

BAB II

LANDASAN TEORI

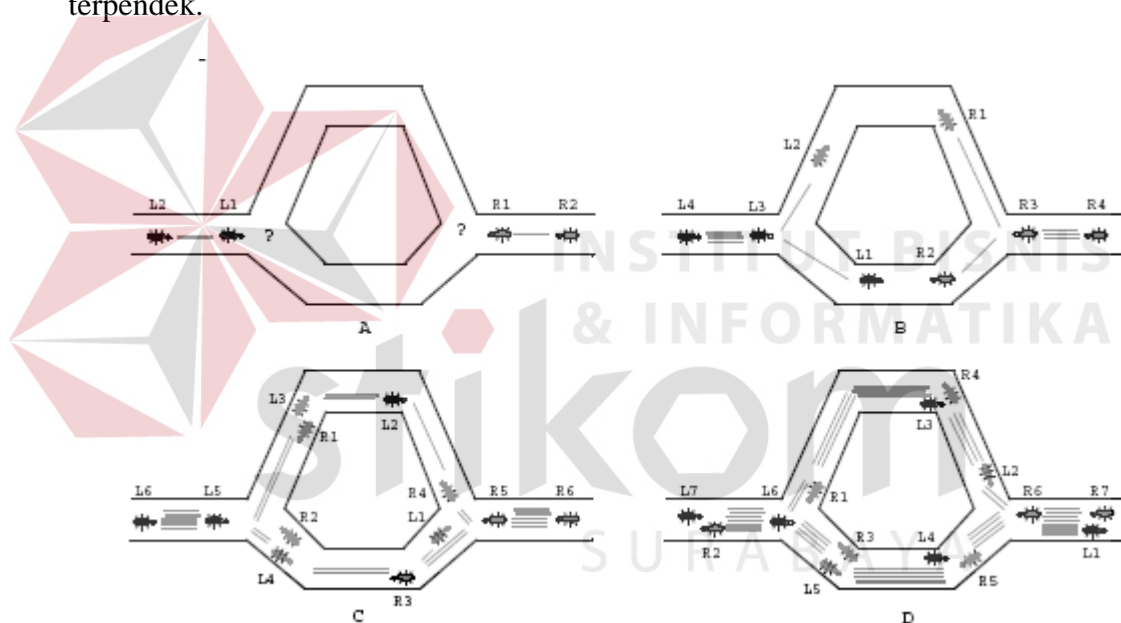
Untuk melakukan perancangan dan pembuatan aplikasi pencarian lokasi terdekat diperlukan pemahaman terhadap teori dan konsep yang mendasarinya, antara lain konsep mengenai *Short Message Service (SMS)*, *Global System for Mobile Communication (GSM)*, dan khususnya dalam pembahasan algoritma Ant System dalam menemukan rute terdekat.

2.1 Ant Colony Optimization (ACO)

Algoritma Ant Colony terinspirasi oleh kebiasaan semut. Semut yang sesungguhnya mampu untuk menemukan jalur yang terpendek dari makanan ke sarangnya. Selama dalam perjalanannya, setiap semut menaburkan *pheromone* ke tanah sebagai jejak atau petunjuk untuk semut lainnya. *Pheromone* adalah semacam zat kimia yang dikeluarkan oleh semut dari dalam tubuhnya. Semut mengikuti jalan berdasarkan jumlah *pheromone* yang sebelumnya telah disebarkan oleh semut lainnya. Cara semut dalam mengeksploitasi *pheromone* untuk menemukan jalur antara dua titik ditunjukkan pada gambar 2.1.

Pada gambar 2.1A menunjukkan semut berada dalam titik keputusan dimana mereka akan memutuskan untuk belok ke kiri atau kekanan. Selama semut tersebut belum mempunyai petunjuk untuk memilih pilihan yg terbaik, mereka memilih secara acak. Dalam rata-rata, separuh semut memutuskan untuk belok kekanan dan separuh lagi memilih ke kiri. Hal ini terjadi pada semut baik itu yang bergerak dari kiri ke kanan (yang diberi label L) dan sebaliknya (yang diberi label R). Gambar 2.1B dan 2.1C menunjukkan semut melalui percabangan secara acak.

