

# MULTI OBJECTIVE ACO (MO-ACC) ALGORITHM UNTUK OPTIMASI STRATEGI MITIGASI BENCANA

Dr. Verry Ronny Palilingan, M.Eng.

UNIMA PRESS

**BUKU REFERENSI**

***MULTI-OBJECTIVE ACO (MO-ACO) ALGORITHM***  
**UNTUK OPTIMASI STRATEGI MITIGASI BENCANA**

**Oleh:**

**DR. VERRY RONNY PALILINGAN, M.ENG**

**UNIMA PRESS**

**ISBN : 978-602-1376-57-7**

***MULTI-OBJECTIVE ACO (MO-ACO) ALGORITHM* UNTUK  
OPTIMASI STRATEGI MITIGASI BENCANA**

Oleh : Dr. Verry Ronny Palilingan, M.Eng

Editor : John Reymon Batmetan, S.Pd, MT

Diterbitkan oleh Unimja Press

UPT Penerbitan/Percetakan Kampus UNIMA di Tondano

Tondano, 95618

Hak Cipta dilindungi Undang Undang .Dilarang memperbanyak buku ini sebahagian atau seluruhnya ,dalam bentuk dan dengan v cara apapun juga,baik secara mekanis atau elektronik ,termasuk foto copi, rekaman dan lain lain tanpa izin tertulis dari penerbit

Edisi Pertama :

Perancang Cover : Nehemia Kusbin

Perwajahan : Sugiono

Dicetak oleh : UNIMA Press

Email : [ronnypalilingan@unima.ac.id](mailto:ronnypalilingan@unima.ac.id)

@Penerbit unima press Tondano

(x+107 hal ) 16 x 23 cm

**ISBN : 978-602-1376-57-7**

## INTISARI

Mitigasi merupakan salah satu bagian penting dalam manajemen bencana. Mitigasi dilihat sebagai upaya menentukan jalur terbaik dalam evakuasi bencana, penentuan jalur distribusi terbaik dan penentuan titik pengungsian. Mitigasi dalam bencana adalah harus dioptimasi dengan penentuan jalur terbaik dan terpendek. Pemilihan jalur terbaik ini ditujukan pada daerah yang padat dengan pemukiman penduduk tempat tinggal warga yang dengan dengan area letusan gunung berapi dan masuk pada area berbahaya. Mitigasi bencana memerlukan system informasi yang akurat dan cepat dalam memberikan respon. Kekhawatiran dinyatakan dalam mitigasi adalah pilihan jalan kurang akurat dan jalan kompleks yang akan dipilih sehingga mengakibatkan korban jiwa. Tesis ini mencoba untuk menyelidiki untuk memilih jalur terpendek di mitigasi letusan gunung berapi Lokon. Penelitian ini mencoba menerapkan formula Algoritma Multiple Objective Ant Colony Optimization dengan memasukkan indikator kecepatan, jarak, tikungan, kepadatan dan titik aman untuk menghitung jalur pendek untuk yang dipilih menggunakan algoritma MO-ACO. Model simulasi dan perhitungan yang dilakukan dengan membuat agregasi indikator yang ada dan terus menghitung dan menentukan optimasi pemilihan jalur mitigasi terbaik menggunakan algoritma MO-ACO. Hasil yang diperoleh untuk menghasilkan solusi dengan jalur pendek peta mitigasi yang terbaik, dan cepat dalam merespon dan penting untuk mengelola bencana. Peta ini menjadi petunjuk dalam manajemen bencana dan navigasi dalam proses mitigasi. peta mitigasi ini menggunakan sistem informasi mitigasi bencana dan terlibat dalam penanggulangan bencana gunung berapi Lokon di Sulawesi Utara.

**Keywords:** multiple-objective, ant colony optimization, mitigation natural disaster, lokon volcano, short path.

## ABSTRACT

Mitigation is one of the most important parts of disaster management. Mitigation is about how to selecting short path for evacuation, logistic relief disaster and shelter of refuge. Mitigation for natural Disaster must be optimizing with selection short path. Selection path of the mitigation is aimed at residential areas close to the volcano and into the danger zone. A mitigation disaster needed an information system with accurate and quickly intervening respond. Concerns expressed in the mitigation is a less accurate path selection and a complex path that will be chosen so as to result in casualties. This thesis tries to investigate to choose the shortest path in the mitigation of the eruption of the Lokon volcano. This study tried to apply the formula by entering a speed indicator, distance, bend, density and secure point to calculate the short path to being selected using MO-ACO algorithm. Model simulations and calculations are done by making the aggregation of existing indicators and continue to calculate and determine the short lpath mitigation using MO-ACO algorithm. Results obtained to produce a solution with the short path mitigation map that is a low budget, and quickly respond to manage disaster. This map became clues in disaster management and navigation in the mitigation process. This mitigation map using for mitigation information system and implicated in disaster management Lokon volcano on North Sulawesi.

**Keywords:** multiple-objective, ant colony optimization, mitigation natural disaster, lokon volcano, short path.

## KATA PENGANTAR

Bersyukur kepada Allah yang maha kuasa dan berdaulat atas segala sesuatu yang diciptakanNya yang melimpahkan berkat, kekuatan dan kemampuan sehingga penulis dapat menyelesaikan penelitian dengan baik, dan dapat menghasilkan tulisan yang dapat menjadi berkat bagi banyak orang yang membacanya.

Buku ini dimaksudkan untuk memenuhi kebutuhan literatur sekaligus sebagai bagian dari panggilan penulis untuk membantu daerah-daerah yang rawan bencana dan memerlukan optimasi dalam mitigasi bencana sehingga dapat mengurangi korban jiwa dan kerugian yang besar serta menghemat dana dalam proses penanggulangan bencana khususnya bencana letusan gunung berapi.

Penulis menyadari bahwa selesainya pembuatan buku ini tidak bisa terlepas dari bantuan yang diberikan oleh Johan R Batmetan,SPd,MT dan berbagai pihak, baik yang bersifat langsung maupun tidak langsung.

Demikian penulisan buku ini dibuat dengan sebaik – baiknya dan menyadari bahwa penulisan buku ini masih memiliki banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu saran dan kritik yang membangun demi penyempurnaan penulisan buku ini akan selalu penulis nantikan.

Akhir kata, penulis mengharapkan buku ini dapat berguna dan bermanfaat sehingga dapat memberikan inspirasi bagi pembacanya.

Manado, November 2018

Penulis

## DAFTAR ISI

INTISARI .....	iii
ABSTRAC .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	viii
DAFTAR TABEL .....	ix
BAB I PENDAHULUAN .....	1
1.1.Latar Belakang .....	1
1.2.Rumusan Masalah .....	3
1.3. Batasan Masalah .....	4
1.4.Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	4
1.6. Keaslian Peneliti8an .....	4
1.7.Sistimatika Penulisan .....	5
BAB II TINJAUAN PUSTAKA .....	6
2.1.Ant Coloni Optimization .....	6
2.2.Algoritme Multiple Objective A nt Colony Optimization (MO-ACO) .....	7
2.3.Mitigasi Bencana Alam .....	9
2.4.Gunung Lokon .....	10
BAB III LANDASAN TEORI .....	17
3.1.Ant Coloni Optimization (ACO) .....	17
3.2.Multiple Objective –Ant Colony Optimization (MO-ACO)	23
3.3 Mitigasi Bencana .....	50
BAB IV METODOLOGI PENELITIAN .....	52
4.1.Tahapan Penelitian .....	52
4.2.Diagram Alir Penelitian .....	55
4.3.Rancangan Skema Penelitian .....	57
4.4 Jadwal Penelitian .....	58
BAB V PEMBAHASAN .....	59
5.1.Peta Kota Tomohon .....	59
5.2.Peta Kebencanaan Gunung Lokon .....	62
5.3.Perhitungan .....	65
5.4.Hasil .....	69.

5.4.1.Segmen Optimasi .....	69
<b>BAB VI KESIMPULAN ,REKOMENDASI DAN DISKSUSI</b>	<b>87</b>
6.1.Kesimpulan .....	87
6.2.Rekomendas .....	87
6.3.Diskusi .....	88
REFERENCES .....	89
LAMPIRAN .....	96



## DAFTAR GAMBAR

Figure 1. Peta Kawasan Rawan Bencana Gunung Api Lokon	11
Figure 2 Ilustrasi Algoritma ACO .....	20
Figure 3. MO ACO Frame work .....	31
Figure 4. Komponen Algoritma MOACO .....	33
Figure 5 Taksonomi algoritma MOACO .....	34
Figure 6 Gambar algoritma MOACO .....	36
Figure 7 Default parameter setting dari algoritma ACO(MMAS)	43
Figure 8 Domain Komponen algoritma MOACO .....	44
Figure 9 Konfigurasi dengan lima tuning runs algoritma MOACO .....	44
Figure 10. Konfigurasi dengan komponen algoritma MOACO	45
Figure 11 Domain parameter setring pada algoritma ACO(MMAS) .....	46
Figure 12. Diagram alir Penelitian .....	56
Figure 13. Rancangan skema metode penelitian .....	57
Figure 14 Peta area kota Tomohon Utara (Sumber : wikimapia.org) .....	59
Figure 15 Peta area desa Tinoor I dan II ( Sumber : wikimapia.org) .....	60
Figure 16.Peta area desa Kinilow (Sumber:wikimapia.org).....	61
Figure 17 Peta area Kelurahan kakaskasen I,II,III (Sumber : wikimapia.org) .....	62
Figure 18.Peta Bencana Gunung Lokon (Sumber BNPB RI,2011	65
Figure 19.Pembagian segmen optimasi area bencana .....	70
Figure 20.Peta mitigasi bencana dengan algoritma MOACO .....	86

## DAFTAR TABEL

Table 1. Rekapitulasi Literatur Review .....	12
Table 2 Jadwal Penelitian .....	58
Table 3 Parameter tetapan setting MOACO algorithm .....	65
Table 4 Komponen pada framework MOACO algorithm Secara otomatis .....	66
Table 5. Contains Components yang digunakan .....	70
Table 6 Parameter $\beta$ value titik Shelter .....	71
Table 7 Parameter $\rho$ value titik Shelter .....	71
Table 8 Problem Parameter titik Shelter .....	72
Table 9 Hasil perhitungan Shelter pada Segmen 1 .....	72
Table 10 Hasil perhitungan shelter pada segmen II .....	73
Table 11 Hasil perhitungan Shelter pada Segmen III .....	73
Table 12 Hasil perhitungan Shelter pada Segmen IV .....	73
Table 13 Rekapitulasi Hasil Algoritma MOACO .....	74
Table 14 Intensitas pheromone tiap Segmen .....	75
Table 15 Data Lokasi Bencana .....	75
Table 16. Contains Components yang digunakan pada jalur Evakuasi .....	76
Table 17. Parameter $\beta$ value pada jalur evakuasi .....	77
Table 18 Parameter $\rho$ value pada jalur evakuasi .....	77
Table 19 Problem Parameter pada jalur evakuasi .....	78
Table 20 Hasil Choosing short path segmen I pada jalur Evakuasi .....	78
Table 21 Hasil Choosing short path segmen II pada jalur Evakuasi .....	78
Table 22 Hasil Choosing Short path Segmen III pada jalur Evakuasi .....	79
Table 23 Hasil Choosing Short path segmen IV pada jalur Evakuasi .....	79
Table 24 Rakapitulasi Hasil MOACO pada jalur evakuasi ..	79
Table 25 Intensitas pheromone tiap segmen .....	80
Table 26 Contains components yang digunakan pada jalur Evakuasi .....	81
Table 27 Parameter $\beta$ value pada jalur logistik.....	81

Table 28.Parameter value pada jalur logistik.....	82
Table 29 Problem parameter pada jalur logistik .....	83
Table 30 Hasil Choosing short path segmen I pada jalur Logistik.....	83
Table 31 Hasil Choosing short path segmen II pada jalur Logistik.....	83
Table 32 Hasil Choosing short path segmen III pada jalur Logistik.....	84
Table 33 Hasil Choosing short path segmen IV pada jalur Logistik .....	84
Table 34 Hasil Rekapitulasi MOACO pada jalur logistik....	84
Table 35 Intensitas pheromone tiap segmen pada jalur Logistik .....	86

## PENDAHULUAN

### 1.1. Latar belakang

Smart Environment merupakan pengelolaan dan perwujudan lingkungan yang berkelanjutan (sustainability), yang dapat dikelola dengan bantuan teknologi informasi. Kemudahan mengakses informasi dari lingkungan sekitar akan membantu mewujudkan lingkungan yang baik dan sehat. Dengan teknologi informasi dapat diwujudkan lingkungan yang cerdas.

Pengelolaan smart environment ini dimaksudkan bagaimana pemerintah dan masyarakat dapat mengelola dan mengakses informasi yang berkaitan dengan bencana alam, mitigasi bencana dan penanganan bencana seperti pemilihan jalur evakuasi dan penanganan pasca bencana. Hal ini dimaksudkan untuk mengurangi jatuhnya korban jiwa dan juga mengurangi penyebaran informasi yang tidak akurat seperti hoax, dan informasi yang simpang siur sehingga menyebabkan kebingungan dan ketidakjelasan. Masyarakat seharusnya informasi yang akurat dan didapatkan dari sumber informasi yang benar dari pihak yang berhak memberi informasi sehingga tidak terjadi *missing information*. Dengan demikian penting untuk dibuat strategy pengelolaan berbasis teknologi informasi yang dapat memanfaatkan model-model algoritma komputasi seperti algoritma *multiple objective ant colony* sebagai solusi dalam pemilihan jalur evakuasi bencana.

Mitigasi bencana dibagi menjadi tiga bagian yaitu tindakan atau kegiatan *detection* (deteksi bencana), *respond* (penanganan saat terjadi bencana) dan *recovery* (penanganan pasca bencana). Ketiga tindakan ini memerlukan informasi yang akurat dan cepat sehingga korban tidak berjatuh. Keberhasilan system mitigasi sangat tergantung pada informasi yang akurat dan cepat baik saat pengolahan informasi maupun saat informasi tersebut sampai kepada pengguna informasi seperti masyarakat di daerah bencana maupun petugas SAR dan lainnya.

Pada bagian yang lain, mitigasi sangat memerlukan efisiensi yang tinggi agar beban anggaran dan sumber daya yang dimiliki

dapat menjadi optimal. Kegiatan yang dapat dibuat menjadi lebih optimal seperti penentuan titik selter penggungi secara cepat dan tepat, mudah dijangkau dengan waktu yang tidak terlalu lama. Hal lain yang dapat dioptimalkan adalah pendistribusian logistic bencana karena hal ini membutuhkan pemilihan jalur tercepat dalam mencapai titik selter penggungi.

Mitigasi sangat penting bagi daerah yang memiliki kerentanan yang tinggi terhadap bencana alam. Bencana alam berupa tsunami, gempa bumi, letusan gunung berapi, dan lainnya memerlukan mitigasi yang akurat dan cepat sehingga meminimalisir kerugian dan jatuhnya korban akibat bencana tersebut. Letak geografis suatu daerah sangat menentukan mitigasi yang akan diterapkan karena akan mempengaruhi cara penanganannya. Letak geografis suatu daerah juga menentukan kerentanan suatu daerah menghadapi bencana alam. Misalnya daerah yang terletak di daerah patahan lembengan bumi sangat rentan terhadap gempa bumi dan tsunami atau daerah yang berada di *ring of fire* bumi akan sangat rentan terhadap bencana letusan gunung berapi karena terdapat banyak gunung berapi di daerah tersebut.

Letak Indonesia secara geografis berada pada bentangan *ring of fire* yang sewaktu-waktu dapat menimbulkan bencana alam seperti gempa bumi, tsunami, gunung menetus, dll. Hal ini memerlukan penanganan dan mitigasi yang baik dan efisien agar tidak terjadi kerugian yang besar bahkan jatuhnya korban yang banyak. Untuk itu diperlukan kajian dan cara maupun strategi yang tepat dalam menangani bencana alam tersebut.

*Ring of fire* yang dimaksud adalah Sabuk Gempa Pasifik yang merupakan daerah berbentuk seperti tapal kuda yang mengelilingi Samudera Pasifik mencakup panjang 40.000 km. Sekitar 90% gempa bumi terjadinya di daerah ini dan 81% gempa bumi terbesar terjadi di sepanjang Cincin Api tersebut [2]. Indonesia masuk ke dalam Sabuk Gempa Pasifik ini sehingga sering terjadi gempa bumi dan letusan gunung berapi. Indonesia merupakan salah satu negara yang terletak pada *ring of fire* dengan tingkat ancaman bencana alam yang tinggi [3]. Tercatat dalam periode tahun 2010-2015 terdapat beberapa

bencana alam yang terkait dengan ring of fire ini seperti gempa bumi, tsunami, letusan gunung api, dan air laut pasang. Di Indonesia pada periode 2010-2015 bencana gempa bumi dan tsunami telah mengakibatkan 512 orang meninggal dunia, 56 orang hilang, 4,264 orang terluka dan 74,498 orang menggungsi, sedangkan letusan gunung berapi mengakibatkan 409 orang meninggal dunia, hilang 4 orang, terluka 2,223 orang dan 183,345 orang menggungsi dan bencana air laut pasang mengakibatkan 17 orang meninggal dunia, hilang 47 orang, terluka 27 orang dan 4,338 orang menggungsi [4]. Di propinsi Sulawesi utara juga merupakan daerah yang termasuk dalam ring of fire sehingga menjadi daerah yang rawan terhadap bencana alam. Bencana alam yang terjadi pada periode 2010-2015 memiliki data seperti bencana gempa bumi dan tsunami telah terjadi 3 kali tanpa ada korban jiwa maupun menggungsi, sedangkan letusan gunung berapi mengakibatkan 4 orang hilang, 5 orang terluka dan 1,625 orang menggungsi dan bencana air laut pasang mengakibatkan 3 orang terluka [4]. Gunung lokon merupakan gunung berapi yang aktif dengan aktivitas kegempaan yang cukup tinggi. Terhitung dalam tahun 2015 telah 6 (enam) kali erupsi dan mengeluarkan abu vulkanik yang membahayakan. Letak gunung lokon berada di kota Tomohon yang menjadi ibu kota daerah kota Tomohon Sulawesi utara dimana terdapat banyak perkampungan dikeliling kaki gunung tersebut.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah di uraikan, dapatlah dirumuskan masalahnya adalah sebagai berikut :

- a. Implementasi Algoritma *Multiple Objective Ant Colony Optimization* dalam penentuan letak selter pengungsi terbaik
- b. Implementasi Algoritma *Multiple Objective Ant Colony Optimization* dalam penentuan jalur distribusi logistik bencana terbaik
- c. Implementasi Algoritma *Multiple Objective Ant Colony Optimization* dalam penentuan jalur evakuasi bencana terbaik

### **1.3 Batasan Masalah**

Dari rumusan masalah yang dikemukakan, maka batasan masalah dapat dibatasi sebagai berikut :

Penelitian ini dilakukan di Gunung Lokon Sulawesi Utara bagian utara dimana kota Tomohon berada terutama di daerah Desa Tinoor, Desa Kinilow, Kelurahan Kakaskasen I, dan Kelurahan Kakaskasen II

### **1.4 Tujuan Penelitian**

Tujuan dilakukannya penelitian ini adalah sebagai berikut :

- a. Mengoptimasi Strategi dan Menganalisis penentuan selter pengungsi terbaik menggunakan metode algoritma *Multiple Objective Ant Colony Optimization*
- b. Mengoptimasi Strategi dan Menganalisis penentuan jalur distribusi logistic terbaik dalam mencapai selter pengungsi menggunakan metode algoritma *Multiple Objective Ant Colony Optimization*
- c. Mengoptimasi Strategi dan Menganalisis penentuan jalur evakuasi terbaik dalam evakuasi korban bencana menggunakan metode algoritma *Multiple Objective Ant Colony Optimization*

### **1.5 Manfaat Penelitian**

- a. Manfaat Teoritis :  
Hasil penelitian ini diharapkan memberi kontribusi secara teoritis pada penelitian yang berkaitan dengan implementasi algoritma *Multiple Objective Ant Colony Optimization* dalam penentuan atau pemilihan jalur tercepat
- b. Manfaat praktis :  
Hasil penelitian ini diharapkan memberi kontribusi kepada :
  - Pemerintah Kota Tomohon dalam mengatur tatakelola bencana alam yang dihadapi
  - Masyarakat kota Tomohon dalam menghadapi bencana alam sehingga menjadi lebih siap dan *aware* dengan lingkungan kebencanaan yang dihadapi.

### **1.6 Keaslian Penulisan**

Hasil penelitian penulis merupakan murni karya penulis. Belum pernah dilakukan penelitian sebelumnya terkait topik dan objek yang sementara penulis teliti. Tulisan hasil karya orang lain pada karya ini penulis jadikan referensi untuk memperdalam pemahaman penulis dalam melakukan penelitian.

### **1.7 Sistematika Penulisan**

Secara garis besar sistematika penulisan pada penelitian ini terdiri dari 6 bab dengan rincian sebagai berikut.

#### **Bab I. Pendahuluan**

Bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, manfaat penelitian, tujuan penelitian, keaslian penelitian dan sistematika penulisan.

#### **Bab II. Tinjauan Pustaka**

Bab ini menjelaskan tentang tinjauan pustaka dari penelitian-penelitian yang telah dilakukan sebelumnya yang berkaitan dengan topik ini .

#### **Bab III. Landasan Teori**

Bab ini menjelaskan tentang teori yang mendukung topik penelitian ini. Teori-teori ini diambil dari buku-buku yang diterbitkan dan digunakan sebagai refensi dalam penulitas penelitian ini.

#### **Bab IV. Metode Penelitian**

Bab ini menjelaskan tentang bahan penelitian, alat dan langkah-langkah penelitian.

#### **Bab V. Pembahasan Hasil Penelitian**

Bab ini menjelaskan tentang keseluruhan hasil penelitian meliputi analisis dan rekomendasi terhadap penentuan shelter pengungsi korban bencana, pemilihan jalur distribusi dan pemilihan jalur tercepat dalam penanganan bencana gunung berapi Lokon di Kota Tomohon Sulawesi Utara.

#### **Bab VI. Penutup**

Bab ini berisi kesimpulan atas seluruh penelitian yang sudah dilakukan dan saran bagi pihak yang berminat dalam mengembangkan hasil analisis ini menjadi aplikasi yang dapat dijalankan baik pada laptop/PC maupun smartphone.



## BAB II TINJAUAN PUSTAKA

### 2.1. Ant Colony Optimization

Ant Colony merupakan sebuah algoritma dalam komputasi yang merupakan bagian dari cabang ilmu Swarm Intelligence. Dharmendra Sutariya menjelaskan swarm intelligence sebagai berikut “Swarm intelligence (SI) is a kind of artificial intelligence that aims to simulate the swarms behavior such as ant colonies, honey bees, bird flocks, particle swarm optimization and artificial immune system etc”[1]. Sehingga dapat dilihat bahwa penelitian-penelitian yang dilakukan menggunakan algoritma Ant Colony memiliki hubungan pendekatan dengan cabang ilmu swarm intelligence dan artificial intelligence.

Seperti penelitian yang dilakukan oleh Dorigo [5] menjelaskan bahwa algoritma Ant Colony merupakan sebuah algoritma metaheuristic. Metaheuristic sendiri merupakan pengembangan dari konsep heuristic untuk menghasilkan solusi yang optimal dengan mengubah perhitungan dengan menggunakan variable independen [6] [7]. Metaheuristic memungkinkan pendekatan yang lebih optimal dalam waktu yang cepat sehingga dapat menghasilkan solusi strategy yang optimal [5] [8].

Algoritma ant colony optimization (ACO) merupakan sebuah turunan dari algoritma ant colony. Ant colony sendiri memiliki beberapa varian seperti Ant System Rank, Ant Max-Min, Quantum ant colony, ant colony optimization, ACS, EAS, etc [9]. Dari semua varian ant colony yang ada, Dorigo menyimpulkan bahwa algoritma Ant Colony Optimization (ACO) merupakan yang terbaik dari varian yang ada [5]. Pendapat tersebut dikuatkan oleh penelitian yang dilakukan oleh Leksono (2009) bahwa ACO memiliki hasil perhitungan yang relative mendekati optimum [10]. Dari hasil penelitian yang telah dilakukan oleh berbagai peneliti, maka tulisan ini mencoba mengangkat Algoritma ACO sebagai acuan untuk dilakukan tindakan lanjutan yakni dengan mengimplementasikannya ke dalam permasalahan mitigasi bencana alam khususnya bencana letusan gunung berapi.

Dari yang dilakukan oleh Batmetan [11] yang mengangkat masalah pemilihan jalur dengan cepat dalam evakuasi bencana letusan gunung lokon, algoritma Ant Colony Optimization (ACO) dapat digunakan dalam pemilihan lajur terbaik dalam evakuasi. Hasil penelitian ini menunjukkan bahwa cara pemilihan jalur evakuasi terbaik pada bencana letusan Gunung Lokon dapat dihitung dengan mudah dan cepat. Dengan algoritma ACO ini, jalur evakuasi dapat dipilih dengan cepat dan cermat sehingga dapat mengurangi jatuhnya korban pada bencana letusan.

## **2.2. Algoritma Multiple Objective Ant Colony Optimization (MO-ACO)**

Algoritma ant colony optimization terinspirasi dari perilaku semut dalam mencari makan. Seperti yang dijelaskan oleh Mora bahwa algoritma ACO merupakan pencarian metaheuristic yang didasarkan pada perilaku semut saat mencari makanan dan membangun jalur terpendek antara jaringannya dan memilih yang terpendek[3]. Jika diperhatikan algoritma ACO digunakan untuk memberi solusi terhadap satu objek permasalahan seperti yang dijelaskan oleh Mora dalam penelitiannya[3]. Lebih lanjut Mora menjelaskan bahwa jika objek yang dihadapi banyak, maka algoritma yang digunakan adalah Multiple Objective Ant Colony Optimization (MOACOs)[3].

