



***ANT COLONY OPTIMIZATION* DALAM PENYELESAIAN
TRAVELLING SALESMAN PROBLEM
MENGUNAKAN MATLAB**

Skripsi

disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Program Studi Matematika

oleh
UNNES
Metrika Indra Buana
UNIVERSITAS NEGERI SEMARANG
4111410008

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2016

PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.



Semarang, 20 Januari 2016

Metrika Indra Buana
NIM 4111410008

UNNES
UNIVERSITAS NEGERI SEMARANG

PENGESAHAN

Skripsi yang berjudul:

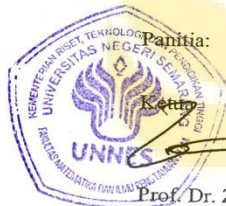
Ant Colony Opimization dalam Penyelesaian *Travelling Salesman Problem*
Menggunakan MATLAB

disusun oleh

Metrika Indra Buana

4111410008

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi Fakultas Matematika
dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang pada tanggal 20
Januari 2016.



Panitia:

Ketua

Prof. Dr. Zaenuri, S.E., M.Si., Akt.
NIP. 196412231988031001

Penguji 1

Dr. Dwijanto, M.S.
NIP. 195804301984031006

Anggota Penguji /Pembimbing

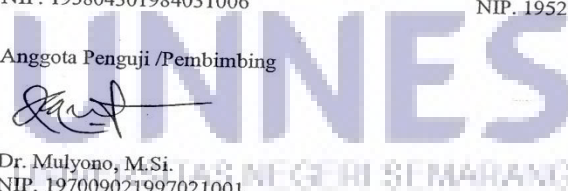
Dr. Mulyono, M.Si.
NIP. 197009021997021001

Sekretaris

Drs. Arief Agoestanto, M.Si.
NIP. 196807221993031005

Penguji 2

Dr. Amin Suyitno, M.Pd.
NIP. 195206041976121001



MOTTO DAN PERSEMBAHAN

Motto

- Siapa ingin mutiara, harus berani terjun di lautan yang dalam. (Ir. Soekarno)
- Saya bangun, saya bimbingan, saya ujian, saya revisi, dan saya menang.

Persembahan

Skripsi ini kupersembahkan untuk :

- Untuk bapak dan ibu yang tercinta yang selalu mendoakan saya.
- Untuk kakak dan adik-adikku yang tersayang.
- Untuk Henny Prasetyani yang selalu sabar menunggu dan menemani.
- Untuk Rectangga, Yoko, Hendra, Fajar, Prakoso, Wawan, Deda, dan semua teman yang selama ini memberikan dukungan, dan semangat kepada saya selama menempuh kuliah di UNNES.
- Untuk keluarga Mamang Uhen dan Mamang 5 yang selalu memberi asupan gizi selama kuliah di UNNES.
- Untuk semua mahasiswa matematika angkatan 2010 yang tercinta.

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT atas segala rahmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "***Ant Colony Opimization dalam Penyelesaian Travelling Salesman Problem Menggunakan MATLAB***".

Selanjutnya penyusun berterima kasih atas bantuan dan peran yang tidak dapat didefinisikan satu persatu pada tahapan penyelesaian skripsi ini, kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri, S.E., M.Si., Akt., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Drs. Mashuri, M.Si, Ketua Prodi Matematika FMIPA Universitas Negeri Semarang.
5. Dr. Mulyono, M.Si., selaku dosen pembimbing yang senantiasa meluangkan waktu untuk membimbing dan memberi masukan serta motivasi selama penyusunan skripsi.
6. Segenap civitas matematika Unnes, khususnya Bapak dan Ibu dosen yang telah memberikan ilmunya dengan tulus.
7. Teman-teman Mentalhouse dan Proxima yang selalu menghibur dan telah berjuang bersama dalam suka dan duka.

8. Seluruh pihak yang turut membantu penyelesaian skripsi yang tidak dapat penyusun sebutkan satu persatu.

Hanya ucapan terima kasih dan doa, semoga apa yang telah diberikan tercatat sebagai amal baik dan mendapatkan balasan dari Allah SWT.

Penyusun menyadari bahwa dalam skripsi ini masih banyak terdapat kesalahan. Oleh karena itu, saran dan kritik sangat penyusun harapkan demi kesempurnaan penyusunan selanjutnya. Semoga skripsi ini dapat memberikan manfaat dan kontribusi dalam kemajuan dunia pendidikan dan secara umum kepada semua pihak yang berkepentingan.

Semarang, 20 Januari 2016

Penyusun



ABSTRAK

Buana, Metrika I. 2016. *Ant Colony Opimization dalam Penyelesaian Travelling Salesman Problem Menggunakan MATLAB*. Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Dr. Mulyono, M.Si.

Kata Kunci : Algoritma *Ant Colony Optimization*; Graph; Optimisasi; Rute Terpendek; *Travelling Salesman Problem*.

Algoritma *Ant Colony Optimization (ACO)* merupakan algoritma pencarian hasil terbaik yang dihasilkan melalui pengamatan terhadap semut. Pada algoritma ACO, semut berfungsi sebagai agen yang ditugaskan untuk mencari solusi terhadap suatu masalah optimisasi. ACO telah diterapkan dalam berbagai bidang, salah satunya adalah untuk mencari solusi optimal pada *Traveling Salesman Problem (TSP)*.

Dalam penelitian ini permasalahan yang diambil adalah untuk mencari rute terpendek dan total jarak optimal dari 32 kota yang ada di Jawa Tengah untuk PT Gowill dalam mendistribusikan sandal jepit, serta pengaplikasian strategi tersebut dengan membangun sebuah program pencarian yang dapat membantu menentukan rute pendistribusiannya.

Dengan bantuan aplikasi Matlab dalam pembuatan program untuk PT Gowill ini, penyelesaian dilakukan secara bertahap dengan algoritma *Ant Colony Optimization* dengan terlebih dahulu melakukan koding Matlab, selanjutnya memasukkan 32 data kota diteruskan dengan pembuatan GUI agar mudah dalam pengoprasian sampai akhirnya diperoleh rute terpendek yang diharapkan. Selain itu, dari rute yang didapatkan jarak optimal yang akan ditempuh seorang salesman adalah 1522 km.



DAFTAR ISI

	Halaman
KATA PENGANTAR	v
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xii
BAB	
1. PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	6
2. TINJAUAN PUSTAKA.....	8
2.1 Graph.....	8
2.2 Optimisasi	12
2.3 Travelling Salesman Problem (TSP)	13
2.4 Ant Colony Optimization.....	17

2.5	MATLAB.....	21
2.6	Flowchart Ant Colony Optimization	22
3.	METODE PENELITIAN	24
3.1	Studi Pustaka.....	24
3.2	Perumusan Masalah	24
3.3	Analisis dan Pemecahan Masalah.....	25
3.4	Penarikan Kesimpulan	26
4.	HASIL DAN PEMBAHASAN	27
4.1	Implementasi Program.....	27
4.2	Implementasi Prosedural.....	28
4.3	Pengujian Program.....	31
4.4	Aplikasi Ant Colony Optimization	32
5.	SIMPULAN DAN SARAN	37
5.1	Simpulan	37
5.2	Saran	38
	DAFTAR PUSTAKA	39
	LAMPIRAN.....	42



