

**MODIFIKASI PENERAPAN ALGORITMA *MULTI-  
VERSE* PADA STUDI KASUS *ASYMMETRIC  
TRAVELING SALESMAN PROBLEM***

**SKRIPSI**

Diajukan untuk memenuhi salah satu syarat guna mencapai gelar  
Sarjana dalam bidang ilmu Teknik Industri

Disusun oleh:

**Nama : Alexander Evan Putra**

**NPM : 2015610046**



**PROGRAM STUDI SARJANA TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KATOLIK PARAHYANGAN  
BANDUNG  
2019**

**FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KATOLIK PARAHYANGAN  
BANDUNG**



Nama : Alexander Evan Putra  
NPM : 2015610046  
Jurusan : Teknik Industri  
Judul Skripsi : MODIFIKASI PENERAPAN ALGORITMA *MULTI-VERSE* PADA  
STUDI KASUS *ASYMMETRIC TRAVELING SALESMAN  
PROBLEM*

**TANDA PERSETUJUAN SKRIPSI**

Bandung, Januari 2019

**Ketua Program Studi Sarjana Teknik Industri**

(Romy Loice, S.T., M.T.)

**Dosen Pembimbing**

(Ignatius A. Sandy, S.Si., M.T..)

**FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS KATOLIK PARAHYANGAN  
BANDUNG**



Nama : Alexander Evan Putra  
NPM : 2015610046  
Jurusan : Teknik Industri  
Judul Skripsi : MODIFIKASI PENERAPAN ALGORITMA *MULTI-VERSE* PADA  
STUDI KASUS *ASYMMETRIC TRAVELING SALESMAN*  
*PROBLEM*

**TANDA PERSETUJUAN SKRIPSI**

Bandung, Januari 2019

**Ketua Program Studi Sarjana Teknik Industri**

(Romy Laise, S.T., M.T.)

**Dosen Pembimbing**

(Ignatius A. Sandy, S.Si., M.T.)



Jurusan Teknik Industri  
Fakultas Teknologi Industri  
Universitas Katolik Parahyangan

## **Pernyataan Tidak Mencontek atau Melakukan Tindakan Plagiat**

Saya, yang bertanda tangan dibawah ini,

Nama : Alexander Evan Putra  
NPM : 2015610046

Dengan ini menyatakan bahwa Skripsi dengan judul :

**“MODIFIKASI PENERAPAN ALGORITMA *MULTI-VERSE* PADA STUDI  
KASUS *ASYMMETRIC TRAVELING SALESMAN PROBLEM*”**

adalah hasil pekerjaan saya dan seluruh ide, pendapat atau materi dari sumber lain telah dikutip dengan cara penulisan referensi yang sesuai.

Pernyataan ini saya buat dengan sebenar-benarnya dan jika pernyataan ini tidak sesuai dengan kenyataan, maka saya bersedia menanggung sanksi yang akan dikenakan kepada saya.

Bandung,

Alexander Evan Putra  
NPM : 2015610046

## ABSTRAK

*Asymmetric Traveling Salesman Problem* (ATSP) adalah permasalahan optimasi dimana terdapat seorang pedagang yang perlu mengunjungi beberapa kota. Pada kasus ATSP, jarak tempuh dari kota A ke kota B tidak sama dengan jarak tempuh dari kota B ke kota A. Tujuan penyelesaian masalah ini adalah untuk meminimasi jarak tempuh yang dialami oleh pedagang itu. *Multi-Verse Optimizer algorithm* (MVO) adalah salah satu metode metaheuristik pernah dicoba untuk menyelesaikan permasalahan ATSP. Namun, hasil dari implementasi MVO dalam menyelesaikan permasalahan ATSP masih kurang begitu baik. Oleh sebab itu, pada penelitian ini akan dilakukan modifikasi penerapan MVO terhadap kasus ATSP. Modifikasi dilakukan dengan menggunakan algoritma *divide and conquer* sebagai dasar. Metode *divide and conquer* berusaha memecah satu permasalahan yang besar menjadi beberapa permasalahan kecil. Lalu, permasalahan kecil yang ada diselesaikan akan digabung lagi untuk mendapat hasil akhir secara keseluruhan. Hasil penerapan modifikasi mampu membuat *best known solution* yang awalnya bernilai lebih besar 22,81% menjadi 8,68% lebih besar dari solusi optimal untuk kasus dengan 45 kota. Pada kasus ATSP dengan 34 kota, hasil penerapan modifikasi mampu membuat *best known solution* yang sebelumnya lebih besar 10,49% menjadi 2,95% lebih besar dari solusi optimal.

## **ABSTRACT**

*Asymmetric Traveling Salesman Problem (ATSP) is an optimization problem about a salesman who is required to visit some cities. In ATSP, the distance from city A to city B is not equal with the distance from city B to city A. The objective function of this problem is to minimize the total distance that must be taken by the salesman. Multi-Verse Optimizer Algorithm (MVO) is one of the metaheuristic methods that has been used to complete ATSP. In the other hand, the results obtained from this method are not very good. Therefore, the content of this research will be the modification of the implementation of MVO for ATSP. Divide and conquer algorithm will be the base of this modification. Divide and conquer algorithm tries to complete a problem by dividing the problem into smaller problems. The smaller problems will be resolved and reunited to obtain the final result of the problem. Taking example of ATSP with 45 cities, this implementation makes the result of the best-known solution decreased from 22,81% optimal solution to 8,68%. Using ATSP with 34 cities, the implementation makes the result of the best-known solution decreased from 10,49% to 2,95% optimal solution.*