Beberapa algoritma MOACO didasarkan pada pendekatan multi colony, dengan subnets yang independent dari kerja semut saat mencari solusi terbaik, sesuai jalur distribusi yang termudah atau dipararelkan tanpa halangan atau larangan, semut-semut tersebut dapat menyelesaikan pencarian mandiri, berbagi saat salah satu telah selesai melakukan pencarian, disinilah set pareto diperoleh[3].

Penelitian yang dilakukan Mora menunjukkan bahwa MOACOs algoritma ini dimulai dengan multi algoritma dua object dan dua pheromone. Dengan demikian terdapat dua fungsi persamaan yang akan dihitung. Pada penelitian Mora, menggunakan persamaan BIAN (Bi-Criterion Ant), MOACS (Multi-Objective Ant Colony System) dan CHAC (Compañía de Hormigas

ACorazadas). Pada penelitian Mora, kasus yang di ujicoba adalah pada kinerja prosesor intel saat menentukan jalur terpendek dalam pengiriman data. Kriteria yang digunakan adalah jarak dan kecepatan[3].

Sedangkan Pada penelitian ini, akan dicoba menggunakan tiga persamaan dengan lima kriteria yang berbeda yakni kecepatan, jarak, tikungan, titik aman, dan kepadatan. Contoh kasus yang akan digunakan adalah mitigasi bencana dalam menentukan titik shelter, jalur terbaik distribusi bencana, dan jalur terbaik evakuasi bencana.

Sejalan dengan penelitian Mora, penelitian yang dilakukan oleh lopez menyebutkan bahwa MOACOs dapat digunakan untuk menyelesaikan masalah TSP. faktanya, algoritma MOACOs banyak dipakai untuk menyelesaikan masalah optimasi dengan bi-objectif TSP (Traveling Salesman Problem). Oleh karena itu, untuk mengawali analisis, Lopez menggunakan MAX-MIN Ant System (MMAS) untuk menentukan parameternya. Kemudian menggunakan pendekatan algoritma ACO untuk menentukan pencarian lokalnya. Langkah berikutnya lopez memasukan alternatif desain baru untuk di analisis menggunakan pendekatan Algoritma MOACO[4].

Penelitian lain yang dilakukan oleh Kötzing, melakukan pendekatan untuk menyelesaikan masalah TSP. Pendekatan yang dilakukan adalah dengan menggabungkan dua algoritma ant colony yaitu MAX-MIN Ant System (MMAS) dan ACO[5]. Pendekatan ini hampir sama seperti MOACOs, tetapi digunakan dalam menyelesaikan masalah pada TSP. Pendekatan yang dilakukan sudah cukup baik dalam mencari solusi TSP.

Pada penelitian ini, akan dicoba menggunakan algoritma MOACO untuk mencari solusi pada masalah mitigasi. Mitigasi ini meliputi pencarian jalur terbaik untuk distribusi logistic, jalur terbaik evakuasi bencana dan penentuan titik shelter pengungsi. Algoritma ini menggunakan pendekatan MOACO dengan tiga persamaan yaitu Biant (Bi-Criterion Ant), MOACS (Multi-Objective Ant Colony System) dan CHAC (Compañía de Hormigas ACorazadas). Dalam perhitungan, algoritma ini menggunakan lima indikator/variable

dalam perhitungan. Hasil terbaik dalam perhitungannya akan di implementasikan untuk menentukan proses mitigasi bencana.

### **2.3. Mitigasi Bencana Alam**

Mitigasi bencana memerlukan sebuah pendekatan yang baik saat menghadapi bencana. Pendekatan ini meliputi *detektion, respond and revocery*. Penelitian yang dilakukan oleh Barzinpour mengemukakan bahwa dalam mitigasi bencana alam salah satu bagian yang penting adalah melokalisasi area bencana. Hal ini untuk mempermudah supply bantuan logistic, TIM penyelamat maupun peralatan yang digunakan dalam operasi penyelamatan[2]. Penentuan shelter pengungsi dalam operasi bencana alam juga menjadi penting. Factor penting dalam penentuan adalah dapat dijangkau oleh pengungsi dan distribusi bantuan serta aman dari bahaya bencana alam. Barzinpour dalam penelitiannya menggunakan model Multiple Objective dalam memetakan area cakupan bencana dan menghitung besaran kerugian yang ditimbulkan serta memetakan jalur distribusi logistis ke area bencana dengan cepat dan murah sehingga menurunkan biaya[2].

Penelitian lain yang dilakukan oleh Batmetan (2016), mengemukakan bahwa mitigasi bencana dapat dilakukan dengan menerapkan algoritma komputasi seperti algoritma ACO [11]. Penelitian tersebut menemukan bahwa penggunaan algoritma ACO dapat memilih dan memetakan dengan cepat jalur evakuasi bencana saat bencana letusan gunung berapi terjadi. Penelitian tersebut mengambil lokasi pada gunung Lokon di kota Tomohon Sulawesi Utara. Penelitian-penelitian ini membuktikan bahwa pendekatan algoritma komputasi dapat diterapkan dalam mengatasi masalah pada mitigasi bencana alam.

Pada penelitian ini, mitigasi dilakukan dengan mengimplementasi algoritma MOACO untuk menentukan letak shelter pengungsi, menentukan jalur distribusi terbaik dan juga menentukan jalur evakuasi terbaik yang harus dilalui. Hal ini diperlukan agar tidak terjadi korban jiwa yang banyak, memberi jalan kepada TIM SAR mencapai lokasi bencana dengan cepat dan

pengiriman logistic dengan cepat sehingga biaya menjadi lebih murah.

#### **2.4. Gunung Lokon**

Gunung lokon merupakan gunung api aktif yang berada di propinsi Sulawesi Utara, tepatnya terletak 45 km arah utara dari Kota Manado. Secara administratif, gunung lokon masuk dalam wilayah administrasi Pemerintah Kota Tomohon. Gunung Lokon termasuk Gunung Api yang aktif mengeluarkan erupsi [12]. Letaknya secara geografis berada pada Lokon  $01^{\circ} 21,5'$  LU dan  $124^{\circ} 47,5'$  BT dan Empung  $01^{\circ} 22'$  LU dan  $124^{\circ} 47,5'$  BT. Gunung lokon memiliki 3 puncak tertinggi yakni puncak Lokon 1579,5 m dml, puncak Empung 1340 m dml dan puncak Tompaluan 1140 m dml [12].

Dari pengamatan yang dilakukan oleh Badan Geologi melalui Pos Pemantau Gunung Berapi, tercatat dalam periode 2014-2015 telah 6 kali meletus mengeluarkan abu vulkanik [12]. Akibat letusan ini, 634 orang di ungsikan ke tempat yang aman agar selamat dari ancaman erupsi gunung lokon ini [4].

Gunung lokon sendiri terdapat kota Tomohon yang berada tepat dibawah kaki gunung tersebut. Selain itu terdapat 12 perkampungan masyarakat di sekeliling kaki gunung lokon. Hal ini selalu menimbulkan bahaya jika terjadi erupsi pada gunung lokon tersebut. 12 perkampungan ini masuk pada area berbahaya jika gunung lokon mengeluarkan abu vulkanik yang membahayakan.

Peta kerawanan bencana letusan gunung Lokon yang dikeluarkan oleh Badan Nasional Penanggulangan Bencana (BNPB) [13] pada tahun 2011 membagi daerah Kawasan Rawan Bencana II dengan daerah yang letaknya terdekat dengan sumber bahaya, sehingga kemungkinan akan terlanda oleh bahaya langsung, berupa luncuran awan panas, lontaran batu (pijar), hujan abu lebat dan lahar [13]. Tanpa memperhitungkan arah tiupan angin pada saat terjadi erupsi, daerah bahaya ini diperkirakan meliputi wilayah dalam radius lk. 3,5 km berpusatkan kawah aktif di puncak G. Lokon. Kawasan Rawan Bencana II ini dibedakan menjadi dua, yaitu :

- a. Kawasan rawan terhadap aliran masa berupa awan panas dan aliran lahar/banjir
- b. Kawasan rawan terhadap material lontaran dan jatuhnya seperti lontaran batu (pijar dan hujan abu lebat).

Kawasan berikutnya adalah Kawasan Rawan Bencana I yang merupakan kawasan yang berpotensi terlanda lahar/banjir, meliputi lembah atau daerah aliran sepanjang sungai-sungai yang berhulu di daerah puncak [13]. Selama erupsi membesar, kawasan ini berpotensi tertimpa material jatuhnya berupa hujan abu dan lontaran batu (pijar). Kawasan ini dibedakan menjadi dua, yaitu :

- a. Kawasan rawan terhadap lahar/banjir. Kawasan ini terletak di sepanjang sungai/ di dekat lembah sungai atau bagian hilir sungai yang berhulu di daerah sekitar kawah.
- b. Kawasan rawan terhadap hujan abu tanpa memperhatikan arah tiupan angin dan kemungkinan terkena lontaran batu (pijar).

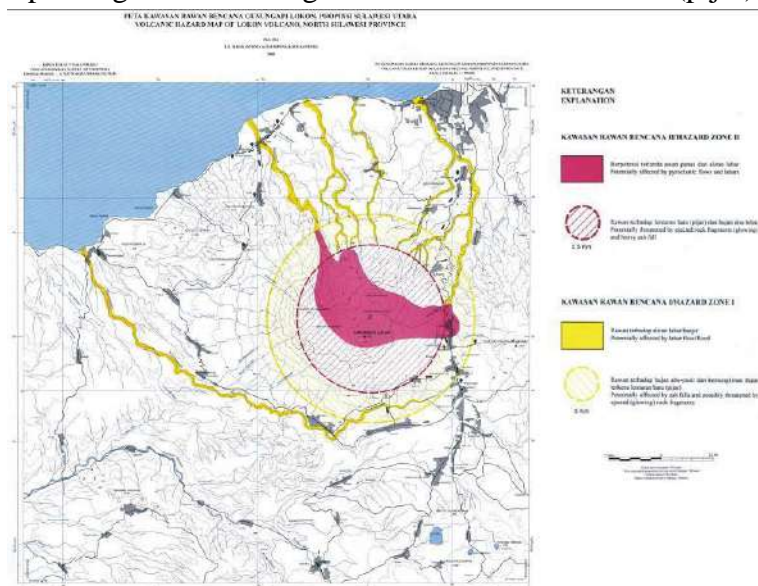


Figure 1 Peta Kawasan Rawan Bencana Gunung api Lokon  
(Sumber : Badan Nasional Penanggulangan Bencana RI, 2015)

Untuk itu diperlukan mitigasi bencana letusan gunung dan juga penanganan saat terjadi bencana seperti tindakan evakuasi dari zona bahaya. Tindakan ini memerlukan perlakuan khusus seperti

pemilihan jalur evakuasi yang cepat dan aman agar tidak terjadi korban jiwa.

*Table 1 Rekapitulasi Literature Review*

Nama, Tahun	Judul	Masalah, tujuan, metode	Hasil Penelitian
Mora., et al., (2013)	Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal	Penelitian ini menggunakan algoritma MOACO dengan 2 variable independen dalam struktur MO dengan menggunakan set pareto. MOACOs algoritma ini dimulai dengan multi algoritma dua object dan dua pheromone. Mora, menggunakan persamaan BIAN (Bi-Criterion Ant), MOACOs (Multi-Objective Ant Colony System) dan CHAC (Compañía de Hormigas Acorazadas). kasus yang di ujicoba pada kinerja prosesor intel saat menentukan jalur terpendek dalam pengiriman data. Kriteria yang digunakan adalah jarak dan kecepatan	Hasil penelitian ini menunjukkan optimasi yang baik dari set solusi yang dihasilkan. Hasilnya adalah terdapat peningkatan optimasi pada kinerja prosesor yang menggunakannya.
Lopez., et al., (2012)	An experimental analysis of design choices of multi-	Penelitian ini menggunakan algoritma MOACOs untuk menyelesaikan	Hasil penelitian ini menunjukkan bahwa algoritma

	objective ant colony optimization algorithms	<p>masalah TSP. faktanya, algoritma MOACOs banyak dipakai untuk menyelesaikan masalah optimasi dengan bi-objectif TSP (Traveling Salesman Problem). mengawali analisis, menggunakan MAX-MIN Ant System (MMAS) untuk menentukan parameternya. Kemudian menggunakan pendekatan algoritma ACO untuk menentukan pencarian lokalnya. Langkah berikutnya dimasukkan alternatif desain baru untuk di analisis menggunakan pendekatan Algoritma MOACO</p>	<p>MOACOs menampilkan solusi yang optimal dalam menyelesaikan masalah TSP. algoritma ini memiliki kualitas yang tinggi dalam menyelesaikan masalah optimasi dengan lebih dari 1 variable independen.</p>
Daniel Angus, et al., (2009)	Multiple objective ant colony optimisation	<p>Penelitian ini mencoba mengevaluasi kinerja algoritma MOACOs baik yang menggunakan single variable independent maupun yang memiliki variable lebih dari 1. Penelitian ini juga mencoba melakukan klasifikasi terhadap</p>	<p>Penelitian ini menghasilkan klasifikasi dalam algoritma MOACO beserta karakteristiknya. Hasil lainnya juga adalah evaluasi terbaik baik yang memiliki</p>



		algoritma MOACO.	single pheromone maupun multiple, terdapat pada algoritma MOACO.
Kötzing., et al., (2012)	Theoretical analysis of two ACO approaches for the traveling salesman problem	Penelitian ini melakukan pendekatan untuk menyelesaikan masalah TSP. Pendekatan yang dilakukan adalah dengan menggabungkan dua algoritma ant colony yaitu MAX-MIN Ant System (MMAS) dan ACO. algoritma ini hampir sama seperti MOACOs, tetapi digunakan dalam menyelesaikan masalah TSP.	Pendekatan yang dilakukan sudah cukup baik dalam mencari solusi TSP. dengan update pheromone dengan tinggi, didapatkan sebuah solusi yang baik bagi penyelesaian masalah TSP. pendekatan ini terbukti efektif dalam penyelesaian masalah TSP.
Barzinpour., et al., (2014)	A multi-objective relief chain location distribution model for urban disaster management	Penelitian mengemukakan bahwa dalam mitigasi bencana alam salah satu bagian yang penting adalah melokalisasi area bencana. Penelitian ini menggunakan model Multiple Objective dalam memetakan area cakupan bencana dan menghitung besaran	Hasil penelitian ini menunjukkan bahwa model Multiple Objective dapat dipakai dalam memetakan area cakupan bencana dan menghitung besaran kerugian yang ditimbulkan

		kerugian yang ditimbulkan serta memetakan jalur distribusi logistik ke area bencana dengan cepat dan murah sehingga menurunkan biaya	serta memetakan jalur distribusi logistik ke area bencana dengan cepat dan murah sehingga menurunkan biaya. Penelitian ini merekomendasikan agar metode ini dapat digunakan dalam mitigasi bencana.
Batmetan, Johan Reimon (2016)	Algoritma <i>Ant Colony Optimization</i> (ACO) untuk Pemilihan Jalur Tercepat Evakuasi Bencana Gunung Lokon Sulawesi Utara	Penelitian ini mengangkat masalah pemilihan jalur dengan cepat dalam evakuasi bencana letusan gunung lokon. Penelitian ini menggunakan algoritma <i>Ant Colony Optimization</i> (ACO) dalam pemilihan jalur terbaik dalam evakuasi.	Hasil penelitian ini menghasilkan sebuah cara pemilihan jalur evakuasi terbaik pada bencana letusan Gunung Lokon. Dengan algoritma ACO ini, jalur evakuasi dapat dipilih dengan cepat dan cermat sehingga dapat mengurangi jatuhnya

			korban pada bencana letusan.
--	--	--	------------------------------------

## BAB III LANDASAN TEORI

### 3.1. Ant Colony Optimization (ACO)

Algoritma Ant Colony Optimization adalah sebuah algoritma metaheuristik yang digunakan untuk pencarian secara mendalam dengan kualitas solusi yang lebih baik dibandingkan dengan metode pencarian konvensional biasa. Ant Colony biasanya digunakan untuk mencari jalur terpendek [11]. Menurut Lopez [8] ACO adalah sebuah pendekatan *NP hard* yang mengkombinasikan masalah optimasi. Lopez mengatakan bahwa pada dasarnya ACO dibangun dari algoritma *Ant Colony System*(ACS) dan *MAX-MIN Ant System*(MMAS). ACO merupakan algoritma pencarian metaheuristik yang terinspirasi dari kelompok semut dalam mencari makanan [11] [7] [8].

Lebih lanjut Sutariya dalam penelitiannya menjelaskan bahwa “Ant Colony Optimization (ACO) is a class of optimization algorithms, inspired by an organized collaborative behavior of ants. Ants are creatures of nature with limited intelligence, which are wandering around their nests to forage for food”[1]. Jadi Ant Colony Optimization adalah sebuah rumpun ilmu yang tergabung dalam swam intelligence dan merupakan bagian dari artificial intelligence.

Ant colony optimization (ACO) merupakan sebuah model yang dikembangkan dengan melihat semut sebagai objek utama pembentuk algoritmanya. Lebih lanjut Sutariya menjelaskan bahwa “The ant colony optimization algorithmic approach models the concept of food foraging, nest building, division of labour, cooperative support, self assembly and cemetery organization of real ants for the meta-heuristic approaches. ACO has been formalized in to a meta-heuristic computational approach” [14]. Untuk mendapatkan makanan, semut menggunakan cara tertentu untuk sampai pada tujuannya seperti dijelaskan sebagai berikut :

”While finding root from nest (source) to food (destination), ants communicate with other ants by depositing traces of pheromone (chemical substance) as they walk along their path. This indirect form of communication is called as stigmergy. As more ants travel

over a particular path, the concentration of pheromone increases along that path. Pheromone along a path gradually evaporates, decreasing their concentration on that path. Among the multiple path between nest and food ants select the single optimal path on the basis of maximum pheromone concentration along the path and some heuristic functions". [14]

ACO merupakan pengembangan dari Ant Colony. Secara informal, ACO bekerja sebagai berikut: pertama kali, sejumlah  $m$  semut ditempatkan pada sejumlah  $n$  titik berdasarkan beberapa aturan inisialisasi (misalnya, secara acak). Setiap semut membuat sebuah tour (yaitu, sebuah solusi jalur evakuasi yang mungkin) dengan menerapkan sebuah aturan transisi status secara berulang kali. Selagi membangun tournya, setiap semut juga memodifikasi jumlah *pheromone* pada edge-edge yang dikunjunginya dengan menerapkan aturan pembaruan *pheromone* local yang telah disebutkan tadi.

Setelah semua semut mengakhiri tour mereka, jumlah *pheromone* yang ada pada edge-edge dimodifikasi kembali (dengan menerapkan aturan pembaruan *pheromone* global). Dalam membuat tour, semut 'dipandu' oleh informasi *heuristic* (mereka lebih memilih edge-edge yang pendek) dan oleh informasi *pheromone*. Sebuah edge dengan jumlah *pheromone* yang tinggi merupakan pilihan yang sangat diinginkan. Kedua aturan pembaruan *pheromone* itu dirancang agar semut cenderung untuk memberi lebih banyak *pheromone* pada edge-edge yang harus mereka lewati. [9]. Algoritma ACO memiliki simulasi yang baik dalam memecahkan masalah optimasi [15].

Chaimongkon Chokpanyasuwan menjelaskan karakteristik ant colony sebagai berikut

"The characteristics of an artificial ant colony include positive feedback, distributed computation and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence and the greedy heuristic helps to find acceptable solutions in the early stages of the search process" [16].

Wang juga menjelaskan algoritma ACO sebagai “Ant colony optimization (ACO) goes through the necessary nodes on the graph to achieve the optimal solution with the objective of minimizing total production costs (TPC)” [6]. Mohammed dalam penelitiannya menjelaskan bahwa ACO dapat memberi solusi yang baik, penjelasannya sebagai berikut

“It provides a good diversification for the search in the solution space. Indeed, on the one hand it imposes lower and upper bounds on the amounts of pheromones to be deposited on the edges of the problem’s graph. It thus mitigates the transition probabilities for the most taken edges. On the other hand, it allows, under the frequency of one iteration over two, to support the pheromones concentration on the edges of the best iteration’s solution although it is not the best overall solution” [17].

Peneliti yang lain yakni Liqiang Liu memberi penjelasan yang lebih komprehensif yakni

“We can give such a model through the above process of analysis and expansion: assuming the food source is very where throughout in the continuous space, the quality of food source is different. At the initial moment, ants of ant colony distribute uniformly in the continuous space and release pheromones according to food sources of their position. The higher the quality of the food source, the more the pheromone ants released. The pheromone is distributed throughout the continuous space in a certain dispersed model, and ants perceive spatial concentration of pheromone intensity, moving to the position of a higher concentration of pheromone in a certain way and achieve the exploration of unknown regions during the move. The movement of the single ant will cause the change of the whole position distribution of ant colony, so that all the ants keep aggregating to the higher quality of food source and search the highest quality of food source in the continuous space eventually. This model is called position distribution model of ant colony foraging” [7]

Menurut Lopez [8], algoritma ACO pada awalnya dikonstruksikan dengan kandidat konstruksi untuk mendapatkan kombinasi masalah optimasi dimulai dengan solusi yang masih kosong. Kemudian ditambahkan dengan komponen solusi untuk mendapatkan kandidat solusi yang lebih menyeluruh dan optimal. Algoritma ACO dapat di ilustrasikan dengan langkah-langkah yang di tampilkan pada gambar 2 berikut ini :

---

```

SetParameters
Initialize(PhInfo)
best-so-far ← ∅
while termination condition not met do
    S ← ∅
    for each ant do
        solution ← ∅
        while solution not completed do
            component ← ConstrPolicy(PhInfo, HeurInfo)
            solution ← solution ∪ component
        end while
        S ← S ∪ {solution}
    end for
    ApplyLocalSearch( S ) # optional
    best-so-far ← FindBest( S ∪ {best-so-far} )
    B ← Select( S ∪ {best-so-far} ) # B ⊆ S ∪ {best-so-far}
    Evaporation( PhInfo, ρ )
    Update(PhInfo, B)
end while
return best-so-far

```

---

*Figure 2 Ilustrasi algoritma ACO*

(Sumber : Manuel Lopez et al, 2004)

Dalam *sigle objective ACO* yang disebut Lopez sebagai QAP, komponen solusi didefinisikan sebagai tanda fasilitas menuju sebuah lokasi. Karena itu, pheromone untuk menunjukkan sebuah lokasi dapat mengikuti persamaan berikut :

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad \text{if } j \in N_i^k$$

..... (1)

Dimana :

$\tau_{ij}(t)$  = informasi pheromone pada iterasi yang sedang berjalan

$\eta_{ij}$  = QAP spesifik pada informasi heuristic

adan  $\beta$  = dua parameter yang menentukan nilai pheromone relative dan informasi heuristic

$\eta_i^k$  = ant tetangga (*neighborhood*) dari ant k.

$p_{ij}^k(t)$  = komponen solusi pada sebuah lokasi.

Untuk melakukan konstruksi solusi langkah tersebut diulangi sampai penentuan neighborhood menjadi kosong sehingga solusi lengkap dapat dihitung. Setelah solusi selesai dikonstruksi, pheromone kemudian di update. Performa evaporasi dilihat dengan penurunan pertama pheromone trails dengan factor yang konstan kemudian ant di ijinakan untuk mendeposit pheromone pada solusi yang telah dikonstruksikan. Setiap pheromone di update dengan mengikuti persamaan berikut :

$$\Delta\tau_{ij}(t + 1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

..... (2)

Dimana parameter  $\rho$  (dengan  $0 \leq \rho < 1$ ) adalah jejak *persistence* (dimana  $1 - \rho$  model evaporasi) dan  $\Delta\tau_{ij}^k(t)$  adalah jumlah pheromone ant k yang ditambahkan kedalam pheromone trail. Dalam AS (ant system) jumlah tersebut didefinisikan sebagai persamaan (3) berikut ini :

$$\Delta\tau_{ij}^k(t) = \begin{cases} 1/f(\phi^k) & \text{if } \phi_i^k = j \text{ in iteration } t \\ 0 & \text{otherwise} \end{cases}$$

..... (3)

Dimana :

$\phi^k$  = kosntruksi solusi dengan ant k

$f(\phi^k)$  = denotes nilai objek dari QAP untuk sebuah solusi  $\phi$ .

Dengan demikian ACO diaplikasikan dalam jalur mitigasi dengan cara sebagai berikut :

- *Pertama* adalah dengan mengkonstruksikan masalah ke dalam sebuah graph  $G = (V, E)$  dengan V himpunan vertek yang merepresentasikan himpunan titik – titik, dan E adalah himpunan dari edge yang merepresentasikan jarak antara dua titik.
- *Kedua*, kendala yang terdapat pada jalur evakuasi yaitu mengunjungi n titik dengan titik-titik yang ada hanya dikunjungi sekali dimana titik awal sama dengan titik akhir. Tujuan dari



pemilihan jalur terpendek, jalur logistic terpendek dan penentuan titik shelter yaitu mencari tour terpendek terhadap  $n$  titik.