Jumlah dari garis putus-putus kasarannya sebanding dengan jumlah *pheromone* yang telah disebar oleh semut di tanah. Selama jalur bawah lebih pendek dari jalur atas, rata-rata lebih banyak semut akan melewatinya, dan maka dari itu *pheromone* terakumulasi dengan cepat. Setelah beberapa periode transisi yang pendek, perbedaan jumlah *pheromone* dalam kedua jalur cukup besar untuk mempengaruhi keputusan semut yang baru datang ke dalam sistem (ditunjukkan pada gambar 2.1D). Mulai dari sekarang, semut yang baru akan lebih memilih ke jalur bawah. Tidak lama lagi semua semut akan menggunakan jalur yang terpendek.



Gambar 2.1. Menunjukkan semut yang sesungguhnya menemukan jalur terpendek.

A. Semut dihadapkan pada titik keputusan, B. Beberapa semut memilih jalan atas dan sebagian memilih jalan bawah secara acak, C. Selama semut berjalan dengan kecepatan yang rata-rata sama, semut akan bereaksi memilih jalan bawah yang lebih pendek, lebih cepat daripada jalur atas yang lebih panjang, D. Jumlah *pheromone* rata-rata lebih tinggi pada jalur terpendek.

2.1.1 Ant System (AS)

Ant System (AS) merupakan algoritma pertama kali dari kumpulan algoritma Ant Colony Optimization (ACO) yang dikembangkan oleh Dorigo, Maniezzo, dan Colorni pada tahun 1991. AS banyak diterapkan untuk mengoptimalkan kombinasi pada permasalahan *Travelling Salesman Problem* dan *Quadratic Assignment Problem*. Dimana dalam AS terdapat sejumlah *artificial ant* yang bekerja sama untuk menemukan solusi dari suatu masalah dengan menukarkan informasi melalui *pheromone* yang disebarkan di garis dalam grafik.

Artificial ant adalah semut buatan yang bekerja seperti layaknya semut yang sesungguhnya sedangkan *pheromone* adalah semacam zat kimia yang dikeluarkan oleh semut sesungguhnya yang digunakan untuk menandai atau memberikan jejak jalan yang telah dilewati untuk memberitahukan kepada semut yang lainnya.

Secara formal kerja dari AS adalah sebagai berikut :

1. *Artificial ant* lebih memilih berpindah dari kota yang memiliki probabilitas state transition yang tinggi yaitu kota yang terhubung oleh jalan yang pendek dengan jumlah *pheromone* yang tinggi. Probabilitas sebuah semut k pergi dari kota i ke kota $j \in \mathcal{N}_i^k$ ketika sedang membangun perjalanannya pada iterasi ke- t dirumuskan sebagai berikut :

$$p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in \mathcal{N}_i^k} a_{il}(t)}$$

Dimana $\mathcal{N}_i^k \subseteq \mathcal{N}_i$ adalah kota tetangga kota i yang belum pernah dikunjungi, $a_{ij}(t)$ adalah tabel *ant-decision* $A_i = [a_{ij}]_{|N_i|}$ dari node i didapat dari komposisi antara *local pheromone trail value* dengan *local heuristic value* yang disebut *Random Proportional Rule* dengan formula sebagai berikut :

$$a_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \forall j \in \mathcal{N}_i$$

Dimana $\tau_{ij}(t)$ adalah jumlah jejak *pheromone* pada jalur (i,j) pada waktu t , $\eta_{ij} = 1/\delta_{ij}$ adalah nilai heuristic value perpindahan dari kota i ke kota j , \mathcal{N}_i adalah kota tetangga kota i , dengan α dan β adalah dua parameter yang mengontrol pembobotan nilai jejak *pheromone* dan nilai heuristic. Peranan dari parameter α dan β adalah jika $\alpha = 0$, maka kota yang terdekat lebih dipilih untuk dikunjungi dan kebalikannya jika $\beta = 0$, maka kota dengan jejak *pheromone* yang terbanyak yang akan dipilih. Setiap *artificial ant* yang telah menyelesaikan perjalanannya, maka *pheromone intensity* dari setiap jalur akan diupdate. Ada 3 cara dalam mengupdate *pheromone intensity* :