## KATA PENGANTAR

Puji dan syukur saya sampaikan kepada Tuhan Yang Maha Kuasa atas kelancaran yang diberikan dalam menyelesaikan penelitian ini. Selama proses penelitian, penulis mendapat banyak dukungan dari banyak pihak. Dukungan yang diterima penulis berasal dari dosen penguji dan pembimbing, keluarga, dan teman. Melalui sarana ini, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Ignatius A. Sandy, S.Si, M.T. selaku dosen pembimbing dalam penelitian ini.
2. Ibu Cynthia Prithadevi Juwono, Ir., M.S., Ibu Cherish Rikardo, S.Si., M.T. dan Bapak Romy Loice, S.T., M.T., selaku dosen yang banyak memberikan masukan dalam pengembangan penelitian yang dilakukan penulis.
3. Ko Hardian yang sudah mau membantu memberikan penjelasan mengenai penelitian yang la buat sebelumnya.
4. Nenek, mama, papa, ii, om, dan cici penulis yang selalu mendukung dan menyemangati selesainya penelitian ini.
5. Hellen dan Lisa yang telah membantu penulis dalam mengarahkan penulis mengenai cara mengutarakan pemikiran dalam tulisan dengan baik.
6. Ko Dyo, Januar, Demas, Jane, Dian, Kenny, Joice dan Verrel yang telah menjadi *partner* dalam menjalankan proses penggarapan skripsi ini.
7. Teman-teman dari kelas D angkatan 2015 dan teman-teman lainnya baik diangkatan 2015 dan 2016.
8. Teman-teman penulis dari SMA maupun Aikido Dojo UNPAR lainnya.
9. Pihak-pihak lain yang tidak disebutkan karena telah terlibat secara langsung maupun tidak langsung selama proses penelitian yang dilakukan.

Penulis berharap Tuhan akan membalas segala bantuan yang diterima penulis. Penulis memohon maaf apabila terdapat kesalahan-kesalahan baik dalam proses penelitian maupun proses penulisan penelitian ini. Penulis sangat

menerima kritik yang positif dan membangun dari pihak-pihak lain. Semoga penelitian ini dapat bermanfaat.

Bandung, 17 Januari 2019

Penulis

## DAFTAR ISI

<b>ABSTRAK</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>KATA PENGANTAR</b> .....	<b>iii</b>
<b>DAFTAR ISI</b> .....	<b>v</b>
<b>DAFTAR TABEL</b> .....	<b>vii</b>
<b>DAFTAR GAMBAR</b> .....	<b>ix</b>
<b>BAB I PENDAHULUAN</b> .....	<b>I-1</b>
I.1 Latar Belakang Masalah.....	I-1
I.2 Identifikasi dan Perumusan Masalah.....	I-3
I.3 Pembatasan Masalah .....	I-5
I.4 Tujuan Penelitian.....	I-5
I.5 Manfaat Penelitian.....	I-5
I.6 Metodologi Penelitian.....	I-6
I.7 Sistematika Penulisan.....	I-8
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>II-1</b>
II.1 <i>Traveling Salesman Problem</i> dan <i>Asymmetric Traveling Salesman Problem</i> .....	II-1
II.2 <i>Multi-Verse Optimizer Algorithm</i> .....	II-3
II.3 <i>Divide and Conquer Algorithm</i> .....	II-6
II.4 <i>Random Key Representation</i> dan <i>Permutational Encoding</i> .....	II-7
II.5 <i>Analysis of Variance</i> .....	II-8
II.6 <i>Flowchart</i> dan <i>Pseudocode</i> .....	II-9
II.7 Verifikasi dan Validasi.....	II-10
<b>BAB III PERANCANGAN ALGORITMA</b> .....	<b>III-1</b>
III.1 Perancangan Modifikasi Penerapan Algoritma MVO untuk Permasalahan ATSP.....	III-1
III.1.1 Pengelompokan Urutan Keseluruhan Kunjungan Kota menjadi Sejumlah Sub-Urutan Kota.....	III-3
III.1.2 Penggunaan Informasi Sub-Urutan Kota untuk Proses Pembangunan Solusi.....	III-4

III.1.3 <i>Local Optimization</i> dengan Menggunakan Algoritma <i>Divide and Conquer</i> .....	III-5
III.2 <i>Encoding</i> dan <i>Decoding</i> .....	III-7
III.3 Diagram Alir Modifikasi Penerapan MVO untuk Permasalahan ATSP.....	III-8
III.3.1 Rancangan Algoritma Utama MVO serta Modifikasinya...	III-10
III.3.2 Algoritma Pengelompokan Sub-Urutan Kota.....	III-14
III.3.3 Algoritma Pencarian Sub-Urutan Terbaik.....	III-15
III.3.4 Algoritma Penyimpanan Data Sub-Urutan Terbaik.....	III-17
III.3.5 Algoritma <i>Local Optimization</i> .....	III-19
III.3.6 Algoritma Pengacakan <i>Best Universe</i> .....	III-20
III.3.7 Algoritma Penerapan Informasi Sub-Urutan Kota.....	III-22
III.3.8 Algoritma Pemilihan Sub-Urutan Kota yang Digunakan...	III-26
III.4 Pengerjaan Modifikasi Penerapan Algoritma MVO secara Manual.....	III-29
<b>BAB IV IMPLEMENTASI ALGORITMA</b> .....	<b>IV-1</b>
IV.1 Verifikasi Program Komputer.....	IV-1
IV.2 Validasi Program Komputer.....	IV-7
IV.3 Implementasi Modifikasi Penerapan MVO pada Kasus ATSP...	IV-8
IV.3.1 Implementasi Modifikasi Penerapan MVO pada Kasus BR17.....	IV-9
IV.3.2 Implementasi Modifikasi Penerapan MVO pada Kasus FTV33.....	IV-13
IV.3.3 Implementasi Modifikasi Penerapan MVO pada Kasus FTV44.....	IV-16
<b>BAB V ANALISIS</b> .....	<b>V-1</b>
V.1 Analisis Rancangan Algoritma Keseluruhan.....	V-1
V.1.1 Analisis Rancangan Algoritma MVO Tanpa Modifikasi.....	V-1
V.1.1 Analisis Rancangan Penerapan Modifikasi MVO.....	V-2
V.2 Analisis Parameter Besar Pemotongan Kota.....	V-2
V.3 Analisis Performansi Modifikasi Penerapan MVO.....	V-4
<b>BAB VI KESIMPULAN DAN SARAN</b> .....	<b>V-1</b>
VI.1 Kesimpulan.....	V-1

