

**PENERAPAN *LIGHTNING SEARCH ALGORITHM*
DENGAN *LOCAL SEARCH* UNTUK
MENYELESAIKAN *ASYMMETRIC TRAVELING*
*SALESMAN PROBLEM***

SKRIPSI

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

Disusun oleh

Nama : Yulius Chandra Gunawan

NPM : 2015610038



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

**FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KATOLIK PARAHYANGAN
BANDUNG**



Nama : Yulius Chandra Gunawan
NPM : 2015610038
Jurusan : Teknik Industri
Judul Skripsi : PENERAPAN *LIGHTNING SEARCH ALGORITHM* DENGAN
LOCAL SEARCH UNTUK MENYELESAIKAN *ASYMMETRIC*
TRAVELING SALESMAN PROBLEM

TANDA PERSETUJUAN SKRIPSI

Bandung, Juli 2019

**Ketua Program Studi Sarjana
Teknik Industri**

A handwritten signature in blue ink, appearing to read "Romy Lofce", is written over a small, faint rectangular stamp.

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

Pembimbing Tunggal

A handwritten signature in black ink, appearing to read "Ignatius A. Sandy", is written over a small, faint rectangular stamp.

(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 di bawah ini,

Nama : Yulius Chandra Gunawan

NPM : 2015610038

dengan ini menyatakan bahwa Skripsi dengan judul :

"PENERAPAN *LIGHTNING SEARCH ALGORITHM* DENGAN *LOCAL SEARCH* UNTUK MENYELESAIKAN *ASYMMETRIC TRAVELING 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, 24 Juli 2019

Yulius Chandra Gunawan
NPM : 2015610038

ABSTRAK

Asymmetric Traveling Salesman Problem (ATSP) adalah suatu permasalahan optimasi dimana terdapat seorang “*salesman*” yang harus mengunjungi beberapa kota dalam satu kali perjalanan. Setiap kota hanya boleh dikunjungi satu kali saja dan pada akhir kunjungan harus kembali ke kota pertama. ATSP merupakan salah satu variasi dari permasalahan *Traveling Salesman Problem* (TSP). Perbedaannya terdapat pada jarak yang ditempuh dari kota i ke j dapat berbeda dengan jarak dari kota j ke i . Tujuan dari ATSP adalah meminimasi jarak total yang ditempuh oleh sang “*salesman*”.

ATSP diselesaikan dengan menggunakan *Lightning Search Algorithm* (LSA) dan *2-Opt Local Search Algorithm*. LSA adalah sebuah algoritma metaheuristik yang terinspirasi dari proses perambatan lidah petir (*step leader*). Terdapat 3 buah parameter pada LSA, yaitu *maximum channel time* (*max_ctime*), *forking probability* (*fork_prob*), dan *Boundaries* (*Bound*). *2-Opt* merupakan algoritma *local search* yang memecah sebuah rute menjadi dua sub-rute, dan salah satu sub-rute dimanipulasi sedemikian rupa yang kemudian kedua sub-rute akan digabungkan kembali.

LSA dengan *local search* telah dirancang dan dapat menyelesaikan permasalahan ATSP. Nilai parameter terbaik didapat dengan metode 2^k *factorial experiment* dan *trial and error*. LSA diimplementasi dan dibandingkan dengan *Elephant Herding Optimization* (EHO), *Harmony Search Algorithm* (HSA), dan *Lion Optimization Algorithm* (LOA). Algoritma LSA mencapai *best known solution* pada kasus BR17, dan menghasilkan solusi yang lebih baik dari LOA pada kasus FTV33, FTV44, FTV55, dan FTV70. LSA juga mampu menghasilkan solusi lebih baik dari HSA tetapi masih lebih buruk dari EHO pada kasus FTV33. Pada kasus FTV44, FTV55, dan FTV70 EHO dan HSA menghasilkan solusi yang lebih baik dari LSA.

ABSTRACT

Asymmetric Traveling Salesman Problem (ATSP) is an optimization problem where there is a “salesman” that has to visit several cities in one trip. Each city can only be visited once and at the end of the trip, the salesman has to come back to the first city he visited. ATSP is one of many variations of Traveling Salesman Problem (TSP). The difference is in the distance travelled from city i to j , could be different from the distance travelled from city j to i . The objective of ATSP is to minimize the total distance travelled by the “salesman”.

ATSP is solved with Lightning Search Algorithm (LSA) and 2-Opt Local Search Algorithm. LSA is a metaheuristic algorithm that was inspired from the propagation of step leader. There are 3 parameters in LSA, those are maximum channel time (max_ctime), forking probability (fork_prob), and Boundaries (Bound). 2-Opt is a local search algorithm that cuts a route into two, so there would be 2 sub-routes, and one of the sub-routes is manipulated in such a way, and then the two sub-routes will be recombined.

