	0%	0%

A = Sangat Baik  
B = Baik  
C = Cukup  
D = Kurang  
E = Sangat Kurang

Tabel III merupakan rangkuman dari pertanyaan-pertanyaan yang dikhususkan untuk pelanggan UMKM. Melalui pertanyaan-pertanyaan ini, kami dapat menyimpulkan bahwa 76.3% (minimal 60.5% sangat baik dan minimal 15.8% baik) pelanggan merasa puas dan terbantu oleh sistem yang dikembangkan. Apabila kita fokuskan ke tujuan penelitian ini yaitu fitur pembagian ongkos kirim maka dapat kita katakan 97.4% pelanggan menyukai dan mendukung fitur ini.

TABEL III  
HASIL KUISIONER KHUSUS PELANGGAN

Topik	A	B	C	D	E
Kemudahan Pemesanan Produk	73.7%	23.7%	2.6%	0%	0%
Kemudahan Tracking Pemesanan	60.5%	28.9%	10.5%	0%	0%
Kemudahan Proses Komunikasi dengan UMKM	71.1%	26.3%	2.6%	0%	0%
Fitur Pembagian Ongkos Kirim	81.6%	15.8%	2.6%	0%	0%

A = Sangat Baik  
B = Baik  
C = Cukup  
D = Kurang  
E = Sangat Kurang

Rangkuman terakhir terdapat pada Tabel IV, dimana tabel ini merupakan tabel pertanyaan-pertanyaan khusus *driver*. Melalui tabel ini dapat kami simpulkan bahwa 77.7% (minimal 66.7% sangat baik dan minimal 11.1% baik) *driver* merasa puas dan terbantu dengan sistem yang dikembangkan. Dimana apabila kita fokuskan ke tujuan utama penelitian ini maka dapat kita simpulkan bahwa hasil pencarian list pengiriman pesanan otomatis disukai oleh *driver*. Walaupun tidak dapat disangkal juga bahwa masih ada 11% *driver* yang merasa hasil ini cukup dan masih harus ditingkatkan.

TABEL IV  
HASIL KUISIONER KHUSUS DRIVER

Topik	A	B	C	D	E
Pemerataan Pengiriman Pesanan	77.8%	11.1%	11.1%	0%	0%
Effisiensi Rute Pengiriman yang Dihilangkan	66.7%	22.2%	11.1%	0%	0%

A = Sangat Baik  
B = Baik  
C = Cukup  
D = Kurang  
E = Sangat Kurang

**B. Ujicoba Pencarian List Pengiriman dengan Algoritma Emperor Penguin Optimizer**

Di dalam pencarian list pengiriman, kita tidak dapat langsung menemukan hasil yang terbaik tetapi beberapa topik ujicoba tetap harus dilakukan. Masing-masing topik akan memiliki beberapa percobaan dimana setiap percobannya telah kami jalankan hingga lima kali.

1. Uji Coba Fungsi Fitnes

Telah dijelaskan pada section 5.C. pada penelitian ini kami mengajukan dua jenis fungsi fitnes. Oleh sebab itu, dengan beberapa data kecil (tabel V) kami lakukan percobaan untuk masing-masing fungsi fitnes.

TABEL V  
HASIL PERCOBAAN FUNGSI FITNES

Jenis Data	Fungsi 1	Fungsi 2
5 Lokasi dan 1 Driver	Rute Pengiriman Efisien	Rute Pengiriman Efisien dan Rata-rata Ongkos Kirim yang dibayarkan lebih kecil dari Fungsi 1
5 Lokasi dan 2 Driver	Rute Pengiriman Efisien	Rute Pengiriman Efisien dan Rata-rata Ongkos Kirim yang dibayarkan lebih kecil dari Fungsi 1
10 Lokasi dan 3 Driver	Rute Pengiriman Efisien	Rute Pengiriman Efisien dan Rata-rata Ongkos Kirim yang dibayarkan lebih kecil dari Fungsi 1