VI.2 Saran..... V-1

**DAFTAR PUSTAKA**

**LAMPIRAN A**



## DAFTAR TABEL

Tabel I.1	Contoh Bentuk Awal <i>Universe</i> Sederhana.....	I-4
Tabel I.2	Contoh Bentuk Awal <i>Universe</i> yang Ada Setelah Melewati Proses Eksplorasi .....	I-4
Tabel I.3	Contoh Bentuk Awal <i>Universe</i> yang Ada Setelah Melewati Proses Eksploitasi.....	I-5
Tabel II.1	ANOVA untuk Satu Faktor.....	II-8
Tabel II.2	Simbol pada <i>Flowchart</i> .....	II-9
Tabel II.3	Istilah Dasar <i>Pseudocode</i> .....	II-10
Tabel III.1	Tampilan Angka Acak yang Terbentuk.....	III-6
Tabel III.2	Tampilan Angka Acak yang Telah Diurutkan .....	III-7
Tabel III.3	Tampilan Pengalokasian Angka Acak .....	III-7
Tabel III.4	Tampilan Proses <i>Decoding</i> .....	III-7
Tabel III.5	Notasi yang Digunakan pada Perancangan Algoritma ....	III-9
Tabel III.6	Matriks Jarak Permasalahan ATSP Sederhana .....	III-29
Tabel III.7	Urutan Kunjungan pada <i>Universe</i> .....	III-30
Tabel III.8	Matriks Modifikasi hasil algo setelah Proses Berakhir .....	III-35
Tabel III.9	Matriks eksplorasi_baru setelah Proses Berakhir .....	III-37
Tabel III.10	Matriks jarak_explorasi_baru setelah Proses Berakhir ....	III-37
Tabel IV.1	Tampilan Program Secara Keseluruhan.....	IV-2
Tabel IV.2	Tampilan Input Program .....	IV-3
Tabel IV.3	Tampilan Matriks D pada Iterasi Lain.....	IV-3
Tabel IV.4	Tampilan Input dan Hasil Pemilihan Sub-Urutan Kota.....	IV-4
Tabel IV.5	Tampilan Pemotongan dan Penyimpanan Data Sub- Urutan.....	IV-5
Tabel IV.6	Tampilan Proses <i>Local Optimization</i> .....	IV-6
Tabel IV.7	Tampilan Proses Penerapan Data Sub-Urutan pada <i>Universe</i> .....	IV-6
Tabel IV.8	Tampilan Proses Pengacakan <i>Best Universe</i> .....	IV-7
Tabel IV.9	Validasi Hasil Jarak Kunjungan Kasus Sederhana.....	IV-8

Tabel IV.10	Besar Pengujian Pemotongan Kota yang Dilakukan.....	IV-9
Tabel IV.11	Rekapitulasi Hasil Modifikasi Penerapan MVO terhadap BR17.....	IV-10
Tabel IV.12	Hasil Pengujian ANOVA terhadap Kasus BR17.....	IV-11
Tabel IV.13	Rekapitulasi Hasil Modifikasi Penerapan MVO terhadap FTV33.....	IV-13
Tabel IV.14	Hasil Pengujian ANOVA terhadap Kasus FTV33.....	IV-14
Tabel IV.15	Rekapitulasi Hasil Modifikasi Penerapan MVO terhadap FTV44.....	IV-16
Tabel IV.16	Hasil Pengujian ANOVA terhadap Kasus FTV44.....	IV-17
Tabel V.1	Rekapitulasi Jumlah Waktu <i>Running</i> Program Komputer yang Dibuat.....	V-3