DAFTAR TABEL

Tabel	Halaman
4.1 Daftar Kabupaten/Kota yang harus dikunjungi	33

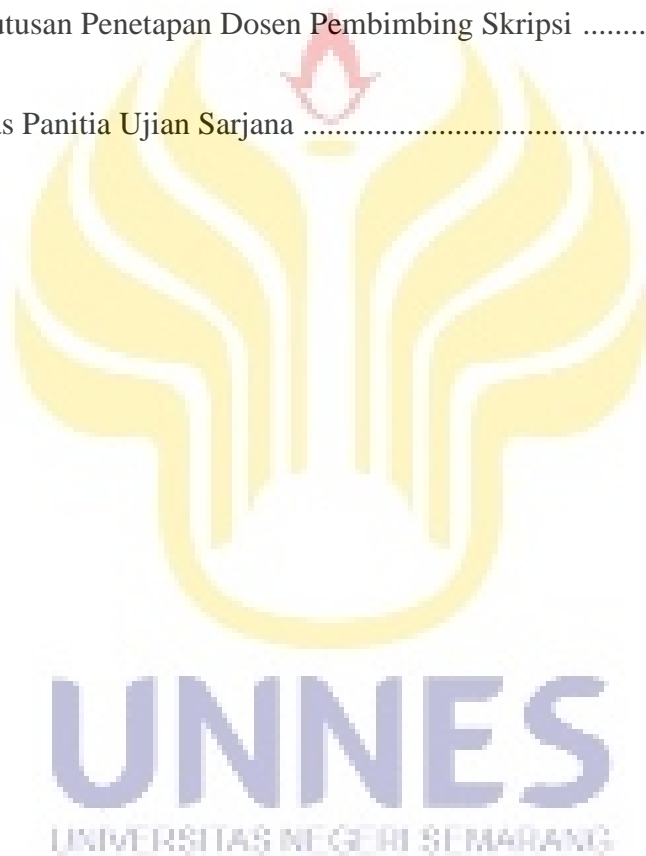


DAFTAR GAMBAR

Gambar	Halaman
2.1 Graph Sederhana, Graph Ganda, dan Graph Semu.....	9
2.2 Ilustrasi masalah TSP.....	14
2.3 Koloni semut mencari mencari sumber makanan	20
2.4 Flowchart algoritma <i>Ant Colony Optimization</i> untuk TSP	23
4.1 Halaman utama program.....	28
4.2 Peringatan jika field tidak diisi	32
4.6 Salah satu graph dengan pengujian terbaik.....	34
4.7 Rute terpendek peta Provinsi Jawa Tengah	35
4.8 Jarak optimal.....	35

DAFTAR LAMPIRAN

Lampiran	Halaman
1 Koordinat X dan Y Kabupaten dan Kota di Jawa Tengah	42
2 Tabel Jarak Antar Kabupaten atau Kota di Jawa Tengah	43
3 Surat Keputusan Penetapan Dosen Pembimbing Skripsi	46
4 Surat Tugas Panitia Ujian Sarjana	47



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi saat ini telah mengalami perkembangan pesat terutama dalam hal proses komputasi yang memungkinkan sebuah masalah dapat diselesaikan dengan bantuan komputer sebagai solusinya. Sejalan dengan itu pula peningkatan ilmu pengetahuan telah melahirkan begitu banyak algoritma-algoritma yang sangat membantu bagi pekerjaan manusia. Terdapat banyak algoritma untuk melakukan pencarian rute terpendek. Pemilihan algoritma yang paling optimum selalu menjadi permasalahan dalam pencarian rute terpendek, karena masing-masing algoritma memiliki kelebihan dan kekurangannya.

Aplikasi pada teori graph banyak digunakan dalam bidang pendistribusian. Dalam aplikasi dunia usaha, distribusi merupakan proses yang sangat berpengaruh pada keberlangsungan suatu usaha. Dengan pengoptimalan biaya distribusi dapat menambah keuntungan bagi perusahaan. Pengoptimalan biaya distribusi dapat dilakukan dengan mencari rute distribusi yang paling efisien. Karena dengan diperolehnya rute yang optimal dapat meminimasi biaya distribusi. Salah satu metode untuk mencari rute terpendek dalam pendistribusian adalah *Travelling Salesmen Problem* (TSP).

Salah satu perusahaan yang memiliki masalah distribusi dalam hal ini masalah *Travelling Salesman Problem* (TSP) adalah pabrik sandal jepit PT

Gowill, yang berada di Kabupaten Pati. Pendistribusian sandal jepit setelah selesai diproduksi akan segera dikirim keseluruh 32 kabupaten atau kota di Jawa Tengah dan kembali lagi ke kota awal untuk memulai kembali produksi sesuai pesanan yang diterimanya.

Travelling Salesman Problem (TSP) adalah sebuah masalah seorang *salesman* yang mempunyai tugas untuk mengirimkan pesanan atau barang kepada klien yang berada di sejumlah tempat yang berbeda di sebuah kota atau daerah. *Salesman* tersebut mempunyai masalah dalam hal menentukan tempat mana yang lebih dulu dikunjungi sedemikian sehingga total jarak dan waktu bepergian diperkecil dan setiap kota hanya boleh dilalui sekali dalam satu perjalanan, dan perjalanan berakhir pada kota awal dimana seorang sales memulai perjalanannya.

Ant Colony Optimization (ACO) terinspirasi oleh perilaku semut dalam menemukan jalur dari sarangnya menuju makanan sebagai tujuannya. Semut adalah makhluk Tuhan yang unik. Semut mampu mencari makan dengan cara yang efektif. Mereka mampu menemukan jarak terpendek antara sumber makanan dengan sarang mereka tanpa menggunakan indera penglihatan. Pada awalnya mereka berjalan mencari makan secara acak. Sepanjang perjalanan mereka meninggalkan jejak dengan menggunakan feromon. Ketika mereka menemukan makanan, mereka akan kembali ke sarang dengan mengikuti jejak feromon yang mereka tinggalkan. Semut yang lain akan mengikuti jejak feromon ini untuk membantu mengambil makanan tersebut. Feromon ini bisa menguap. Semakin jauh jarak suatu jalan yang dibentuk feromon, maka akan semakin lama semut melintasi jalan tersebut sehingga feromon yang berada di jalan tersebut akan lebih

cepat berkurang karena menguap dibandingkan dengan jalan feromon yang memiliki jarak lebih pendek. Karena feromon yang berjarak lebih jauh lebih sedikit daripada feromon yang berjarak lebih pendek, maka semut-semut yang lain tidak terlalu tertarik untuk mengikuti jalan tersebut, sehingga semut-semut akan cenderung mengambil jalan yang paling pendek.

Permasalahan *Travelling Salesman Problem* (TSP) berkembang terus secara nyata. Dengan adanya berbagai metode pendekatan untuk menyelesaikan masalah optimisasi, peneliti menggunakan *Ant Colony Optimization* (ACO) untuk memecahkan masalah *Travelling Salesman Problem* (TSP) dengan bantuan *software* MATLAB.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka rumusan masalah yang timbul dalam penulisan skripsi ini adalah :

1.2.1 Bagaimana langkah-langkah penyelesaian masalah TSP dengan penerapan - *Ant Colony Optimization* (ACO) menggunakan *software* MATLAB pada pendistribusian sandal jepit di PT Gowill?