LSA with local search had been designed and could solve the ATSP. The best parameters' value was obtained with 2^k factorial experiment and trial and error. LSA was implemented and then compared with Elephant Herding Optimization (EHO), Harmony Search Algorithm (HSA), and Lion Optimization Algorithm (LOA). LSA reached the best known solution in the BR17 case, and produce better solutions than LOA in the cases of FTV33, FTV44, FTV55, and FTV70. LSA also producing a better solution than HSA, but worse than EHO in the FTV33 case. In the cases of FTV44, FTV55, and FTV70, EHO and HSA produce better solutions than LSA.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas berkat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan *Lightning Search Algorithm* dengan *Local Search* untuk Menyelesaikan *Asymmetric Traveling Salesman Problem*”. Skripsi ini disusun dalam rangka memenuhi salah satu syarat guna mencapai gelar sarjana dalam bidang ilmu Teknik Industri.

Selama penulisan skripsi ini, banyak dukungan dan bantuan yang penulis dapatkan, baik secara langsung maupun tidak langsung, sehingga penulis mampu menyelesaikan skripsi dengan baik. Oleh sebab itu, penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua serta kakak dan adik penulis yang sudah memberikan doa, dan selalu mendukung dan membantu segala hal yang penulis butuhkan.
2. Bapak Ignatius A. Sandy, S.Si., M.T. selaku dosen pembimbing, yang telah memberikan ilmu, waktu, tenaga, dan nasihat kepada penulis selama proses pembuatan skripsi ini.
3. Bapak Y. M. Kinley Aritonang, Ph.D. dan Bapak Hanky Franciscus, S.T., M.T. selaku dosen penguji proposal yang telah memberikan kritik dan saran yang membantu penulis dalam pembuatan skripsi.
4. Bapak Hanky Franciscus, S.T., M.T. dan Ibu Titi Iswari, S.T., M.Sc., M.B.A. selaku dosen penguji dalam sidang skripsi.
5. Elvina yang selalu menemani, membantu, menghibur, dan mendukung penulis selama pembuatan skripsi.
6. Teman-teman terdekat penulis, yaitu Gilland Dyell, Evelyn, Eder Varian, Natasya, Garinsa Fantio, Winny Wirianta, Stacia Marella, Giovano Alberto, dan Felicia Naomi yang sudah menyemangati dan mewarnai kehidupan perkuliahan penulis.
7. Teman-teman D2, yaitu Hastomo, Adi Anjoyo, Vieri Gunawan, Kevin Grahadian, dan Yogi Chandres yang selalu menghibur penulis.
8. Teman-teman kelas C angkatan 2015 atas kebersamaan, perjuangan, pengalaman dan dukungan selama masa perkuliahan.

9. Tim Asisten Pemrograman Komputer dan Tim Asisten Simulasi Sistem atas pengalaman dan dukungannya selama ini.
10. Seluruh pihak lain yang terlibat selama masa penyusunan skripsi, masa perkuliahan, dan masa pengembangan diri penulis yang tidak dapat disebutkan satu per satu.

Penulis sadar atas keterbatasan dan kekurangan pada penelitian ini. Oleh sebab itu, penulis sangat terbuka untuk menerima kritik dan saran yang dapat berguna bagi penelitian ini. Penulis berharap penelitian ini dapat digunakan bagi seluruh pihak yang terkait dan mampu menjadi referensi dalam penelitian selanjutnya di masa depan.

Bandung, Juli 2019

Penulis

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	I-1
I.1 Latar Belakang Masalah.....	I-1
I.2 Identifikasi dan Rumusan Masalah	I-2
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
BAB II TINJAUAN PUSTAKA	II-1
II.1 <i>Traveling Salesman Problem (TSP)</i>	II-1
II.2 <i>Asymmetric Traveling Salesman Problem (TSP)</i>	II-1
II.3 <i>Metaheuristic Algorithm</i>	II-2
II.4 <i>Lightning Search Algorithm (LSA)</i>	II-3
II.4.1 <i>Projectile dan Perambatan Step Leader</i>	II-3
II.4.2 Sifat-sifat <i>Projectile</i>	II-3
II.4.3 Pemodelan <i>Projectile</i> dan Pergerakan <i>Step Leader</i>	II-5
II.5 <i>Local Search: 2-Opt Algorithm</i>	II-8
II.6 <i>Permutation Encoding & Random Key Encoding</i>	II-9
II.7 2^k <i>Factorial Experimental Design</i>	II-9
II.8 <i>Analysis of Variance (ANOVA)</i>	II-10
BAB III PERANCANGAN ALGORITMA	III-1
III.1 Identifikasi Elemen ATSP dan LSA.....	III-1
III.2 <i>Lightning Search Algorithm</i> pada ATSP	III-3