## DAFTAR GAMBAR

Gambar I.1	Metodologi Penelitian.....	I-6
Gambar I.2	Metodologi Penelitian.....	I-9
Gambar II.1	Ilustrasi <i>White Hole</i> , <i>Black Hole</i> , dan <i>Wormhole</i> .....	II-3
Gambar II.2	Ilustrasi Penggunaan <i>Divide and Conquer</i> dalam Menyelesaikan Masalah.....	II-6
Gambar II.3	Contoh <i>Encoding</i> dengan Menggunakan <i>Random Key</i>	II-7
Gambar III.1	Diagram Alir Algoritma Utama.....	III-10
Gambar III.2	Diagram Alir Algoritma Pengelompokan Sub-Urutan Kota.....	III-13
Gambar III.3	Diagram Alir Algoritma Pencarian Sub-Urutan Terbaik	III-16
Gambar III.4	Diagram Alir Algoritma Penyimpanan Sub-Urutan Terbaik.....	III-17
Gambar III.5	Diagram Alir Algoritma <i>Local Optimization</i> .....	III-19
Gambar III.6	Diagram Alir Pengacakan <i>Best Universe</i> .....	III-20
Gambar III.7	Diagram Alir Penerapan Informasi Sub-Urutan Kota....	III-22
Gambar III.8	Diagram Alir Pemilihan Sub-Urutan Kota yang Digunakan.....	III-26
Gambar III.9	Diagram Alir Penyesuaian Hasil Eksplorasi.....	III-27
Gambar IV.1	Hasil Pengujian Fisher's LSD pada Kasus BR17.....	IV-11
Gambar IV.2	<i>Interval Plot</i> pada Kasus BR17.....	IV-12
Gambar IV.3	Hasil Pengujian Fisher's LSD pada Kasus FTV33.....	IV-14
Gambar IV.4	<i>Interval Plot</i> pada Kasus FTV33.....	IV-15
Gambar IV.5	Hasil Pengujian Fisher's LSD pada Kasus FTV44.....	IV-17
Gambar IV.6	<i>Interval Plot</i> pada Kasus FTV44.....	IV-18

# BAB I

## PENDAHULUAN

Pada bagian ini akan dijelaskan mengenai tahapan pendahuluan dalam pengembangan modifikasi penerapan algoritma MVO dalam menyelesaikan permasalahan ATSP. Bagian ini akan berisi latar belakang masalah, identifikasi dan perumusan masalah, batasan masalah, tujuan dan manfaat penelitian. Selain itu juga akan dijabarkan metodologi penelitian serta sistematika penulisan penelitian.

### I.1 Latar Belakang Masalah

Salah satu permasalahan yang membutuhkan pencarian *global optimum* sebagai solusinya adalah *traveling salesman problem*. Pada kasus *traveling salesman problem*, terdapat penjual yang mau memasarkan produknya ke beberapa tempat. Masing-masing tempat memiliki posisinya masing-masing dan memiliki jarak tempuh tertentu dengan tempat-tempat lainnya. Penyelesaian kasus ini dimaksudkan untuk mencari rute kunjungan untuk penjual sehingga penjual itu mampu mengunjungi seluruh tempat dengan jarak tempuh tersingkat.

*Traveling Salesman Problem* sendiri dibagi menjadi dua jenis, yaitu *Symmetric TSP* dan *Asymmetric TSP* (Gutin dan Punnen, 2002). Perbedaan antara *Symmetric TSP* dan *Asymmetric TSP* adalah jarak tempuh antara titik A ke titik B ketika penjual bergerak dari titik A ke titik B maupun sebaliknya. Bila jarak antara titik A ke titik B maupun sebaliknya memiliki nilai yang sama, maka persoalan tersebut termasuk ke dalam kasus *Symmetric TSP*. Namun, apabila jarak antara titik A ke titik B tidak sama dengan jarak antara titik B ke titik A, maka persoalan tersebut termasuk ke dalam kasus *Asymmetric TSP*.

*Asymmetric TSP* (ATSP) dapat muncul karena beberapa penyebab. Kondisi penutupan jalan yang membuat pengguna jalan harus melewati jalur lain untuk menuju lokasi yang diinginkan maupun kebijakan satu jalur yang dibuat pemerintah untuk memecah kemacetan merupakan contoh penyebab terjadinya ATSP. Penerapan ATSP dalam kasus TSP mampu mengeliminasi asumsi bahwa

jarak setiap tempat dalam kondisi pulang-pergi memiliki jarak yang sama. Oleh sebab itu, ATSP mampu merepresentasikan kondisi nyata dengan lebih baik.

Menurut Ascheuer, Grotschel, dan Abdel-Hamid (1999), *Asymmetric Traveling Salesman Problem* adalah salah satu jenis dari *NP-Hard Problem* (*Non-deterministic Polynomial-time Hard Problem*). Hal ini berarti semakin besar lingkup permasalahan yang dihadapi, maka waktu untuk mencari penyelesaian akan semakin lama dengan usaha yang semakin besar. Ascheuer, Grotschel, dan Abdel-Hamid (1999) juga mengatakan bahwa penggunaan metode eksak mampu menyelesaikan permasalahan dengan hasil yang optimal dan dengan waktu yang masuk akal apabila permasalahan yang dihadapi juga tidak terlalu besar. Namun, apabila waktu yang ada terbatas, maka hanya metode heuristik yang mampu menyelesaikan permasalahan tersebut.

Terdapat beberapa jenis metode eksak yang ada, seperti *dynamic programming* dan *branch and bound*. Menurut Talbi (2009), metode eksak dapat diterapkan pada masalah yang sederhana. Semakin kompleks sebuah permasalahan, semakin lama waktu yang diperlukan untuk menyelesaikan masalah tersebut. Oleh sebab itu, pendekatan heuristik cocok digunakan untuk menghasilkan solusi yang baik dengan waktu yang masuk akal untuk menjawab permasalahan-permasalahan kompleks.

Metode heuristik dapat diklasifikasikan menjadi dua jenis (Talbi, 2009). Kedua klasifikasi tersebut adalah *specific heuristics* dan *metaheuristics*. *Specific heuristic* didesain untuk menyelesaikan sebuah permasalahan khusus. Sedangkan, metode metaheuristik merupakan algoritma umum yang dapat digunakan untuk menyelesaikan hampir semua permasalahan optimasi. Metode metaheuristik dapat dilihat sebagai metodologi umum yang levelnya lebih tinggi yang dapat digunakan sebagai panduan dalam menyelesaikan suatu permasalahan optimasi yang spesifik.