a. *Ant-cycle*

Pada *ant-cycle*, *pheromone* diupdate ketika semua semut telah membangun solusi dalam perjalanan ke kota tujuan. Formula yang digunakan adalah sebagai berikut :

$$\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Dimana $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$, m adalah jumlah semut pada setiap iterasi t dan ρ adalah besar koefisien kebusukan jejak *pheromone* yang bernilai $1 > \rho > 0$, $\Delta\tau_{ij}$ adalah *pheromone evaporation* yang dihitung untuk setiap semut k pada jalur i,j dan pada iterasi ke t dengan formula :

$$\Delta\tau_{ij}^k(t) = \begin{cases} 1/L^k(t) & \text{if } (i, j) \in T^k(t) \\ 0 & \text{if } (i, j) \notin T^k(t) \end{cases}$$

Dimana $T^k(t)$ adalah jalur perjalanan yang ditempuh oleh semut k pada iterasi ke t dan $L^k(t)$ adalah panjangnya jarak perjalanan yang ditempuh oleh semut k .

Secara garis besar algoritma AS adalah sebagai berikut:

1. Inisialisasi

Set $t := 0$ *{iterasi}*

Set $NC := 0$ *{cycle iterasi}*

Untuk setiap jalur, set nilai initial $\tau := c$ untuk intensitas *pheromone*

dan $\Delta\tau := 0$

Tempatkan m semut pada kota n yang dikehendaki.

2. Set $s := 1$ *{s adalah tabu list index}*

For $k := 1$ to m do

Tempatkan kota asal pada semut ke k pada $\text{tabu}_k(s)$.

3. Ulangi sampai semua semut menemukan solusi

Set $s := s + 1$

For $k := 1$ to m do

Pilih kota j sebagai kota tujuan dengan probabilitas $p_{ij}^k(t)$

Pindahkan semut ke k kedalam kota j

Masukan kota j ke dalam $\text{tabu}_k(s)$

4. For $k := 1$ to m do

Pindahkan semut ke k dari $\text{tabu}_k(n)$ ke $\text{tabu}_k(1)$ (kembali ke asal)

Hitung panjang L_k perjalanan yang telah dilakukan oleh semut k

Update perjalanan yang terpendek yang ditemukan

5. Update intensitas *pheromone* pada setiap jalur

Set $t := t+n$

Set $NC := NC+1$

6. If($NC < NC_{max}$) and (not stagnation behavior) then

Kosongkan semua tabu list

Ulangi langkah ke-2

Else

Cetak perjalanan terpendek

Hentikan iterasi

b. *Ant-density*

Formula untuk mengupdate intensitas *pheromone* pada *ant-density* hampir sama dengan *ant-cycle*, perbedaannya terletak pada waktu mengupdate intensitas *pheromone* yaitu semut akan langsung mengupdate setiap jalur yang dilewatinya tanpa menunggu sampai menyelesaikan perjalanannya.

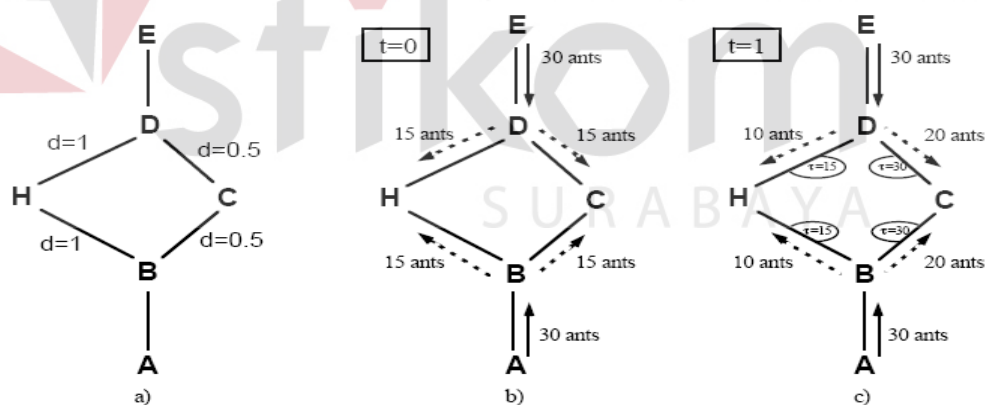
Perbedaan berikutnya adalah pada formula perhitungan *pheromone evaporation*. $\Delta\tau_{ij}^k = 1$ apabila semut ke-k pergi dari kota i ke kota j, jika tidak $\Delta\tau_{ij}^k = 0$.