III.3 Pencarian Solusi dengan LSA.....	III-4
III.3.1 Inisialisasi Populasi Awal.....	III-6
III.3.2 <i>Forking 2</i>	III-10
III.3.3 Percobaan Perpindahan Posisi Step Leader	III-11
III.4 Perancangan LSA untuk ATSP	III-16
III.4.1 Algoritma Utama (Algoritma A).....	III-16
III.4.2 Algoritma Inisialisasi Populasi Awal (Algoritma B).....	III-20
III.4.3 Algoritma <i>getRoute</i> (Algoritma Z).....	III-22
III.4.4 Algoritma <i>getEnergy</i> (Algoritma X).....	III-24
III.4.5 Algoritma <i>Forking 2</i> (Algoritma C)	III-26
III.4.6 Algoritma Pencarian Posisi <i>Space & Lead Projectile</i> (Algoritma D).....	III-27
III.4.7 Algoritma <i>Forking 1</i> (Algoritma E)	III-31
III.4.8 Algoritma Perpindahan Step Leader (Algoritma F)	III-33
III.4.9 Algoritma Pemilihan Step Leader (Algoritma A)	III-34
III.5 Validasi Algoritma LSA	III-35
III.6 Perancangan Program.....	III-37
III.6.1 Verifikasi Program.....	III-37
III.6.2 Verifikasi Program Algoritma	III-43
III.7 Percobaan Implementasi	III-47
III.8 Perancangan Algoritma <i>Local Search</i>	III-48
III.9 Perbandingan Hasil Algoritma LSA Tanpa <i>Local Search</i> dengan Hasil Algoritma dengan <i>Local Search</i>	III-54
BAB IV IMPLEMENTASI ALGORITMA	IV-1
IV.1 Penentuan Parameter <i>Lightning Search Algorithm</i>	IV-1
IV.2 Pengujian dan Penentuan Nilai Parameter LSA.....	IV-6
IV.2.1 Pengujian dan Penentuan Nilai Parameter pada kasus BR17	IV-6
IV.2.2 Pengujian dan Penentuan Nilai Parameter pada kasus FTV33	IV-7
IV.2.3 Pengujian dan Penentuan Nilai Parameter pada kasus FTV44	IV-10
IV.2.4 Pengujian dan Penentuan Nilai Parameter pada kasus FTV55	IV-13

IV.2.5 Pengujian dan Penentuan Nilai Parameter pada kasus FTV70.....	IV-15
IV.3 Implementasi dan Perbandingan Hasil Algoritma	IV-18
BAB V ANALISIS	IV-1
V.1 Analisis <i>Encoding</i> dan <i>Decoding</i>	IV-1
V.2 Analisis <i>Local Search</i>	IV-3
V.3 Analisis Parameter <i>Lightning Search Algorithm</i>	IV-4
V.3.1 Analisis Parameter <i>max_ctime</i>	IV-4
V.3.2 Analisis Parameter <i>fork_prob</i>	IV-5
V.3.3 Analisis Parameter <i>Bound</i>	IV-6
V.4 Analisis Performansi Algoritma LSA.....	IV-7
BAB VI KESIMPULAN DAN SARAN	IV-1
VI.1 Kesimpulan.....	IV-1
VI.2 Saran.....	IV-2
DAFTAR PUSTAKA	
LAMPIRAN	
RIWAYAT HIDUP PENULIS	

DAFTAR TABEL

Tabel II.1	Contoh Tabel ANOVA.....	II-11
Tabel III.1	Elemen-elemen ATSP	III-2
Tabel III.2	Elemen-elemen LSA.....	III-3
Tabel III.3	Elemen ATSP yang Diakomodasi LSA	III-4
Tabel III.4	Contoh Vektor Posisi <i>Transition Projectile</i> untuk ATSP	III-7
Tabel III.5	Contoh Matriks <i>Transition Projectile</i> untuk ATSP	III-7
Tabel III.6	Contoh Populasi Awal.....	III-8
Tabel III.7	Contoh Hasil <i>Encoding</i>	III-8
Tabel III.8	Contoh Hasil Pengurutan Bilangan Acak	III-9
Tabel III.9	Contoh Hasil <i>Decoding</i>	III-9
Tabel III.10	Contoh Matriks Jarak.....	III-10
Tabel III.11	Perhitungan Jarak Populasi Awal.....	III-10
Tabel III.12	Hasil Inisialisasi Populasi Awal.....	III-10
Tabel III.13	<i>Step Leader</i> yang Akan Mengalami <i>Forking 2</i>	III-11
Tabel III.14	Hasil Eliminasi <i>Step Leader</i> dan Penambahan <i>Step Leader</i> Baru	III-10
Tabel III.15	Hasil Pencarian Posisi Oleh <i>Lead Projectile</i>	III-13
Tabel III.16	Hasil Pencarian Posisi Oleh <i>Space Projectile</i>	III-14
Tabel III.17	Contoh Posisi <i>Forked Projectile</i>	III-15
Tabel III.18	Contoh Perubahan Posisi <i>new_pos</i>	III-15
Tabel III.19	Hasil Perpindahan Posisi <i>Step Leader</i>	III-16
Tabel III.20	Keterangan Notasi pada Algoritma Utama.....	III-19
Tabel III.21	Keterangan Notasi pada Algoritma Inisialisasi Populasi Awal....	III-22
Tabel III.22	Keterangan Notasi pada Algoritma <i>getRoute</i>	III-24
Tabel III.23	Keterangan Notasi pada Algoritma <i>getEnergy</i>	III-26
Tabel III.24	Keterangan Notasi pada Algoritma <i>Forking 2</i>	III-26
Tabel III.25	Keterangan Notasi pada Algoritma Pencarian Posisi <i>Space & Lead Projectile</i>	III-31
Tabel III.26	Keterangan Notasi pada Algoritma <i>Forking 1</i>	III-33
Tabel III.27	Keterangan Notasi pada Algoritma Perpindahan <i>Step Leader</i> ...	III-33