Metode metaheuristik ditujukan untuk tiga tujuan utama, yaitu menyelesaikan masalah dengan lebih cepat, menyelesaikan *problem* yang besar, dan membuat algoritma yang *robust* (Talbi, 2009). Sebagian besar metode metaheuristik meniru contoh dari peristiwa yang ada di alam. Beberapa contoh peristiwa yang digunakan sebagai inspirasi pengembangan metode metaheuristik adalah koloni semut, koloni lebah, dan lain-lain.

Metode metaheuristik sendiri telah digunakan untuk menyelesaikan ATSP. Beberapa metode yang telah digunakan adalah *A New Genetic Algorithm* (Nagata dan Soler, 2012), *Harmony Search Algorithm* (Kevin, 2017), *Elephant Herding Optimization* (Santosa, 2017) dan *Football Game Algorithm* (Raharja, 2017). Ketiga metode metaheuristik tersebut berhasil memberikan solusi yang baik dalam menyelesaikan ATSP. Metode-metode metaheuristik lain juga dapat digunakan untuk menyelesaikan ATSP dengan harapan membuahkan hasil yang lebih baik dari metode yang telah digunakan sebelumnya.

*Multi-Verse Optimizer algorithm* sendiri telah digunakan untuk menyelesaikan *Asymmetric Traveling Salesman Problem* (Gunardi, 2018). *Multi-Verse Optimizer algorithm* dicoba untuk menyelesaikan tiga kasus *benchmark* yaitu BR17 dengan mendapatkan *best known solution*, FTV33 dengan hasil 10,49% dari *best known solution* dan FTV44 dengan hasil 22,81% dari *best known solution* (*best known solution* pada kasus FTV33 adalah 1324 satuan jarak dan pada kasus FTV44 adalah 1613 satuan jarak). Hasil *benchmarking* pada penelitian tersebut menunjukkan bahwa *Multi-verse Optimizer algorithm* mampu menunjukkan hasil yang lebih baik daripada penggunaan *Harmony Search Algorithm* dalam menyelesaikan ATSP. Namun, hasil penerapan *Football Game Algorithm* dan *New Genetic Algorithm* masih lebih baik daripada *Multi-verse Optimizer algorithm*.

Pengimplementasian *Multi-Verse Optimizer algorithm* untuk menyelesaikan ATSP belum mampu menyaingi performa penerapan beberapa algoritma lain yang telah disebutkan. Oleh sebab itu, harapan untuk meningkatkan performa *Multi-Verse Optimizer algorithm* dalam menyelesaikan ATSP muncul. Cara yang ingin diterapkan adalah dengan melakukan modifikasi penerapan *Multi-Verse Optimizer algorithm* terhadap penyelesaian ATSP.

## **I.2 Identifikasi dan Perumusan Masalah**

Penyelesaian masalah *Asymmetric Traveling Salesman Problem* dapat dilakukan dengan memunculkan seluruh kombinasi kota tujuan yang ada, lalu dicari kombinasi yang mampu menghasilkan jarak terpendek. Namun, pengecekan seluruh kombinasi yang ada akan memakan waktu yang sangat lama apabila jumlah kota yang harus dikunjungi semakin banyak. Proses yang memakan waktu inilah yang dicoba dihindari dengan menerapkan metode metaheuristik.

Pada metode *Multi-Verse Optimizer algorithm*, setiap *universe* akan memiliki beberapa objek didalamnya. Pada penerapan yang dilakukan oleh Gunardi (2018), setiap *universe* berarti sebuah urutan kunjungan dan setiap objek pada *universe* mewakili suatu urutan kota. Pada proses pergerakan objek yang dirancang pada algoritma *Multi-Verse Optimizer Algorithm*, setiap objek yang terdapat *universe* mengalami perubahan nilai yang menyebabkan perubahan urutan kota pada tiap iterasi. Namun, perubahan nilai objek terjadi secara individual. Hal ini membuat penerapan algoritma ini seperti mencari-cari urutan kota hanya secara *random*.

Berikut ini adalah penjelasan singkat mengenai perubahan nilai objek yang terjadi secara individual. Algoritma *Multi-Verse Optimizer* akan menjalankan serangkaian langkah untuk melakukan perubahan-perubahan nilai awal dari objek-objek pada *universe*. Perubahan nilai dapat terjadi atau tidak bergantung dari langkah yang dirancang oleh MVO. Berikut ini adalah Tabel I.1 yang berisi nilai awal dari tiga *universe* sederhana. Urutan *universe* 1 hingga *universe* 3 dianggap telah diurutkan dari urutan kota yang paling baik hingga paling buruk. Lalu, urutan kunjungan yang terbentuk merupakan hasil *decoding* angka acak yang terbentuk.

Tabel I.1 Contoh Bentuk Awal *Universe* Sederhana

	Objek1	Objek2	Objek3	Objek4	Objek5	Objek6	Objek7	Urutan Kunjungan
<i>Universe 1</i>	0,25	0,52	0,98	0,02	0,57	0,65	0,4	4-1-7-2-5-6-3
<i>Universe 2</i>	0,54	0,72	0,62	0,29	0,09	0,41	0,03	7-5-4-6-1-3-2
<i>Universe 3</i>	0,03	0,96	0,93	0,42	0,63	0,22	0,55	1-6-4-7-5-3-2

Setelah terdapat *universe-universe* awal, selanjutnya akan dilakukan proses eksplorasi yang dirancang pada algoritma MVO. Algoritma MVO akan melakukan eksplorasi dengan menukar-nukar nilai dari satu *universe* ke *universe* lain. Namun, *universe* yang memiliki performa lebih baik akan cenderung mengirimkan objek ke *universe* yang lebih rendah performanya. Berikut ini adalah Tabel I.2 yang berisi contoh hasil eksplorasi yang dilakukan algoritma MVO terhadap contoh pada Tabel I.1.