- *Ketiga*, pemberian nilai intensitas jejak semut (*Pheromone*) dan informasi heuristik. Pemberian nilai *Pheromone* ( $\tau_{rs}$ ) dalam jalur evakuasi dilakukan saat semut mengunjungi titik  $s$  setelah mengunjungi titik  $r$ . Informasi heuristik ( $\eta_{rs}$ ) merupakan informasi yang merepresentasikan kualitas suatu edge antara titik  $r$  dan titik  $s$ , informasi ini dihitung sebelum algoritma dijalankan. Dengan  $\eta_{rs} = \frac{1}{d_{rs}}$ ,  $d_{rs}$  adalah jarak antara titik  $r$  dan titik  $s$ .
- *Keempat*, (*tour construction*). Sebuah tour dibangun dengan mengaplikasikan prosedur sederhana sebagai berikut : Inisialisasi, ditempatkan  $m$  semut di  $n$  titik menurut aturan tertentu, kemudian semut mengaplikasikan *state transition rule* secara iteratif. Semut membangun lintasan sebagai berikut. Pada titik  $r$ , semut memilih secara probabilistik titik  $s$  yang belum dikunjungi menurut intensitas *Pheromone* ( $\tau_{rs}$ ) pada edge antara titik  $r$  ke titik  $s$ , serta informasi heuristik lokal yang ada, yaitu panjang sisi (edge). Semut secara probabilistik lebih menyukai titik yang dekat dan terhubung dengan tingkat *Pheromone* yang tinggi. Untuk membangun jalur terpendek yang mungkin, setiap semut mempunyai suatu bentuk memori yang disebut *tabu list*. *Tabu list* digunakan untuk menentukan himpunan titik yang masih harus dikunjungi pada setiap langkah dan untuk menjamin terbentuknya jalur terpendek yang mungkin. Selain itu semut bisa melacak kembali lintasannya, ketika sebuah lintasan itu diselesaikan. Setelah semua semut membangun sebuah tour, *Pheromone* di-update dengan cara mengurangi tingkat *Pheromone* oleh suatu faktor konstanta dan kemudian semut meletakkan *Pheromone* pada edge yang dilewati. *Update* dilakukan sedemikian rupa sehingga edge dari lintasan yang lebih pendek dan dilewati banyak semut menerima jumlah *Pheromone* yang lebih banyak. Karena itu pada iterasi algoritma yang berikutnya akan mempunyai probabilitas yang lebih tinggi untuk dipilih.

### 3.2. Multiple Objective – Ant Colony Optimization (MO-ACO)

Multiple objective optimization didefinikan oleh Daniel Angus, et al sebagai optimisasi secara simultan dari fungsi multiple objective [18]. Dalam taxonomi MOACO, algoritma ini merupakan golongan algoritma metaheuristic yang memiliki fungsi colony lebih dari satu ant colony [19] [20]. Dengan demikian pheromonanya didapatkan dari lebih dari satu pheromone yang merupakan bagian dari colony masing-masing.

Algoritma MOACO menambahkan angka pada komponen algoritma kedalam algoritma *metaheuristic* ACO dengan menghubungkannya kedalam *multi-objective problem* [19]. Dengan mengasumsikan bahwa algoritma MO-ACO menggunakan *multi pheromone information*, dimana setiap informasi pheromone memiliki perbedaan yang mendefinisikan objek dan berat yang dimilikinya sehingga menjadi satu nilai saja. Nilai tersebut sama dengan berat yang digunakan dalam menghitung agregat object dalam skala pada *multi objective problem* [8].

Jika diasumsikan arbitrary *multi objective colony (MOCO) problem* dengan Q objective seperti yang dijelaskan Manuel Lopez [8], maka ant *k* ditambahkan dengan *j* pada solusi partial akan mengikuti probability untuk menghitung pheromone dengan persamaan berikut :

$$p_{S_j}^k = \frac{[\prod_{q=1}^Q (\tau_{S_{qj}}^q)^{\lambda q}]^{\beta}}{\left( [\prod_{q=1}^Q (\tau_{S_{ql}}^q)^{\lambda q}]^{\alpha} \cdot [\prod_{q=1}^Q (\eta_{S_{ql}}^q)^{\lambda q}]^{\beta} \right)} \quad \text{if } j \in$$

$$N_S^k \dots\dots\dots (4)$$

Dimana :

S = {S<sub>1</sub>, S<sub>2</sub>, ..., S<sub>Q</sub>}, merupakan state factor pada *current partial solution*

S<sub>q</sub> = vector yang dinilainya ditambahkan dengan each objective q=1, ..., Q yang diberikan komponen solusi

N<sub>S</sub><sup>k</sup> = *feasible neighborhood* dari ant *k* pada *current state vector*

τ<sub>S<sub>qj</sub></sub><sup>q</sup> = Informasi pheromone dari *j* yang diberikan S<sub>q</sub> untuk q<sup>th</sup> objective pada *current iteration*

$\eta_{sqj}^q$  = informasi heuristic pada j didapatkan dari  $S_q$  yang terhubung dengan objective  $q^{\text{th}}$   
 $\Lambda_q$  = nilai berat dari objective  $q^{\text{th}}$  dan dapat dilihat sebagai komponen  $q^{\text{th}}$  vector berat  $\lambda$ .

Pada performa terbaik ACO, Strategy update pheromone pada single colony dilakukan dengan cara memilih satu solusi yang terbaik atau sedikit dari solusi yang terbaik yang ditetapkan dari *current iteration* atau sejak algoritmanya dimulai (*best so far strategy*). Strategy ini dilakukan dengan memilih pheromone yang dominan dan direkomendasi untuk digunakan [19].

Strategy dalam melakukan update pheromone multi colony, hampir sama dengan strategi yang dipakai pada single colony. Untuk menjalankan spesialisasi colony, setiap deposit pheromone hanya digunakan untuk satu colony saja [8]. Pemilihan pheromone dipilih yang dominan menggunakan metode *straightforwardly adapted* pada pendekatan multi colony. Penggunaan ini merujuk pada *ant pereto set* yang kandidatnya didapatkan dari distribusi ant antara colony dan diijinkan melakukan deposit pheromone [8].

Pada penelitian ini, untuk mendapatkan mitigasi pada penentuan lokasi shelter pengungsi, jalur distribusi terbaik dan jalur evakuasi terbaik, maka peneliti menggunakan framework MOACO buatan dari Manuel López-Ibáñez and Thomas Stützle [21]. Framework ini memiliki langkah-langkah sebagai berikut :

#### 1. Konfigurasi Framework MOACO

Untuk melakukan konfigurasi pada algoritma MOACO seperti yang dijelaskan Manuel Lopez [21] adalah :  
 “Pertama, pada single objective ACO, informasi pheromone terkait dengan fungsi objektif yang adalah komponen solusi berkualitas tinggi dimana menerima lebih banyak pheromone. Dalam konteks multi-objective, fungsi objective adalah multi-dimensi dan tidak skalar, dan hanya ada urutan parsial antara solusi. Selain itu, dalam beberapa masalah, bisa masuk akal untuk memiliki komponen solusi yang berbeda untuk masing-masing objective, oleh karenanya, asosiasi matriks pheromonenya berbeda. Misalnya, dalam aplikasi

ACO untuk multi objective dengan masalah penjadwalan, satu matriks pheromone dapat mewakili pekerjaan  $x$  hubungan pekerjaan dan posisi lainnya  $\times$  hubungan pekerjaan. Dalam hal ini, kedua matriks dapat diperbarui dengan jumlah pheromone yang sama karena maknanya berbeda. Dalam masalah lain seperti BTSP, baik matriks pheromone mewakili komponen solusi yang sama (tepi), dan karenanya, pheromone perlu diperbarui baik menggunakan solusi yang berbeda atau menggunakan jumlah pheromone yang berbeda. Atau, bisa masuk akal juga menggunakan matriks pheromone tunggal. Jika beberapa pheromone atau matriks heuristik yang digunakan, biasanya dikumpulkan selama pembangunan solusi dengan cara bobot. Namun, ada perbedaan yang kuat antara algoritma MOACO dengan dilakukannya cara agregasi ini dan berapa banyak bobot yang digunakan. Sebuah pertanyaan desain lebih lanjut adalah berapa semut yang dipilih untuk menyetorkan pheromone. Algoritma MOACO memilih beberapa (atau semua) solusi nondominated, atau memilih solusi terbaik terhadap objective yang terkait dengan matriks pheromone yang diperbarui. Akhirnya, beberapa algoritma MOACO menggunakan beberapa koloni semut. Kami mengambil pandangan bijaksana terhadap komponen dari desain algoritma MOACO, yang memungkinkan kita untuk memilih abstrak dari pilihan desain tertentu.”

Manuel Lopez et all [21] telah mengidentifikasi beberapa komponen algoritmik yang sesuai dengan alternatif desain yang dijelaskan di atas. Mereka membagi komponen ini ke dalam tiga kelompok utama: (i) yang berkaitan dengan definisi pheromone dan pembangunan solusi; (ii) yang berkaitan dengan update dari pheromone; dan (iii) yang berhubungan dengan penggunaan beberapa koloni. Lopez memeriksa komponen ini secara lebih rinci dalam bagian berikut.

a) Komponen MOACO Algorithm untuk konstruksi solusi

Menurut kajian yang dilakukan Manuel Lopez et all [21], komponen MOACO Algorithm dapat di konstruksi dengan solusi sebagai berikut :

“Salah satu keputusan desain yang paling penting ketika menerapkan algoritma ACO untuk sebuah masalah adalah definisi informasi pheromone (dan heuristik). Dalam konteks multi-objective, masalah ini diperparah oleh kenyataan bahwa ada beberapa cara untuk mengevaluasi solusi tunggal, dan oleh fakta bahwa beberapa solusi optimal mungkin (Pareto) diterapkan”.

Menurut kajian Lopez et all [21] pertanyaan desain ini dapat dijawab oleh komponen algoritmik berikut :

*Tunggal / Multi Pheromone / Informasi Heuristic.* Lopez menjelaskan bahwa mungkin memiliki satu (tunggal) atau beberapa (multiple) matriks baik untuk informasi pheromone  $\tau$  atau, jika berlaku, informasi heuristic  $\eta$ . Dalam kasus matriks multiple heuristic, setiap matriks dikaitkan dengan objective yang berbeda, seperti  $\eta^1$  dan  $\eta^2$  sesuai dengan informasi heuristic dari masing-masing objective. Untuk beberapa masalah, hal itu tidak mungkin untuk menentukan informasi heuristic untuk setiap objective secara independen, jadi kita akan dipaksa untuk menggunakan matriks heuristic tunggal. Demikian pula, dalam kasus matriks multiple pheromone,  $\tau^1$  dan  $\tau^2$  akan dikaitkan dengan masing-masing objective; prosedur untuk memilih solusi yang akan digunakan untuk memperbarui setiap matriks harus menegakkan perbedaan ini entah bagaimana [21].

Lebih lanjut lopez et all menjelaskan bahwa selama mengkonstruksi solusi, setiap kali kita memiliki beberapa matriks, kita perlu agregat matriks-matriks tersebut menjadi matriks tunggal. Metode agregasi didefinisikan oleh komponen Agregasi, yang dijelaskan di bawah ini. Bisa dibayangkan untuk menggunakan metode agregasi yang berbeda untuk pheromone dan informasi heuristic; Namun, kita tidak mengeksplorasi kemungkinan ini di sini [21].

*Aggregation.* Nilai-nilai dari multi pheromone (atau heuristic) matriks harus dikumpulkan (aggregate) ke nilai pheromone tunggal (atau heuristic). Lopez et all [21] mengidentifikasi tiga alternatif dalam literature yaitu :

- Weighted sum :

$$\tau_{ij} = (1 - \lambda)\tau_{ij}^1 + \lambda\tau_{ij}^2 \quad \text{and} \quad \eta_{ij} = (1 - \lambda)\eta_{ij}^1 + \lambda\eta_{ij}^2. \quad (5)$$

- Weighted product :

$$\tau_{ij} = (\tau_{ij}^1)^{(1-\lambda)} \cdot (\tau_{ij}^2)^\lambda \quad \text{and} \quad \eta_{ij} = (\eta_{ij}^1)^{(1-\lambda)} \cdot (\eta_{ij}^2)^\lambda. \quad (6)$$

- Random :

Pada setiap langkah pembangunan, diberi nomor acak seragam  $U(0, 1)$ , semut memilih yang pertama dari dua matriks jika  $U(0, 1) < 1 - \lambda$ ; selain itu memilih matriks lainnya.

Dalam tiga metode agregasi yang dijelaskan di atas, ada weight  $\lambda$  yang menjadi bias agregasi terhadap satu objective atau yang lain. Set bobot  $\Lambda$  didefinisikan oleh komponen  $N^{\text{weights}}$  dan *NextWeight*.

$N^{\text{weights}}$  dan *NextWeight*. Himpunan bobot didefinisikan dalam interval  $[0, 1]$  sebagai

$$\Lambda = \{\lambda_i = 1 - (i - 1)/N^{\text{weights}} - 1\}, \quad i = 1, \dots, N^{\text{weights}}$$

dimana  $N^{\text{weights}} = |\Lambda|$  adalah parameter dari framework.

Komponen *NextWeight* menentukan berat tertentu digunakan oleh semut pada iterasi tertentu. Opsi diuji untuk *NextWeight* yang baik bahwa semua semut menggunakan bobot yang sama pada iterasi tertentu (satu-berat-per-iterasi), atau bahwa semua bobot yang digunakan pada setiap iterasi (all-bobot-per-iterasi). Dalam kasus satu-berat-per-iterasi, berat digunakan dalam iterasi yang berurutan berikut memerintahkan urutan unsur-unsur  $\Lambda$ , dan urutan terbalik ketika berat badan terakhir dalam urutan tercapai. Dalam kasus all-bobot-per-iterasi, ketika jumlah semut  $N^a$  lebih besar dari  $N^{\text{weights}}$ , beberapa semut akan menggunakan berat yang sama. Sebuah *speed-up* jelas adalah untuk menghitung agregasi dari matriks pheromone hanya sekali per berat per iterasi. Ini menjelaskan perilaku untuk koloni tunggal [21].

#### b) MOACO Algorithm untuk Pheromone Update

Untuk menjelaskan cara update pheromone, Manuel Lopez et al menjelaskan sebagai berikut :

“Mengingat satu set  $A^{\text{upd}}$  solusi calon memperbarui informasi pheromone, komponen *PheromoneUpdate* yang memutuskan solusi

memperbarui informasi pheromone dan bagaimana. Parameter  $N^{\text{upd}}$  menentukan berapa banyak solusi digunakan untuk memperbarui setiap matriks pheromone” [21].

Menurut Manuel Lopez et all [21], untuk melakukan update pheromone, komponen PheromoneUpdate harus mempertimbangkan beberapa alternatif sebagai berikut:

- *Solusi Nondominated*. Solusi yang digunakan untuk memperbarui informasi pheromone adalah nondominated solusi dalam  $A^{\text{upd}}$ . Ketika ada lebih nondominated solusi dari  $N^{\text{upd}}$ , harus diterapkan mekanisme pemotongan dari SPEA2 untuk memilih hanya solusi  $N^{\text{upd}}$ . Pada prinsipnya, adalah mungkin untuk menggabungkan pheromone nondominated ini Metode update dan matriks multiple pheromone dengan menggunakan solusi yang sama untuk memperbarui kedua matriks selama update berbeda di setiap matriks pheromone.
- *Best-of-objective*. Mekanisme ini pertama memilih dari  $A^{\text{upd}}$  yang memiliki  $N^{\text{upd}}$  solusi terbaik terhadap masing-masing objective. Di kasus matriks multiple pheromone, setiap matriks pheromone diperbarui menggunakan solusi  $N^{\text{upd}}$  terkait dengan objective yang sesuai. Jika tidak,  $2 \cdot N^{\text{upd}}$  digunakan untuk solusi memperbarui matriks pheromone tunggal.
- *Best-of-Objective -per-Weight*. Untuk setiap berat  $\lambda$  dan masing-masing objective, ada daftar solusi terbaik  $N^{\text{upd}}$  untuk objective yang dihasilkan menggunakan  $\lambda$ . Dalam kasus tertentu jika  $\lambda = 0$ , maka kita hanya tetap menggunakan daftar untuk objective pertama, dan kami melakukan hal yang sama untuk  $\lambda = 1$  dan objective kedua. Bila menggunakan matriks multiple pheromone, setiap matriks diperbarui hanya menggunakan solusi dari daftar yang terkait dengan objective yang sama. Dalam kasus matriks pheromone tunggal, kedua daftar yang digunakan untuk update. Oleh karena itu, dalam kasus tertentu dari dua bobot, metode ini setara untuk best-of-objective. Metode best-of-objective -per-weight yang digunakan oleh algoritma mACO-1 dan mACO- 2, dan kami memasukkannya untuk kelengkapan. Namun, tidak

jelas bagaimana pendekatan ini harus diperluas ke beberapa koloni, karena solusi mungkin dipertukarkan antara koloni dengan bobot yang berbeda.

Metode di atas tidak menjelaskan apakah  $A^{upd}$  terdiri dari best-so-far atau solusi iterasi terbaik, karena ini ditentukan oleh algoritma ACO mendasari syarat tertentu. Dengan demikian saat menggunakan algoritma MOACO, pheromone dapat diupdate menggunakan salah satu metode diatas berdasarkan jenis algoritmanya.

c) MOACO Algorithm untuk Multiple Coloni

Penelitian yang dilakukan Manuel Lopez [21] menunjukkan banyaknya algoritma MOACO yang mengusulkan penggunaan beberapa koloni semut, menggunakan berbagai definisi tentang apa itu koloni. Salah satu Pendekatan yang populer adalah untuk menentukan dua koloni, masing-masing memiliki informasi pheromone sendiri. Koloni ini mungkin menjadi solusi yang digunakan untuk memperbarui informasi pheromone. Beberapa semut, yang kadang-kadang dikatakan milik "Extra" koloni, mengumpulkan informasi pheromone dari dua koloni lainnya. Menurut Manuel Lopez, pendekatan ini sebenarnya setara untuk menggunakan salah satu koloni dengan beberapa matriks pheromone, yang dikumpulkan dengan cara pembobotan yang berbeda menggunakan allweights- per-iterasi pilihan, dan tampaknya tidak tepat mengatakan bahwa semut milik koloni yang berbeda hanya karena mereka menggunakan bobot yang berbeda. Sedangkan arsitektur multi-koloni dari Iredi yang dijelaskan Manuel Lopez [21] mendefinisikan koloni sebagai sebuah kelompok semut yang terkait dengan informasi pheromone tertentu, sehingga semut dari setiap koloni membangun solusi hanya menurut informasi pheromone koloni mereka. Pada beberapa kasus informasi pheromone, setiap koloni memiliki dua matriks pheromone, satu untuk setiap tujuan. Ketika set dari bobot yang digunakan, masing-masing koloni menetapkan sendiri bobot  $\Lambda_c$ . Arsitektur multi-koloni ini memungkinkan kita untuk menggeneralisasi semua komponen



sebelumnya. Oleh karenanya, Manuel Lopez mengadopsinya untuk dijadikan kerangka kerja (*framework*) algoritma MOACO. Selain itu, koloni bekerja sama dengan menggunakan arsip umum solusi nondominated untuk mendeteksi yang didominasi. Lebih lanjut kerjasama diberlakukan dengan bertukar solusi untuk memperbarui informasi pheromone . Pengaturan ini dikendalikan oleh Komponen MultiColonyUpdate dijelaskan di bawah. Nomor koloni diberikan oleh komponen  $N^{col}$ , dan komponen dijelaskan di bawah hanya berpengaruh ketika  $N^{col}$  lebih besar dari satu.

MultiColonyWeights. Dalam kasus beberapa koloni seperti yang dijelaskan Manuel Lopez, diciptakan satu set bobot  $\Lambda_c$  dari ukuran  $N^{weights}$  untuk setiap koloni  $c$ . Set ini dibangun dengan mengikuti dua alternatif : menguraikan dan interval yang tumpang tindih. Dalam kedua kasus, pertama dihasilkan jumlah yang diperlukan bobot yang merata dalam interval  $[0, 1]$ . Kemudian, untuk menguraikan interval, set ini dibagi menjadi sub interval yang menguraikan kondisi sama per koloni, yaitu  $c, i = ((c - 1) \cdot N^{weights} + (i - 1)) / (N^{weights} \cdot N^{col}), i = 1, \dots, N^{weights}, c = 1, \dots, N^{col}$ . Pada kasus interval yang tumpang tindih, sub-interval tumpang tindih dengan 50% dan karenanya,  $\Lambda_c$  dan  $\Lambda_{c+1}$  dibagi 50% dari bobot mereka.

MultiColonyUpdate. Manuel Lopez [21] menjelaskan bahwa dalam beberapa kasus koloni, solusi yang dihasilkan oleh semua koloni pada iterasi saat ini disimpan dalam arsip  $A^{iter}$ , sehingga semua koloni berkontribusi untuk mendeteksi dan menghapus solusi dominasi. Set  $A^{upd}$  kemudian dibangun dari solusi nondominated tersisa di  $A^{iter}$  atau dari arsip dari semua solusi nondominated pernah ditemukan, tergantung pada algoritma ACO yang mendasarinya. Sesudah ini langkah, solusi di  $A^{upd}$  ditugaskan kembali ke setiap koloni untuk memperbarui informasi pheromone . Metode dasar, disebut update dengan asal (*region*), memberikan setiap solusi dari  $A^{upd}$  ke koloni asli. Untuk menegakkan kerjasama yang lebih, memungkinkan solusi koloni pertukaran. Salah satu metode pertukaran, yang disebut update dengan wilayah, bekerja dengan cara membagi  $A^{upd}$  di bagian yang sama di antara koloni sedemikian rupa

sehingga cara di setiap koloni sesuai dengan salah satu wilayah ruang obyektif. Kedua pengaturan, update dengan asal dan pembaruan menurut wilayah.

d) MOACO Framework

```

1: for each colony  $c \in \{1, \dots, N^{col}\}$  do
2:   InitializePheromoneInformation()
3:    $\Lambda_c := \text{MultiColonyWeights}()$ 
4: end for
5: InitializeHeuristicInformation()
6:  $A^{bf} := \emptyset$ 
7:  $iter := 0$ 
8: while not stopping criteria met do
9:    $A^{iter} := \emptyset$ 
10:  for each colony  $c \in \{1, \dots, N^{col}\}$  do
11:    for each ant  $k \in \{1, \dots, N^a\}$  do
12:       $\lambda := \text{NextWeight}(\Lambda_c, k, iter)$ 
13:       $\tau := \begin{cases} \text{Aggregation}(\lambda, \{\tau_c^1, \tau_c^2\}) & \text{if multiple } [\tau] \\ \tau_c & \text{if single } [\tau] \end{cases}$ 
14:       $\eta := \begin{cases} \text{Aggregation}(\lambda, \{\eta^1, \eta^2\}) & \text{if multiple } [\eta] \\ \eta & \text{if single } [\eta] \end{cases}$ 
15:       $s := \text{ConstructSolution}(\tau, \eta)$ 
16:       $s := \text{WeightedLocalSearch}(s, \lambda)$  // Optional
17:       $A^{iter} := \text{RemoveDominated}(A^{iter} \cup \{s\})$ 
18:    end for
19:  end for
20:   $A^{bf} := \text{RemoveDominated}(A^{bf} \cup A^{iter})$ 
21:  for each colony  $c \in \{1, \dots, N^{col}\}$  do
22:     $A_c^{upd} := \text{MultiColonyUpdate}(A^{upd})$ 
23:    PheromoneUpdate( $A_c^{upd}, N^{upd}$ )
24:  end for
25:   $iter := iter + 1$ 
26: end while
27: Output:  $A^{bf}$ 

```

Figure 3 MOACO framework

(sumber : Manuel Lopez et all, 2012)

Tabel merangkum komponen algoritmik yang didefinisikan atas dan domain algoritma MOACO. Beberapa pengaturan hanya signifikan untuk nilai-nilai tertentu saja dari pengaturan lainnya. Sebagai contoh, sebuah agregasi Metode ini hanya diperlukan jika ada beberapa pheromone atau matriks heuristik. Kami mengusulkan Algoritma 2 sebagai cara untuk mengintegrasikan komponen ini ke dalam kerangka kerja MOACO fleksibel untuk optimasi bi-objektif. Ini mengikuti garis dasar yang metaheuristik ACO. Pertama, algoritma menginisialisasi pheromone informasi (fungsi

InitializePheromoneInformation, baris 2) dan himpunan bobot masing-masing koloni (fungsi MultiColonyWeights, jalur 3), sedangkan informasi heuristic diinisialisasi hanya sekali, karena hal itu dibagikan oleh berbagai koloni. Arsip semua solusi nondominated pernah ditemukan (terbaik-jadi-jauh arsip, Abf) dan counter iterasi yang diinisialisasi di jalur 6 dan 7, masing-masing. Pada setiap iterasi, setiap koloni membangun solusi  $N_a$  menurut sendiri Informasi pheromone, yang mungkin mungkin digabungkan berat suatu? dari mengatur sendiri bobot (garis 13-15). Tergantung pada pengaturan dari NextWeight,? mungkin sama berat badan atau yang berbeda untuk setiap semut di koloni. dalam singleobjective algoritma ACO, solusi sering ditingkatkan dengan berarti dari pencarian lokal. Oleh karena itu, kami telah menyertakan opsional Prosedur WeightedLocalSearch (line 16) yang meningkatkan suatu solusi dengan menerapkan pencarian lokal untuk jumlah agregasi tertimbang fungsi objektif, menggunakan berat yang sama seperti itu digunakan untuk membangun solusi. Solusi baru ditambahkan ke arsip iterasi saat bersama oleh semua koloni Aiter (Line 17). Setelah semua semut dari seluruh koloni telah selesai membangun solusi, arsip Abf terbaik-jadi-jauh diperbarui dengan solusi nondominated ditemukan dalam iterasi saat Aiter (Fungsi RemoveDominated, line 20). Update dari pheromone terdiri dari dua tahap. Pertama, MultiColonyUpdate mendistribusikan arsip solusi untuk update (Aupd) antara koloni. Kedua, prosedur PheromoneUpdate (line 23) memutuskan bagaimana solusi ditugaskan untuk setiap koloni memperbarui Informasi pheromone koloni. Algoritma MOACO terus sampai sejumlah iterasi atau batas waktu tercapai. Gambar 3 grafis menggambarkan hubungan antara berbagai komponen dari kerangka MOACO.

ALGORITHMIC COMPONENTS OF THE PROPOSED MOACO FRAMEWORK.