1.2.2 Bagaimana bentuk rute terpendek dengan jarak optimal dari permasalahan yang diberikan?

1.3 Batasan Masalah

Dari latar belakang dan rumusan masalah yang telah dijelaskan, pencarian rute terpendek dibatasi pada salah satu jenis algoritma yang digunakan dalam metode heuristik, yaitu ACO. Batasan masalah yang diperlukan dalam penelitian yaitu :

1.3.1 Masukan yang diperlukan berupa graph yang terdiri dari jumlah titik, nama, dan koordinat titik. Letak titik dibangkitkan secara manual menggunakan koordinat titik kabupaten atau kota di Jawa Tengah.

1.3.2 Diasumsikan bahwa setiap kota yang terhubung selalu ada jalan langsung yang menghubungkan antara satu kota dengan kota lainnya.

1.3.3 Keluaran yang dihasilkan adalah pelaporan proses ACO.

1.3.4 Fungsi-fungsi yang dibutuhkan adalah sebagai berikut;

1.3.4.1 Fungsi untuk membuat graph.

1.3.4.2 Fungsi probabilitas untuk menentukan titik selanjutnya.

1.3.4.3 Fungsi untuk menyeleksi jarak terpendek.

1.3.5 Program dibangun menggunakan bahasa pemrograman Matlab.

1.3.6 Program hanya digunakan pada pengiriman pertama, pengiriman selanjutnya menyesuaikan kota mana saja yang kembali memesan.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang akan diteliti di atas, maka tujuan dari penelitian ini adalah sebagai berikut :

1.4.1 Merancang sistem penyelesaian masalah TSP dengan penerapan *Ant Colony Optimization* (ACO) menggunakan *software* MATLAB pada pendistribusian sandal jepit di PT Gowill.

1.4.2 Memberikan bentuk rute terpendek dengan jarak optimal dari permasalahan yang diberikan.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat berguna bagi berbagai pihak, antara lain sebagai berikut :

1.5.1 Pribadi

Bagi pihak pribadi manfaat dari penelitian ini adalah untuk menerapkan ilmu pengetahuan yang telah diperoleh dalam bentuk penelitian tugas akhir berupa karya tulis ilmiah dalam rangka penyelesaian studi strata 1 sebagai syarat memperoleh gelar Sarjana Sains di Jurusan Matematika Universitas Negeri Semarang.

1.5.2 Mahasiswa

Bagi pihak mahasiswa manfaat dari penelitian ini adalah untuk menambah ilmu pengetahuan tentang penelitian matematika khususnya dalam bidang TSP beserta aplikasinya dan dapat digunakan sebagai bahan referensi untuk penelitian selanjutnya.

1.5.3 Lembaga

Bagi pihak lembaga yang terkait, manfaat dari penelitian ini adalah sebagai bahan referensi untuk data penelitian selanjutnya dengan pemanfaatan yang lebih baik lagi.

1.5.4 Masyarakat

Bagi pihak masyarakat manfaat dari penelitian ini adalah menambah wawasan dalam pencarian rute terpendek sehingga dapat meningkatkan nilai ekonomis serta dapat memberikan pengetahuan tentang optimasi penjadwalan.

1.6 Sistematika Penulisan

Penulis skripsi disusun dalam tiga bagian utama, yaitu bagian awal, bagian inti, dan bagian akhir skripsi.

1.6.1 Bagian Awal

Dalam penulisan skripsi ini, bagian awal berisi halaman judul, pernyataan, pengesahan, motto dan persembahan, kata pengantar, abstrak, daftar isi, daftar gambar, daftar tabel, dan daftar lampiran.

1.6.2 Bagian Inti

Bagian inti dari penulisan skripsi ini adalah isi skripsi yang terdiri dari lima bab, yaitu:

BAB 1 : PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, sistematika penulisan.

BAB 2 : LANDASAN TEORI

Berisi tentang tinjauan pustaka yang berfungsi sumber atau alat dalam memahami permasalahan yang berkaitan dengan teori graph, optimisasi, teori TSP, teori mengenai ACO, dan Matlab.

BAB 3 : METODE PENELITIAN

Memuat prosedur atau langkah-langkah yang dilakukan dalam penelitian ini meliputi menentukan masalah, merumuskan masalah, studi pustaka analisis dan pemecahan masalah, penarikan simpulan.

BAB 4 : HASIL DAN PEMBAHASAN

Bab ini membahas tentang analisis kerja dari program Matlab. Pada bagian ini mengulas analisis hasil pengujian terhadap sistem yang dibandingkan dengan kebenaran dan kesesuaiannya dengan kebutuhan program yang dibuat.

BAB 5 : PENUTUP

Berisi kesimpulan dari penulisan skripsi ini dan saran.

1.6.3 Bagian Akhir

Berisi daftar pustaka yang digunakan oleh penulis sebagai acuan penulisan yang didalamnya terdapat informasi tentang buku dan literatur lain yang digunakan dalam skripsi ini serta lampiran yang mendukung.



BAB II

TINJAUAN PUSTAKA

2.1 Graph

Graph merupakan bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. Representasi visual dari graph adalah dengan menyatakan obyek sebagai noktah, bulatan, atau titik. Sedangkan hubungan antar obyek dinyatakan dengan garis. Graph merupakan salah satu model matematika yang kompleks dan cukup sulit, akan tetapi bisa juga menjadi solusi yang sangat bagus untuk masalah tertentu. Saat ini teori graph semakin berkembang dan menarik karena keunikan dan banyak sekali penerapannya. Salah satu alasan perkembangan teori graph yang begitu pesat adalah aplikasinya yang sangat luas dalam kehidupan sehari-hari maupun dalam berbagai bidang ilmu (Budayasa, 2007).

2.1.1 Definisi Graph

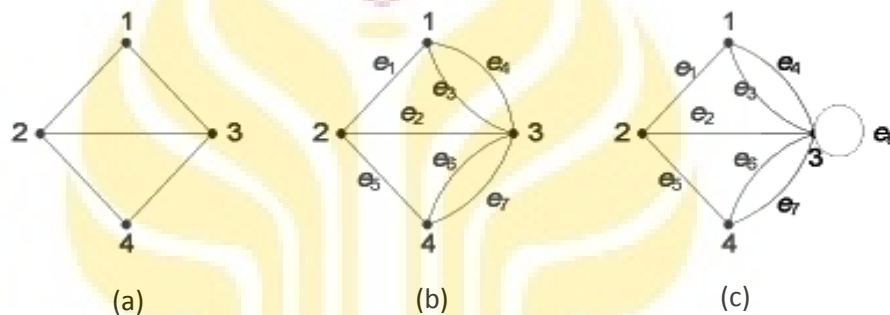
Graph merupakan representasi dari suatu masalah yang digambarkan sebagai sekumpulan titik atau simpul (*vertex*) yang dihubungkan dengan sekumpulan garis atau sisi (*edge*). Secara singkat suatu graph dapat ditulis sebagai $G = (V, E)$ yang dalam hal ini:

V = Himpunan berhingga dan tidak kosong dari simpul – simpul *vertices* .
 $= \{v_1, v_2, \dots, v_n\}$

E = Himpunan sisi yang menghubungkan sepasang simpul.

$$= \{e_1, e_2, \dots, e_n\}$$

Simpul pada graph dapat dinomori dengan huruf seperti v, w, \dots , dengan bilangan asli $1, 2, 3, \dots$, atau gabungan keduanya. Sisi (*edge*) yang menghubungkan dua simpul (*vertex*) v_i dan v_j , dan dinyatakan dengan pasangan (v_i, v_j) atau dengan lambang e_1, e_2 , dengan kata lain, jika e adalah sisi yang menghubungkan simpul v_i dan v_j , maka e dapat ditulis sebagai $e = (v_i, v_j)$.