Tabel I.2 Contoh Bentuk *Universe* yang Ada Setelah Melewati Proses Eksplorasi

	Objek1	Objek2	Objek3	Objek4	Objek5	Objek6	Objek7	Urutan Kunjungan
<i>Universe 1</i>	0,25	0,52	0,98	0,93	0,57	0,65	0,4	1-7-2-5-6-4-3
<i>Universe 2</i>	0,54	0,25	0,62	0,29	0,02	0,41	0,65	5-2-4-6-1-3-7
<i>Universe 3</i>	0,03	0,62	0,93	0,42	0,4	0,22	0,55	1-6-5-4-7-2-3

Pada Tabel I.2, terlihat bahwa urutan kota pada setiap *universe* mampu mengalami perubahan. Hal ini sejalan dengan prinsip eksplorasi dimana setiap kemungkinan yang ada dicoba-coba untuk melihat apakah ada hasil yang lebih baik atau tidak. Selanjutnya, dilakukan proses eksploitasi data *best universe* (dalam penjelasan ini, *best universe* masih merupakan *universe 1*). Proses eksploitasi MVO mengacu pada nilai *best universe*. Namun, masih terdapat variabel lain seperti TDR yang dapat menambahkan dan mengurangi nilai dari *best universe*. Hal ini mengakibatkan perubahan nilai pada *universe* selain *best universe* dapat mengalami perubahan nilai secara drastis. Berikut ini adalah Tabel I.3 yang berisi contoh hasil penerapan eksploitasi MVO pada *universe*.

Tabel I.3 Contoh Bentuk *Universe* yang Ada Setelah Melewati Proses Eksploitasi

	Objek1	Objek2	Objek3	Objek4	Objek5	Objek6	Objek7	Urutan Kunjungan
<i>Universe 1</i>	0,25	0,52	0,98	0,93	0,57	0,65	0,4	1-7-2-5-6-4-3
<i>Universe 2</i>	0,54	0,76	1,06	0,29	0,02	0,31	0,65	5-4-6-1-7-2-3
<i>Universe 3</i>	0,03	1,45	0,76	0,42	0,11	1,21	0,55	1-5-4-7-3-6-2

Pada Tabel I.3, terlihat bahwa urutan kunjungan *universe 2* dan *3* mengalami perubahan dari proses eksplorasi. Namun, tidak terlihat adanya perubahan yang didasarkan pada *best universe* yang ada (*universe 1*). Padahal, prinsip dari proses eksploitasi adalah untuk mendalami informasi pada *best universe* untuk dilihat apakah kemungkinan urutan kota yang ada di sekitar *best universe* memiliki jawaban yang lebih baik atau tidak. Hal inilah yang menyebabkan mengapa algoritma MVO dikatakan hanya seperti mencari urutan kota secara acak saja.

Lantas, muncul pemikiran bahwa apakah terdapat sebuah cara yang dapat membuat pencarian urutan kota yang lebih baik tidak perlu dilakukan *secara random*. Informasi yang ingin dibawa merupakan informasi-informasi yang memiliki nilai yang baik untuk diaplikasikan pada objek-objek di dalam *universe*. Pembuatan informasi yang baik ini diharapkan mampu membuat pencarian urutan kota dapat mencari urutan kota tidak secara *random*.

Metode *divide and conquer* (Brassard, G dan Bratley, P., 1996) menawarkan cara memecah masalah yang ada menjadi beberapa kelompok yang lebih kecil. Memecah masalah yang besar menjadi beberapa masalah yang lebih kecil diharapkan mampu mempermudah penyelesaian masalah secara keseluruhan. Contoh sederhana dalam menggunakan metode *divide and conquer*

adalah mengurutkan 12 bilangan agar memiliki urutan terbesar ke terkecil secara manual. Memecah 12 bilangan menjadi 3 kelompok, lalu menyelesaikan masalah pada kelompok itu dan menggabungkan kelompok yang ada untuk diurutkan kembali akan memudahkan proses pencarian solusi yang dilakukan. Berikut ini adalah Gambar 1.1 yang berisi contoh penerapan metode *divide and conquer* seperti pada contoh yang ada.



Gambar 1.1 Penerapan Metode *Divide and Conquer* pada Kasus Pengurutan Angka

Dengan menggunakan metode *divide and conquer*, diharapkan muncul informasi pengelompokan yang sudah sangat baik dibandingkan dengan kelompok-kelompok lain. Informasi yang didapatkan selanjutnya akan digunakan untuk membangun solusi. Diharapkan, pencarian solusi dapat lebih terarah dengan adanya informasi tambahan mengenai pengelompokan-pengelompokan yang dianggap sudah baik.