Component	Domain	Description
$[\tau]$	{ single, multiple }	Definition of pheromone matrices
$[\eta]$	{ single, multiple }	Definition of heuristic matrices
Aggregation	{ weighted sum, weighted product, random }	How weights are used to aggregate different matrices
$N^{weights}$	$N^+$	Number of weights (per colony)
NextWeight	{ one weight per iteration (1wpi), all weights per iteration (awpi) }	How weights are used at each iteration
PheromoneUpdate	{ nondominated solutions (ND), best-of-objective (BO), best-of-objective-per-weight (BOW) }	Which solutions are selected for updating the pheromone matrices
$N^{sp}$	$N^+$	Number of solutions that update each $[\tau]$ matrix
$N^{col}$	$N^+$	Number of colonies
MultiColonyWeights	{ disjoint, overlapping }	How weights are partitioned among colonies
MultiColonyUpdate	{ origin, region }	How solutions are assigned to colonies for update

Figure 4 Komponen algoritma MOACO

(sumber : Manuel Lopez at all, 2012)

## 2. Design MOACO Algorithm

Manuel Lopez [21] mendesain kerangka algoritma MOACO dengan cara konfigurasi kerangka tertentu dari meniru desain algoritma MOACO yang ada. Tujuannya adalah untuk menghasilkan algoritma yang lebih fleksibel dengan mengidentifikasi komponen konfigurasi algoritma tertentu dalam rangka penyesuaian pilihan desain algoritma dalam setiap kasus. Secara khusus, desain ini didasari oleh algoritma ACO, inisialisasi pheromone atau memperbarui jumlah, informasi heuristik, atau masalah-dependent speed-up.

Penelitian ini secara khusus mengadopsi desain yang dibuat oleh Manuel Lopez dalam mendesain algoritma MOACO. Hasil yang didapatkan, mengikuti pola framework MOACO yang dibuat oleh Manuel Lopez. Tabel II merangkum konfigurasi framework MOACO yang terdapat dalam ulasan ini.

Algorithm	$N^{nd}$	$[\tau]$	$[\eta]$	Aggregation	$N^{weights}$	PheromoneUpdate	$N^{nd}$
MOAQ [4, 30]	1	1	2	-	2 ( $\Lambda = \{0, 1\}$ )	nondominated solutions	$\infty$
BicriterionAnt [7]	any	2	2	weighted product	$N^A$	nondominated solutions	$\infty$
MACS [6]	1	1	2	weighted product	$N^A$	nondominated solutions	$\infty$
COMPETants [26]	1	2	2	weighted sum	3 ( $\Lambda = \{0, 0.5, 1\}$ )	best-of-objective	any
P-ACO [5]	1	2	1, 2	weighted sum	$N^A$	best-of-objective	2
mACO-1 [9]	1	2	2	$\left\{ \begin{array}{l} \text{random } (\tau) \\ \text{weighted sum } (\eta) \end{array} \right.$	3 ( $\Lambda = \{0, 0.5, 1\}$ )	best-of-objective-per-weight	1
mACO-2 [9]	1	2	2	weighted sum	3 ( $\Lambda = \{0, 0.5, 1\}$ )	best-of-objective-per-weight	1
mACO-3 [9]	1	1	1	-	-	nondominated solutions	$\infty$
mACO-4 [9]	1	2	1	random ( $\tau$ )	1 ( $\Lambda = \{0.5\}$ )	best-of-objective	1

Figure 5 Taksonomi algoritma MOACO  
(sumber Manuel Lopez at all, 2012)

#### a) MOAQ

Manuel Lopez [21] menjelaskan bahwa meskipun beberapa Tujuan Ant-Q (MOAQ) awalnya dirancang untuk masalah dengan preferensi yang diberikan agar tujuan (optimasi leksikografis), tetapi secara umum MOAQ dirancang untuk masalah bi-objective dalam hal optimalisasi algoritma Pareto [22]. Garcia-Martinez menggunakan matriks pheromone tunggal dan ganda matriks heuristik, satu untuk setiap objective. Untuk pheromone update, mereka menggunakan semua solusi nondominated. Garcia-Martinez et al [22] membagi semut menjadi dua kelompok, dan setiap kelompok hanya menggunakan informasi heuristik sesuai dengan satu tujuan. Hal tersebut sama seperti Manuel Lopez dalam frameworknya yang dilakukan dengan menggunakan bobot  $\Lambda = \{0, 1\}$  untuk menggabungkan (*aggregate*) dua matriks heuristik, sehingga setengah dari semut menggunakan setiap berat badan. Seperti satu set bobot efektif berarti bahwa tidak ada agregasi yang sebenarnya terjadi, tapi bukannya setengah dari semut menggunakan satu matriks heuristik dan setengah lainnya menggunakan yang lain [21].

b) bicriterionAnt

Salah satu bagian dalam algoritma MOACO yang dijelaskan Manuel Lopez [21] disebut BicriterionAnt, dengan multi pheromone dan matriks heuristik, digabungkan dengan produk tertimbang (*weighted*). Setiap  $k$  semut menggunakan bobot yang berbeda dengan  $\lambda_k$  untuk menggabungkan matriks pheromone; dengan demikian, ada banyak bobot sebagai semut ( $N^{\text{weights}} = N^a$ ). Para penulis dari BicriterionAnt menyarankan untuk memperbarui matriks pheromone menggunakan solusi nondominated dalam iterasi saat ini. Selain itu, memperbarui kedua matriks pheromone dengan jumlah yang sama. Seperti update tidak menghasilkan matriks pheromone identik karena, dalam masalah tersebut ditangani oleh mereka, matriks pheromone merupakan komponen solusi yang berbeda seperti yang telah dijelaskan. Ketika menerapkan BicriterionAnt ke BTSP, Garcia-Martinez et al [22] menggunakan nilai fungsi tujuan dari masing-masing tujuan untuk memperbarui matriks yang sesuai  $\Delta\tau^k = 1 / f^k(s_a)$ . Manuel Lopez melakukan hal yang sama dalam rangka untuk kombinasi dari beberapa matriks pheromone dan pembaruan pheromone nondominated [21].

Selain algoritma di atas, diusulkan penggunaan beberapa koloni, dan menentukan koloni sebagai sekelompok semut yang membangun solusi menurut informasi pheromone mereka sendiri. Dalam hal ini, algoritma BicriterionAnt multi-koloni mirip dengan pendekatan multi-start dengan beberapa kekhasan. Pertama, koloni yang berbeda mengkhususkan diri di berbagai daerah perbatasan Pareto dengan menggunakan set yang berbeda dari bobot untuk agregat informasi pheromone. Kedua, koloni bekerja sama untuk mendeteksi solusi dominasi dengan menjaga solusi dalam arsip umum. Ketiga, koloni juga dapat bekerja sama dengan bertukar solusi. Sebagaimana dijelaskan dalam Bagian III-C, ini cocok dengan pendekatan multi-koloni diadopsi dalam framework MOACO buatan Manuel Lopez [21].

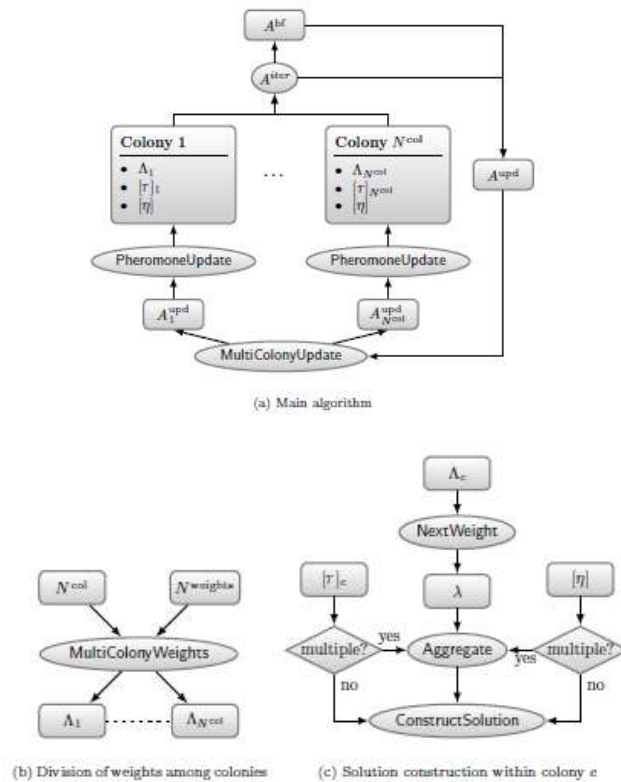


Figure 6 Gambar algoritma MOACO  
(sumber: Manuel Lopez, 2012)

### c) Multiple Ant Colony System

Beberapa Ant Colony System (MACS) seperti yang dijelaskan Manuel Lopez [21] menggunakan satu matriks heuristik untuk setiap tujuan dan matriks pheromone tunggal. Matriks heuristik dikumpulkan oleh produk tertimbang, dan setiap semut menggunakan berat yang berbeda. Selain itu, informasi pheromone diperbarui dengan solusi nondominated. MACS dapat dilihat sebagai varian MOAQ, seperti yang didefinisikan oleh Garc'ia- Mart'inez et al [22], yang menggunakan lebih dari dua bobot untuk mengumpulkan informasi heuristik. Di sisi lain, satu-satunya perbedaan antara MACS dan single-koloni BicriterionAnt adalah jumlah matriks pheromone. Akibatnya, sangat mudah untuk mendefinisikan multi-koloni MACS, seperti yang Manuel Lopez lakukan dalam framework MOACO ciptaannya.

#### d) COMPETants

COMPETants seperti yang dijelaskan Manuel Lopez [21] disajikan sebagai pendekatan multi-koloni, dengan satu koloni untuk setiap tujuan. Setiap koloni memiliki satu pheromone dan matriks heuristik. Setiap koloni membangun solusi secara independen, kecuali untuk sejumlah semut (disebut "mata-mata"), yang agregat dua matriks pheromone oleh jumlah tertimbang (dengan  $\lambda = 0,5$ ) baik pertama menggunakan atau matriks heuristik kedua, sehingga menciptakan dua solusi. Akhirnya, sejumlah semut dari setiap koloni digunakan untuk memperbarui matriks pheromone dari setiap koloni.

Manuel Lopez merumuskan COMPETants dalam rangka MOACO dengan menggunakan dua pheromone dan heuristik matriks, yang dikumpulkan oleh jumlah tertimbang dan tiga bobot  $\Lambda = \{0, 0,5, 1\}$ . Dengan demikian, COMPETants adalah pendekatan single-koloni dalam framework yang dibuat, dan itu sangat mudah untuk menentukan varian formulasi ini dengan jumlah sewenang-wenang koloni. Demi kesederhanaan, dalam framework MOACO, jumlah semut sama dibagi dengan jumlah bobot, dan jumlah semut per berat badan tidak berubah selama menjalankan, seperti dalam proposal awal; Namun demikian, fitur ini dapat ditambahkan ke kerangka sebagai komponen tambahan. Selain itu, setidaknya dalam TSP bi-obyektif, tidak pernah menggabungkan informasi heuristik, seperti dalam COMPETants asli, menyebabkan kualitas buruk di tengah depan Pareto. Oleh karena itu, kita tidak menangani khusus informasi heuristik untuk semut menggunakan  $\lambda = 0,5$  tapi matriks heuristik dikumpulkan dengan cara yang sama seperti matriks pheromone. Akhirnya, setiap matriks pheromone diperbarui dengan  $N^{\text{upd}}$  menghasilkan solusi terbaik untuk tujuan yang sesuai, yang merupakan *best-of-objective* pada pengaturan untuk PheromoneUpdate dalam framework MOACO.

#### e) Pareto Ant Colony Optimization

Pareto Ant Colony Optimization (P-ACO) seperti yang dijelaskan Manuel Lopez [21] menggunakan beberapa matriks



pheromone, satu untuk setiap tujuan digabungkan dengan cara jumlah tertimbang. Seperti di BicriterionAnt dan MACS, berat yang berbeda terkait dengan setiap semut. Selain itu, matriks pheromone diperbarui dengan solusi yang terbaik dan kedua terbaik untuk masing-masing tujuan, yang pada dasarnya adalah sama dengan memperbarui metode yang digunakan oleh COMPETants, dan dalam framework MOACO sesuai dengan best-of-objective pengaturan untuk Pheromone-Update dan  $N^{upd} = 2$ . Dalam masalah yang dipecahkan, tidak ada definisi yang jelas tentang informasi heuristik untuk masing-masing tujuan, oleh karenanya, Manuel Lopez menggunakan matriks heuristik tunggal. Namun, RAPP menggunakan beberapa matriks heuristik, satu untuk setiap tujuan, yang digabungkan dengan cara yang sama sebagai matriks pheromone. Dalam penelitian Lopez sebelumnya, menunjukkan bahwa ada perbedaan penting antara menggunakan satu atau dua matriks heuristik di P-ACO untuk TSP bi-objektif [23]. Secara khusus, Lopez mengamati bahwa matriks heuristik tunggal mengarah ke Pareto depan sangat sempit dan karenanya, framework MOACO menggunakan beberapa matriks heuristik ketika instantiating P-ACO. Meskipun demikian, framework MOACO bisa meniru kedua varian tersebut [21].

f) mACO Varian 1 (mACO-1)

Pada penelitian Manuel Lopez [21], terdapat usulan empat alternatif untuk desain algoritma MOACO. Varian pertama, mACO-1, digambarkan menggunakan beberapa koloni: satu koloni per obyektif, dan koloni tambahan yang membangun solusi dengan menggabungkan matriks pheromone dari dua koloni lainnya menyusul agregasi acak. Setiap koloni menggunakan informasi heuristik dari tujuan yang sesuai, sedangkan koloni ekstra agregat matriks heuristik. Dalam framework MOACO ini, Manuel Lopez merumuskan koloni tunggal dengan beberapa pheromone dan heuristik matriks dikumpulkan menggunakan tiga bobot  $\Lambda = \{0, 0,5, 1\}$ . Pada mACO-1 asli, informasi pheromone dari setiap koloni diperbarui dengan solusi terbaik yang dihasilkan oleh koloni tujuan

yang sesuai dari koloni yang sama. Selain itu, algoritma solusi terbaik yang dihasilkan oleh koloni ekstra untuk setiap tujuan, dan menggunakan koloni untuk memperbarui matriks pheromone yang sesuai dari dua koloni lainnya. Karena setiap koloni dalam bahasa aslinya sesuai dengan berat yang berbeda  $\lambda$  dalam formulasi yang dihasilkan, metode pembaruan ini sesuai dalam framework untuk best-of-objective -per-berat dengan  $N^{\text{upd}} = 1$ .

g) mACO Varian 2 (mACO-2)

Varian kedua (mACO-2) memiliki sedikit perbedaan dengan mACO-1 yaitu hanya pada agregasi pheromone. Dalam mACO-2, menurut Manuel Lopez [21] matriks pheromone dikumpulkan dengan menjumlahkan matriks pheromone dari masing-masing tujuan. Ketika algoritma ACO yang mendasari adalah skala invarian, seperti AS, MMAS dan ACS, ini setara dengan weight  $\lambda = 0,5$ . Manuel Lopez mengemukakan penelitian sebelumnya pada TSP bi-tujuan menunjukkan bahwa tidak ada perbedaan yang signifikan dalam kualitas antara mACO-1 dan mACO-2 [23].

h) mACO Varian 3 (mACO-3)

Varian ketiga seperti yang dijelaskan oleh Manuel Lopez menggunakan matriks pheromone tunggal, yang diperbarui dengan menggunakan semua solusi nondominated (baik dari iterasi-terbaik arsip atau arsip terbaik-jadi-jauh) [21]. Pendekatan mACO-3 menekankan bahwa setiap nilai pheromone yang terkait dengan komponen solusi, diperbarui paling banyak sekali, meskipun banyak solusi menampungnya. Hal ini berbeda dengan algoritma lain seperti MOAQ, MACS, dan BicriterionAnt, yang menggunakan update pheromone nondominated. Namun, Manuel Lopez tidak menemukan keuntungan apapun dalam persyaratan khusus ini, karena kerangka ini tidak memasukkannya demi kesederhanaan. Informasi heuristik juga matriks tunggal. Dalam masalah di mana ada informasi heuristik tersedia untuk setiap tujuan, ini dikumpulkan sebelum menjalankan ke dalam matriks heuristik tunggal [21].

i) mACO Varian 4 (mACO-4)

Pada varian terakhir yang dijelaskan oleh Manuel Lopez [21], ada satu pheromone matriks per objektif, dan ini selalu digabungkan

dengan cara acak sama seperti untuk m ACO-1, yang dalam prakteknya sesuai dengan agregasi acak dengan *single weight*  $\lambda = 0,5$ . Namun, ada matriks heuristik tunggal, seperti dalam mACO-3. Akhirnya, setiap matriks pheromone diperbarui dengan solusi terbaik untuk setiap tujuan, yang merupakan definisi best-of-objective pembaruan pheromone digunakan dalam kerangka kerja ini [21].

j) New Design Alternatif

Manuel Lopez [21] menjelaskan bahwa meskipun jumlah algoritma MOACO berbeda yang diusulkan dalam literatur, ada sejumlah yang jauh lebih besar dari kombinasi yang belum dijelajahi komponen algoritmik tersebut. Secara khusus, varian multi-koloni dapat didefinisikan untuk semua algoritma. Agregasi acak hanya diuji dengan  $\lambda = 0,5$  sejauh ini. Selain itu, semua algoritma Ulasan atas penggunaan semua bobot yang tersedia di setiap iterasi (allweights-per-iterasi). Dalam penelitian yang dilakukan Manuel Lopez untuk bi-tujuan QAP, mengusulkan bahwa semua semut menggunakan bobot yang sama dalam satu iterasi, dan berat berikutnya dalam urutan di iterasi nex (satu-berat-per-iterasi) [21]. Oleh karena itu, dapat dengan mudah membangun varian baru yang paling algoritma pada Tabel II. varian baru seperti dapat dianggap algoritma MOACO baru.

Namun, harapannya bahwa banyak dari mereka tidak akan menyebabkan setiap terobosan signifikan. Di sisi lain, masalah tertentu yang sedang diselesaikan dapat mempengaruhi pilihan desain terbaik. Dalam kasus apapun, itu akan menjadi upaya besar untuk menguji mereka satu per satu untuk menemukan desain terbaik. Sebaliknya, Manuel Lopez mengusulkan untuk secara otomatis menemukan desain terbaik untuk masalah tertentu dengan (offline) konfigurasi otomatis dari *framework* MOACO [21]. Ada beberapa metode otomatis untuk konfigurasi offline algoritma optimasi single-objektif. Dalam penelitian Manuel Lopez [21] telah menambah Iterated F-Ras (I / F-Ras), untuk kasus multi-objective dengan menggunakan ukuran kualitas unary. Pada bagian berikutnya, kita menerapkan pendekatan ini untuk kerangka dijelaskan dalam makalah ini dan melakukan analisis rinci dari hasil.

### 3. Automatic Configuration MOACO Framework

Ada begitu banyak algoritma yang terkait dengan algoritma MOACO ini, sehingga perlu dieksploitasi konfigurasi algoritma secara otomatis untuk mendapatkan varian MOACO yang efisien sesuai konteksnya. Dengan demikian diperlukan sebuah cara untuk menjadikannya otomatis. Tujuan otomatis (offline) konfigurasi ini adalah untuk menemukan algoritma yang terbaik sesuai pengaturan parameter dari algoritma-algoritma yang ada untuk memecahkan masalah pada kasus-kasus yang ada. Framework yang dibuat Manuel Lopez bertujuan untuk menemukan Instansiasi yang baik dari metaheuristik sebuah ruang besar desain potensial. Selain itu, memperpanjang Ide untuk pertama kalinya dengan konteks multi-objective [21]. Framework tersebut menunjukkan bahwa dalam bagian ini terdapat konfigurasi otomatis dari framework MOACO yang fleksibel sehingga memungkinkan untuk menemukan hasil algoritma MOACO lebih baik untuk BTSP daripada yang tersedia dalam literatur.

Manuel Lopez juga mempelajari strategi konfigurasi yang berbeda, dan memeriksa hasil beberapa konfigurasi otomatis independen yang berjalan [21]. Metode konfigurasi otomatis yang digunakan seperti yang dijelaskan Manuel Lopez di sini adalah I / F-Ras, yang merupakan state-of-the-art otomatis metode konfigurasi yang mampu menangani terus menerus, kategoris dan parameter bersyarat. Manuel Lopez menggunakan pelaksanaan I / F-Ras yang disediakan oleh paket Irace. I / F-Ras bergantian antara menghasilkan konfigurasi calon baru dan melakukan balapan untuk membuang yang berkinerja terburuk. Dalam perlombaan, konfigurasi calon dijalankan pada satu Misalnya pada suatu waktu. I / F-Ras menggunakan tes Friedman diikuti dengan analisis post-test untuk membuang konfigurasi setiap kali ada bukti statistik yang cukup bahwa mereka melakukan lebih buruk daripada yang lain. Ketika hanya sejumlah kecil dari konfigurasi tetap dalam lomba, lomba berhenti. Sebuah ras baru dimulai dengan konfigurasi terbaik sebelumnya ditemukan dan dengan calon baru konfigurasi yang

dihasilkan dari konfigurasi terbaik menggunakan model probabilistik sederhana. Proses konfigurasi otomatis akan berhenti setelah mencapai angka (iterasi) maksimum yang diberikan (nomor dari berjalan atau batas waktu). Versi I / F-Ras saat ini dirancang untuk single objective masalah optimasi, dan karenanya, membutuhkan kriteria evaluasi yang memberikan nilai tunggal untuk masing-masing berjalan dari konfigurasi. Manuel Lopez menerapkan I / F-Race ke multi-objective dengan konteks cara pengukuran kualitas unary, yang menetapkan nilai kualitas tunggal untuk satu set nondominated. Manuel Lopez menguji dua langkah unary sebagai kriteria evaluasi I / F-Race, yaitu hyper volume1 dan (aditif) epsilon mengukur. Hyper volume adalah volume objective ruang lemah didominasi oleh satu set nondominated dan dibatasi oleh titik referensi yang ketat didominasi oleh semua Pareto optimal vektor obyektif. Semakin besar hypervolume tersebut, maka makin baik sesuai set nondominated. Aditif epsilon ukuran memberikan nilai minimum yang harus dikurangi dari semua tujuan dari satu set nondominated sehingga lemah mendominasi set referensi. Set referensi ini biasanya set nondominated dari semua solusi yang dikenal. Sebuah epsilon kecil Nilai ukuran lebih disukai [21].

Framework ini menggunakan algoritma ACO sebagai dasarnya, dan Manuel Lopez menggunakan MMAS sebagai ditetapkan untuk TSP [21]. Secara khusus, Manuel Lopez menggunakan pengaturan default yang diuraikan pada Tabel III, dengan menetapkan  $\Delta\tau = 1$  untuk Jumlah pheromone yang disimpan oleh semut, dan tidak menggunakan daftar calon untuk pembangunan solusi [21]. Manuel lopez juga menggabungkan pseudo-acak Aturan pilihan aksi ACS, yang memungkinkan untuk greedier konstruksi solusi. Parameter  $q_0$  mengontrol keserakahan tindakan aturan pilihan pseudo-random. Nilai dari  $q_0 = 0$  menonaktifkan dan beralih kembali ke MMAS asli. Itu framework MOACO diimplementasikan dalam C, dan mendasari algoritma ACO berasal dari ACOTSP. Kode ini dikompilasi dengan gcc [21].

Manuel Lopez [21] membagi upaya konfigurasi dalam tiga tahap:

- Pertama, mempelajari apakah mungkin untuk secara otomatis menemukan desain MOACO baru. Oleh karena itu, hanya mengkonfigurasi komponen multi-objective dalam framework MOACO, dan membandingkan hasilnya dengan algoritma yang dijelaskan di dalam literatur.
- Kedua, menilai berapa banyak perbaikan mungkin dicapai dengan mengkonfigurasi pengaturan parameter algoritma ACO yang mendasari. Melakukannya dengan otomatis konfigurasi pengaturan algoritma ACO dari salah satu konfigurasi yang ditemukan pada tahap sebelumnya.
- Akhirnya, bertanya apakah konfigurasi dua-tahap ini memiliki pendekatan lebih baik daripada mengkonfigurasi semua komponen dan parameter sekaligus menggunakan upaya konfigurasi setara untuk dua tahap sebelumnya.

Parameter	Value
$N^*$	$24 \cdot \lfloor n/100 \rfloor$
$\rho$	$0.02 (n < 300), 0.05 (n \geq 300)$
$q_0$	0
$\alpha$	1
$\beta$	2

*Figure 7 Default parameter setting dari algoritma ACO (MMAS)  
(sumber: Manuel Lopez, 2012)*

#### a) Konfigurasi Komponen Multiple Objective

Dari percobaan pertama yang dilakukan Manuel Lopez, tujuannya adalah untuk menemukan desain baru, mudah-mudahan lebih baik dari algoritma MOACO untuk BTSP. Manuel Lopez mencari desain dalam mode otomatis dengan mengkonfigurasi komponen multi-objective framework MOACO, sambil tetap tetap mendasari pengaturan parameter algoritma ACO. Pengaturan algoritma ACO tetap diberikan dalam Tabel III dan domain konfigurasi komponen multi-objective dijelaskan pada Tabel IV. Budget konfigurasi diatur 1000 set berjalan dari framework MOACO. Manuel lopez melakukan lima kali pengulangan independen dari proses konfigurasi menggunakan hypervolume

sebagai kriteria evaluasi, dan lima pengulangan menggunakan ukuran epsilon unary [21].