2. Uji Coba Parameter Algoritma Emperor Penguin  
Setelah kami menemukan fungsi fitnes yang tepat, masing-masing algoritma evolutionary juga membutuhkan pengaturan parameter yang tepat untuk setiap permasalahan yang dikerjakan. Begitu hal nya dengan algoritma emperor penguin, agar dapat mencari solusi dengan baik maka beberapa parameter harus diujicobakan sehingga algoritma dapat berjalan secara

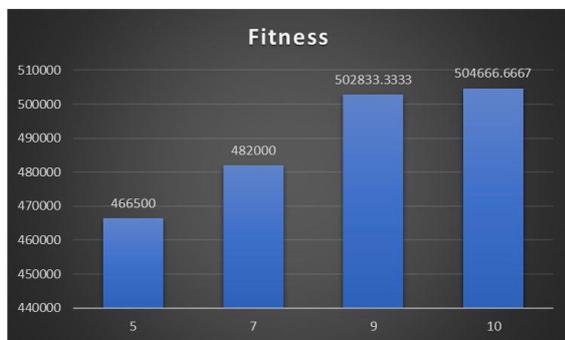
optimal dan efisien. Tabel VI merupakan tabel parameter yang dicoba beserta hasil yang berupa nilai fitness (rata-rata dari lima kali percobaan) dimana semakin kecil nilai fitness maka semakin baik solusi yang dihasilkan.

TABEL VI  
HASIL PERCOBAAN PARAMETER EPO

M	f	l	Max Iterasi	n	Fitness
1	2.4	1.7	800	80	218.000
2	2.4	1.7	800	80	196.666
3	2.4	1.7	800	80	213.166
4	2.4	1.7	800	80	240.500
5	2.4	1.7	800	80	208.000
2	2.2	1.7	800	80	222.333
2	2.5	1.7	800	80	230.833
2	2.6	1.7	800	80	230.500
2	2.8	1.7	800	80	197.666
2	2.4	1.6	800	80	228.500
2	2.4	1.8	800	80	215.666
2	2.4	1.9	800	80	225.000
2	2.4	1.7	400	80	229.666
2	2.4	1.7	600	80	219.000
2	2.4	1.7	1000	80	222.500
<b>2</b>	<b>2.4</b>	<b>1.7</b>	<b>800</b>	<b>50</b>	<b>181.250</b>
2	2.4	1.7	800	100	195.625

3. Uji Coba 50 Lokasi Pengiriman

Dengan berbekalkan fungsi fitness dan parameter yang telah diuji sebelumnya. Kami melakukan percobaan untuk 50 lokasi pengiriman (nilai fitness yang ditampilkan adalah rata-rata dari lima kali percobaan) dengan pilihan beberapa *driver* (5-10 *driver*).



Gambar 5. Hasil Percobaan 50 Lokasi Pengiriman

Berdasarkan hasil yang ditunjukkan pada gambar 5, maka dapat kami simpulkan semakin sedikit *driver* akan menghasilkan rute yang semakin baik akan tetapi dikarenakan keterbatasan waktu pengiriman hal ini belum tentu semakin baik. Sedangkan ketika *driver* telah mencapai batas maksimum (dalam gambar ini 9 dan 10) maka perubahan nilai fitness optimal yang dihasilkan tidak terpaut jauh mengingat kinerja *driver* menjadi tidak maksimal dikarenakan jumlah pengiriman yang terlalu sedikit.

VII. KESIMPULAN

Setelah melalui berbagai ujicoba baik ujicoba kelayakan sistem yang dihasilkan di penelitian ini atau ujicoba terkait pencarian rute otomatis menggunakan algoritma emperor

penguin maka dapat disimpulkan bahwa untuk keseluruhan sistem yang dihasilkan pengguna merasa puas dan terbantu. Hal ini dapat dibuktikan berdasarkan hasil kuisioner kelayakan dimana diatas 75% pengguna menyatakan sistem yang dihasilkan sangat baik atau baik. Terkait fokus utama penelitian ini dapat disimpulkan bahwa pencarian list pengiriman pesanan yang disertai dengan rute pengiriman dan menerapkan konsep pembagian ongkos kirim berjalan dengan efektif dan menghasilkan rute yang optimal. Hal tersebut dibuktikan dengan kuisioner *driver* dan kuisioner pengguna dimana berdasarkan 77% *driver* pemerataan pengiriman pesanan dan rute pengirimannya telah optimal dan efisien sedangkan berdasarkan 97.4% pelanggan fitur pembagian ongkos kirim sangat disukai dan membantu meringankan pelanggan dalam pembayaran ongkos kirim.