Tabel III.28	Keterangan Notasi pada Algoritma Pemilihan <i>Step Leader</i>	III-34
Tabel III.29	Kasus ATSP untuk Validasi Program	III-43
Tabel III.30	Hasil Uji Coba pada Kasus BR17 dan FTV33	III-47
Tabel III.31	Hasil Uji Coba Kedua pada Kasus FTV33.....	III-48
Tabel III.32	Keterangan Notasi pada Algoritma 2-Opt.....	III-50
Tabel III.33	Hasil Uji Coba dengan 2-Opt pada Kasus BR17 dan FTV33.....	III-54
Tabel III.34	Data yang Digunakan pada t-test	III-55
Tabel IV.1	Perancangan Eksperimen dengan 2^3 <i>Factorial Design</i>	IV-5
Tabel IV.2	Hasil Eksperimen pada Kasus BR17	IV-7
Tabel IV.3	Hasil Eksperimen pada Kasus FTV33	IV-7
Tabel IV.4	Hasil Eksperimen pada Kasus FTV44	IV-10
Tabel IV.5	Hasil Eksperimen pada Kasus FTV55	IV-14
Tabel IV.6	Hasil Eksperimen pada Kasus FTV70	IV-15
Tabel IV.7	Rekapitulasi Nilai Parameter.....	IV-18
Tabel IV.8	Implementasi Algoritma LSA dengan <i>Local Search</i> pada Kasus <i>Benchmark</i>	IV-18
Tabel IV.9	Perbandingan Performansi LSA dengan Algoritma Lainnya.....	IV-20

DAFTAR GAMBAR

Gambar I.1	Flowchart Metodologi Penelitian	I-7
Gambar II.1	Ilustrasi LSA dalam Mencari Global Optimum	II-4
Gambar II.2	Flow Chart LSA	II-7
Gambar II.3	2-Opt Algorithm: (a) Tur Awal (b) Tur Setelah 2-Opt.....	II-8
Gambar II.4	Random Key Encoding.....	II-9
Gambar II.5	Contoh Kombinasi Setiap Level Faktor.....	II-10
Gambar III.1	Langkah-langkah Besar dalam Pencarian Solusi oleh LSA	III-5
Gambar III.2	Proses Decoding	III-9
Gambar III.3	Flow Chart Algoritma Utama LSA untuk ATSP.....	III-18
Gambar III.4	Algoritma Inisialisasi Populasi Awal Bagian 1	III-20
Gambar III.5	Algoritma Inisialisasi Populasi Awal Bagian 2	III-21
Gambar III.6	Algoritma getRoute.....	III-23
Gambar III.7	Algoritma getEnergy	III-25
Gambar III.8	Algoritma Forking 2	III-27
Gambar III.9	Algoritma Pencarian Posisi Space & Lead Projectile Bagian 1.....	III-28
Gambar III.10	Algoritma Pencarian Posisi Space & Lead Projectile Bagian 2.....	III-29
Gambar III.11	Algoritma Pencarian Posisi Space & Lead Projectile Bagian 3.....	III-30
Gambar III.12	Algoritma Forking 1	III-32
Gambar III.13	Algoritma Perpindahan Step Leader.....	III-34
Gambar III.14	Algoritma Pemilihan Step Leader Terbaik	III-35
Gambar III.15	Ilustrasi Algoritma 2-Opt yang Dirancang	III-49
Gambar III.16	Algoritma 2-Opt	III-51
Gambar III.17	Algoritma Utama dengan 2-Opt	III-52
Gambar III.18	Uji t 2 Sampel	III-55
Gambar IV.1	Iterasi pada Kasus BR17.....	IV-2
Gambar IV.2	Iterasi pada Kasus FTV33.....	IV-2

Gambar IV.3 Iterasi pada Kasus FTV44	IV-3
Gambar IV.4 Iterasi pada Kasus FTV55	IV-4
Gambar IV.5 Iterasi pada Kasus FTV70	IV-4
Gambar IV.6 <i>Interval Plot</i> Kasus FTV33 dengan Nilai Bound 100, 600, dan 1000	IV-8
Gambar IV.7 <i>Interval Plot</i> Kasus FTV33 dengan Tambahan Bound 50 dan 150	IV-9
Gambar IV.8 <i>Interval Plot</i> Kasus FTV33 dengan Tambahan Bound 125 dan 175	IV-10
Gambar IV.9 <i>Interval Plot</i> Kasus FTV44 dengan Nilai Bound 100, 600, dan 1000	IV-11
Gambar IV.10 <i>Interval Plot</i> Kasus FTV44 dengan Tambahan Bound 50 dan 150	IV-12
Gambar IV.11 <i>Interval Plot</i> Kasus FTV33 dengan Tambahan Bound 75 dan 125	IV-12
Gambar IV.12 <i>Interval Plot</i> Kasus FTV33 dengan Tambahan Hasil Bound 1200	IV-13
Gambar IV.13 <i>Interval Plot</i> Kasus FTV33 dengan Nilai max_ctime 1% - 5%.	IV-15
Gambar IV.14 <i>Interval Plot</i> Kasus FTV70 dengan Nilai Bound 100, 600, dan 1000	IV-16
Gambar IV.15 <i>Interval Plot</i> Kasus FTV70 dengan Tambahan Bound 50 dan 150	IV-17
Gambar IV.16 <i>Interval Plot</i> Kasus FTV70 dengan Tambahan Bound 75 dan 125	IV-17