Gambar 2.1 Graph Sederhana (a), Graph Ganda (b), Graph Semu (c)

2.1.2 Jenis-Jenis Graph

Graph dapat dikelompokkan menjadi beberapa jenis bergantung pada sudut pandang pengelompokannya. Pengelompokan graph dapat dipandang dari ada tidaknya sisi ganda atau sisi gelang (*loop*), dan berdasarkan pada orientasi arah pada sisi.

Berdasarkan ada atau tidak adanya sisi ganda atau sisi gelang (*loop*) pada suatu graph, maka secara umum graph dapat dibedakan menjadi dua jenis, yaitu:

2.1.2.1 Graph Sederhana

Graph sederhana adalah graph yang tidak mengandung gelang maupun sisi ganda. Pada gambar 2.1 gambar (a) merupakan contoh graph sederhana.

2.1.2.2 *Graph tak-sederhana*

Graph tak-sederhana adalah graph yang mengandung gelang atau sisi ganda. Ada dua macam graph tak-sederhana, yaitu graph ganda dan graph semu. Graph ganda yaitu graph yang apabila pada graph tersebut terdapat sisi ganda, pada gambar 2.1 gambar (b) merupakan contoh dari graph ganda, sedangkan graph semu adalah graph yang apabila pada graph tersebut terdapat gelang, pada gambar 2.1 gambar (c) merupakan contoh dari graph semu. Sisi (*edge*) pada graph dapat mempunyai orientasi arah. Berdasarkan orientasi arah pada sisi (*edge*), secara umum graph dapat dibedakan menjadi dua jenis, yaitu:

2.1.2.3 *Graph tak berarah*

Graph tak berarah adalah graph yang isinya (*edge*-nya) tidak mempunyai orientasi arah. Pada graph tak berarah, urutan pasangan simpul (*vertex*) yang dihubungkan oleh sisi (*edge*) tidak diperhatikan, jadi $v_j, v_k = (v_k, v_j)$ adalah sisi (*edge*) yang sama.

2.1.2.4 *Graph berarah*

Graph berarah adalah graph yang setiap sisinya (*edge*) diberikan orientasi arah dan urutan pasangan simpul pada graph berarah diperhatikan, sehingga pada graph berarah $v_j, v_k = (v_k, v_j)$.

2.1.3 Definisi *Walk, Trail, Path, dan Cycle*

Misalkan G adalah sebuah graph. Sebuah *jalan* (*walk*) di G adalah sebuah barisan berhingga (tak kosong) $W = \{v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k\}$ yang suku-sukunya bergantian simpul dan sisi, sedemikian hingga v_{i-k} dan v_i adalah simpul-simpul akhir dari sisi e_i , untuk $1 \leq i \leq k$. Kita katakan W adalah sebuah

jalan dari simpul v_0 ke simpul v_k , atau jalan- (v_0, v_k) . Simpul v_0 dan simpul v_k berturut-turut disebut *simpul awal* dan *simpul akhir* W . Sedangkan simpul-simpul $v_1 v_2 \dots v_{k-1}$ disebut *simpul-simpul internal* W , dan k disebut *panjang jalan* W . Perhatikan bahwa panjang jalan W adalah banyaknya sisi dalam W . Sebuah simpul G , mungkin saja muncul lebih dari satu kali dalam jalan W , begitu juga dengan sebuah sisi G , boleh muncul lebih dari satu kali dalam jalan W . Jika simpul awal dan simpul akhir sama maka W disebut *jalan tertutup*, sebaliknya jika simpul awal dan simpul akhir berbeda maka W disebut *jalan terbuka*. Jika semua sisi dalam jalan W berbeda, maka W disebut *jejak (trail)*. Jika semua simpul dan semua sisi dalam jalan W berbeda, maka W disebut *lintasan (path)*. Lintasan dari simpul u ke simpul v dapat ditulis sebagai lintasan- (u,v) . Jejak tertutup disebut *sirkuit*. *Sikel (cycle)* adalah sebuah jejak tertutup yang simpul awal dan semua simpul internalnya berbeda. Banyaknya sisi dalam suatu sikel disebut panjang dari sikel tersebut. Sikel dengan panjang k (memuat sebanyak k sisi) disebut *k-sikel*, disimbolkan dengan C_k . Pada penelitian ini *sikel (cycle)* yang melewati semua simpul disebut dengan rute.

2.1.4 Definisi Jarak

Jarak adalah bilangan yang menunjukkan seberapa jauh suatu benda berubah posisi melalui suatu rute tertentu. Dalam penelitian ini, jarak dapat diartikan sebagai panjang *sikel* dalam sebuah graph-bobot yaitu jumlah bobot minimum semua sisi pada *sikel* tersebut.

2.2 Optimisasi

Optimisasi merupakan suatu upaya sistematis untuk memilih elemen terbaik dari suatu kumpulan elemen yang ada. Di dalam konteks matematika, optimisasi ini bisa dinyatakan sebagai suatu usaha sistematis untuk mencari nilai minimum atau maksimum dari suatu fungsi. Dengan kata lain, optimisasi merupakan proses mencari nilai terbaik berdasarkan fungsi tujuan dengan daerah asal yang telah didefinisikan. Fungsi ini secara sederhana dapat dinyatakan dengan:

$$\min/\max f(x)$$

sebagai contoh adalah fungsi kuadrat $f(x) = x^2$, dimana x anggota bilangan real ($x \in R$). di dalam contoh ini, $f(x) = x^2$ merupakan fungsi tujuannya, sedangkan x adalah daerah asal yang didefinisikan sebagai anggota bilangan real.

Konsep optimisasi sudah dipakai sejak jaman prasejarah. Hal ini dapat dibuktikan dengan adanya saluran-saluran air yang ditemukan di situs-situs prasejarah. Saluran-saluran air ini dipakai untuk mengoptimalkan penggunaan air. Hal ini mengindikasikan bahwa konsep optimisasi merupakan bagian dari kehidupan manusia sejak lama. Permasalahan pengaturan air masih dijumpai dalam masyarakat masa kini, hanya saja penyelesaiannya sudah menggunakan metode optimisasi yang modern.

Meskipun konsep optimisasi sudah sangat lama digunakan, tetapi metode optimisasi pertama, yang mengacu pada teknik yang terstruktur, yang diakui adalah *steepest descent*. Istilah optimisasi diperkenalkan oleh George Dantzig yang mengembangkan algoritma *simplex* untuk menyelesaikan masalah *linear*

programming. Istilah *programming* di sini tidak mengacu pada pemrograman komputer, tetapi lebih pada program pelatihan dan penjadwalan logistik yang diadakan oleh pihak militer Amerika dimana masalah-masalah tersebut menjadi focus riset yang dilakukan oleh Dantzig. *Linear programming* sendiri merupakan metode untuk menyelesaikan fungsi linear, baik fungsi tujuan maupun fungsi batasannya (*constraint*).