Pada penelitian ini, *asymmetric traveling salesman problem* akan dicoba untuk diselesaikan dengan memodifikasi penerapan *Multi-Verse Optimazion algorithm*. Diharapkan, modifikasi penerapan *Multi-Verse Optimazion algorithm*

dapat menyelesaikan *Asymmetric Traveling Salesman Problem* dengan hasil yang lebih baik daripada penerapan *Multi-Verse Optimazion algorithm* yang telah dilakukan sebelumnya. Oleh karena itu, rumusan masalah pada penelitian ini adalah:

1. Bagaimana cara menerapkan metode *divide and conquer* kepada *Multi-Verse Optimizer algorithm* untuk menyelesaikan *Asymmetric Traveling Salesman Problem*?
2. Apakah ukuran pengelompokan kota yang dilakukan berpengaruh kepada hasil implementasi modifikasi yang dilakukan?
3. Bagaimana perbandingan hasil modifikasi penerapan *Multi-Verse Optimizer algorithm* dengan *Multi-Verse Optimizer algorithm* tanpa modifikasi dalam menyelesaikan *Asymmetric Traveling Salesman Problem*?

### **I.3 Pembatasan Masalah**

Batasan dari masalah yang telah dijelaskan pada identifikasi masalah di dalam penelitian ini adalah:

1. Kasus yang digunakan terbatas hanya pada kasus *benchmark*.
2. Waktu proses pencarian solusi bukan tolak ukur performansi algoritma dalam penelitian ini.
3. Jumlah pemotongan maksimum yang dilakukan adalah 10.

### **I.4 Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah dijabarkan pada bagian sebelumnya, berikut ini adalah tujuan penelitian yang akan dilakukan, yaitu:

1. Mengimplementasikan metode *divide and conquer* terhadap *Multi-Verse Optimazion algorithm* dalam menyelesaikan kasus *benchmark* pada kasus *Asymmetric Traveling Salesman Problem*.
2. Mengetahui pengaruh ukuran pengelompokan pada modifikasi *Multi-Verse Optimazion algorithm* terhadap performansinya.
3. Membandingkan performansi modifikasi penerapan *Multi-Verse Optimazion algorithm* dengan *Multi-Verse Optimizer algorithm* yang telah dirancang sebelumnya.

### **I.5 Manfaat Penelitian**

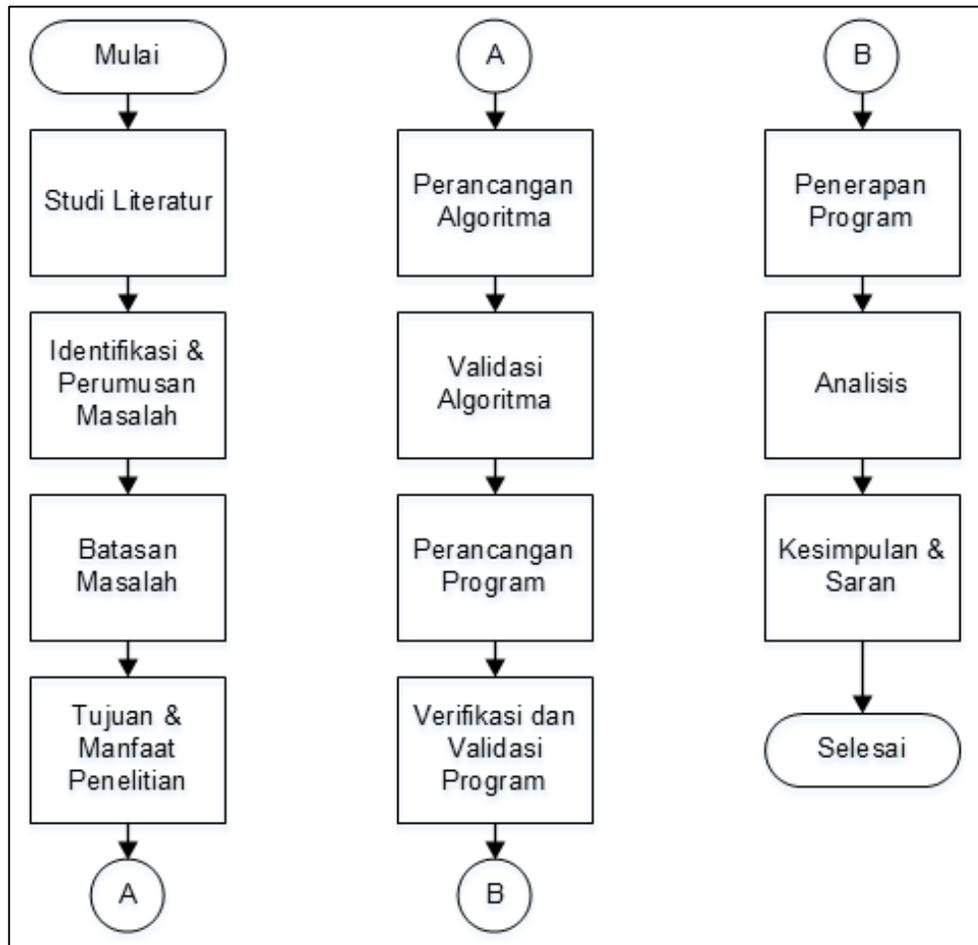
Penelitian yang baik hendaknya memiliki tujuan-tujuan tertentu agar memberikan dampak yang baik terhadap pihak-pihak yang terlibat. Oleh sebab itu, berdasarkan penelitian yang dirancang, manfaat yang diharapkan oleh peneliti untuk diperoleh dari penelitian ini adalah:

1. Menambah pengetahuan mengenai modifikasi penerapan *Multi-Verse Optimazion algorithm* dalam menyelesaikan kasus *Asymmetric Traveling Salesman Problem*.
2. Menambah referensi penelitian dalam menyelesaikan kasus *Asymmetric Traveling Salesman Problem*.