c. *Ant-quantity*

Pada dasarnya *ant-quantity* hampir sama dengan *ant-density* perbedaannya terletak pada cara mengupdate intensitas *pheromone*, dimana pada *ant-quantity* kuantitas *pheromone* yang ditaburkan oleh *artificial ant* dari kota i ke kota j sebanyak $1/d_{ij}$ dimana d_{ij} adalah jarak dari kota i ke kota j. Formula untuk mengupdate *pheromone intensiti* adalah $\Delta\tau_{ij}^k = 1/d_{ij}$ apabila semut ke-k pergi dari kota i ke kota j, jika tidak $\Delta\tau_{ij}^k = 0$.

Setelah membangun solusi perjalanan, semua *artificial ant* akan ‘mati’ dengan kata lain *artificial ant* akan di hapus dari sistem.

2. Agar *artificial ant* membangun perjalanan secara legal, perpindahan ke kota yang sudah pernah disinggahi tidak diperbolehkan sampai menyelesaikan perjalanan. Setiap kota yang disinggahi oleh *artificial ant* akan disimpan dalam ingatannya yang disebut dengan *tabu list*. Dengan mengeksplorasi memori yang dimiliki oleh setiap semut maka setiap semut dapat membangun solusi yang mungkin.
3. Setiap *artificial ant* menyebarkan sejumlah *pheromone* pada jalur dalam perjalanannya yang menerangkan seberapa dekat perjalanannya (dalam kata lain, jalur yang menjadi jalur perjalanan yang dekat adalah jalur yang memiliki jumlah *pheromone* yang lebih banyak). Pada gambar 2.2 menunjukkan prinsip kerja AS.



Gambar 2.2. Prinsip kerja Ant System.

Pada gambar 2.2a menunjukkan inisial graph beserta jarak antar kota. Pada gambar 2.2b pada waktu $t=0$ masih belum ada terdapat jejak pada semua jalur maka dari itu *artificial ant* memilih untuk jalur kiri dan kanan dengan probabilitas yang sama. Pada gambar 2.2.c pada $t=1$ jalur dengan jarak yang

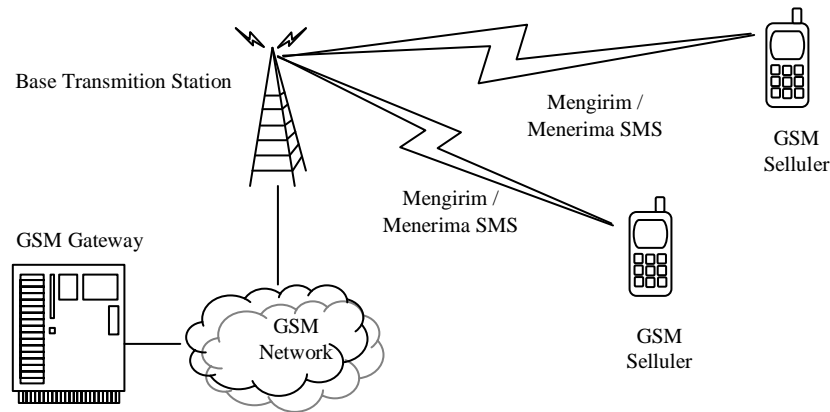
pendek memiliki intensitas pheromone yang lebih besar sehingga sebagian besar semut melewatinya.