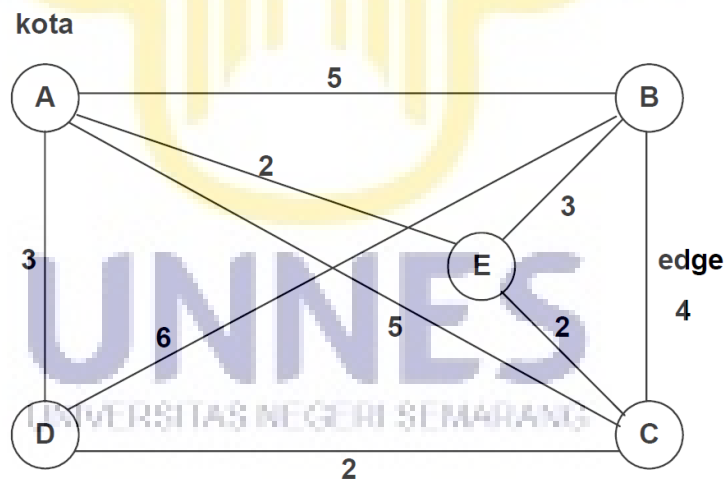
Dalam perkembangan selanjutnya, optimisasi sangat berkaitan dengan perkembangan komputer karena proses optimisasi ini umumnya dijalankan di komputer. Di awal-awal perkembangannya, penelitian optimisasi hanya dilakukan secara intensif untuk menyelesaikan permasalahan-permasalahan penting di bidang militer yang melibatkan teknologi tinggi. Tetapi ketika harga komputer semakin terjangkau, penelitian di bidang optimasi berkembang sangat pesat. Dalam dua dekade terakhir, banyak sekali metode optimisasi baru yang lahir.

Optimisasi dipakai di hampir semua bidang ilmu antara lain bidang teknik, sains, ilmu sosial, ekonomi dan bisnis. Banyak permasalahan di bidang teknik, sains dan ekonomi yang dapat dinyatakan sebagai permasalahan optimisasi seperti meminimalkan biaya, waktu dan resiko atau memaksimalkan keuntungan dan kualitas.

2.3 Travelling Salesman Problem (TSP)

Menurut Helsgaun dan Purnomo (2014:62), *Travelling salesman problem* (TSP) merupakan permasalahan kombinatorial yang banyak dipelajari. TSP adalah suatu kondisi dimana seorang *salesman* keliling yang harus mengunjungi n kota dengan aturan bahwa dia harus mengunjungi setiap kota hanya sebanyak satu

kali, meminimalisasi total jarak perjalanannya dan pada akhirnya dia harus kembali ke kota asalnya. Banyak permasalahan yang dapat direpresentasikan dalam bentuk TSP. Persoalan ini sendiri menggunakan representasi graph untuk memodelkan persoalan yang diwakili sehingga lebih memudahkan penyelesaiannya. Diantara permasalahan yang dapat direpresentasikan dengan TSP adalah pencarian rute bus sekolah untuk mengantarkan siswa, pengambilan tagihan telepon, efisiensi pengiriman surat atau barang, perancangan pemasangan pipa saluran dan lain-lain. Persoalan yang muncul adalah bagaimana cara mengunjungi node (simpul) pada graph dari titik awal ke setiap titik-titik lainnya tepat satu kali dengan bobot minimum (biaya paling murah) dan kembali lagi ke titik awal.



Gambar 2.2 Ilustrasi masalah TSP

Menurut Johnson dan McGeoch (1995), TSP dapat didefinisikan sebagai berikut: ada satu set kota $1, c_2, c_3, \dots, c_N$ dan $d_{c_i c_j}$ adalah jarak antara kota ke- i dan kota ke- j . Tujuannya adalah menemukan urutan π dari rumus di bawah ini untuk mendapatkan nilai yang paling minimal

$$\sum_{i=1}^{N-1} d_{c_{\pi i} \cdot c_{\pi i+1}} + d_{(c_{\pi N} \cdot c_{\pi 1})} \quad (1)$$

Hasil dari rumus di atas disebut sebagai panjang dari perjalanan seorang *salesman* mengunjungi kota-kota sesuai urutan π dari rumus di atas, dimana setelah mengunjungi semua kota *salesman* tersebut akan kembali ke kota awal. Dalam hal ini jika kasusnya *symmetric* TSP maka $d_{c_i c_j} = d_{c_j c_i}$ jika *asymmetric* TSP maka $d_{c_i c_j} \neq d_{c_j c_i}$ untuk $1 \leq i, j \leq N$.

Permasalahan TSP ini termasuk dalam kategori permasalahan *non deterministic polynomial time* artinya belum ditemukan waktu polinomial yang dibutuhkan untuk memecahkan masalah tersebut, misal : untuk algoritma A, jumlah kota N, waktu yang dibutuhkan T. Maka untuk jumlah kota 2N tidak bisa dikatakan membutuhkan waktu 2T untuk algoritma A. Hal ini mengakibatkan 2 alternatif pengembangan algoritma. Yang pertama algoritma yang berusaha menemukan perjalanan yang mendekati optimal tetapi dapat dilakukan dengan waktu yang singkat. Yang kedua mengembangkan optimasi algoritma yang benar-benar mencari solusi yang terbaik dari permasalahan TSP.

Jika hanya dua kota yang menjadi tujuan dalam suatu perjalanan, maka masalah perjalanan ini bisa dikatakan mudah sekali. Untuk *symmetric* TSP perjalanan tiga kota juga masalah yang mudah dipecahkan. Jumlah kombinasi perjalanan yang dihasilkan untuk kasus *Asymmetric* adalah $n - 1 !$. Untuk melihat kenapa itu bisa terjadi, pilihlah satu kota yang mana saja secara bebas sebagai kota pertama, kemudian ada $n - 1$ kota yang bisa dipilih sebagai kota yang kedua untuk dikunjungi, $n - 2$ kota yang bisa dipilih sebagai kota yang

ke-tiga untuk dikunjungi, begitu seterusnya sampai semua kota habis dikunjungi dan kembali ke kota awal.

Menurut Dakin (1997), pada kasus *symmetric* TSP, jumlah kombinasi perjalanannya adalah setengah dari jumlah kombinasi perjalanan *asymmetric* TSP yaitu $\frac{n-1!}{2}$.

2.3.1 Sejarah Travelling Salesman Problem

Permasalahan matematika tentang *Traveling Salesman Problem* dikemukakan pada tahun 1800 oleh matematikawan Irlandia William Rowan Hamilton dan matematikawan Inggris Thomas Penyngton. Bentuk dari TSP pertama dipelajari oleh para matematikawan dapat ditemukan di *Graph Theory* tahun 1736-1936 oleh N. L. Biggs, E. K. Lloyd, dan R. J. Wilson, di Oxford, tahun 1976.

Diawali oleh Karl Menger di Vienna dan Harvard. Setelah itu permasalahan TSP dipublikasikan oleh Hassler Whitney dan Merrill Flood di Princeton. Penelitian secara detail dari hubungan antara Menger dan Whitney, dan perkembangan TSP sebagai sebuah topik studi dapat ditemukan pada makalah Alexander Schrijver's "*On The History of Combinatorial Optimization* (sejak 1960)".

2.3.2 Contoh Perkembangan Masalah yang Muncul

Kode program komputer yang dibuat untuk menyelesaikan persoalan TSP telah berkembang semakin baik dari tahun ke tahun. Tanda yang paling mencolok dari perkembangan metode untuk menyelesaikan persoalan TSP adalah

bertambahnya jumlah simpul (*node*) yang dapat diselesaikan, mulai dari solusi persoalan 49 kota yang dikembangkan oleh Dantzig, Fulkerson, dan Johnson's pada tahun 1954 sampai kepada solusi persoalan 24.978 kota pada tahun 2004. Data-data tersebut didapat dari National Imagery and Mapping Agency, sebuah database nasional yang menyimpan nama-nama fitur geografi.