DAFTAR PUSTAKA

[1] A. N. Sloss and S. Gustafson, "2019 Evolutionary Algorithms Review," Jun. 2019.

[2] Z. Dasgupta Dipankarand Michalewicz, "Evolutionary Algorithms — An Overview," in *Evolutionary Algorithms in Engineering Applications*, Z. Dasgupta Dipankarand Michalewicz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 3–28.

[3] D. Whitley, "An overview of evolutionary algorithms: practical issues and common pitfalls," *Inf. Softw. Technol.*, vol. 43, no. 14, pp. 817–831, 2001, doi: [https://doi.org/10.1016/S0950-5849\(01\)00188-4](https://doi.org/10.1016/S0950-5849(01)00188-4).

[4] T. Bäck and H.-P. Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–23, 1993, doi: 10.1162/evco.1993.1.1.1.

[5] C. Pickerling, H. Armanto, and E. R. Setyaningsih, "Symbiotic organisms search algorithm for scheduling laboratory sessions in University," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, 2017, pp. 1–6, doi: 10.1109/CAIPT.2017.8320698.

[6] Zulkarnaen, H. Budianto, and H. Armanto, "Algoritma Improved Symbiotic Organism Search (I-SOS) sebagai Pendukung Keputusan dalam Penentuan Dosen Pengampu Matakuliah," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 1, no. 4, pp. 279–286, Feb. 2020, doi: 10.35746/jtim.v1i4.64.

[7] M. S. Hossain, A. S. Tanim, S. S. Choudhury, S. M. A. I. Hayat, M. N. Kabir, and M. M. Islam, "An Efficient Solution to Travelling Salesman Problem using Genetic Algorithm with Modified Crossover Operator," *Emit. Int. J. Eng. Technol.*, vol. 7, no. 2, Dec. 2019, doi: 10.24003/emitter.v7i2.380.

[8] M. Yousefikhoshbakh, "Solving the Traveling Salesman Problem: A Modified Metaheuristic Algorithm," *Complexity*, vol. 2021, p. 6668345, 2021, doi: 10.1155/2021/6668345.

[9] C.-M. Pintea, P. C. Pop, and C. Chira, "The generalized traveling salesman problem solved with ant algorithms," *Complex Adapt. Syst. Model.*, vol. 5, no. 1, p. 8, 2017, doi: 10.1186/s40294-017-0048-9.

[10] H. Abdulkarim and I. F. Alshammari, "Comparison of Algorithms for Solving Traveling Salesman Problem," *Int. J. Eng. Adv. Technol.*, vol. ISSN, pp. 2249 – 8958, Nov. 2015.

[11] J. Li, Q. Sun, M. Zhou, X. Yu, and X. Dai, "Colored Traveling Salesman Problem and Solution," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 9575–9580, 2014, doi: <https://doi.org/10.3182/20140824-6-ZA-1003.01403>.

[12] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowledge-Based Syst.*, vol. 159, pp. 20–50, 2018, doi: <https://doi.org/10.1016/j.knosys.2018.06.001>.

[13] M. L. Napoli, *Beginning Flutter: A Hands On Guide to App Development*. Wiley, 2019.

[14] S. Alessandria and B. Kayfitz, *Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart*. Packt Publishing, 2021.

[15] E. Windmill, *Flutter in Action*. Manning, 2020.

[16] C. Zaccagnino, *Programming Flutter: Native, Cross-Platform Apps the Easy Way*. Pragmatic Bookshelf, 2020.