Component	Domain	Constraint
$\tau$	{ single, multiple }	
$\eta$	{ single, multiple }	
Aggregation	{ weighted sum, weighted product, random }	only if multiple $\tau$ or $\eta$
$N^{weights}$	{ 2, 3, $N^2/3, N^2/2, N^2$ }	(per colony)
NextWeight	{ one weight per iteration, all weights per iteration }	
PheromoneUpdate	{ ND, BO, BOW* }	* only with $N^{col} = 1$
$N^{upd}$	{ 1, 2, 5, 10 }	
$N^{col}$	{ 1, 2, 3, 5, 10 }	
MultiColonyWeights	{ disjoint, overlapping }	only if $N^{col} > 1$
MultiColonyUpdate	{ origin, region }	only if $N^{col} > 1$

Figure 8 Domain komponen algoritma MOACO

(sumber: Manuel Lopez, 2012)

Konfigurasi tertentu yang ditemukan diberikan dalam Tabel V dan VI untuk hypervolume dan ukuran epsilon masing-masing. Semua sepuluh konfigurasi menemukan penggunaan beberapa matriks heuristik, sejumlah besar koloni, update dengan wilayah dan agregasi oleh produk tertimbang. Kebanyakan konfigurasi menggunakan beberapa matriks pheromone diperbarui oleh best-of-objective (BO), dan satu-weight-per-iterasi (1wpi).

Component	Run 1	Run 2	Run 3	Run 4	Run 5
$\tau$	single	multiple	multiple	multiple	multiple
$\eta$	multiple	multiple	multiple	multiple	multiple
Aggregation	w. product	w. product	w. product	w. product	w. product
$N^{weights}$	$N^2/3$	3	$N^2/2$	$N^2/3$	2
NextWeight	1wpi	1wpi	1wpi	1wpi	1wpi
PheromoneUpdate	ND	BO	BO	BO	BO
$N^{upd}$	1	2	2	2	1
$N^{col}$	10	5	10	10	10
MultiColonyWeights	disjoint	overlap	disjoint	overlap	overlap
MultiColonyUpdate	region	region	region	region	region

Figure 9 Konfigurasi dengan lima tuning runs algoritma MOACO

(sumber: Manuel Lopez, 2012)

Component	Run 1	Run 2	Run 3	Run 4	Run 5
$ V $	multiple	single	multiple	multiple	multiple
$ V $	multiple	multiple	multiple	multiple	multiple
Aggregation Weights	w-product	w-product	w-product	w-product	w-product
Nac/Weight	2	$N^2/3$	$N^2/2$	$N^2$	$N^2$
PhasomonyUpdate	1wpi	1wpi	1wpi	1wpi	1wpi
$N^{upd}$	BD	ND	BCI	BCI	BCI
$N^{col}$	1	1	2	2	2
$N^{col}$	10	10	5	10	10
MultiColonyWeights	overlap	overlap	overlap	overlap	overlap
MultiColonyUpdate	nigon	nigon	nigon	nigon	nigon

Figure 10 Konfigurasi dengan komponen algoritma MOACO  
(sumber: Manuel Lopez, 2012)

Manuel Lopez menerapkan 10 konfigurasi ini untuk contoh uji, dan melakukan 15 berjalan independen satu sama konfigurasi dengan biji acak yang berbeda. Dan membuat perbandingan, dengan mengevaluasi algoritma MOACO dari literature. Manuel Lopez juga mengevaluasi kualitas set nondominated ditemukan di tes ini berjalan dengan cara baik hypervolume dan ukuran epsilon. Hasilnya mengarah pada kesimpulan yang sama secara independen dari ukuran kualitas yang digunakan dalam analisis. Selain itu, tidak ada perbedaan yang signifikan menurut uji tanda, antara konfigurasi saat ditemukan menggunakan hypervolume sebagai kriteria evaluasi I / F-Ras dan yang ditemukan saat menggunakan ukuran epsilon. Dengan demikian metode ini mampu menemukan desain baru yang melampaui-the-art-of arus dalam literatur MOACO. Tidak mengherankan, desain baru ini mirip dengan varian multi-koloni BicriterionAnt. Namun, itu termasuk perbedaan yang signifikan, seperti penggunaan pembaruan terbaik-ofobjective, nilai kecil  $N^{upd}$ , lebih dari satu semut per weight, dan satu-weight-per-iterasi (1wpi) [21].

b) Konfigurasi underlying ACO Algorithm

Dalam percobaan berikutnya, Manuel Lopez [21] secara sembarangan memilih salah satu konfigurasi yang ditemukan ketika mengkonfigurasi komponen framework multi-objective MOACO, melaksanakan konfigurasi otomatis dari pengaturan dari algoritma ACO yang mendasari. Domain pengaturan algoritma ACO dijelaskan pada Tabel VII. Bukannya mengkonfigurasi langsung jumlah semut per koloni  $N^a$ , mengkonfigurasi parameter  $a_f$  pengganti yang



mendefinisikan jumlah semut dalam ketergantungan dari ukuran contoh  $n$ , yaitu,  $N^a = 6 \cdot a_f \cdot \lfloor n / 100 \rfloor$ . Kemudian kalikan faktor ini dengan enam agar dapat menentukan  $N^{\text{weights}}$  baik sebagai  $N^a / 3$  atau  $N^a / 2$  dalam percobaan kemudian [21].

Manuel Lopez melakukan lima kali pengulangan independen dari proses konfigurasi menggunakan hypervolume sebagai kriteria evaluasi, dan setiap pengulangan dihentikan setelah 1000 berjalan dari framework MOACO. Sebagai perbandingan, digunakan juga mengkonfigurasi cara pengaturan algoritma ACO dari versi multi-koloni BicriterionAnt, yang ditemukan menjadi algoritma MOACO terbaik dari literatur di bagian sebelumnya [21].

Component	Domain
$a_f$	$\{1, 2, \dots, 25\}$ where $N^a = 6 \cdot a_f \cdot \lfloor n / 100 \rfloor$
$\rho$	$(0.01, 0.99)$
$q_0$	$(0.95, 0.99)$ or $q_0 = 0$
$\alpha$	$(0, 6)$
$\beta$	$(0, 6)$

Figure 11 Domain parameter setting pada algoritma ACO (MMAS) (sumber: Manuel Lopez, 2012)

Muncul pertanyaan apakah tidak akan lebih baik untuk mengkonfigurasi kedua komponen multi-tujuan dan pengaturan parameter ACO dari framework MOACO sekaligus, bukannya berturut-turut seperti yang dilakukan di atas. Untuk menjawab pertanyaan ini, kita melaksanakan konfigurasi otomatis dari semua parameter dari kerangka MOACO sekaligus, dengan domain yang dijelaskan dalam Tabel IV dan VII, dan dengan budget 2000 percobaan, yaitu setara dengan usaha gabungan dihabiskan di dua-tahap proses konfigurasi otomatis sebelumnya. Konfigurasi yang dihasilkan diberikan sebagai bahan tambahan. Hasil yang disajikan di bawah ini menunjukkan bahwa ada perbedaan yang signifikan dalam mendukung secara otomatis mengkonfigurasi semua parameter sekaligus.

Untuk perbandingan konfigurasi yang ditemukan di berbagai prosedur konfigurasi otomatis, dilakukan 15 berjalan independen

satu sama konfigurasi pada contoh uji. Manuel Lopez mengevaluasi hasil dengan menggunakan ukuran hypervolume, dan sesuai boxplots. Plot menunjukkan bahwa secara otomatis mengkonfigurasi pengaturan algoritma ACO dari BicriterionAnt sangat meningkatkan kinerja, dan untuk kasus dengan 500 kota, itu melampaui kualitas konfigurasi MOACO dengan pengaturan algoritma standar ACO. Meskipun demikian, dengan secara otomatis mengkonfigurasi pengaturan dari algoritma ACO yang mendasari framework MOACO, peningkatan hypervolume bahkan lebih luar biasa, jelas mengalahkan konfigurasi terbaik dari BicriterionAnt. Plot tidak menunjukkan perbedaan yang jelas antara otomatis mengkonfigurasi framework MOACO menggunakan pendekatan dua tahap atau dengan mengkonfigurasi semua parameter sekaligus. Namun, konfigurasi diperoleh dengan pendekatan dua tahap mendapatkan hypervolume lebih rendah, yaitu lebih buruk, daripada sepenuhnya dikonfigurasi dalam 80% dari berjalan. Tanda-test menegaskan bahwa perbedaan ini secara statistik signifikan [21].

#### 4. MOACO Framework dan Local Search

Penelitian yang dilakukan Manuel Lopez [21] memberikan hasil yang disajikan di atas secara meyakinkan menunjukkan bahwa konfigurasi otomatis mengarah ke desain yang lebih baik dari algoritma MOACO daripada yang disajikan dalam literatur. Meskipun sejumlah besar pekerjaan pada algoritma MOACO tidak mempertimbangkan penggunaan pencarian lokal, diketahui bahwa, pada algoritma BTSP, MOACO dengan pencarian lokal jauh mengungguli tanpa pencarian lokal. Oleh karena itu, Manuel Lopez mengulangi konfigurasi otomatis hibridisasi framework MOACO dan pencarian lokal [21].

##### a) Experimental setup

Pencarian lokal yang digunakan di sini adalah perbaikan berulang algoritma didasarkan pada lingkungan 2-exchange (2-opt). Pencarian lokal diterapkan untuk setiap solusi yang dibangun oleh semut. The BTSP dikonversi menjadi TSP single-objective dengan cara jumlah agregasi tertimbang dari dua matriks jarak. Jika semut

menggunakan vektor bobot untuk membangun solusi, pencarian lokal selanjutnya akan menggunakan vektor bobot yang sama. Jika tidak, jika semut tidak menggunakan beban untuk konstruksi solusi, weight masih ditugaskan untuk setiap semut berikut pengaturannya tetapi mereka hanya digunakan oleh pencarian lokal. Pencarian lokal juga memanfaatkan teknik speed-up standar untuk TSP, misalnya daftar memerintahkan tepi calon ukuran 20 dihitung untuk setiap vektor berat. Semut menggunakan daftar calon yang berbeda dari ukuran 20 untuk konstruksi, yang diperoleh dengan menyortir tepi sesuai dengan dominasi Peringkat [21].

Parameter sisa framework MOACO sama dengan pengaturan sebelumnya tanpa pencarian lokal (Tabel III, IV, dan VII) kecuali bahwa: (i) nilai default  $\rho$  adalah 0,2; (ii) jumlah semut tidak tergantung pada ukuran contoh dan sekarang dihitung sebagai  $N^a = 6 \cdot a_f$ ; dan (iii) batas waktu masing-masing berjalan sekarang  $4 \cdot (n / 100)^2$ . Sebagai contoh pelatihan, kami menghasilkan 10 BTSP seragam acak contoh Euclidean untuk setiap  $n = \{500, 600, 700, 800, 1000\}$  node (60 kasus total). Perbandingan dan evaluasi konfigurasi diperoleh menggunakan satu set yang berbeda dari 15 contoh uji, 3 contoh setiap ukuran  $n$  [21].

b) Automatic configuration MOACO+ls

Manuel Lopez [21] melaksanakan empat setup konfigurasi otomatis yang terpisah dari MOACO + ls kerangka kerja untuk BTSP sebagai berikut:

- 1) Melakukan konfigurasi komponen multi-objective framework MOACO dan menggunakan pengaturan default untuk algoritma ACO yang mendasarinya.
- 2) Melakukan konfigurasi pengaturan dari algoritma ACO yang mendasari dan menggunakan salah satu desain MOACO ditemukan pada langkah sebelumnya.
- 3) Melakukan konfigurasi semua pengaturan dari framework MOACO sekaligus menggunakan dua kali budget eksperimen seperti untuk dua langkah sebelumnya. Tujuan dari konfigurasi ini adalah untuk menyelidiki apakah pendekatan konfigurasi dua

tahap lebih atau kurang efektif daripada mengkonfigurasi semua parameter sekaligus.

- 4) Melakukan konfigurasi pengaturan algoritma ACO dari BicriterionAnt + ls untuk perbandingan dengan hasil lainnya.

Manuel Lopez melakukan lima kali pengulangan setiap pengaturan konfigurasi otomatis, dan budget dari setiap pengulangan diatur ke 1000 percobaan (2000 untuk mengkonfigurasi semua pengaturan sekaligus). Alat konfigurasi menggunakan ukuran hypervolume sebagai kriteria evaluasi. Hal ini cukup untuk mengatakan di sini bahwa konfigurasi menunjukkan variabilitas yang lebih besar dari pengaturan, dijelaskan oleh fakta bahwa pencarian lokal mengurangi efek parameter lainnya.

Konfigurasi ditemukan secara otomatis dievaluasi pada contoh uji dengan melakukan 15 berjalan independen satu sama konfigurasi, dan menghitung ukuran hypervolume set nondominated yang dihasilkan. Hasil yang didapatkan mengikuti kesimpulan yang diuraikan dalam bagian sebelumnya. Pertama, kinerja BicriterionAnt sangat meningkat dengan mengkonfigurasi pengaturan algoritma ACO nya. Kedua, meskipun peningkatan ini, secara otomatis dikonfigurasi BicriterionAnt tidak mengungguli desain otomatis menemukan kerangka MOACO dengan pengaturan algoritma standar ACO. Selain itu, secara otomatis mengkonfigurasi pengaturan algoritma ACO dari suatu desain yang baik mengarah untuk lebih meningkatkan kualitas, jelas mengalahkan hasil terbaik dari BicriterionAnt. Akhirnya, meskipun boxplots tidak secara visual menunjukkan perbedaan yang jelas antara proses konfigurasi dua tahap vs sepenuhnya mengkonfigurasi MOACO + ls kerangka sekaligus, tanda-test menunjukkan bahwa ada probabilitas signifikan lebih besar, sekitar 0,84, untuk memperoleh hypervolume lebih besar dengan konfigurasi diperoleh dengan metode yang terakhir. Hal ini sesuai dengan kesimpulan yang dicapai saat otomatis mengkonfigurasi kerangka MOACO tanpa pencarian lokal [21].

Hasil ini jelas menunjukkan bahwa telah ditemukan desain baru dari algoritma MOACO yang mengungguli keadaan

sebelumnya seni dalam literatur MOACO untuk BTSP. Lebih penting lagi, Manuel Lopez telah menemukan desain ini secara semi-otomatis, di mana sebagian besar upaya telah dihabiskan pada definisi pilihan desain alternatif, meninggalkan tugas mencari desain yang benar untuk karenanya, berisi alat-alat otomatis dan. Fakta bahwa beberapa pilihan desain tidak pernah muncul dalam beberapa konfigurasi memberikan indikasi yang jelas bahwa mereka tidak berkontribusi terhadap kinerja algoritma. Di sisi lain, variabilitas desain menunjukkan bahwa ada beberapa alternatif desain yang menghasilkan kualitas yang sama dari hasil [21].

Dengan demikian, penelitian ini akan menggunakan framework yang ditemukan oleh Manuel Lopez. Penelitian ini akan mencoba melakukan implementasi framework ini untuk memecahkan masalah pada objek mitigasi bencana.

### **3.3. Mitigasi Bencana**

Mitigasi bencana merupakan manajemen bencana secara umum. Mitigasi memiliki tiga bagian yaitu, proses pendeteksian (*detection*), proses penanganan saat terjadi bencana (*respons*), dan penanganan pasca bencana (*recovery*) [24] [25]. Pendeteksian membahas menyangkut bagaimana secara dini bencana dalam dapat dideteksi sehingga dapat membangun *awardness* dan kesiap-siagaan dalam menghadapi bencana alam [26]. System deteksi ini biasa dikenal dengan istilah *early warning system* [27] [24].

Sedangkan respon adalah bagian dimana bencana dihadapi dan dilakukan penganggulangan. Kegiatan yang dilakukan seperti melakukan evakuasi dan penyelamatan korban bencana melalui jalur evakuasi yang dipilih. Cara menentukan jalur evakuasi haruslah dipilih jalur paling cepat untuk dilalui [28]. Salah satu cara yang dapat digunakan untuk menentukan jalur evakuasi adalah dengan menggunakan algoritma komputasi seperti algoritma ant colony [11] [29].

Proses terakhir dalam mitigasi bencana adalah recovery. Proses ini adalah proses pemulihan pasca bencana alam seperti bagaimana penyaluran logistic ke shelter pengungsian dimana para

korban bencana ditangani sementara [30] [31]. Pada proses ini, dapat dilakukan penentuan shelter pengungsi secara cepat dan dapat dijangkau dalam waktu yang cepat [32] [33]. Factor lain yang harus dipertimbangkan adalah pemilihan jalur distribusi pengiriman bantuan yang dapat dikirim secara cepat dan tepat di shelter pengungsian [33] [34] [35].

Menurut Iman Satyarno (2016), tujuan dari mitigasi adalah untuk mengurangi jumlah korban jiwa dan mengurangi kerugian harta benda [36]. Manajemen kebencanaan dapat dibagi menjadi (dua) bagian besar yaitu sebelum (*berofe*) dan Sesudah (*after*) bencana terjadi. Pada bagian *before* kegiatan yang dapat dilakukan adalah mitigasi dan *reduction* sedangkan bagian *after* adalah kegiatan untuk memberi respond an melakukan rehabilitasi dan *reconstruction* [36]. Pada penelitian ini focus utama yang dipelajari adalah proses pada bagian mitigasi bencana saja. Bagian lainnya dapat dibahas secara terpisah.

## BAB IV METODOLOGI PENELITIAN

Bab ini membahas tentang metode yang digunakan dalam usulan penelitian ini. Teknik pengumpulan data yang digunakan dijelaskan pada bagian pertama. Selanjutnya analisis kebutuhan system yang akan digunakan dibahas dalam penelitian diberikan pada bagian dua. Analisis Perencanaan pada bagian ketiga. Waktu dan jadwal penelitian diberikan pada bagian empat. Kerangka penelitian dalam penelitian ini dijelaskan pada bagian kelima.

### 4.1. Tahapan Penelitian

#### 4.1.1. Pengumpulan Data

Pada awal penelitian ini, dimulai dengan pengumpulan data.

##### *a. Studi Literatur*

Pada bagian ini, dilakukan studi literature agar didapatkan informasi yang berkaitan dengan smart environment yang membahas tentang mitigasi bencana dan algoritma multiple objective ant colony optimization (MO ACO). Informasi yang terkait dikumpulkan dari berbagai sumber ilmiah seperti jurnal, buku (*textbook*), disertasi, thesis, laporan kebencanaan dan sumber-sumber lainnya baik cetak maupun *electronic*.

##### *b. Observasi/Studi Lapangan*

Pada bagian ini, data dikumpulkan dengan cara peneliti melakukan pengamatan langsung di lapangan. Daerah yang diamati adalah daerah yang rawan bencana seperti desa tinoor, desa kinilow, kelurahan kakaskasen I dan kakaskasen II. Pengamatan juga dilakukan pada Badan Penanggulangan Bencana Daerah (BPBD) kota Tomohon tentang penentuan titik shelter pengungsi dan jalur distribusi logistic bencana. Selain itu, peneliti juga melakukan observasi pada Pos Pemantau Gunung Lokon untuk melakukan pengamatan aktivitas gunung serta mempelajari system mitigasi yang dimiliki saat terjadi bencana letusan, mempelajari aliran informasi yang dikeluarkan mulai dari proses mitigasi sampai informasi tersebut sampai kepada masyarakat.

### *c. Interview/Wawancara*

Interview atau wawancara ini, dilakukan terhadap kepala Badan penanggulangan bencana daerah dan juga para staf pegawai yang ada termasuk TIM SAR yang dimiliki. Selain itu, wawancara dilakukan terhadap para staf yang ada di pos pemantau aktivitas gunung Lokon serta masyarakat yang ada di daerah rawan bencana. Informasi yang didapatkan berhubungan dengan system mitigasi bencana terkait dengan deteksi, respond dan recovery saat terjadi bencana letusan.

#### **4.1.2. Analisis Kebutuhan Sistem**

Pada tahapan ini, analisis dilakukan dengan dengan menetapkan indicator-indikator yang akan digunakan untuk analisis smart enviromentnya. Selain itu, ditetapkan beberapa asumsi yang akan digunakan dalam penggunaan algoritma multiple objective ant colony optimization (MO-ACO). Beberapa parameter awal ditetapkan sesuai kebutuhan yang akan analisis menggunakan algoritma tersebut. Pada bagian ini juga akan di analisis kebutuhan perhitungan system dan langkah-langkah yang akan digunakan dalam perhitungan. Pada bagian ini juga akan dicoba melihat kebutuhan yang spesifik dalam mitigasi bencana termasuk cara-cara yang dipakai pemerintah dalam melakukan mitigasi dan penanganan bencana.

#### **4.1.3. Perancangan Sistem**

Pada tahapan ini, dilakukan perancangan model graph daerah rawan bencana berdasarkan peta kebencanaan yang ada. Pada tahapan ini, dilihat daerah-daerah yang akan terpapar letusan gunung lokon. Kemudian akan dilakukan asumsi titik-titik yang akan dilalui dan oleh pengungsi maupun TIM SAR bahkan pengiriman jalur logistic bencana. Hal ini diperkukan untuk memudahkan perhitungan dan analisis menggunakan algoritma MO-ACO. Pada tahapan ini, akan dilakukan pengkonstrusian pheromone dan penetapan parameter yang akan digunakan dalam algoritma MO-ACO. Penetapan dimulai dengan penggambaran titik-titik bencana dalam



mapping kemudian akan dikonversi menjadi model graph, sehingga mudah untuk dilakukan perhitungan.

#### **4.1.4. Pengembangan Sistem**

Pada tahapan ini, dilakukan proses analisis menggunakan algoritma Multiple Objective Ant Colony Optimization, analisis ini menggunakan persamaan-persamaan yang ada pada algoritma MOACO tersebut. Analisis ini juga dilakukan sesuai tahapan analisis dan perhitungan yang ada pada algoritma tersebut.

Dalam tahapan pengembangan ini, akan dilakukan perhitungan menggunakan algoritma MO-ACO dengan tiga langkah sederhana yakni :

- 1) Konstruksi solusi yang akan dicapai
- 2) Local search
- 3) Update pheromone

Setelah tiga langkah tersebut dilakukan, maka langkah berikutnya adalah melakukan assessment terhadap strategy mitigasi bencana. Hasilnya dapat berupa model smart environment yang dapat diterapkan dalam penanganan bencana alam.

Pada tahap Analisis ini, system akan dikembangkan dengan membuat perhitungan dan proses perhitungan tersebut akan menggunakan alat bantu untuk perhitungan algoritma yakni berupa alat dan bahan dalam bentuk *hardware* dan *software* dengan spesifikasi sebagai berikut :

- ❑ Alat
  - Perangkat Keras
    - Intel Dual Core 2x1,7 Ghz dan memory DDR 2 : 2 GB
  - Perangkat Lunak
    - Perhitungan menggunakan Ms.excel 2013 dan matlab 2014

#### **4.1.5. Pengujian Sistem**

Pada tahapan ini, pengujian dilakukan dengan menguji hasil analisis yang telah didapatkan apakah telah sesuai dan baik. Tahapan ini akan ditarik kesimpulan apakah algoritma MOACO tersebut baik untuk di implementasikan dalam system mitigasi bencana atau tidak.

Apakah algoritma MOACO secara objektif dapat meningkatkan efisiensi biaya logistic dan distribusi penyaluran logistic bencana kepada para pengungsi atau tidak. Hasil perhitungan yang didapatkan sebelumnya akan digunakan lebih lanjut untuk melakukan assessment terhadap strategy smart envoroment mitigasi bencana. Hasilnya diharapkan dapat diterapkan dengan lebih efisien dan efektif.

Hasil perhitungan komputasi ini juga akan dilakukan validasi dengan menggunakan *benchmark* pada peta mitigasi yang tersedia. Peta mitigasi yang dimaksud adalah peta mitigasi yang dikeluarkan oleh Badan Penanggulangan Bencana Daerah Kota Tomohon. Proses validasi dimaksudkan apakah hasil perhitungan komputasi tersebut dapat terjadi optimasi pada proses mitigasinya sehingga hasilnya dapat menjadi lebih akurat.

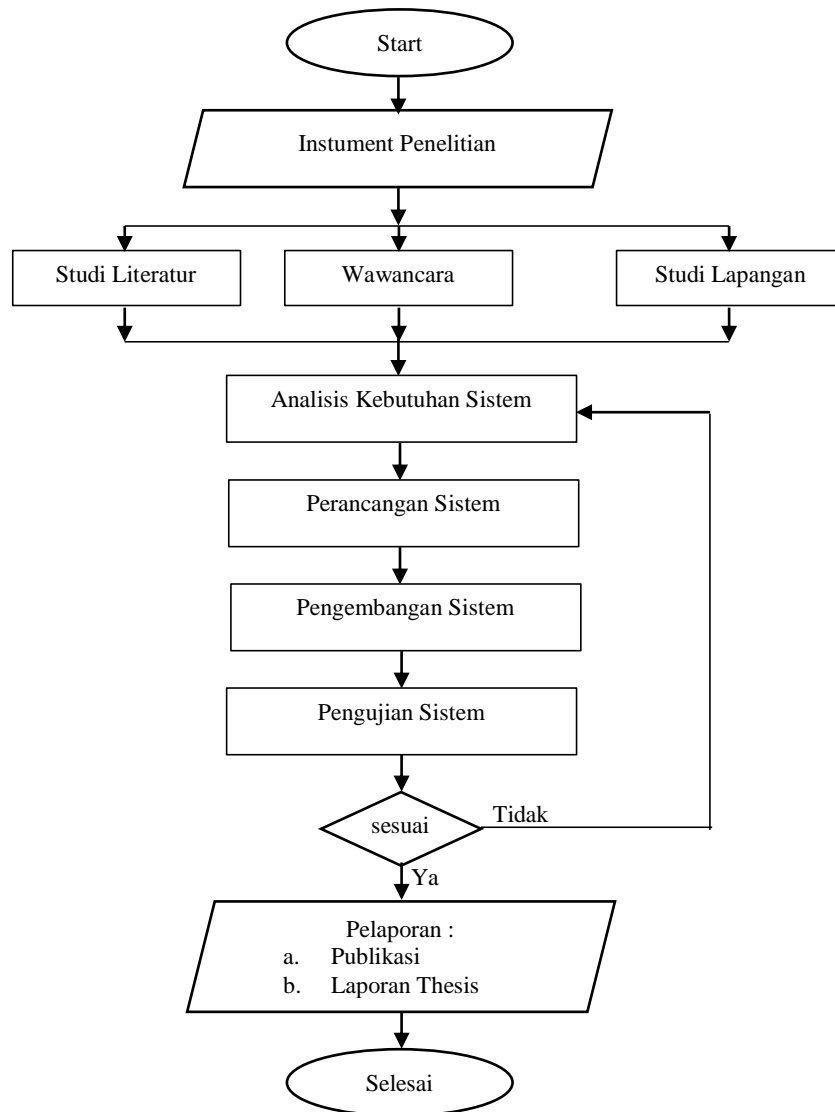
#### **4.1.6. Pelaporan**

Pada tahapan ini, peneliti melakukan perampungan laporan yang ditulis dalam bentuk format pelaporan thesis. Kemudian hasil penulisan akan dipublikasikan pada jurnal Nasional terakreditasi maupun jurnal international.