2.4 Ant Colony Optimization (ACO)

Menurut Dorigo dan Gambardella (dalam Leksono, 2009), *Ant Colony Optimization (ACO)* diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut. Sedangkan Algoritma Semut adalah sebuah algoritma dalam kelompok ACO yang diaplikasikan untuk masalah TSP. Algoritma ini tersusun atas sejumlah m semut yang bekerjasama dan berkomunikasi secara tidak langsung melalui komunikasi *Pheromone*.

Secara informal, ACO bekerja sebagai berikut : Setiap semut memulai tournya melalui sebuah titik yang dipilih secara acak (setiap semut memiliki titik awal yang berbeda). Secara berulang kali, satu persatu titik yang ada dikunjungi oleh semut dengan tujuan untuk menghasilkan sebuah tour. Pemilihan titik-titik yang akan dilaluinya didasarkan pada suatu fungsi probabilitas, dinamai aturan transisi status (*state transition rule*), dengan mempertimbangkan *visibility* (invers dari jarak) titik tersebut dan jumlah *Pheromone* yang terdapat pada ruas yang menghubungkan titik tersebut. Semut lebih suka untuk bergerak menuju ke titik-titik yang dihubungkan dengan ruas yang pendek dan memiliki tingkat *Pheromone* yang tinggi (Dorigo, M., dan Gambardella, L. M., 1997). Setiap semut memiliki sebuah memori, dinamai *tabulist*, yang berisi semua titik yang telah

dikunjunginya pada setiap tour. *Tabulist* ini mencegah semut untuk mengunjungi titik-titik yang sebelumnya telah dikunjungi selama tour tersebut berlangsung, yang membuat solusinya mendekati optimal.

Setelah semua semut menyelesaikan tour mereka dan *tabulist* mereka menjadi penuh, sebuah aturan pembaruan *Pheromone* global (*global Pheromone updating rule*) diterapkan pada setiap semut. Penguapan *Pheromone* pada semua ruas dilakukan, kemudian setiap semut menghitung panjang tour yang telah mereka lakukan lalu meninggalkan sejumlah *Pheromone* pada edge-edge yang merupakan bagian dari tour mereka yang sebanding dengan kualitas dari solusi yang mereka hasilkan. Semakin pendek sebuah tour yang dihasilkan oleh setiap semut, jumlah *Pheromone* yang ditinggalkan pada edge-edge yang dilaluinya pun semakin besar. Dengan kata lain, edge-edge yang merupakan bagian dari tour-tour terpendek adalah edge-edge yang menerima jumlah *Pheromone* yang lebih besar. Hal ini menyebabkan edge-edge yang diberi *Pheromone* lebih banyak akan lebih diminati pada tour-tour selanjutnya, dan sebaliknya edge-edge yang tidak diberi *Pheromone* menjadi kurang diminati. Dan juga, rute terpendek yang ditemukan oleh semut disimpan dan semua *tabu list* yang ada dikosongkan kembali.

Peranan utama dari penguapan *Pheromone* adalah untuk mencegah stagnasi, yaitu situasi dimana semua semut berakhir dengan melakukan tour yang sama. Proses di atas kemudian diulangi sampai tour-tour yang dilakukan mencapai jumlah maksimum atau sistem ini menghasilkan perilaku stagnasi dimana sistem ini berhenti untuk mencari solusi alternatif.

2.4.1 Aturan Transisi Status (*State Transition Rule*)

Aturan transisi status yang digunakan oleh AS dinamai *random-proportional rule* (Dorigo, M., Maniezzo, V., dan Coloni, A., 1996), yang ditunjukkan oleh persamaan (2.2). P_{rs}^k merupakan probabilitas dari semut k pada titik r yang memilih untuk menuju ke titik s .

$$P_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^\alpha [\eta_{rs}]^\beta}{\sum_{u \in J_r^k} [\tau_{ru}]^\alpha [\eta_{ru}]^\beta} & \text{untuk } s \in J_r^k \\ 0 & \text{untuk } s \text{ lainnya} \end{cases} \dots\dots\dots(2.3)$$

Dimana τ_{rs} adalah jumlah *Pheromone* yang terdapat pada edge antara titik r dan titik s , $(\eta_{rs}) = \frac{1}{d_{rs}}$ adalah *visibility* (invers dari jarak d_{rs}) dimana $d_{rs} = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2}$ (jika hanya diketahui koordinat titiknya saja). α adalah sebuah parameter yang mengontrol bobot (weight) relatif dari *Pheromone* dan β adalah parameter pengendali jarak ($\alpha > 0$ dan $\beta > 0$).

J_r^k adalah himpunan titik yang akan dikunjungi oleh semut k yang sedang berada pada titik r (untuk membuat solusinya mendekati optimal), dan pada persamaan (2.1) kita mengalikan *Pheromone* pada edge (r,s) dengan nilai *visibility* yang sesuai (η_{rs}). Dengan cara ini kita memilih edge yang lebih pendek dan memiliki jumlah *Pheromone* yang lebih besar.

2.4.2 Aturan Pembaruan Feromon (*Update Pheromone Trail*)

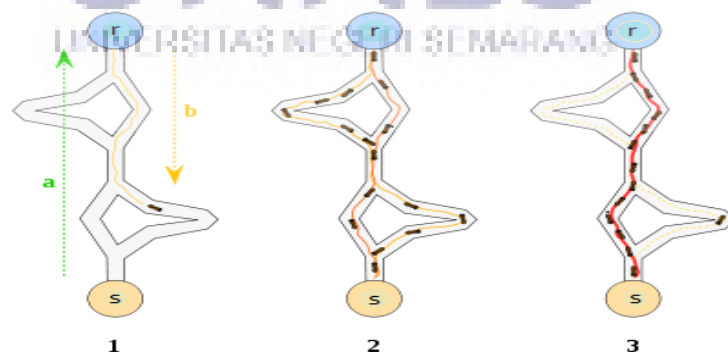
Setelah semua semut menyelesaikan tour-nya masing-masing maka *Pheromone* di-update. Dalam Algoritma Semut, aturan pembaruan *Pheromone global* (Dorigo, M., Maniezzo, V., dan Coloni, A., 1996) diimplementasikan menurut persamaan 2.2 sebagai berikut :

$$\tau_{rs} \leftarrow 1 - \rho \tau_{rs} + \frac{m}{k=1} \Delta\tau_{rs}^k \dots\dots\dots(2.4)$$

$$\text{dengan } \Delta\tau_{rs}^k = \begin{cases} \frac{1}{C^k} & \text{jika } r, s \in \text{tour yang dilakukan oleh semut } k \\ 0 & \text{sebaliknya} \end{cases} \dots\dots(2.5)$$

Dimana C^k , panjang tour yang dilalui oleh semut k. $0 < \rho \leq 1$ adalah sebuah parameter tingkat *evaporasi Pheromone*, dan m adalah jumlah dari semut.

Untuk memastikan bahwa semut mengunjungi n titik yang berbeda, diberikan *tabulist* pada masing-masing semut, yaitu sebuah struktur data yang menyimpan titik-titik yang telah dikunjungi semut dan melarang semut mengunjungi kembali titik-titik tersebut sebelum mereka menyelesaikan sebuah tour. Ketika sebuah tour selesai, *tabulist* digunakan untuk menghitung solusi yang ditemukan semut pada tour tersebut. *Tabulist* kemudian dikosongkan dan semut kembali bebas memilih titik tujuannya pada tour berikutnya. $Tabu_k$ adalah *tabulist* untuk semut k. $Tabu_k(r)$ adalah elemen ke- r dari $Tabu_k$, yaitu titik ke- r yang dikunjungi semut k pada suatu tour.