DAFTAR LAMPIRAN

Lampiran A: Proses Validasi Algoritma

Lampiran B: Kasus *Benchmark*

BAB I

PENDAHULUAN

I.1 Latar Belakang Masalah

Pada dunia industri saat ini, banyak sekali masalah yang terkait dengan transportasi. Fokus dari masalah transportasi ini biasanya adalah bagaimana seseorang atau sebuah perusahaan dapat meminimasi biaya yang harus dikeluarkan untuk transportasi tersebut. Sebagai contoh adalah *travel agent* yang harus menentukan rute perjalanan untuk konsumen yang akan melakukan tur. *Travel agent* harus bisa memilih rute terbaik agar setiap tempat yang menjadi tujuan akan didatangi dan jarak yang ditempuh minimum. Dengan jarak yang minimum tersebut, *travel agent* dapat mengurangi biaya yang harus dikeluarkan untuk transportasi tambahan.

Permasalahan yang telah disebutkan pada contoh sebelumnya merupakan salah satu contoh dari *traveling salesman problem* (TSP). TSP merupakan pemilihan rute dari seorang *salesman* yang dimulai dari lokasi awal, yang kemudian mengunjungi sekumpulan kota yang telah ditentukan, dan kembali ke lokasi awal dengan cara sedemikian rupa sehingga total jarak yang ditempuh minimum, dan setiap kota hanya didatangi satu kali saja (Gutin & Punnen, 2002). TSP termasuk ke dalam *combinatorial optimization problem*, yang artinya permasalahan tersebut memiliki sejumlah solusi yang *feasible* (Winston, 2004). Menurut Brest dan Zerovnik (2003), TSP juga termasuk ke dalam kategori *NP-hard problem* (*Non-deterministic Polynomial-time Hard Problem*) yang artinya semakin kompleks suatu permasalahan, waktu yang dibutuhkan untuk mendapatkan hasil optimal dari masalah tersebut akan semakin lama.

Menurut Gutin dan Punnen (2002), permasalahan TSP dapat dibagi menjadi dua kelas, yaitu *Symmetric TSP* (STSP) dan *Asymmetric TSP* (ATSP). Perbedaan STSP dengan ATSP terdapat pada jarak antar kota. Sebagai contoh, jika jarak dari kota *i* ke kota *j* dan dari kota *j* ke kota *i* sama, maka permasalahan tersebut termasuk ke dalam STSP. Jika jarak kota *i* ke kota *j* dengan kota *j* ke kota *i* berbeda, permasalahan tersebut adalah permasalahan ATSP.

ATSP merupakan suatu permasalahan yang paling mirip dengan keadaan di dunia nyata, karena di dunia nyata sendiri terdapat banyak rute yang menghubungkan dua lokasi. Jika seseorang memilih rute A, belum tentu orang tersebut akan menggunakan rute A untuk kembali ke lokasi awal. Dengan demikian, jarak yang ditempuh orang tersebut saat pergi dan saat kembali akan berbeda. Perbedaan rute pergi dan rute kembali ini dapat disebabkan oleh beberapa hal seperti penutupan jalan yang memaksa orang untuk menggunakan rute lain, adanya larangan untuk melewati rute tersebut pada jam-jam tertentu, dan lain-lain.

Menurut Talbi (2009), untuk suatu permasalahan yang mirip dengan keadaan dunia nyata, tidak dapat digunakan metode analitik untuk menyelesaikan permasalahan tersebut dengan waktu yang terbatas. Hal ini dikarenakan metode analitik memakan waktu yang sangat lama untuk mendapatkan solusi yang optimal. Oleh sebab itu, untuk menyelesaikan ATSP yang merupakan salah satu permasalahan yang mirip dengan dunia nyata, dapat dikembangkan metode alternatif yang menggunakan pendekatan heuristik atau metaheuristik untuk mencari solusi yang dianggap baik dan dapat diterima.

I.2 Identifikasi dan Rumusan Masalah

Penyelesaian masalah ATSP dapat dilakukan dengan menggunakan metode analitik dan metode heuristik. Contoh metode analitik yang dapat digunakan untuk menyelesaikan ATSP adalah *branch and bound*, dan *dynamic programming*. Namun, jika jumlah kota sangat banyak, mencari solusi menggunakan metode analitik akan memakan waktu yang sangat lama sekali. Sehingga pencarian solusi dari rute terpendek dapat dilakukan dengan menggunakan metode heuristik.