#### **4.2. Diagram Alir Penelitian**

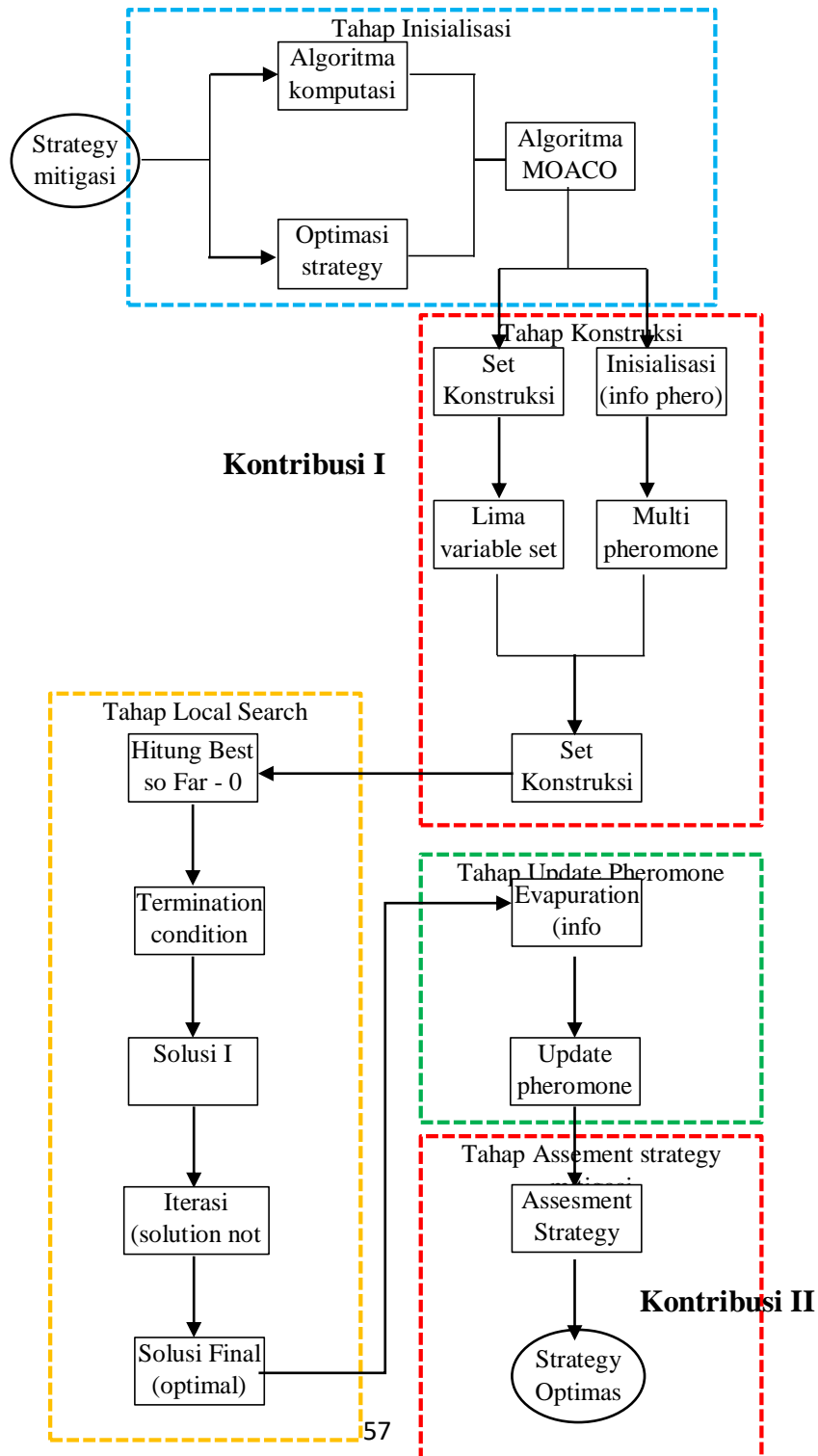
Secara garis besar, diagram alir metodologi penelitian ini dilakukan dan disajikan sebagai berikut :

Figure 12 Diagram alir penelitian



### 4.3. Rancangan Skema Metode Penelitian

Figure 13 Rancangan skema metode penelitian



#### 4.4. Jadwal Penelitian

Penelitian ini akan dilakukan dengan jadwal penelitian sebagai berikut :

*Table 2 Jadwal penelitian*

No.	Kegiatan	Juni 2016				Juli 2016				Agustus 2016				September 2016			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengumpulan Data																
	a.Studi Literatur	√	√	√	√	√	√										
	b.Studi Lapangan	√	√	√	√	√	√										
2	Analisis Kebutuhan Sistem				√	√	√	√									
3	Perancangan Sistem							√	√	√	√						
4	Pengembangan Sistem										√	√	√	√			
5	Pengujian Sistem												√	√	√	√	
6	Pelaporan																
	a. Publikasi														√	√	√
	b. Laporan Thesis														√	√	√
7	Ujian Pendaran																√

## BAB V PEMBAHASAN

### 5.1. Peta Kota Tomohon

#### 5.1.1. Profil Tomohon Utara

Tomohon bagian utara adalah bagian dari Kota Tomohon, salah satu wilayah administrative di Sulawesi utara. Wilayah ini merupakan area yang sangat subur dengan suhu udara  $16^{\circ}\text{C} - 28^{\circ}\text{C}$ . Wilayah ini juga didominasi oleh perkebunan sayur mayur, sawah, peternakan (sapi, Babi dan Ayam) dan perkebunan bunga hias serta area wisata. Wilayah ini sebagian besar sekitar 70% berada dibawah kaki gunung Lokon sebelah utara dan Timur. Empat perkampungan penduduk yaitu Desa Tinoor, Kinilow, Desa Kakaskasen I, dan Kakaskesen II berada pada zona letusan gunung berapi Lokon. 2 (dua) desa yakni desa Tinoor dan Kinilow masuk pada zona berbahaya dan masuk pada prioritas penangan bencana dan evakuasi.

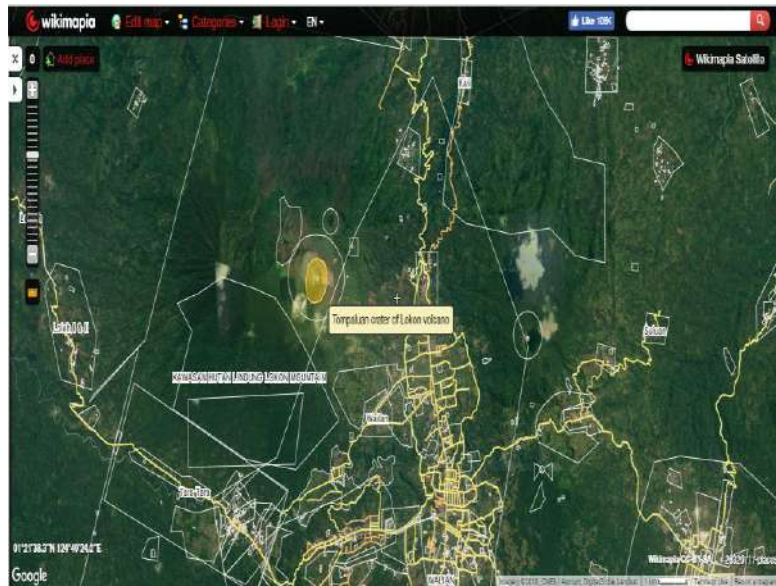


Figure 14 Peta area kota Tomohon Utara (sumber : wikimapia.org)

### 5.1.2 Profil Desa Tinoor I&II

Desa Tinoor terdiri dari Tinoor I dan Tinoor II yang letaknya saling berdekatan. Awalnya kedua desa tersebut merupakan sebuah desa saja, tetapi dimekarkan menjadi dua desa sehingga menjadi desa Tinoor I dan desa Tinoor II. Kedua desa ini merupakan batas sebelah barat area administrative pemerintah kota Tomohon dengan kabupaten Minahasa. Seluruh wilayahnya berada di lereng atau kaki gunung Lokon. Desa Tinoor masuk menjadi area berbahaya dan menjadi area prioritas penanganan bencana Gunung Lokon. Sebagian besar sekitar 80% areanya merupakan area perkebunan dan pertanian. Daerahnya secara topologi adalah dataran tinggi dengan medan yang sulit dilalui. Jalan penghubungnya merupakan jalan desa dengan banyak tikungan tajam dan berbahaya. Peta area kedua desa Tinoor tersebut dapat dilihat pada peta berikut ini.

Untuk mengantisipasi bencana letusan gunung Lokon, diberi beberapa petunjuk jalur evakuasi. Petunjuk jalur evakuasi ini dibuat hanya dengan perkiraan saja tanpa perhitungan yang memadai sehingga hasilnya masih belum akurat.

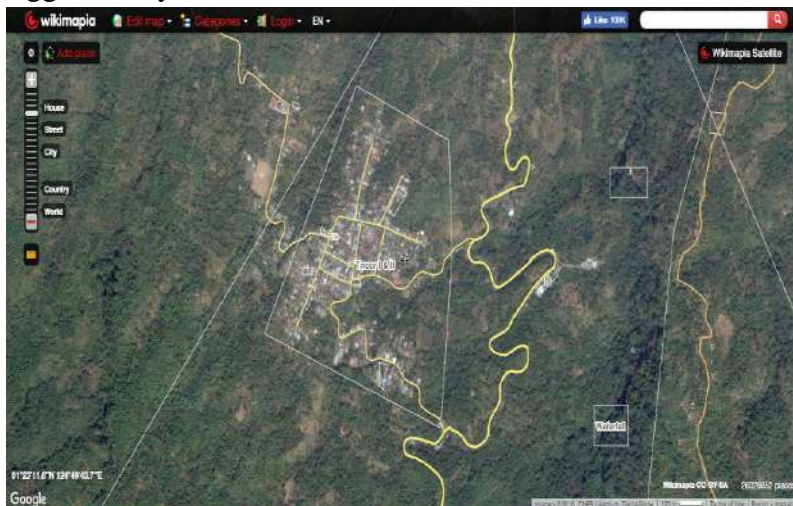


Figure 15 Peta area desa Tinoor I dan II (sumber : wikimapia.org)

### 5.1.3 Profil Desa Kinilow

Desa kinilow termasuk salah satu desa yang berada di wilayah Tomohon Utara. Sebagian besar areanya berada di lereng atau kaki

gunung Lokon yang terjal dan berbahaya. Areanya masuk pada kategori zona berbahaya karena letaknya yang sangat dekat dengan pusat letusan atau kawah gunung Lokon. Daerah ini merupakan prioritas penanganan bencana letusan gunung Lokon. Daerah ini selain pemukiman warga, juga merupakan area perkebunan, pertanian dan juga peternakan warga yang subur dan hijau.

Untuk mengantisipasi bencana letusan gunung Lokon, diberi beberapa petunjuk jalur evakuasi. Petunjuk jalur evakuasi ini sama seperti desa Tinooor dibuat hanya dengan perkiraan saja tanpa perhitungan yang memadai sehingga hasilnya masih belum akurat.

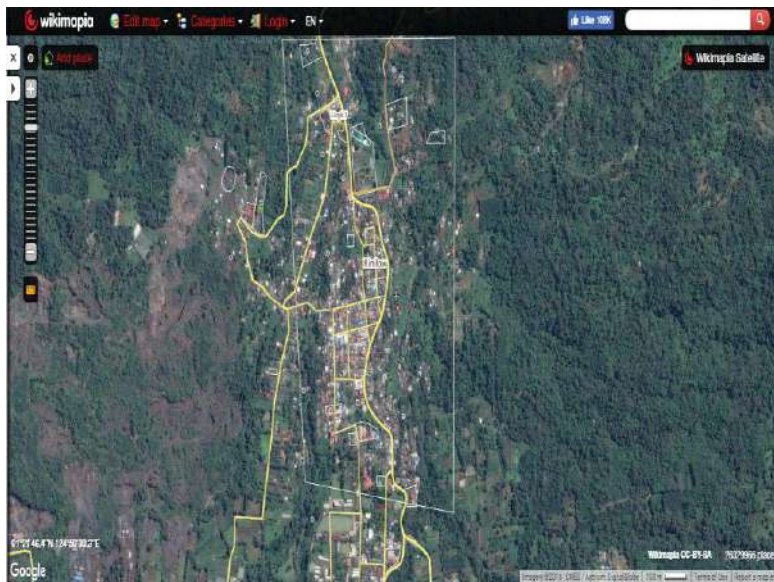


Figure 16 Peta area Desa Kinilow (sumber : wikimapia.org)

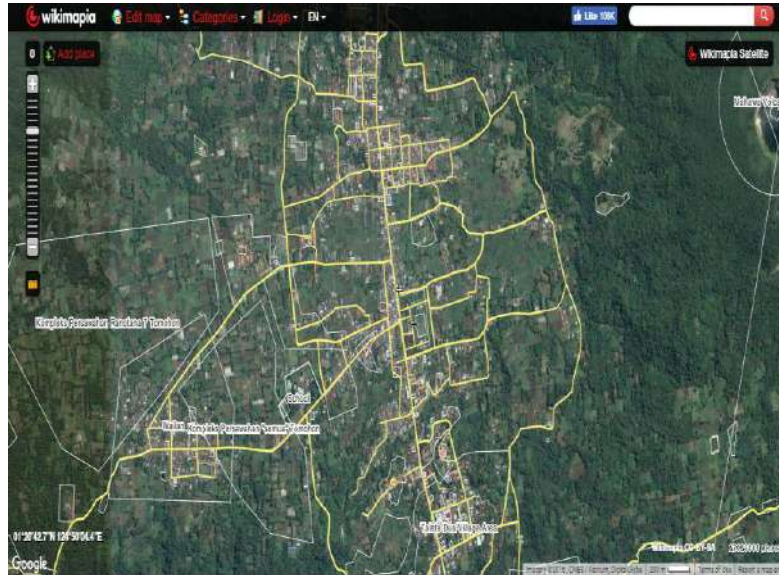
#### 5.1.4 Profil Kota Tomohon Kakaskasen I, II, dan III

Kelurahan Kakaskasen I, II, dan III merupakan wilayah kota Tomohon bagian utara. Wilayah ini berada di area landai di kaki gunung Lokon. Sebagian besar wilayahnya merupakan area persahawan yang subur, pertanian, peternakan dan perkebunan bunga hias yang khas kota Tomohon. Pemukiman warga termasuk pemukiman yang cukup padat dengan populasi penduduk yang cukup besar. Daerah ini merupakan area letusan dengan jarak minimal 5 km dari kawah letusan. Area ini merupakan area yang



dapat digunakan sebagai area aman jika terjadi letusan. Juga dapat digunakan sebagai titik aman untuk menempatkan shelter pengungsi yang dapat digunakan sewaktu terjadi bencana. Daerah yang terbilang cukup aman adalah daerah kakaskasen II dan III.

Untuk mengantisipasi bencana letusan gunung Lokon, diberi beberapa petunjuk jalur evakuasi. Petunjuk jalur evakuasi ini sama seperti desa Tinoor dibuat hanya dengan perkiraan saja tanpa perhitungan yang memadai sehingga hasilnya masih belum akurat.



*Figure 17 Peta area kelurahan kakaskasen I,II,III (sumber : wikimapia.org)*

## **5.2. Peta Kebencanaan Gunung Lokon**

Berdasarkan catatan BNPD Kota Tomohon, Gunung Lokon memiliki karakteristik tersendiri. Berdasarkan catatan sejarah erupsi, pada umumnya erupsi Gunung Lokon berupa erupsi abu disertai lontaran batu pijar, kadang-kadang mengeluarkan lava pijar dan awan panas. Erupsinya berlangsung beberapa hari. Bila terjadi erupsi besar, maka bahaya utama erupsi G. Lokon atau bahaya primer (bahaya langsung akibat erupsi) adalah luncuran awan panas, lontaran piroklastik (bom vulkanik, lapili, pasir dan abu) dan mungkin aliran lava. Sedangkan bahaya sekunder (bahaya tidak

langsung dari erupsi) adalah lahar hujan yang terjadi setelah erupsi apabila turun hujan lebat di sekitar puncak. Gejala G. Lokon menjelang meletus pada umumnya berupa menebalnya asap kawah, tingginya berfluktuasi antara 400 - 600 m di atas bibir kawah. Makin lama asap tersebut makin menebal dan suatu saat akan berubah warna menjadi kelabu, yang menandakan bahwa material berukuran abu sudah terbawa keluar [37].

Gunung Lokon memiliki 1(satu) kawah yang diberi nama Tompaluan dan memiliki 2 (dua) puncak yaitu puncak Lokon dan puncak Empung. Letak secara geografis yaitu puncak lokon :  $01^{\circ}21,5'LU$  dan  $124^{\circ}47,5'BT$  dan puncak Empung:  $01^{\circ}22'LU$  dan  $124^{\circ}47,5'BT$ . Ketinggian puncaknya adalah Lokon 1579,5m dpl, Empung 1340m dpl dan Tompaluan 1140m dpl dan terletak di wilayah administrative Kota Tomohon Sulawesi Utara.

Gunung lokon memiliki perioda erupsi sebelum tahun 1800 selang waktu erupsi sangat lama (400 tahun), tetapi sesudah 1949 menunjukkan peningkatan frekuensi yang sangat tajam, selang waktu erupsi bervariasi antara 1 - 4 tahun, rata-rata 3 tahun. Erupsi besar terakhir terjadi tahun 1991 sisanya bersakala kecil dan sedang [12].

Geomorfologi kompleks Lokon-Empung dibagi menjadi 4 satuan, yaitu Satuan Geomorfologi Kerucut, Kawah, Punggungan Rendah & Bergelombang serta Geomorfologi Dataran. Satuan Geomorfologi kerucut menempati daerah sekitar tubuh G. Lokon dan G. Empung. G. Lokon mempunyai puncak yang datar tanpa kawah dengan kemiringan antara  $30^{\circ}$  -  $70^{\circ}$ . Sedangkan G. Empung mempunyai dua buah kerucut terpancung, yaitu Empung Muda di bagian barat dan Empung Tua di bagian timur, yang masing-masing mempunyai kawah di puncaknya. Pola aliran sungainya adalah radier dengan lembah yang berbentuk "V", dengan tebing yang relatif curam. Vegetasi penutupnya berupa alang-alang yang cukup tebal. Satuan geomorfologi kawah terdapat di kawah Tompaluan dan Kawah Empung. Kawah Tompaluan merupakan kawah paling aktif saat ini yang terbentuk sekitar tahun 1828, sedangkan Kawah Empung tidak aktif lagi. Satuan geomorfologi Perbukitan Rendah & Bergelombang menempati sebagian besar lerang kompleks Lokon -

Empung, merupakan morfologi yang membentuk punggung yang landai serta bergelombang, sudut lerengnya  $< 30^{\circ}$ . Batuan pembentuknya berupa piroklastik dan lava. Sebagian besar daerah ini dimanfaatkan sebagai lahan pertanian. Satuan geomorfologi Dataran menempati sepanjang pantai bagian utara, sekitar daerah Malalayang dan dataran tinggi Kakaskasen pada elevasi lebih kurang 800 m. Umumnya digunakan sebagai daerah persawahan dan perkebunan kelapa [37].

Jenis gempa yang terekam di G. Lokon terdiri dari gempa Vulkanik A, Vulkanik B, Tektonik, Letusan dan Hembusan. Sebaran episenter gempa vulkanik untuk periode Mei – Juni 2008 terkonsentrasi di kawah Tompaluan dengan radius kurang dari 1 km. Disamping itu terdapat beberapa gempa dengan episenter berada di di barat laut kawah. Hiposenter gempa-gempa tersebut berkedalaman 50 m hingga 10000 m di bawah Kawah Tompaluan. Berdasarkan hiposenternya, terdapat dua kelompok gempa yang dominan yaitu masing-masing pada kedalaman 0,05 - 1 km di bawah Kawah Tompaluan dan 2 – 3 km di bawah Kawah Tompaluan. Sepanjang pengamatan (Mei – Juni 2008) tidak terlihat migrasi hiposenter dari dalam ke dangkal [37].

Pemantauan G. Lokon dilakukan secara menerus baik visual maupun kegempaannya dari Pos Pengamatan yang terletak di Kakaskasen, Kota Tomohon. Pengamatan visual dilakukan dengan mengamati cuaca dan aktivitas G. Lokon, meliputi tinggi, warna dan tekanan asap yang keluar dari kawah G. Lokon. Pemantauan kegempaan dilakukan dengan menempatkan 4 sensor seismometer di tubuh G. Lokon [37].

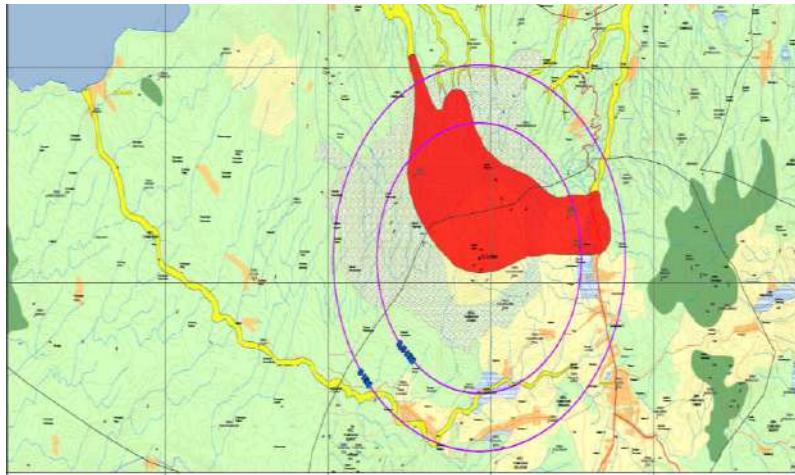


Figure 18 Peta Bencana Gunung Lokon (sumber : BNPB RI, 2011)

Gunung Lokon memiliki kawasan rawan bencana sebanyak 2 (dua) area seperti terlihat pada gambar diatas yaitu area I terbentang antara 5 km dari puncak gunung dan area II yang terbentang antara 3,5 km dari puncak gunung dan masuk kategori area berbahaya dan menjadi prioritas penanganan bencana.

### 5.3. Perhitungan

Pada penelitian ini, peneliti menggunakan dan mengadopsi framework MOACO algorithm dari Manuel Lopez, dengan parameter tetapan sebagai berikut :

Table 3 Parameter tetapan setting MOACO algorithm

Parameter	Value
$N^a$	$24 \cdot \lceil \eta / 100 \rceil$
P	$0.02 (\eta < 300), 0.05 (\eta \geq 300)$
$q_0$	0
A	1
B	2

Kemudian untuk melakukan perhitungan, peneliti melakukan konfigurasi komponen yang digunakan sesuai dengan setting pada framework MOACO algorithm secara otomatis yaitu :

Table 4 Komponen pada framework MOACO algorithm secara otomatis

Komponen	Domain	Constraint
$[\tau]$	{single, multiple}	
$[\eta]$	{single, multiple}	
Aggregation	$\begin{cases} \text{weighted sum} \\ \text{weighted product} \\ \text{random} \end{cases}$	Only if multiple $\tau$ or $\eta$
$N^{\text{weight}}$	{2,3, $N^a/3$ , $N^a/2$ , $N^a$ }	(per colony)
NextWeight	$\begin{cases} \text{one weight per iteration,} \\ \text{all weight per iteration} \end{cases}$	
pheromoneUpdate	$\begin{cases} ND, \\ BO, \\ BOW^* \end{cases}$	* only with $N^{\text{col}} = 1$
$N^{\text{upd}}$	{1, 2, 5, 10}	
$N^{\text{col}}$	{1, 2, 3, 5, 10}	
MultiColonyWeights	{disjoint, overlapping}	Only if $N^{\text{col}} > 1$
MultiColonyUpdate	{origin, region}	Only if $N^{\text{col}} > 1$

#### Factor Batasan (variable) Perhitungan

Dalam melakukan perhitungan, peneliti menggunakan faktor-faktor batasan (*variable*) yang digunakan dalam perhitungan menggunakan algoritma MOACO sebagai berikut :

Variable yang digunakan dalam penelitian ini merupakan asumsi-asumsi yang digunakan dalam melakukan perhitungan. Asumsi- asumsi ini kemudian disebut kriteria dalam perhitungan. Kriteria- kriteria tersebut antara lain :

- 1) Panjang Rute, di ukur dalam satuan km (kilometer). Variable ini merupakan jarak antara node-node yang dituju dan menyatakan jarak yang harus ditempuh dalam melintasi node-node tersebut. Algoritma MOACO ini akan menghitung jarak terbaik dan terdekat yang harus dilalui sampai pada titik aman evakuasi yakni

shelter pengungsi. Jarak ini merupakan salah satu variable yang penting dalam penggunaan algoritma MOACO ini.