Gambar 2.3 (1) Semut menemukan makanan (2) Koloni semut mencari jarak terpendek (3) Koloni semut mendapat jarak terpendek

2.5 MATLAB

MATLAB adalah sebuah lingkungan komputasi numerical dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh The MathWorks, MATLAB memungkinkan manipulasi matriks, pemplotan fungsi dan data, implementasi, pembuatan antarmuka pengguna, dan pengantarmukaan dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah kotak kakas (*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Sebuah paket tambahan, Simulink, menambahkan simulasi grafis multiranah dan Desain Berdasar-Model untuk sistem terlekat dan dinamik. Pada tahun 2004, MathWorks mengklaim bahwa MATLAB telah dimanfaatkan oleh lebih dari satu juta pengguna di dunia pendidikan dan industri.

2.5.1 Sejarah MATLAB

MATLAB diciptakan pada akhir tahun 1970-an oleh Cleve Moler, yang kemudian menjadi Ketua Departemen Ilmu Komputer di Universitas New Mexico. Ia merancangya untuk memberikan akses bagi mahasiswa dalam memakai LINPACK dan EISPACK tanpa harus mempelajari Fortran. Karyanya itu segera menyebar ke universitas-universitas lain dan memperoleh sambutan hangat di kalangan komunitas matematika terapan. Jack Little, seorang insinyur, dipertemukan dengan karyanya tersebut selama kunjungan Moler ke Universitas Stanford pada tahun 1983. Menyadari potensi komersialnya, ia bergabung dengan Moler dan Steve Bangert. Mereka menulis ulang MATLAB dalam bahasa pemrograman C, kemudian mendirikan The MathWorks pada tahun 1984 untuk

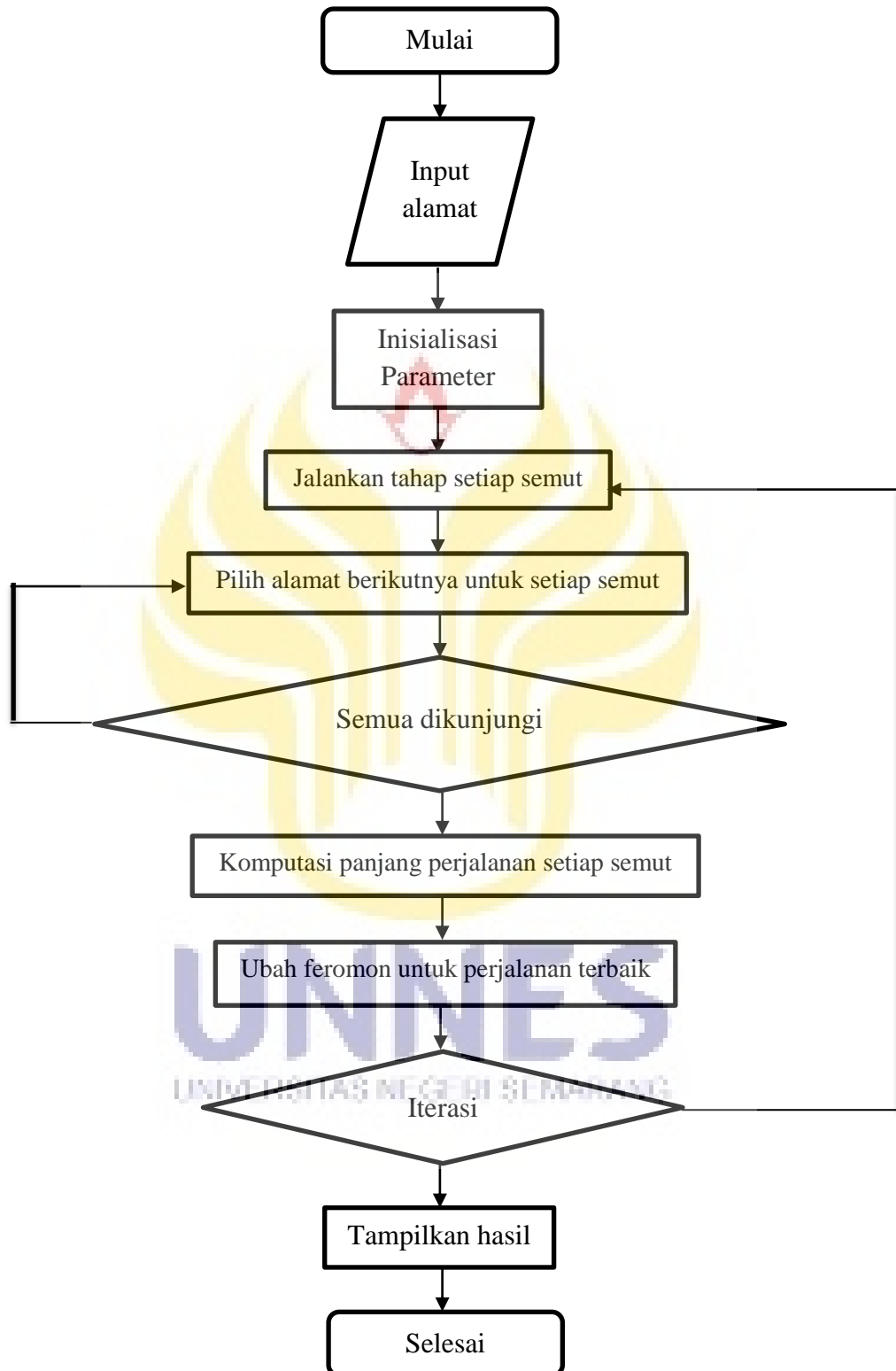
melanjutkan pengembangannya. Pustaka yang ditulis ulang tadi kini dikenal dengan nama JACKPAC. Pada tahun 2000, MATLAB ditulis ulang dengan pemakaian sekumpulan pustaka baru untuk manipulasi matriks, atau yang dikenal dengan LAPACK.

MATLAB pertama kali diadopsi oleh insinyur rancangan kontrol (yang juga spesialisasi Little), tapi lalu menyebar secara cepat ke berbagai bidang lain. Kini juga digunakan di bidang pendidikan, khususnya dalam pengajaran aljabar linear dan analisis numerik, serta populer di kalangan ilmuwan yang menekuni bidang pengolahan citra.

2.6 Flowchart Ant Colony Optimization

Flowchart menggambarkan berbagai aliran aktivitas dalam system yang akan dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. *Flowchart* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Flowchart digunakan untuk memperjelas perancangan dan algoritma yang akan dibuat, *flowchart* dari algoritma dapat dilihat pada gambar 2.5:



Gambar 2.4 Flowchart Ant Colony Optimization untuk penyelesaian TSP

BAB V

KESIMPULAN DAN SARAN

Pada bab ini diuraikan beberapa kesimpulan dari pengujian aplikasi dan beberapa saran yang dapat menyempurnakan perancangan dalam yang telah dibuat.