Metode heuristik dapat dibagi ke dalam dua jenis, yaitu metaheuristik dan heuristik problema-spesifik. Metaheuristik dapat digunakan secara umum dan dapat digunakan untuk hampir semua permasalahan optimasi. Sedangkan heuristik problema-spesifik hanya bisa digunakan untuk masalah yang spesifik. Solusi yang dihasilkan dari metode heuristik ini pun belum tentu optimal, melainkan hanya solusi yang dianggap baik dan dapat diterima (Talbi, 2009).

Beberapa metode metaheuristik telah digunakan untuk menyelesaikan permasalahan ATSP, seperti *Harmony Search Algorithm* (Kevin, 2016), *Elephant*

Herding Optimization (Santosa, 2017), dan *Lion Optimization Algorithm* (Elim, 2018). Penyelesaian kasus ATSP tidak hanya dapat menggunakan metode metaheuristik yang telah disebutkan sebelumnya, melainkan dapat dikembangkan sistem penyelesaian ATSP dengan menggunakan metaheuristik lainnya, yang diharapkan dapat menghasilkan solusi yang lebih baik dari metode metaheuristik yang sudah digunakan.

Lightning Search Algorithm (LSA) dirancang oleh Hussain Shareef, Ahmad Asrul Ibrah, dan Ammar Hussein Mutlag pada tahun 2015. LSA merupakan sebuah algoritma metaheuristik yang terinspirasi dari mekanisme perambatan lidah petir (*step leader*) menggunakan partikel yang sangat cepat, yang disebut sebagai *projectile*. *Projectile* akan mencari jalur untuk *step leader* melakukan perambatan. Terdapat tiga jenis *projectile* yang terdapat dalam LSA, yaitu *transition projectile* yang menginisiasikan populasi dari solusi-solusi awal, *space projectile* dan *lead projectile* akan melakukan eksplorasi dan eksploitasi untuk mencari solusi yang dianggap baik. Selain itu, LSA juga memiliki parameter-parameter yang dapat diatur sedemikian rupa agar LSA dapat menghasilkan solusi terbaik. Sebagai contoh, terdapat parameter *maximum channel time*, *forking probability*, dan lainnya yang dibahas pada bab-bab berikutnya.

LSA sebelumnya telah digunakan untuk menyelesaikan 23 *benchmark functions* dan satu buah kasus optimasi berupa kasus *Symmetric TSP*. Dalam menyelesaikan kasus-kasus tersebut, LSA dibandingkan dengan *Differential Search Algorithm* (DSA), *Bactracking Search Algorithm* (BSA), *Firefly Algorithm* (FFA), *Particle Swarm Optimization* (PSO), dan *Harmony Search Algorithm* (HSA). *Benchmark functions* yang digunakan memiliki karakteristik seperti *linear*, *non-linear*, *continuous*, *discontinuous*, *unimodal*, *multimodal*, *convex*, *non-convex*, *separable*, dan *non-separable*. Pada *benchmark functions* dengan karakteristik *unimodal* dan *separable*, LSA menunjukkan performansi yang dapat diterima. Pada *benchmark functions* dengan karakteristik *unimodal* dan *non-separable*, LSA memiliki performansi yang baik. Pada *benchmark functions* dengan karakteristik *multimodal* dan *separable*, LSA mampu menunjukkan performansi yang baik. Pada *benchmark functions* dengan karakteristik *multimodal* dan *non-separable high- & low-dimensional* mampu menunjukkan performansi yang memuaskan. Pada kasus *Symmetric TSP*, LSA mampu

menghasilkan solusi yang paling baik di antara kelima algoritma lainnya. Secara umum, LSA memberikan hasil yang lebih baik dibandingkan dengan kelima metode metaheuristik lainnya dengan tingkat konvergensi yang tinggi.

Local search algorithm merupakan sebuah algoritma yang termasuk ke dalam kategori *single-solution based search*. Artinya, *local search* melakukan manipulasi dan melakukan perubahan hanya pada sebuah solusi saja. *Local search* melakukan pencarian solusi di sekitar solusi yang ada, dan berfungsi untuk melakukan pencarian *local optima*. Contoh dari *local search algorithm* adalah, *hill climbing*, dan 2-opt (Talbi, 2009). 2-opt pertama kali digagas oleh G. A. Croes pada tahun 1958 dan merupakan algoritma *local search* yang paling sering digunakan untuk permasalahan TSP. 2-opt menghasilkan variasi solusi yang lebih kecil untuk permasalahan ATSP dibandingkan dengan *local search* lainnya.