- 2) Kecepatan, di ukur dalam satuan km/jam (kilometer per jam). Variable ini menyatakan kecepatan rata-rata yang dapat ditempuh dalam menembuh jarak dari node-node tersebut. Kecepatan ini merupakan kecepatan rata-rata yang bersifat akumulasi tanpa melihat pertimbangan lain seperti tikungan dan banyaknya hambatan yang dihadapi. Algoritma ini mencoba untuk memilih kecepatan yang baik dalam memilih jalur evakuasi yang harus dilalui.
- 3) Tikungan, di ukur dalam satuan jumlah tikungan. Ini menyatakan berapa banyak rintangan yang dihadapi melalui suatu jalur. Makin banyak tikungan yang terdapat dalam sebuah jalur, maka kecepatan dalam melakukan evakuasi akan berkurang dan mempengaruhi kinerja proses evakuasi pada umumnya. Algoritma MOACO ini akan memilih jalur yang tikungannya lebih sedikit untuk menentukan jalur evakuasi yang harus dilalui.
- 4) Kepadatan, di ukur dari banyaknya kepadatan dari sebuah jalur. Variable ini menghitung banyaknya orang dan tingkat kepadatan yang dapat mengakibatkan kemacetan dari sebuah jalur yang mungkin dilalui, sehingga dapat menghambat kecepatan dan efektifitas dalam melakukan proses evakuasi. Algoritma MOACO ini akan memilih jalur yang tingkat kepadatannya rendah sehingga kecepatan dan efektivitas evakuasi akan menjadi lebih baik.
- 5) Titik aman, di ukur dari banyaknya titik aman yang akan dilalui pada sebuah jalur. Titik aman ini didapatkan dari peta bencana yang tersedia kemudian dapat ditentukan jalur yang tingkat titik amannya banyak. Algoritma MOACO ini memasukan titik aman dalam perhitungan agar jalur yang terpilih harus benar-benar aman untuk dilalui. Algoritma MOACO ini akan memilih jalur yang titik amannya banyak sehingga aman untuk dilewati dari bencananya sendiri. Hal ini dimaksudkan agar pada proses evakuasi, tidak terjadi jatuhnya korban pada saat proses evakuasi dan proses evakuasinya sendiri dapat dengan mudah diketahui

oleh masyarakat terutama yang terkena dampak letusan gunung berapi.

Untuk menghitung determinan agregasi informasi heuristic, digunakan persamaan yang digunakan dalam algoritma MOACO framework dengan tiga pilihan persamaan yaitu : Weighted sum :

$$\tau_{ij} = (1 - \lambda)\tau_{ij}^1 + \lambda\tau_{ij}^2 \text{ and } \eta_{ij} = (1 - \lambda)\eta_{ij}^1 + \lambda\eta_{ij}^2 \quad (7)$$

▪ Weighted product :

$$\tau_{ij} = (\tau_{ij}^1)^{(1-\lambda)} \cdot (\tau_{ij}^2)^\lambda \text{ and } \eta_{ij} = (\eta_{ij}^1)^{(1-\lambda)} \cdot (\eta_{ij}^2)^\lambda. \quad (8)$$

▪ Random :

Pada tiap langkah pengembangan, diberikan angka random yang seragam  $U(0, 1)$ , semut dipilih pertama pada dua matriks jika  $U(0, 1) < 1 - \lambda$ ; kemudian dipilih matriks berikutnya.

Pada ketiga metode agregasi yang dijelaskan diatas, terdapat weight  $\lambda$  yang mengakibatkan bias agregasi pada satu objective atau lainnya. Set of weights  $\Lambda$  didefinisikan dengan komponen  $N^{weights}$  and  $NextWeight$ .

$N^{weights}$  dan  $NextWeight$ . The set weight dif definisikan dalam interval  $[0, 1]$  sebagai :

$$\Lambda = \{\lambda_i = 1 - (i - 1)/N^{weights} - 1\}, \quad i = 1, \dots, N^{weights} \quad (7)$$

Where  $N^{weights} = |\Lambda|$  are the parameters of the framework.

Persamaan diatas digunakan pada solusi determinan parsial yang akan mengikuti kemungkinan untuk menghitung pheromone dengan menggunakan persamaan berikut :

$$p_{Sj}^k = \frac{\left[ \prod_{q=1}^Q (\tau_{S_{qj}}^q)^{\lambda q} \right]^\beta}{\left( \left[ \prod_{q=1}^Q (\tau_{S_{ql}}^q)^{\lambda q} \right]^\alpha \cdot \left[ \prod_{q=1}^Q (\eta_{S_{ql}}^q)^{\lambda q} \right]^\beta \right)} \quad \text{if } j \in N_S^k \quad (8)$$

Untuk menghitung, digunakan framework MOACO secara otomatis [21]. Framework ini bekerja dengan menggunakan opsi variasi yang tersedia mendapatkan hasil yang optimal.

Pada bagian ini, kami mengimplementasikan algoritma MOACO untuk mendapatkan hasil terbaik. Data yang dihitung merupakan data yang kompleks. Untuk memudahkan maka dibagi

menjadi empat segmen sehingga memudahkan perhitungan optimasi. Diharapkan pembagian segmen ini dapat memfasilitasi perhitungan sehingga mendapatkan hasil terbaik. Untuk mengeksekusi kebutuhan waktu, digunakan metode CPU Time. Hal ini diperlukan untuk menjelaskan gap antara To execute the time required we used the CPU time method. weighted sum of objective value dan ideal value. Gap ini kemudian akan disebut sebagai  $V_{moy}$ , dengan menggunakan persamaan :

$$V_{moy} = \frac{\sum_{i=0}^{p-1} \lambda_i (f_i - f_{i*})}{\sum_{i=0}^{p-1} \lambda_i} \quad (9)$$

Dimana :

$\rho$  = Jumlah Objectives ( $p = 7$ ).

$\lambda_i$  = objective dengan prioritas  $i$  weight =  $(p - i + 1)$ .

$f_i$  = objective dengan prioritas  $i$  value.

$f_{i*}$  = prioritas objective  $i$  ideal value, diperoleh dengan melakukan minimalisasi objective ini di bawah batasan kaku

Sedangkan untuk menjelaskan penyempurnaan solusi, digunakan persamaan dibawah ini :

$$\%Im = \frac{V_{moy}(Sol.Insiatif) - V_{moy}(Sol.Finale)}{V_{moy}(Sol.Insiatif)} \quad (10)$$

Dimana ,  $\%Im$  = Initial solution percentage improvement

## 5.4. Hasil

### 5.4.1 Segmen optimasi

Untuk melakukan optimasi pada area bencana gunung lokon, maka area tersebut dibagi menjadi empat segmen untuk memudahkan perhitungan. Gambar dibawah ini dapat menjelaskan empat segmen tersebut.



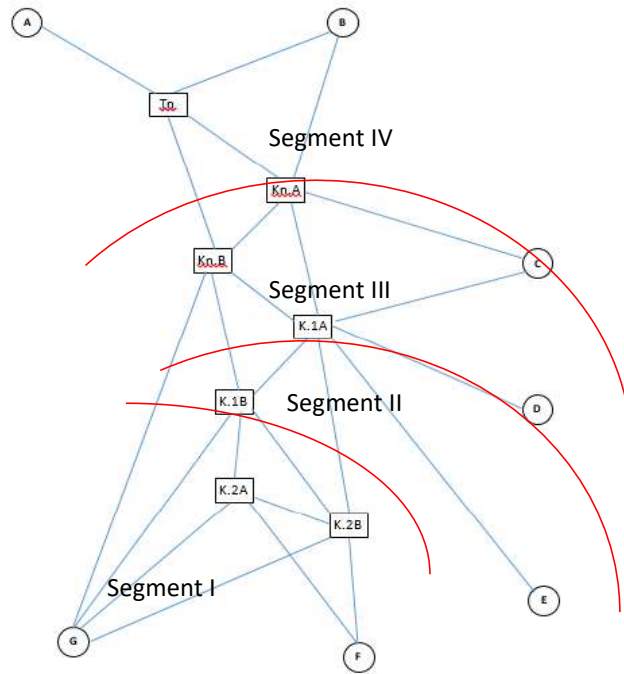


Figure 19 Pembagian segmen optimasi area bencana

#### 5.4.2 Perhitungan Titik Shelter terbaik

Untuk menghitung titik shelter terbaik dengan algoritma MOACO, digunakan perhitungan pada ke empat segmen dengan ke 5 (lima) kriteria seperti yang ditampilkan pada table 3 dibawah ini. Ke lima kriteria tersebut adalah kenyataan sebenarnya yang hitung secara cepat dan tepat.

Table 5 Contains Components yang digunakan

Jalur segmen	Jarak	Kecepatan	Tikungan	Kepadatan	Titik Aman
Segment I	8	5.33	6	9	8
Segment II	12	4.33	8	10	7
Segment III	11	4.08	9	9	9
Segment IV	13	3.75	12	8	10

Dengan melakukan lima kali running untuk melakukan experiment dan mendapatkan nilai fix  $\alpha$  value. Nilai value  $\alpha$  adalah 1 dan merupakan nilai terbaik. Kemudian dilakukan seleksi terhadap nilai  $\beta$  value sebagaimana yang didefinisikan. Setelah itu, dilakuak 5 (lima) kali run pada algoritma tersebut dan dapat didefinisikan  $\beta$  value sebagai nilai yang optimal dan dapat digunakan sebagai parameter dalam prosedur yang digunakan dalam experiment ini. Nilai  $\beta$  value dapat dilihat pada table dibawah ini.

*Table 6 Parameter  $\beta$  value titik Shelter*

Run	B					
	0.01	0.9	1	2	3	5
1	1.33	1.56	1.73	1.61	1.72	1.82
2	1.53	1.29	1.12	1.11	1.38	1.18
3	1.16	1.18	1.15	1.06	1.29	1.12
4	1.47	1.63	1.54	1.29	1.46	1.69
5	1.65	1.87	1.21	1.36	1.42	1.72
Average	1.43	1.51	1.35	1.29	1.45	1.51
Minimum from average	1.29					
Minimum	1.06					

Kami melakukan setting parameter (tetapan parameter)  $\beta$  value menjadi 0.2. Setting ini didapatkan dari rata-rata minimum value dan rata-rata secara keseluruhan yang didapatkan. Dengan demikian kita memiliki nilai set  $\alpha = 1$  and  $\beta = 2$ , kemudian kita mendefinisikan  $\rho$  value. Dibuat juga 5 (lima) kali runs sehingga didapatkan minimum cost dan juga rata-rata minimum cost. Table dibawah ini menjelaskan parameter  $\rho$  value corresponds.

*Table 7 Parameter  $\rho$  value titik Shelter*

Run	P				
	0.1	0.3	0.6	0.7	0.8
1	0.14	1.29	0.12	1.16	0.22
2	1.18	0.15	0.85	0.19	1.16

3	0.16	0.18	0.76	0.12	0.21
4	0.18	1.15	0.97	0.87	0.15
5	1.12	0.16	0.1	0.74	1.12
Average	0.556	0.586	0.56	0.616	0.572
Minimum from average	0.556				
Minimum	0.1				

Akhirnya didapatkan nilai-nilai parameter terbaik yang akan digunakan dalam perhitungan algoritma. Nilai-nilai tersebut ditampilkan pada table dibawah ini :

*Table 8 Problem parameter titik Shelter*

<b>Parameter</b>	<b>Value</b>
#colony	#objective = 7
# ant	# path
# iteration	700
P	0.6
A	1
B	2

*Table 9 Hasil perhitungan Shelter pada Segmen I*

<b>Category A1</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	1.23	0.06	51.2
P2	1.3	0.08	47.8
P3	1.24	0.05	44.2
P4	1.48	0.06	48.8
P5	1.54	0.07	50.2
P6	1.3	0.04	45.9
Average	1.35	0.06	48.02

*Table 10 Hasil perhitungan Sherter pada Segmen II*

<b>Category A2</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	1.65	0.07	63.4
P2	1.65	0.05	52.1
P3	1.76	0.08	48.2
P4	1.92	0.2	43.8
P5	1.58	0.07	63.2
P6	1.36	0.06	68.7
Average	1.65	0.09	56.57

*Table 11 Hasil perhitungan Sherter pada Segmen III*

<b>Category A3</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	1.34	0.16	62.3
P2	1.72	0.8	61.8
P3	1.54	0.12	53.6
P4	1.28	0.56	65.2
P5	1.85	0.89	54.9
P6	1.25	0.62	59.2
Average	1.50	0.53	59.50

*Table 12 Hasil perhitungan Sherter pada Segmen IV*

<b>Category A4</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	1.7	0.47	65.2
P2	1.65	0.56	54.2
P3	1.45	0.54	58.1
P4	1.86	0.87	62.4
P5	1.36	0.69	52.4
P6	1.47	0.38	55.8

Average	1.58	0.59	58.02
---------	------	------	-------

*Table 13 Rekapitulasi Hasil Algoritma MOACO*

Category	Initial. Sol	Final. Sol	% Im	CPU (s)
	Vmoy	Vmoy		
A1	1.35	0.06	0.96	48.02
A2	1.65	0.09	0.95	56.57
A3	1.50	0.53	0.65	59.50
A4	1.58	0.59	0.63	58.02

Solusi akhir yang didapatkan menggunakan kuantitas dan waktu eksekusi, dibangun sebuah pendekatan yang lebih fisible dalam mendapatkan peta logistic bencana. Untuk mendapatkan solusi yang berkualitas, dilihat secara hati-hati pada bagian jalur terpendek dan seragam dalam sebuah segmen kemudian dilanjutkan ke segmen berikutnya. Bagian segmen pertama merupakan diversifikasi yang diadopsi algoritma ini kemudian digunakan koloni awal dari koloni yang berbeda dengan koloni tersebut untuk mendapatkan solusi. Tujuan yang terkait dengan masing-masing koloni solusi memulai setiap semut dari koloni yang berbeda dari satu koloni ke koloni lain sangat diperhitungkan. Dalam menentukan solusi awal dari koloni, jumlah maksimum iterasi yang dibutuhkan untuk meningkatkan solusi yang tersedia. Jumlah iterasi tidak melebihi seratus iterasi, dan juga berdampak pada pengurangan waktu komputasi. Metode yang diusulkan harus memungkinkan memiliki kecepatan yang baik sehingga memungkinkan untuk perencanaan untuk menjalankan sejumlah besar tes set. Kriteria yang telah ditetapkan terkait dengan koloni tersebut diharapkan mendapatkan solusi yang baik dan tepat.

Dari hasil perhitungan menggunakan algoritma MOACO, dapat dilihat bahwa titik shelter terbaik sebagaimana terlihat pada segmen I oleh 96% merupakan terbesar dengan peringkat pertama, kemudian segmen II sebesar 95% diikuti dengan segmen III sebesar 65% dan terakhir adalah segmen IV sebesar 63%. Hal ini mengisyaratkan bahwa dalam menentukan shelter terbaik harus

dilakukan pada segmen area I dan II karena cukup aman dan dapat dicapai dengan cepat. Sementara segmen III dan IV merupakan area berbahaya sehingga tidak menjadi prioritas. Yang menjadi prioritas adalah segmen I dan alternatifnya adalah segmen II.

Setelah mendapatkan hasil, maka dilakukan update pheromone dengan teknik origin dan region. Untuk mengupdate intensitas pheromone tiap segmen setelah diperbaharui dengan pembaharuan global sehingga maka didapatkan hasil sebagai berikut:

*Table 14 Intensitas pheromone tiap segmen*

RUN	Segmen I	Segmen II	Segmen III	Segmen IV
1	0.009	0.312595	0.1396	0.06795
2	0.02595	0.010	0.254	0.009
3	0.1396	0.1035	0.010	0.010
4	0.02797	0.010	0.010	0.010
5	0.009	0.010	0.358	0.0938

### 5.4.3 Perhitungan Jalur Evakuasi terbaik

Dari area bencana, daerah yang paling dekat dan masuk pada daerah rawan bencana adalah Desa Tinoor. Desa Tinoor sendiri berada di area berbahaya yang letaknya hanya 500 m - 2 km dari pusat letusan atau kawah letusan gunung lokon. Desa ini menjadi prioritas penanganan dan harus mendapatkan prioritas evakuasi saat terjadi letusan.

Secara spesifik dapat di uraikan data desa yang rawan terhadap bencana

*Table 15 Data Lokasi bencana*

Simbol	Area Desa	Jarak ke pusat Letusan	Kategori
A	Desa Tinoor	500m – 2 km	Berbahaya
B	Desa Kinilow	1 km – 2,8 km	Berbahaya
C	Desa Kakaskasen I	2,5 km – 3,5 km	Siaga

D	Desa Kakaskasen II	3,6 km - 4 km	Siaga
E	Rindam Kakaskasen II	4 km – 5 km	Aman

Pada penelitian ini, lokasi asal korban yang akan di evakuasi diasumsikan berada di area prioritas penanganan yaitu Desa Tinoor dan akan di evakuasi melalui pemilihan jalur evakuasi tercepat. Lokasi berikut yang akan dilakukan evakuasi adalah daerah rawan dan berbahaya berikutnya adalah Desa Kinilow yang letaknya berdekatan dengan Desa Kinilow. Kedua desa inilah yang dalam penelitian ini menjadi focus utama untuk dilakukan proses evakuasi.

Untuk menghitung jalur distribusi terbaik dengan algoritma MOACO, digunakan perhitungan pada ke empat segmen dengan ke 5 (lima) kriteria seperti yang ditampilkan pada table 3 dibawah ini. Ke lima kriteria tersebut adalah kenyataan sebenarnya yang hitung secara cepat dan tepat.

*Table 16 Contains Components yang digunakan pada jalur evakuasi*

Jalur segmen	Jarak	Kecepatan	Tikungan	Kepadatan	Titik Aman
Segment I	6	3.6	5	6	3
Segment II	10	3.7	9	9	7
Segment III	9	3.2	6	7	9
Segment IV	10	2.5	7	9	9

Dengan melakukan lima kali running untuk melakukan experiment dan mendapatkan nilai fix  $\alpha$  value. Nilai value  $\alpha$  adalah 1 dan merupakan nilai terbaik. Kemudian dilakukan seleksi terhadap nilai  $\beta$  value sebagaimana yang didefinisikan. Setelah itu, dilakuak 5 (lima) kali run pada algoritma tersebut dan dapat didefinisikan  $\beta$  value sebagai nilai yang optimal dan dapat digunakan sebagai parameter dalam prosedur yang digunakan dalam experiment ini. Nilai  $\beta$  value dapat dilihat pada table dibawah ini.

Table 17 Parameter  $\beta$  value pada jalur evakuasi

Run	B					
	0.01	0.9	1	2	3	5
1	0.65	0.24	0.35	0.44	0.42	0.51
2	0.23	0.47	0.39	0.65	0.36	0.48
3	0.87	0.56	0.47	0.73	0.54	0.59
4	1.31	1.23	1.32	1.38	1.62	1.38
5	1.32	1.29	1.07	1.34	1.44	1.23
Average	0.88	0.76	0.72	0.91	0.88	0.84
Minimum from average	0.72					
Minimum	0.23					

Kami melakukan setting parameter (tetapan parameter)  $\beta$  value menjadi 0.2. Setting ini didapatkan dari rata-rata minimum value dan rata-rata secara keseluruhan yang didapatkan. Dengan demikian kita memiliki nilai set  $\alpha = 1$  and  $\beta = 2$ , kemudian kita mendefinisikan  $\rho$  value. Dibuat juga 5 (lima) kali runs sehingga didapatkan minimum cost dan juga rata-rata minimum cost. Table dibawah ini menjelaskan parameter  $\rho$  value corresponds.

Table 18 Parameter  $\rho$  value pada jalur evakuasi

Run	P				
	0.1	0.3	0.6	0.7	0.8
1	0.65	0.87	0.62	0.76	0.87
2	0.87	0.23	0.66	0.73	0.67
3	0.23	0.65	0.37	0.48	0.65
4	0.42	0.43	0.55	0.62	0.76
5	1.02	0.27	0.28	0.43	0.56
Average	0.638	0.49	0.496	0.604	0.702
Minimum from average	0.49				
Minimum	0.23				



Akhirnya didapatkan nilai-nilai parameter terbaik yang akan digunakan dalam perhitungan algoritma. Nilai-nilai tersebut ditampilkan pada table dibawah ini :

*Table 19 Problem Parameter pada jalur evakuasi*

<b>Parameter</b>	<b>Value</b>
#colony	#objective = 7
# ant	# path
# iteration	700
P	0.6
A	1
B	2

*Table 20 Hasil choosing short path segmen I pada jalur evakuasi*

<b>Category A1</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	0.76	0.09	55.6
P2	0.65	0.2	53.8
P3	0.53	0.08	47.8
P4	0.62	0.07	42.9
P5	0.59	0.1	50.8
P6	0.45	0.08	65.2
Average	0.60	0.10	52.68

*Table 21 Hasil choosing short path segmen II pada jalur evakuasi*

<b>Category A2</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	1.95	0.36	74.2
P2	1.86	0.2	55.2
P3	1.69	0.45	54.9
P4	1.89	0.19	50.6
P5	1.74	0.22	45.9
P6	1.79	0.15	61.3
Average	1.82	0.26	57.02

Table 22 Hasil Choosing Short Path Segmen III pada jalur evakuasi

Category A3	Initial. Vmoy	Sol	Final. Vmoy	Sol	CPU (s)
P1	2.69		0.15		76.3
P2	2.78		0.19		56.9
P3	2.95		0.14		55.8
P4	2.81		0.29		61.4
P5	2.86		0.67		62.8
P6	2.95		0.78		60.2
Average	2.84		0.37		62.23

Table 23 Hasil choosing short path segmen IV pada jalur evakuasi

Category A4	Initial. Vmoy	Sol	Final. Vmoy	Sol	CPU (s)
P1	1.8		0.02		65.2
P2	1.94		0.15		54.2
P3	1.67		0.09		58.1
P4	1.85		0.35		62.4
P5	1.73		0.36		52.4
P6	1.58		0.04		55.8
Average	1.76		0.17		58.02

Table 24 Rekapitulasi Hasil MOACO pada jalur evakuasi

Category	Initial. Sol Vmoy	Final. Sol Vmoy	% Im	CPU (s)
A1	0.60	0.10	0.83	52.68
A2	1.82	0.26	0.86	57.02
A3	2.84	0.37	0.87	62.23
A4	1.76	0.17	0.90	58.02

Solusi akhir segmen optimasi jalur evakuasi yang didapatkan menggunakan kuantitas dan waktu eksekusi, dibangun sebuah pendekatan yang lebih fisible dalam mendapatkan peta jalur evakuasi bencana. Untuk mendapatkan solusi yang berkualitas, dilihat secara

hati-hati pada bagian jalur terpendek dan seragam dalam sebuah segmen kemudian dilanjutkan ke segmen berikutnya. Bagian segmen pertama merupakan diversifikasi yang diadopsi algoritma ini kemudian digunakan koloni awal dari koloni yang berbeda dengan koloni tersebut untuk mendapatkan solusi. Tujuan yang terkait dengan masing-masing koloni solusi memulai setiap semut dari koloni yang berbeda dari satu koloni ke koloni lain sangat diperhitungkan. Dalam menentukan solusi awal dari koloni, jumlah maksimum iterasi yang dibutuhkan untuk meningkatkan solusi yang tersedia. Jumlah iterasi tidak melebihi seratus iterasi, dan juga berdampak pada pengurangan waktu komputasi. Metode yang diusulkan harus memungkinkan memiliki kecepatan yang baik sehingga memungkinkan untuk perencanaan untuk menjalankan sejumlah besar tes set. Kriteria yang telah ditetapkan terkait dengan koloni tersebut diharapkan mendapatkan solusi yang baik dan tepat.

Dari hasil perhitungan menggunakan algoritma MOACO, dapat dilihat bahwa jalur terpendek sebagaimana terlihat pada segmen IV oleh 90% merupakan terbesar dengan peringkat pertama, maka segmen III sebesar 87%, di ikuti segmen II sebesar 86% dan I sebesar 83%. Hal ini mengisyaratkan bahwa dalam melakukan optimasi pemilihan jalur evakuasi harus dilakukan pada segmen area IV dan III karena dapat merupakan area paling berbahaya. Sementara segmen I dan II merupakan area yang relative aman sehingga bukan menjadi prioritas. Yang menjadi prioritas adalah segmen IV dan alternatifnya adalah segmen III.

Setelah mendapatkan hasil, maka dilakukan update pheromone dengan teknik origin dan region. Untuk mengupdate intensitas pheromone tiap segmen setelah diperbaharui dengan pembaharuan global sehingga maka didapatkan hasil sebagai berikut:

*Table 25 Intensitas pheromone tiap segmen*

RUN	Segmen I	Segmen II	Segmen III	Segmen IV
1	0.018	0.412587	0.2399	0.04792
2	0.22592	0.018	0.258	0.018
3	0.3295	0.3039	0.017	0.019

4	0.05792	0.018	0.018	0.018
5	0.0013	0.018	0.259	0.2938

#### 5.4.4 Perhitungan Jalur distribusi terbaik

Untuk menghitung jalur distribusi terbaik dengan algoritma MOACO, digunakan perhitungan pada ke empat segmen dengan ke 5 (lima) kriteria seperti yang ditampilkan pada table 3 dibawah ini. Ke lima kriteria tersebut adalah kenyataan sebenarnya yang hitung secara cepat dan tepat.

*Table 26 Contains components yang digunakan pada jalur logistik*

Jalur segmen	Jarak	Kecepatan	Tikungan	Kepadatan	Titik Aman
Segment I	7	4.33	4	7	4
Segment II	14	3.33	8	10	8
Segment III	11	3.08	4	8	7
Segment IV	12	2.75	2	6	8

Dengan melakukan lima kali running untuk melakukan experiment dan mendapatkan nilai fix  $\alpha$  value. Nilai value  $\alpha$  adalah 1 dan merupakan nilai terbaik. Kemudian dilakukan seleksi terhadap nilai  $\beta$  value sebagaimana yang didefinisikan. Setelah itu, dilakuak 5 (lima) kali run pada algoritma tersebut dan dapat didefinisikan  $\beta$  value sebagai nilai yang optimal dan dapat digunakan sebagai parameter dalam prosedur yang digunakan dalam experiment ini. Nilai  $\beta$  value dapat dilihat pada table dibawah ini.