5.1 Simpulan

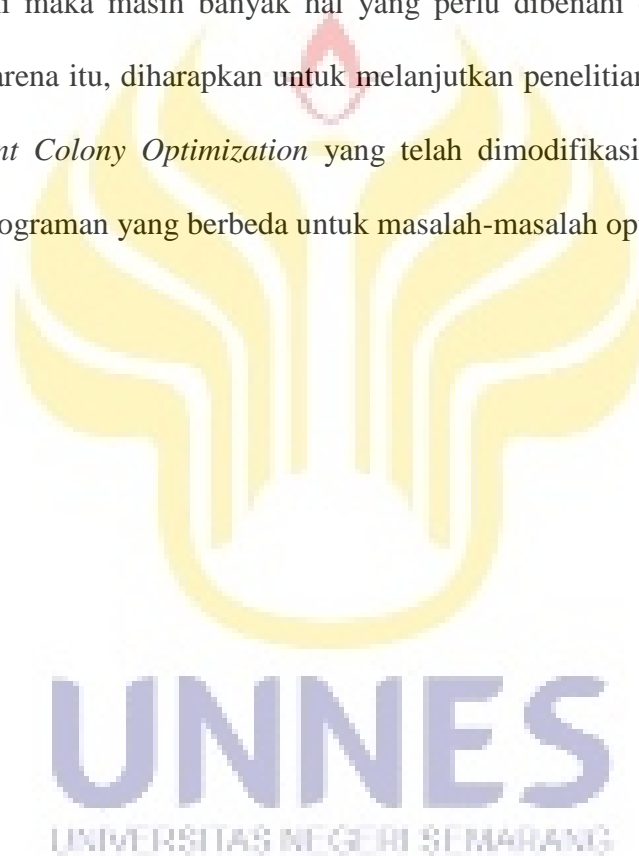
Dari hasil penelitian dan pembahasan pada bab IV dapat ditarik kesimpulan sebagai berikut.

5.1.1 Aplikasi untuk pencarian rute terpendek dibuat dengan menggunakan *software* MATLAB R2008a. Tahapan-tahapan yang dilakukan dalam pembuatan program adalah perancangan sistem, pembuatan desain tampilan program dengan menggunakan GUI kemudian dilakukan *coding* yang sesuai dengan algoritma *Ant Colony Optimization* pada MATLAB R2008a agar desain program bisa berfungsi. Setelah proses selesai maka dilakukan pengujian terhadap aplikasi yang dibuat. Selanjutnya memasukkan 32 data kabupaten atau kota di Jawa Tengah yang disusun secara matematis pada *Microsoft Excel* 2013 dan memasukkan parameter yang telah ditentukan oleh pengguna. Setelah semua proses selesai aplikasi dapat digunakan untuk pencarian rute terpendek khususnya pada kabupaten atau kota di Jawa Tengah.

5.1.2 Berdasarkan hasil pengujian sistem untuk 32 data kabupaten atau kota di Jawa Tengah diperoleh jarak optimal untuk algoritma *Ant Colony Optimization* sebesar 1522 km.

5.2 Saran

Penulis menyadari masalah optimisasi yang dapat diselesaikan dengan algoritma *Ant Colony Optimization* selain masalah *Travelling Salesman Problem* masih banyak lagi dan algoritma ini dapat dimodifikasi sesuai dengan aplikasi masalah yang akan diselesaikan. Karena keterbatasan penulis dalam mengerjakan penelitian ini maka masih banyak hal yang perlu dibenahi dan disempurnakan lagi. Oleh karena itu, diharapkan untuk melanjutkan penelitian tentang penerapan algoritma *Ant Colony Optimization* yang telah dimodifikasi dan menggunakan bahasa pemrograman yang berbeda untuk masalah-masalah optimasi lainnya.



DAFTAR PUSTAKA

- Astri, Dhanika. 2006. *Model Optimasi Rotasi Penugasan Awak Pesawat Menggunakan Metode Algoritma Genetika (Studi Kasus: Pesawat Boeing 737 Pada PT X)*. Universitas Indonesia, Indonesia, Departemen Teknik Industri.
- Dorigo, Marco dan Coloni, Alberto dan Maniezzo, Vittorio. 1996. *The Ant System: Optimization By A Colony Of Cooperating Agents*. <http://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.10SMC96.pdf>. Didownload pada tanggal 04 Januari 2008.
- Dorigo, M., dan Gambardella, L. M. 1997. *Ant Colonies for The Traveling Salesman Problem*. Tech.Rep/IRIDIA/1996-003, Université Libre de Bruxelles, Belgium.
- Fajar, S., Ping, M., Taufiq, H., Amy, F. 2007. *Perbandingan Performansi Algoritma Genetika dan Algoritma Semut untuk Penyelesaian Shortest Path Problem*. Bali : Seminar Nasional Sistem dan Informatika 2007.
- Hardiyanti, Y., Purwanto. 2013. *Penerapan Algoritma Genetik dalam Penyelesaian Travelling Salesman Problem With Precedence Constraints (TSPPC)*. Malang : Universitas Negeri Malang.
- Hilda, D. 2011. Materi Teori Graph. (<http://dhaahilda.blogspot.com/2011/12/materi-teori-graph.html>) diunduh pada tanggal 05 Januari 2015 pada pukul 13.26.

- Leksono, Agus. 2009. *Algoritma Ant Colony Optimization (ACO) untuk Menyelesaikan Traveling Salesman Problem (TSP)*. Skripsi. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Diponegoro. [Online] Tersedia: [eprints.undip.ac.id/7314/1/Tugas_Akhir_\(full\).pdf](http://eprints.undip.ac.id/7314/1/Tugas_Akhir_(full).pdf) [22 Juni 2014]
- Mindaputra, E. 2009. *Penggunaan Algoritma Ant Colony System dalam Travelling Salesman Problem (TSP) pada Eka Jaya Motor*. Skripsi. Semarang : FMIPA Universitas Diponegoro.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms (2nd ed.)*. London: The MIT Press.
- Munir, Rinaldi. 2012. *Matematika Diskrit*. Revisi kelima. Bandung: Informatika Bandung.
- Muttakhiroh, I'ing. 2007. *Menentukan Jalur Terpendek Menggunakan Algoritma Semut*. Skripsi. Universitas Islam Indonesia, Indonesia, Fakultas Teknologi Industri.
- Pan, J., & Dingwer, Wang. 2006. *An Ant Colony Optimization Algorithm for Multiple Travelling Problem*. Shenyang : Northeastern University.
- Pradhana, Chandra. 2013. Analisis Graph Sebagai Model Sistem dalam Menentukan Rute Antar Jemput di Perusahaan Jasa B-CLEAN Laundry. (<https://pradhablog.wordpress.com/2013/07/26/jurnal/>) diunduh pada tanggal 17 Januari 2015 pukul 17.30.

Purnomo, Hindriyanto D. 2014. *Cara Mudah Belajar Metode Metaheuristik Menggunakan Matlab*. Yogyakarta: Penerbit Gava Media.

Siang, J.J. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Penerbit Andi.

Sukrisnha, Pande G. 2013. Mari Belajar Dasar Matlab. (<https://torikiri.wordpress.com/2013/09/25/mari-belajar-dasar-matlab/>)
diunduh pada tanggal 08 April 2015 pada pukul 00.30.

Susilo, B., Efendi, R., Maulinda, S. 2011. *Implementasi dan Analisis Kinerja Algoritma Ant System (AS) dalam Penyelesaian Multiple Travelling Salesman Problem (MTSP)*. Yogyakarta : *Jurnal Nasional Aplikasi Teknologi Informasi 2011 (SNATI 2011)*.

Zhong, W., Shan, H., Chen, Z., Xia, L. 2014. *Multiple Traveling Salesman Problem with Precedence Constraints Based on Modified Dynamic Tabu Artificial Bee Colony Algorithm*. *Journal of Information & Computational Science*, 11:4 (2014), hal. 1225-1232.