Metode metaheuristik yang digunakan dalam penelitian ini adalah *Lightning Search Algorithm* (LSA). Shareef et. al. (2015) mengatakan bahwa LSA dirancang untuk dapat menyelesaikan permasalahan optimasi. Selain itu, dari hasil *benchmark* TSP yang telah dilakukan sebelumnya, LSA mampu memberikan hasil yang lebih baik dari algoritma lainnya. *Local search* yang akan digunakan pada penelitian ini adalah 2-opt, yang mampu memberikan hasil yang baik untuk TSP. Sehingga dengan potensi yang dimiliki, pada penelitian kali ini akan digunakan LSA dengan 2-opt *local search* untuk menyelesaikan kasus ATSP. Hasil penelitian akan dibandingkan dengan hasil yang didapat oleh algoritma lainnya, yaitu *Elephant Herding Optimization* (Santosa, 2017), *Harmony Search Algorithm* (Kevin, 2016), dan *Lion Optimization Algorithm* (Elim, 2018). Ketiga algoritma tersebut dipilih sebagai pembanding karena ketiga algoritma tersebut telah menyelesaikan kasus *benchmark* yang digunakan dalam penelitian ini. Oleh karena itu, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan *Lightning Search Algorithm* dengan *local search* dalam menyelesaikan kasus *Asymmetric Traveling Salesman Problem*?
2. Bagaimana pengaruh dari parameter-parameter *Lightning Search Algorithm* terhadap performansinya?
3. Bagaimana perbandingan performansi *Lightning Search Algorithm* dengan *local search*, dibandingkan dengan performansi *Harmony*

Search Algorithm (Kevin, 2016), *Elephant Herding Optimization Algorithm* (Santosa, 2017), dan *Lion Optimization Algorithm* (Elim, 2018)?

I.3 Pembatasan Masalah

Berdasarkan hasil identifikasi dan perumusan masalah pada bagian III, maka batasan masalah perlu didefinisikan. Batasan masalah dalam penelitian ini adalah:

1. Kasus yang digunakan dalam penelitian hanya kasus *benchmark* dari TSPLIB dengan kode kasus BR17, FTV33, FTV44, FTV55, dan FTV70.
2. Waktu penyelesaian masalah atau pencarian solusi tidak menjadi ukuran performansi algoritma dalam penelitian ini.

I.4 Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah:

1. Melakukan implementasi terhadap *Lightning Search Algorithm* dengan *local search* untuk menyelesaikan *Asymmetric Traveling Salesman Problem*.
2. Mengetahui pengaruh parameter-parameter pada *Lightning Search Algorithm* terhadap performansi yang dihasilkan.
3. Membandingkan performansi dari *Lightning Search Algorithm* dengan *local search* terhadap performansi *Harmony Search Algorithm* (Kevin, 2016), *Elephant Herding Optimization Algorithm* (Santosa, 2017), dan *Lion Optimization Algorithm* (Elim, 2018), pada kasus *Asymmetric Travelling Salesman Problem*.

I.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

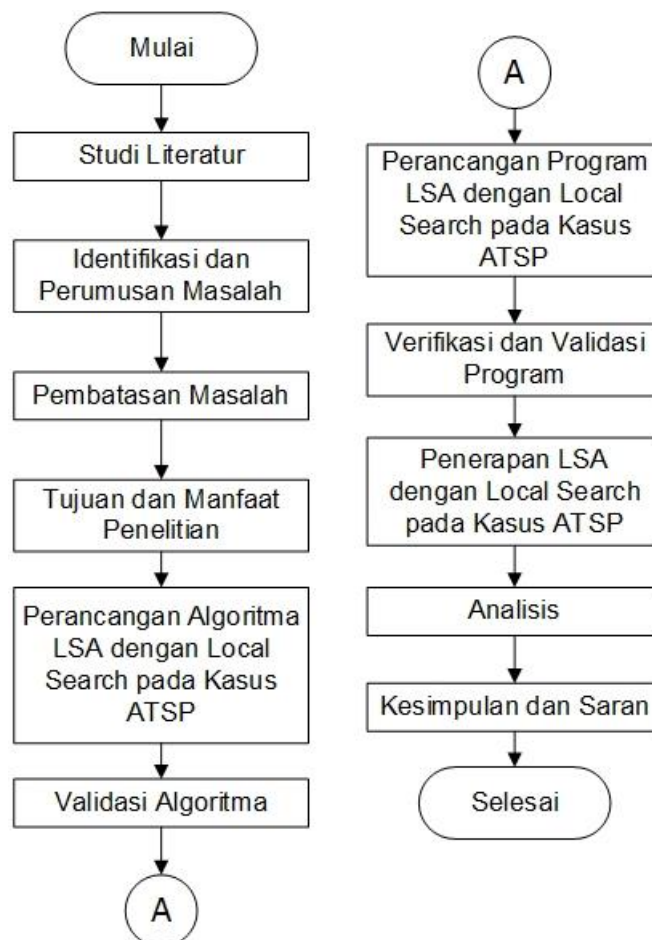
1. Menambah pengetahuan dan wawasan mengenai penyelesaian kasus *Asymmetric Travelling Salesman Problem*.
2. Menambah referensi untuk penelitian mengenai *Lightning Search Algorithm* dan *Asymmetric Travelling Salesman Problem*.