*Table 27 Parameter  $\beta$  value pada jalur logistik*

Run	B					
	0.01	0.9	1	2	3	5
1	1.13	1.32	1.35	1.16	1.36	1.17
2	1.05	1.29	1.12	1.11	1.38	1.18
3	1.16	1.18	1.15	1.06	1.29	1.12
4	1.14	1.15	1.18	1.07	1.46	1.15
5	1.17	1.02	1.21	1.06	1.42	1.08

Average	1.13	1.19	1.20	1.09	1.38	1.14
Minimum from average	1.09					
Minimum	1.02					

Kami melakukan setting parameter (tetapan parameter)  $\beta$  value menjadi 0.2. Setting ini didapatkan dari rata-rata minimum value dan rata-rata secara keseluruhan yang didapatkan. Dengan demikian kita memiliki nilai set  $\alpha = 1$  and  $\beta = 2$ , kemudian kita mendefinisikan  $\rho$  value. Dibuat juga 5 (lima) kali runs sehingga didapatkan minimum cost dan juga rata-rata minimum cost. Table dibawah ini menjelaskan parameter  $\rho$  value corresponds.

*Table 28 Parameter  $\rho$  value pada jalur logistik*

Run	$\rho$				
	0.1	0.3	0.6	0.7	0.8
1	0.14	1.29	0.12	1.16	0.22
2	1.18	0.15	1.01	0.19	1.16
3	0.16	0.18	1.02	0.12	0.21
4	0.18	1.15	1.03	1.08	0.15
5	1.12	0.16	0.1	0.29	1.12
Average	0.556	0.586	0.656	0.568	0.572
Minimum from average	0.556				
Minimum	0.1				

Akhirnya didapatkan nilai-nilai parameter terbaik yang akan digunakan dalam perhitungan algoritma. Nilai-nilai tersebut ditampilkan pada table dibawah ini :

*Table 29 Problem parameter pada jalur logistik*

<b>Parameter</b>	<b>Value</b>
#colony	#objective = 7
# ant	# path
# iteration	700
P	0.6
A	1
B	2

*Table 30 Hasil choosing short path segmen I pada jalur logistik*

<b>Category A1</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	1.16	0.06	51.2
P2	1.3	0.02	47.8
P3	1.24	0.03	54.2
P4	1.48	0.2	48.8
P5	1.12	0.01	52.4
P6	1.3	0.03	54.5
Average	1.27	0.06	51.48

*Table 31 Hasil choosing short path segmen II pada jalur logistik*

<b>Category A2</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	2.65	0.62	61.7
P2	2.87	0.46	47
P3	2.96	0.33	52.87
P4	2.87	0.39	49.85
P5	2.79	0.72	56.32
P6	2.88	0.68	73.2
Average	2.84	0.53	56.82

Table 32 Hasil choosing short path segmen III pada jalur logistik

<b>Category A3</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	2.34	0.16	62.3
P2	2.67	1.08	61.8
P3	2.88	0.12	53.6
P4	2.54	0.12	65.2
P5	2.74	1.07	54.9
P6	2.9	0.52	59.2
Average	2.68	0.51	59.50

Table 33 Hasil CHOOSING SHORT PATH SEGMENT IV pada jalur logistik

<b>Category A4</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>CPU (s)</b>
P1	2.8	0.05	65.2
P2	2.54	1.13	54.2
P3	2.76	0.07	58.1
P4	2.93	1.02	62.4
P5	2.87	0.23	52.4
P6	2.12	0.08	55.8
Average	2.67	0.43	58.02

Table 34 Hasil Rekapitulasi MOACO pada jalur logistik

<b>Category</b>	<b>Initial. Sol Vmoy</b>	<b>Final. Sol Vmoy</b>	<b>% Im</b>	<b>CPU (s)</b>
A1	1.27	0.06	0.95	51.48
A2	2.84	0.53	0.81	56.82
A3	2.68	0.51	0.81	59.50
A4	2.67	0.43	0.84	58.02

Solusi akhir pemilihan segmen jalur logistic yang didapatkan menggunakan kuantitas dan waktu eksekusi, dibangun sebuah pendekatan yang lebih fisible dalam mendapatkan peta logistic bencana. Untuk mendapatkan solusi yang berkualitas, dilihat secara hati-hati pada bagian jalur terpendek dan seragam dalam sebuah segmen kemudian dilanjutkan ke segmen berikutnya. Bagian segmen pertama merupakan diversifikasi yang diadopsi algoritma ini kemudian digunakan koloni awal dari koloni yang berbeda dengan koloni tersebut untuk mendapatkan solusi. Tujuan yang terkait dengan masing-masing koloni solusi memulai setiap semut dari koloni yang berbeda dari satu koloni ke koloni lain sangat diperhitungkan. Dalam menentukan solusi awal dari koloni, jumlah maksimum iterasi yang dibutuhkan untuk meningkatkan solusi yang tersedia. Jumlah iterasi tidak melebihi seratus iterasi, dan juga berdampak pada pengurangan waktu komputasi. Metode yang diusulkan harus memungkinkan memiliki kecepatan yang baik sehingga memungkinkan untuk perencanaan untuk menjalankan sejumlah besar tes set. Kriteria yang telah ditetapkan terkait dengan koloni tersebut diharapkan mendapatkan solusi yang baik dan tepat.

Dari hasil perhitungan menggunakan algoritma MOACO, dapat dilihat bahwa jalur terpendek sebagaimana terlihat pada segmen I oleh 95% merupakan terbesar dengan peringkat pertama, maka segmen IV sebesar 84% dan yang terakhir adalah segmen II dan III sebesar 81%. Hal ini mengisyaratkan bahwa dalam melakukan support logistik harus dilakukan pada segmen area I dan IV karena dapat dicapai dengan cepat. Sementara segmen II dan III perlu waktu untuk mencapai daerah daerah tersebut sehingga bukan menjadi prioritas. Yang menjadi prioritas adalah segmen I dan alternatifnya adalah segmen IV.

Setelah mendapatkan hasil, maka dilakukan update pheromone dengan teknik origin dan region. Untuk mengupdate intensitas pheromone tiap segmen setelah diperbaharui dengan pembaharuan global sehingga maka didapatkan hasil sebagai berikut:



Table 35 Intensitas pheromone tiap segmen pada jalur logistik

	Segmen I	Segmen II	Segmen III	Segmen IV
1	0.009	0.312595	0.1396	0.06795
2	0.12595	0.009	0.454	0.009
3	0.1396	0.1035	0.009	0.009
4	0.06795	0.009	0.009	0.009
5	0.009	0.009	0.258	0.1935

Hasil perhitungan algoritma MOACO dapat diperlihatkan dengan peta mitigasi bencana sebagai berikut :

Figure 20 Peta mitigasi bencana dengan algoritma MOACO



## **BAB VI KESIMPULAN ,REKOMENDASI DAN DISKUSI**

### **6.1 Kesimpulan**

Hasil analisis optimasi dengan menggunakan algoritma MOACO dapat digunakan untuk mencari dan mengoptimasi jalur terbaik dalam mitigasi bencana gunung berapi. Indikator yang dipakai dalam perhitungan algoritma MOACO terbukti sangat baik dalam mengoptimasi jalur pada area bencana. Pada hasil ini, dilakukan pengabaian terhadap kriteria yang nilainya kecil atau nol. Kesimpulan dilakukan setelah mendapatkan optimasi jalur terbaik kemudian ditetapkan jalur alternatifnya. System memberi rekomendasi terhadap jalur utama dan jalur alternatif sebagai pilihan kedua jika jalur utama tidak digunakan.

Hasil analisis algoritma MOACO dapat digunakan sebagai strategy dalam menyelesaikan masalah optimasi mitigasi bencana. Dalam implementasinya, perlu diperhatikan jumlah koloni yang digunakan dalam algoritma dan juga parameter serta jumlah struktur pheromonen dalam algoritma tersebut. Hasil analisis menunjukkan pada algoritma MOACO sangat efektif dalam melakukan optimasi pada mitigasi bencana pada daerah yang memiliki populasi yang banyak. Algoritma ini dapat digunakan secara efektif dalam membuat peta mitigasi bencana dan jalur yang akan digunakan saat bencana terjadi.

### **6.2 Rekomendasi**

Rekomendasi yang dapat diberikan adalah :

1. Para pemangku kepentingan terutama pemerintah melalui dinas terkait agar dapat menggunakan hasil analisis ini sebagai pantolan dalam mengoptimasi mitigasi bencana di daerah rawan bencana.
2. Informasi peta mitigasi bencana dapat diakses oleh masyarakat umum agar masyarakat mendapat informasi mitigasi yang akurat dan terpercaya.

3. Pentingnya melakukan edukasi kepada masyarakat di area bencana agar lebih peduli dalam menghadapi kejadian bencana dengan mengikuti panduan dan system mitigasi yang telah diatur.

### **6.3 Diskusi**

Dalam penelitian ini, perlu ditindak lanjuti dengan membuat aplikasi yang mudah diakses oleh berbagai pihak seperti operator, pembuat keputusan (pemerintah) dan juga masyarakat umum. Penelitian ini belum membahas tentang aplikasinya dan diharapkan dapat dilanjutkan oleh peneliti yang lain dalam merancang dan membangun aplikasi untuk mitigasi bencana ini. Hal ini disebabkan penelitian ini membahas tentang optimasi yang dapat dilakukan dan dihasilkan oleh algoritma MOACO ini dalam menyelesaikan masalah mitigasi bencana.

## REFERENCES

- [1] SIBIS, SIBIS New eEurope Indicator Handbook, European Commission publications., 2003.
- [2] A. A. A. I W. Sudarsana, "Model Matematika Untuk Sistem Evakuasi Tsunami Kota Palu (SET-KP) Berbasis Jalur Terpendek Dan Waktu Evakuasi Minimum," *Online Journal of Natural Science*, pp. 39-53, 2013.
- [3] M. I. Era Madona, "Aplikasi Metode Nearest Neighbor Pada Penentuan Jalur Evakuasi Terpendek Untuk Daerah Rawan Gempa Dan Tsunami," *Jurnal Elektron Vol 5 No. 2*, pp. 45-51, 2013.
- [4] Badan Nasional Penanggulangan Bencana RI, "Data Bencana Indonesia," BNPB, Jakarta, 2015.
- [5] S. T. Marco Dorigo, *Ant Colony Optimization*, London England and Cambridge, Massachusetts: A Bradford Book and The MIT Press, 2004.
- [6] X. L. F. S. W. Jin Feng Wang, "A Graph-Based Ant Colony Optimization Approach for Process Planning," *Hindawi Publishing Corporation The Scientific World Journal*, pp. 1-11, 2014.
- [7] Y. D. J. G. Liqiang Liu, "Ant Colony Optimization Algorithm for Continuous Domains Based on Position Distribution Model of Ant Colony Foraging," *Hindawi Publishing Corporation The Scientific World Journal*, pp. 1-9, 2014.
- [8] I. I. M. López, *Multi Objective Ant Colony Optimization*, Stanford, California: TECHNISCHE UNIVERSITÄTDARMSTADT and Universidad de Granada, 2004.
- [9] E. Verdianto, "Perancangan Sistem Penentuan Rute Terpendek Jalur Evakuasi Tsunami Dengan Algoritma Ant Colony Studi Kasus: Belawan," *Ilmu Komputer Universitas Sumatra Utara*, Medan, 2013.

- [10] A. Leksono, "Algoritma Ant Colony Optimization (Aco) Untuk Menyelesaikan Traveling Salesman Problem (TSP)," Fakultas FMIPA Universitas Diponegoro, Semarang, 2009.
- [11] J. R. Batmetan, "Algoritma Ant Colony Optimization (ACO) untuk Pemilihan Jalur Tercepat Evakuasi Bencana Gunung Lokon Sulawesi Utara," *AITI UKSW*, 2016.
- [12] R. I. Badan Geologi, "Laporan Aktivitas Gunung Berapi Lokon Sulawesi Utara," Badan Geologi Kementerian ESDM RI, Jakarta, 2015.
- [13] B. N. P. B. (BNPB), "Peta Topografi Dan Kawasan Rawan Bencana Gunung Lokon," BNPB, Jakarta, 2011.
- [14] P. K. Dharmendra Sutariya, "A Survey Of Ant Colony Based Routing Algorithms For Manet," *European Scientific Journal*, pp. 82-91, 2013.
- [15] J. A. A. Fransisca Arvevia I A, "Path Selection In Emergency Evacuation Using Quantum Ant-Colony Algorithm," *Teknik Informatika, Fakultas Teknik, Universitas Telkom*, pp. 1-8, 2014.
- [16] T. B. R. P. Chaimongkon Chokpanyasuwan, "Ant Colony Optimization for Load Management Based on Load Shifting in the Textile Industry," *American Journal of Applied Sciences*, pp. 142-154, 2015.
- [17] Y. B. Mohammed Taha Benslimane, "Ant Colony Algorithm for the Multi-Depot Vehicle Routing Problem in Large Quantities by a Heterogeneous Fleet of Vehicles," *INFOR*, pp. 31-40, 2013.
- [18] C. Daniel Angus, "Multiple Objective Ant Colony Optimisation," *Swarm Intell (2009)*, no. November 2008, pp. 69-85, 2009.
- [19] S. T. Manuel López-Ibáñez, "An experimental analysis of design choices of multi-objective ant colony optimization algorithms," *Swarm Intelligence*, vol. 6, no. 3, pp. 207-232, 2012.

- [20] M. E. R. Yassine Saji, "Multi-Objective Ant Colony Optimization Algorithm to Solve a Nurse Scheduling Problem," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 8, pp. 311-320, 2013.
- [21] M. L.-I. a. T. Stützle, "The Automatic Design of Multi-Objective Ant Colony Optimization Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, p. 861–875, 2012.
- [22] P. G.-S. J. J. M. P. A. C. A. M. Mora, "Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal," *Soft Comput*, vol. 17, p. 1175–1207, 2013.
- [23] M. L.-I. a. T. Stützle, "The impact of design choices of multiobjective ant colony optimization algorithms on performance: An experimental study on the biobjective TSP," IRIDIA - Technical Report Series, Technical Report No. TR/IRIDIA/2010-003, Bruxelles, Belgium, 2010.
- [24] D. T. T. L. e. a. Dickson K. W. Chiu, "Alert based disaster notification and resource allocation," *Inf Syst Front*, vol. 12, p. 29–47, 2010.
- [25] e. a. Marco Lu'tzenberger, "A common approach to intelligent energy and mobility services in a smart city environment," *J Ambient Intell Human Comput*, vol. 6, p. 337–350, 2015.
- [26] e. a. Albert Ali Salah, "Multimodal identification and localization of users in a smart environment," *J Multimodal User Interfaces*, vol. 2, p. 75–91, 2008.
- [27] X. Zhang, "Development Of A Mixed-Flow Optimization System For Emergency Evacuation In Urban Networks," Faculty of the Graduate School of the University of Maryland, College Park, Maryland, 2012.
- [28] H. Lyonnais, "Metode Pencarian Lintasan Terpendek Graf untuk Evakuasi Bencana," *Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung*, pp. 1-5, 2012.

- [29] S. S. A. B. R. K. S. C. Jayadeva, "Ants find the shortest path: a mathematical proof," *Swarm Intell*, vol. 7, p. 43–62, 2013.
- [30] C. Coffrin, "Decision support for disaster management through hybrid optimization," *Constraints*, vol. 20, p. 490–491, 2015.
- [31] M. S. J. e. a. Ali Bozorgi-Amiri, "A modified particle swarm optimization for disaster relief logistics under uncertain environment," *Int J Adv Manuf Technol*, vol. 60, p. 357–371, 2012.
- [32] C.-H. Hsu, "Ubiquitous Intelligence and Computing: building smart environment in real and cyber space," *J Ambient Intell Human Comput*, vol. 3, p. 83–85, 2012.
- [33] E. E. B. K. Linet Özdamar, "Emergency Logistics Planning in Natural Disasters," *Annals of Operations Research*, vol. 129, p. 217–245, 2004.
- [34] V. W. H. N. S. Thushari Silva, "Disaster mitigation and preparedness using linked open data," *J Ambient Intell Human Comput*, vol. 4, p. 591–602, 2013.
- [35] P. F. Stefan Propp, "ViSE – A Virtual Smart Environment for Usability Evaluation," *IFIP International Federation for Information Processing*, p. 38–45, 2010.
- [36] I. Satyarno, "Earth Quake Disaster Mitigation," Atma Jaya, Yogyakarta, 2016.
- [37] W. A. J. B. N. Irwan Iftadi, "Perancangan Peta Evakuasi Menggunakan Algoritma Floyd-Warshall untuk Penentuan Lintasan Terpendek: Studi Kasus," *Performa*, pp. 95-104, 2011.
- [38] Y. Z. L. Guo Li, "Comprehensive Optimization of Emergency Evacuation Route and Departure Time under Traffic Control," *The Scientific World Journal*, pp. 1-12, 2014.
- [39] I. Fauzy, "Penggunaan Algoritma Dijkstra Dalam Pencarian Rute Tercepat Dan Terpendek, Studi Kasus Pada Jalan Raya Wilayah Blok M Dan Kota," UIN Syarif Hidayatullah, Jakarta, 2011.
- [40] M. L.-I. Infante, MultiObjective Ant Colony Optimization,

Granda and Stanford: Universidad de Grannada, 2004.

- [41] A. A. Z. W. Liu Hongbo, "A Multi-swarm Approach to Multi-objective Flexible Job-shop Scheduling Problems," *Fundamenta Informaticae*, vol. 95, no. 95, pp. 1-25, 2009.
- [42] A. I. S. G. A. Chandramohan, "Multi-Objective Optimization Using Ant Colony Optimization for Construction Projects," *National Institute of Technology Calicut, India*, pp. 84-96, 2010.
- [43] N. C. W. C. Teerapun Saeheaw, "Application of Ant colony optimization for Multi-objective Production Problems," *World Academy of Science, Engineering and Technology*, vol. 36, pp. 655-660, 2009.
- [44] C.-L. Y. Rong-Hwa Huang, "Solving a multi-objective overlapping flow-shop scheduling," *Int J Adv Manuf Technol*, vol. 42, pp. 955-962, 2009.
- [45] L. K. V. G. Krasimira Genova, "A Survey of Solving Approaches for Multiple Objective Flexible Job Shop Scheduling Problems," *Cybernetics And Information Technologies*, vol. 15, no. 2, pp. 3-22, 2015.
- [46] O. C. F. H. C. Garcí'a-Martí'nez, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP," *European Journal of Operational Research*, vol. 180, pp. 116-148, 2007.
- [47] C. O. J. F. M. C. Pablo Ortega, "Multiple Ant Colony System For A Vrp With Time Windows And Scheduled Loading," *Ingeniare. Revista chilena de ingeniería*, vol. 17, no. 3, pp. 393-403, 2009.
- [48] D. S. Y. J. Sirisha Devi, "Multi Objective Optimization Problem resolution based on Hybrid Ant-Bee Colony for Text Independent Speaker Verification," *I.J. Modern Education and Computer Science*, vol. 1, pp. 55-63, 2015.
- [49] R. M. E. L. L. G. A.E. Rizzoli, "Ant colony optimization for real-world vehicle routing problems From theory to applications," *Swarm Intell*, vol. 1, p. 135-151, 2007.



- [50] D. D. Vivek S. Borkar, "A novel ACO algorithm for optimization via reinforcement and initial bias," *Swarm Intell*, vol. 3, pp. 3-34, 2009.
- [51] M. B. T. S. Z. Y. M. D. Prasanna Balaprakash, "Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem," *Swarm Intell*, vol. 3, p. 223–242, 2009.
- [52] F. N. H. R. C. W. Timo Kötzing, "Theoretical analysis of two ACO approaches for the traveling salesman problem," *Swarm Intell*, vol. 6, pp. 1-21, 2012.
- [53] A. M. A. A. A. F. Khalid M. Salama, "Multiple pheromone types and other extensions to the Ant-Miner classification rule discovery algorithm," *Swarm Intell*, vol. 5, p. 149–182, 2011.
- [54] T. S. M. B. Paola Pellegrini, "A critical analysis of parameter adaptation in ant colony optimization," *Swarm Intell*, vol. 6, p. 23–48, 2012.
- [55] D. I. Candra Dewi, "Map Visualization of Shortest Path Searching of Government Agency Location Using Ant Colony Algorithm," (*IJCSIS*) *International Journal of Computer Science and Information Security*, vol. 11, no. 11, pp. 19-23, 2013.
- [56] Y. D. a. J. G. Liqiang Liu, "Ant Colony Optimization Algorithm for Continuous Domains Based on Position Distribution Model of Ant Colony Foraging," *The Scientific World Journal*, pp. 1-19, 2014.
- [57] L. M. A. C. G. C. L. Bacon, "Developing a smart environment for crisis management training," *Ambient Intell Human Comput*, vol. 4, p. 581–590, 2013.
- [58] E. A. Nik Bessis, "Preface for a special issue on ‘‘Smart environments and collective computational intelligence for disaster management’’," *J Ambient Intell Human Comput*, vol. 4, p. 533–534, 2013.
- [59] T. S. Heiko Rosnagel, "Secure Mobile Notifications of

- Civilians in Case of a Disaster," *H. Leitold and E. Markatos*, p. 33 – 42, 2006.
- [60] G. B. P. F. Stefan Propp, "Task Model-Based Usability Evaluation for Smart Environments," *P. Forbrig and F. Paternò*, p. 29–40, 2008.
- [61] J. H. M. P. G. Y. Petri Pulli, "User interaction in smart ambient environment targeted for senior citizen," *Med Biol Eng Comput*, vol. 50, p. 1119–1126, 2012.
- [62] T. F. I.D.I.D. Ariyasingha, "A Performance Study for the Multi-objective Ant Colony Optimization Algorithms on the Job Shop Scheduling Problem," *International Journal of Computer Applications*, vol. 132, no. 14, pp. 1-8, 2015.
- [63] M. L.-I. a. T. S. Leonardo C. T. Bezerra, "Automatic Generation of Multi-objective ACO Algorithms for the Bi-objective Knapsack," IRIDIA - Technical Report , Series Technical Report No. TR/IRIDIA/2012-013, Bruxelles, Belgium, 2012.
- [64] M. L.-I. a. T. Stützle, "An Analysis of Algorithmic Components for Multiobjective Ant Colony Optimization: A Case Study on the Biobjective TSP," IRIDIA - Technical Report Series, Technical Report No. TR/IRIDIA/2009-019, Bruxelles, Belgium, 2009.
- [65] M. L.-I. a. T. Stützle, "Automatic Configuration of Multi-Objective ACO Algorithms," IRIDIA - Technical Report Series, Technical Report No. TR/IRIDIA/2010-011, Bruxelles, Belgium, 2010.
- [66] E. F. G. M. C. G. L. S. B. Leonardo C.T. Bezerra, "Analyzing the impact of MOACO components: An algorithmic study on the multi-objective shortest path problem," *Expert Systems with Applications*, vol. 40, p. 345–355, 2013.

## MULTI OBJECTIVE ACO (MO-ACC) ALGORITHM UNTUK OPTIMASI STRATEGI MITIGASI BENCANA

Mitigasi merupakan salah satu bagian penting dalam manajemen bencana. Mitigasi dilihat sebagai upaya menentukan jalur terbaik dalam evakuasi bencana, penentuan jalur distribusi terbaik dan penentuan titik pengungsian. Mitigasi dalam bencana adalah harus dioptimasi dengan penentuan jalur terbaik dan terpendek. Pemilihan jalur terbaik ini ditujukan pada daerah yang padat dengan pemukiman penduduk tempat tinggal warga yang dengan dengan area letusan gunung berapi dan masuk pada area berbahaya. Mitigasi bencana memerlukan system informasi yang akurat dan cepat dalam memberikan respon.

Buku ini mengkaji dan menyelidiki untuk memilih jalur terpendek di mitigasi letusan gunung berapi Lokon. Penelitian ini mencoba menerapkan formula Algoritma Multiple Objective Ant Colony Optimization dengan memasukkan indikator kecepatan, jarak, tikungan, kepadatan dan titik aman untuk menghitung jalur pendek untuk yang dipilih menggunakan algoritma MO-ACO. Hasil yang diperoleh untuk menghasilkan solusi dengan jalur pendek peta mitigasi yang terbaik, dan cepat dalam merespon dan penting untuk mengelola bencana. Peta ini menjadi petunjuk dalam manajemen bencana dan navigasi dalam proses mitigasi. Peta mitigasi ini menggunakan sistem informasi mitigasi bencana dan terlibat dalam penanggulangan bencana gunung berapi Lokon di Sulawesi Utara.



**Dr. Verry Ronny Palilingan, M, Eng** Lahir di Manado

28 Januari 1961. Dosen Tetap di Jurusan Pendidikan Teknologi Informasi dan Komunikasi. Staf Pengajar S2 Prodi Pendidikan Teknologi dan Kejuruan Pasca sarjana UNIMA.

Menyelesaikan Pendidikan S1 tahun 1984 di IKIP Negeri Manado , S2 di Universite Aix Marseille , master Information and communication dan S3 di Univesitas Negeri Jakarta Prodi Manajemen Pendidikan.

Terlibat dalam penelitian penelitian unggulan Simliptabmas dan membuat artikel prociding di IOP , Atlantic Press . dan Jurnal yang dimuat di Scopus. Mengikuti kegiatan seminar/workshop dalam kajian Competitive Intelligence di Bad Nauhem Frankfurt Germany dan Universitas Peking di Beijing China.

ISBN 978 602 1376 57 7



9 786021 376577