### **I.6 Metodologi Penelitian**

Dalam melakukan penelitian ini, dibutuhkan suatu metodologi penelitian. Metodologi penelitian akan menggambarkan dengan jelas mengenai tahapan-tahapan penelitian yang dilakukan. Metodologi penelitian juga bermanfaat untuk membimbing jalannya suatu penelitian agar tidak menyimpang. *Flow chart* mengenai metodologi penelitian ini terdapat pada Gambar 1.2. Berikut ini adalah metodologi penelitian yang dilakukan.

1. Studi Literatur:  
Pada bagian ini, dilakukan studi mengenai algoritma MVO dan persoalan ATSP. Studi dilakukan dengan mempelajari sumber-sumber terkait.
2. Identifikasi dan Perumusan Masalah:  
Setelah mempelajari literatur terkait, masalah yang ada perlu diidentifikasi keadaannya. Setelah mengetahui masalah yang dihadapi dalam penelitian, selanjutnya dilakukan perumusan masalah yang perlu dijawab pada penelitian.
3. Penentuan Batasan Masalah:  
Pembatasan masalah dilakukan agar penelitian tidak meluas dan lebih terfokus mengenai masalah yang dihadapi.
4. Penentuan Tujuan dan Manfaat Penelitian:  
Penentuan tujuan dan manfaat penelitian ditujukan agar hasil penelitian dapat terukur dan memiliki dampak positif kepada pihak-pihak yang terkait.



Gambar I.2 Metodologi Penelitian

5. Perancangan Modifikasi Penerapan Algoritma MVO untuk Menyelesaikan ATSP:  
Dari studi literatur yang dilakukan, penyesuaian algoritma yang ada perlu dilakukan agar dapat menyelesaikan ATSP.
6. Validasi Rancangan Implementasi:  
Validasi hasil rancangan diperlukan untuk mengecek apakah ada kesalahan dalam perancangan. Tujuan lainnya adalah untuk memastikan rancangan implementasi yang dibuat benar-benar mampu menyelesaikan permasalahan ATSP. Bila hasil validasi berhasil, maka proses berlanjut ke tahap selanjutnya. Bila gagal, perancangan implementasi algoritma perlu dilakukan kembali.

7. Perancangan Program Hasil Implementasi:  
Dari hasil perancangan implementasi, dibuatlah program yang sesuai. Program yang dibuat akan menghasilkan *output* yang sesuai dengan yang diharapkan.
8. Verifikasi dan Validasi Program:  
Program yang sudah dibuat perlu diverifikasi dan divalidasi untuk memastikan kesesuaian dengan rancangan algoritma yang telah dibuat. Bila verifikasi dan validasi program telah dilakukan, maka proses berlanjut ke tahap selanjutnya. Bila gagal, maka proses pembuatan program perlu diulang.
9. Penerapan Program:  
Program yang ada digunakan untuk menyelesaikan masalah-masalah yang digunakan dalam penelitian untuk dilihat hasilnya.
10. Analisis:  
Analisis dilakukan untuk melihat performa rancangan algoritma untuk menyelesaikan masalah bila dibandingkan dengan algoritma-algoritma lainnya yang telah menyelesaikan masalah yang sama. Lalu, analisis mengenai pengaruh parameter-parameter algoritma terhadap hasil pengolahan juga dilakukan.
11. Kesimpulan dan Saran:  
Penelitian yang dilakukan hendaknya menjawab pertanyaan-pertanyaan yang muncul pada awal penelitian. Lalu, saran diberikan untuk penelitian sejenis yang akan dikerjakan selanjutnya supaya hasil penelitian selanjutnya dapat lebih baik.

### **I.7 Sistematika Penulisan**

Penulisan laporan penelitian ini terdiri dari 6 bagian utama. Berikut ini adalah sistematika penulisan laporan penelitian modifikasi penerapan algoritma MVO dalam menyelesaikan ATSP:

### **BAB I PENDAHULUAN**

Bab ini berisi tentang pendahuluan dan latar belakang dari latar belakang penelitian, identifikasi dan perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan laporan penelitian.

## **BAB II TINJAUAN PUSTAKA**

Bab ini membahas mengenai dasar dari penelitian yang dilakukan. Tinjauan yang dimaksud adalah algoritma *Multi-Verse Optimizer*, algoritma *divide and conquer*, metode *two exchange*, *random key representation* dan *permutational encoding*, serta pengujian *analysis of variance*.

## **BAB III PERANCANGAN ALGORITMA**

Perancangan modifikasi algoritma dilakukan pada bagian ini. langkah-langkah perancangan yang dilakukan adalah merancang modifikasi penerapan algoritma MVO dengan menggunakan prinsip *divide and conquer*, perancangan diagram alir modifikasi, serta pengerjaan algoritma yang dibuat secara manual.

## **BAB IV IMPLEMENTASI ALGORITMA**

Algoritma yang telah dirancang pada bagian sebelumnya diimplementasikan ke dalam program komputer. Lalu, dilakukan pemecahan masalah *benchmark*. Bagian-bagian pada bab ini adalah verifikasi dan validasi implementasi program komputer dan implementasi kasus *benchmark* terhadap algoritma yang dibuat.

## **BAB V ANALISIS**

Hasil perancangan algoritma dan implementasi untuk menyelesaikan permasalahan ATSP dilakukan pada bagian ini. Bagian ini berisi analisis rancangan algoritma secara keseluruhan, analisis parameter besar pemotongan kota, serta analisis performansi modifikasi yang dibuat.

## **BAB VI KESIMPULAN DAN SARAN**

Kesimpulan yang ditarik diperoleh berdasarkan seluruh rangkaian penelitian yang dilakukan. Lalu, saran yang diberikan ditujukan kepada penelitian serupa pada masa yang akan datang.