I.6 Metodologi Penelitian

Dalam melakukan sebuah penelitian perlu adanya metodologi penelitian yang berguna untuk membimbing jalannya penelitian. Metodologi penelitian juga menjelaskan tahapan dari penelitian yang dilakukan. Gambar 4 merupakan *flow chart* dari metodologi penelitian. Berikut ini akan dijelaskan mengenai metodologi penelitian yang dilakukan:

1. Studi Literatur.
Langkah pertama dalam penelitian ini adalah melakukan studi literatur. Studi yang dilakukan adalah mengenai algoritma LSA, dan permasalahan ATSP, yang dikutip melalui referensi jurnal ataupun buku.
2. Identifikasi dan Perumusan Masalah.
Identifikasi masalah dilakukan berdasarkan hasil studi literatur. Kemudian dilanjutkan dengan merumuskan masalah dari hasil identifikasi masalah.
3. Pembatasan Masalah.
Pembatasan masalah dilakukan agar penelitian tidak menyimpang dan tetap pada tujuan awalnya.
4. Penentuan Tujuan dan Manfaat Penelitian.
Setelah dilakukan perumusan dan pembatasan masalah, ditentukan tujuan dan manfaat dari penelitian agar penelitian menjadi terarah dan hasil penelitian dapat terukur serta bermanfaat bagi pihak terkait.
5. Perancangan Penerapan Algoritma LSA untuk permasalahan ATSP.
Pada bagian ini dilakukan penyesuaian algoritma LSA berdasarkan hasil studi literatur. Penyesuaian ini dilakukan dengan harapan didapatkan solusi yang lebih baik untuk menyelesaikan masalah ATSP.
6. Validasi Rancangan Algoritma
Validasi rancangan algoritma dilakukan untuk mengetahui apakah ada kesalahan pada saat melakukan perancangan algoritma. Selain itu, validasi rancangan algoritma juga berguna untuk mengetahui apakah algoritma yang dibuat dapat menyelesaikan permasalahan ATSP atau tidak.
7. Perancangan Program
Setelah rancangan algoritma dinyatakan valid, dilakukan perancangan program menggunakan program MATLAB.

8. Verifikasi dan Validasi Program
Program yang dibuat harus diverifikasi dan divalidasi berdasarkan algoritma yang telah dibuat sebelumnya. Jika sudah diverifikasi dan divalidasi, maka penelitian lanjut ke tahap berikutnya. Jika gagal, maka kembali ke perancangan program.
9. Penerapan Program
Program yang dibuat kemudian diterapkan untuk menyelesaikan masalah ATSP. Penerapan ini menggunakan kasus *benchmark* yang terdapat pada *TSP Library*.
10. Analisis
Hasil dari penerapan program dianalisis untuk melihat performansi dari algoritma yang telah dibuat. Kemudian performansi tersebut akan dibandingkan dengan performansi dari algoritma lain yang telah digunakan untuk menyelesaikan kasus *benchmark* ATSP.



Gambar I.4. Metodologi Penelitian

11. Kesimpulan dan Saran

Kesimpulan diberikan berdasarkan hasil dari analisis, dan diberikan pula saran untuk penelitian berikutnya.

I.7 Sistematika Penulisan

Laporan penelitian ini akan dibagi ke dalam 6 bab. Berikut ini akan dijelaskan tentang sistematika penulisan laporan penelitian penerapan algoritma LSA dengan *local search* untuk menyelesaikan ATSP:

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang masalah, identifikasi dan perumusan masalah, pembatasan masalah, asumsi penelitian, tujuan penelitian, manfaat penelitian, dan yang terakhir adalah sistematika penulisan

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori yang digunakan dalam melakukan penelitian. Teori-teori yang dibahas yaitu, *traveling salesman problem*, *metaheuristic algorithm*, *lightning search algorithm*, algoritma 2-Opt, *random-key encoding*, perancangan eksperimen, dan *analysis of variance*.

BAB III PERANCANGAN ALGORITMA

Bab ini membahas tentang proses perancangan algoritma LSA dengan *local search*, yang dimulai dari identifikasi elemen ATSP dan LSA, proses pencarian solusi, merancang algoritma serta *flow chart*, serta melakukan pengerjaan secara manual untuk validasi algoritma.

BAB IV IMPLEMENTASI ALGORITMA

Pada bab ini dibahas tentang penerapan algoritma menggunakan *software* komputer, yaitu MATLAB. Program algoritma LSA akan diverifikasi dan validasi, yang kemudian dilakukan implementasi pada kasus *benchmark*. Berdasarkan hasil implementasi, dilakukan pengujian parameter dari algoritma LSA.

BAB V ANALISIS

Bab ini akan berisi tentang analisis dari rancangan algoritma, analisis perancangan eksperimen, analisis hasil uji parameter algoritma LSA, dan analisis perbandingan hasil algoritma LSA dengan *local search* dengan algoritma lainnya.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini, akan ditarik kesimpulan berdasarkan dari hasil penelitian yang dilakukan, dan pemberian saran. Kesimpulan akan menjawab hasil dari rumusan masalah, dan saran yang diberikan adalah untuk penelitian serupa di masa depan.