2.2 Global System for Mobile Communication (GSM)

GSM adalah teknologi seluler digital yang dikembangkan pertamakali oleh Group Spatial Mobile di Eropa pada tahun 1982. GSM merupakan teknologi komunikasi seluler berbasis digital dan bersifat global dengan SIM (Subscriber Identification Module) sebagai kartu identitas pengguna yang mencerminkan nomor pelanggan. GSM menyediakan banyak fungsi antara lain : kotak suara (*voice mail*), fasilitas *call divert* (mengalihkan panggilan ke nomor telepon lain), CLI (*Calling Line Identification*), SMS (*Short Message Service*), MMS (*Multimedia Message Service*), dan fasilitas-fasilitas lainnya.

2.3 Short Message Service (SMS)

Short Message Service (SMS) adalah fasilitas yang dimiliki oleh GSM yang memungkinkan pengguna *mobile phone* yang menggunakan kartu GSM dapat mengirim dan menerima pesan-pesan singkat sebanyak maksimal 160 karakter. Prinsip kerja dari SMS ini adalah setiap provider GSM mempunyai satu atau lebih SMS Center yang berfungsi untuk menerima dan meneruskan pesan dari pengirim ke pelanggan tujuan. Dengan adanya SMS Center ini, pelanggan dapat mengetahui status dari pesan SMS yang telah dikirim, apakah telah sampai atau gagal diterima oleh nomor tujuan. Pesan yang gagal dikirim akan disimpan dalam SMS Center sampai *period-validity* terpenuhi. Gambar 2.3 menunjukkan proses pengiriman dan penerimaan SMS.



Gambar 2.3. Diagram Blok pengiriman dan penerimaan SMS

Untuk dapat menerima dan mengirim pesan, sebelumnya harus melakukan koneksi ke SMSC (*Short Message Service Center*). Ada beberapa cara untuk melakukan koneksi ke SMSC antara lain :

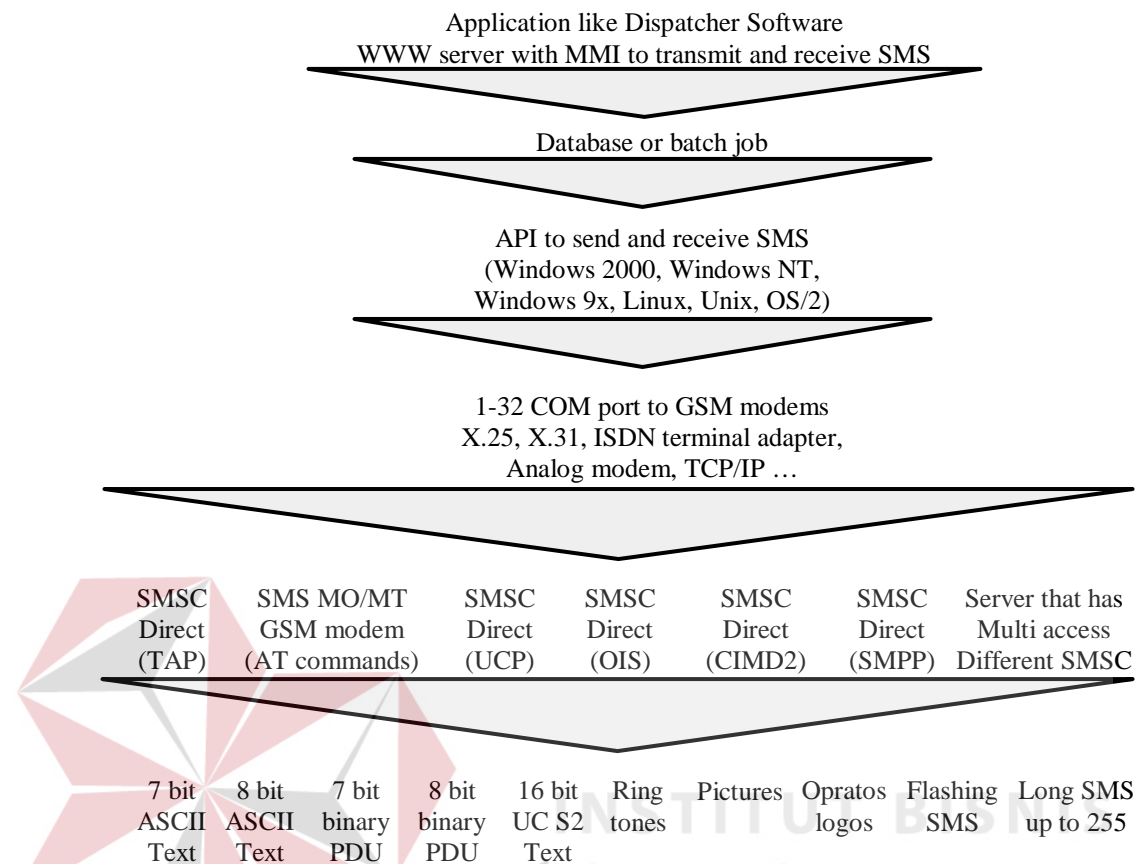
- a. Menggunakan sebuah terminal baik berupa GSM modem atau handphone.

Cara ini adalah yang paling mudah tetapi memiliki kekurangan antara lain jumlah pesan yang dikirim permenit sangat terbatas (sekiar 6-10 pesan per menit). Untuk mengantisipasi hal ini biasanya digunakan lebih dari satu terminal.

- b. Koneksi langsung ke SMSC.

Untuk melakukan koneksi ke SMSC diperlukan protokol penghubung. Protokol yang umum digunakan adalah UCP, SMPP, CIMD2, OIS, dan TAP. Masing-masing operator GSM menyediakan tipe protokol yang berbeda-beda. Dengan koneksi langsung ke SMSC, pesan dapat dikirim dengan jumlah banyak, dapat mencapai sekitar 600 pesan per menit bergantung pada kapasitas dari SMSC itu sendiri.

- c. Menggunakan Software Bantu.



Gambar 2.4 Tipe koneksi SMS Center

Berikut adalah daftar SMS Center pada beberapa operator GSM di Indonesia :

Tabel 2.1 Daftar SMS Center

Operator GSM	Nomor SMSC
Satelindo	62816124
Excelcomindo	62818445009
Telkomsel	6281100000
IM3	62855000000

Terdapat dua mode untuk mengirim dan menerima SMS, yaitu mode text dan mode PDU (*Protokol Data Unit*). Tetapi sistem mode text tidak didukung oleh semua operator GSM di Indonesia dan kebanyakan dari terminal atau handphone juga tidak mendukungnya.

a. Mode Text

Mode ini adalah cara termudah untuk mengirim pesan. Pada mode text pesan yang dikirim tidak dilakukan konversi. Teks yang dikirim tetap dalam bentuk aslinya dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Mode teks sebenarnya adalah hasil encode yang direpresentasikan dalam format PDU. Kekurangannya mode ini adalah tidak dapat menyisipkan gambar atau nada dering ke dalam pesan yang akan dikirim.

b. Mode PDU.

Mode PDU adalah pesan dalam format hexadesimal octet dan semi-desimal octet dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Kelebihan menggunakan format PDU adalah dapat melakukan encoding sendiri, tetapi tentunya harus didukung oleh hardware dan operator GSM, melakukan kompresi data, menambahkan nada dering dan gambar pada pesan yang akan dikirim, menambahkan header kedalam pesan seperti timestamp, nomor SMSC, dan meta-informasi lainnya. Beberapa encoding yang umum digunakan adalah PCCP437, PCDN, 8859-1, IRA, dan GSM.

2.4 Travelling Salesman Problem

Travelling Salesman Problem (TSP) merupakan salah satu permasalahan optimasi kombinasi dalam hal ini TSP dikhususkan dalam permasalahan pencarian jarak terdekat. Secara umum TSP dapat didefinisikan sebagai berikut : terdapat beberapa N titik dan sejumlah E jalur yang menghubungkan antar titik. Dengan d_{ij} adalah panjang jalur $(i,j) \in E$ yang merupakan jarak antara titik i dengan titik j , dengan $i, j \in N$. TSP adalah suatu permasalahan dalam menemukan jarak terpendek dari sirkuit Hamiltonian dalam graph $G = (N,E)$, dimana sirkuit

Hamiltonian dalam graph G adalah perjalanan terpendek dengan mengunjungi sekali dan hanya sekali titik $n \in |N|$ dalam grafik G dan panjang perjalanannya diperoleh dari menjumlahkan panjang dari setiap jalur yang dilewatinya.

