PENCARIAN RUTE TERDEKAT ANTAR KECAMATAN MENGGUNAKAN ALGORITMA SEMUT (*ANT ALGORITHM*) DENGAN *JAVA 2 MICRO EDITION* (J2ME)

Suzuki Syofian*, Cholid Basy Tommy**

ABSTRAK

Seseorang sering melakukan perjalanan dari tempat satu kota ke kota lain dengan mempertimbangkan efisiensi, waktu dan biaya, sehingga akurasi yang diperlukan dalam menentukan rute terpendek ke lokasi tujuan. Hasil penentuan rute terdekat akan menjadi pertimbangan dalam pengambilan keputusan untuk menunjukkan rute yang akan ditempuh. Masalah tersebut telah mengundang banyak solusi sementara solusi yang diharapkan adalah solusi untuk kompleksitas algoritma. Desain ini bertujuan untuk membangun sebuah aplikasi untuk menentukan rute terpendek antara tempat di handpone mobile menggunakan algoritma semut. Algoritma semut digunakan dalam pembuatan aplikasi ini adalah metode pemecahan masalah dengan menggunakan sifat koloni hewan semut. Untuk metode ini dirancang menggunakan model air terjun dan untuk aplikasi manufaktur dengan menggunakan bahasa pemrograman Java 2 Micro Edition (J2ME). Hasil dari desain ini adalah rute pencarian solusi program yang akan menargetkan lokasi terdekat bersama dengan nama dan departemen transportasi umum

Keywords : *Sistem koloni semut, Algoritma semut, Java 2 Micro Edition (J2ME), Aplikasi manufaktur*

1. PENDAHULUAN

Perkembangan alat berteknologi tinggi semakin mendominasi kehidupan manusia. Contohnya, komputer dan *Handphone*, dimana kedua alat itu diharapkan dapat membantu meringankan pekerjaan manusia baik dalam bidang pendidikan, industri dan kehidupan sehari-hari sebagai alat komunikasi.

Komputer dan *handphone* bukan hanya sebagai alat komunikasi pada umumnya. Misalnya *handphone* berteknologi tinggi, kebanyakan orang pada saat ini menggunakan *handphone* tersebut selain untuk komunikasi sering juga digunakan untuk pencarian lokasi dalam perjalanan dari satu tempat atau kota ke tempat yang lain dengan mempertimbangkan efisiensi waktu dan biaya sehingga diperlukan ketepatan dalam menentukan jalur terpendek antar suatu kota. Mencari suatu lokasi di kota besar seperti Tangerang tidaklah mudah, khususnya bagi masyarakat yang belum mengenal betul kondisi lalu-lintas dan jalan-jalan yang ada di kota tersebut. Hal ini disebabkan karena informasi petunjuk jalan yang mengarahkan ke tempat tujuan yang kurang dan juga kurangnya tempat pusat informasi yang tersedia. Kebanyakan masyarakat masih kesulitan mencari lokasi seperti Rumah Sakit, SPBU, tempat ibadah, kampus, dan tempat umum lainnya. Jika kebutuhan pencarian itu mendesak, maka informasi yang cepat dan tepat sangat dibutuhkan. Dengan menggunakan media *Handphone*, maka informasi lokasi suatu tempat bisa didapatkan kapan saja dan dimana saja. Tujuan penelitian ini adalah membuat aplikasi pencarian rute terdekat yang dapat diakses melalui media *Handphone* menggunakan

algoritma semut (*Ant Algorithm)* dengan bahasa pemrograman *Java 2 Micro Edition* (J2ME).

1.2 Perumusan Masalah

Mencari suatu lokasi di kota besar seperti Tangerang tidaklah mudah, dan jika kebutuhan pencarian itu mendesak, maka informasi yang cepat dan tepat sangat dibutuhkan. Masalah yang akan dibahas dalam penelitian ini adalah bagaimana membuat aplikasi pencarian informasi untuk menentukan rute terdekat suatu lokasi pada media *Handphone* dengan jarak sebagai parameternya, sehingga diharapkan mendapatkan informasi yang lebih cepat dan efisien oleh pengguna jalan?

1.3 Batasan Masalah

Pencarian jalur terpendek dibatasi pada salah satu jenis algoritma yang digunakan dalam metode heuristik, yaitu *Ant Algorithm* dengan batasan-batasan sebagai berikut:

1. Bobot yang ditentukan hanyalah berdasarkan bobot jarak (km) antar kota, dengan mengabaikan bobot-bobot lainnya. Sehingga jalur terpendek ditentukan berdasarkan jarak terpendek antar kota.
2. Waktu tempuh dicari dari jarak terdekat antar kota per kecepatan rata-rata kendaraan.
3. Pengukuran waktu tempuh berdasarkan pada kecepatan rata-rata kendaraan roda empat (mobil) dengan kecepatan rata-rata 25 km/jam. Rute yang akan dicari oleh fasilitas ini hanya sebatas kecamatan kota Tangerang.

2. LANDASAN TEORI

2.1 Algoritma Semut

Algoritma semut merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik melalui grafik. Algoritma ini terinspirasi oleh perilaku semut dalam menemukan jalur dari koloninya menuju makanan. Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Oleh karena itu, ketika seekor semut menemukan jalur yang bagus (jalur yang pendek) dari koloni ke sumber makanan, semut lainnya akan mengikuti jalur tersebut, dan akhirnya semua semut akan mengikuti sebuah jalur tunggal. Mengingat prinsip algoritma yang didasarkan pada perilaku koloni semut dalam menemukan jarak perjalanan paling pendek tersebut, *Ant colony* sangat tepat digunakan untuk diterapkan dalam penyelesaian masalah optimasi, salah satunya adalah untuk menentukan jalur terpendek. (Dorigo, M. & Gambardella L.M, 1996 : 205)

Agar semut mendapatkan jalur optimal, diperlukan beberapa proses:

1. Pada awalnya, semut berkeliling secara acak, hingga menemukan makanan.
2. Ketika menemukan makanan mereka kembali ke koloninya sambil memberikan tanda dengan jejak *feromon*.
3. Jika semut-semut lain menemukan jalur tersebut, mereka tidak akan bepergian dengan acak lagi, melainkan akan mengikuti jejak tersebut.
4. Kembali dan menguatkannya jika pada akhirnya mereka pun menemukan makanan.

5. Seekor semut yang secara tidak sengaja menemukan jalur optimal akan menempuh jalur ini lebih cepat dari rekan-rekannya, melakukan *round-trip* lebih sering, dan dengan sendirinya meninggalkan *feromon* lebih banyak dari jalur-jalur yang lebih lambat ditempuh.
6. *Feromon* yang berkonsentrasi tinggi pada akhirnya akan menarik semut-semut lain untuk berpindah jalur, menuju jalur paling optimal, sedangkan jalur lainnya akan ditinggalkan.
7. Pada akhirnya semua semut yang tadinya menempuh jalur yang berbeda-beda akan beralih ke sebuah jalur tunggal yang ternyata paling optimal dari sarang menuju ke tempat makanan.

Secara umum langkah-langkah dalam *Ant algorithm* adalah :

a. Inisialisasi harga parameter-parameter algoritma.
   Parameter-parameter yang diinisialisasikan adalah :
   - Intensitas jejak semut antar kota dan perubahannya (τij)
   - Banyak kota (n) termasuk koordinat (x,y) atau jarak antar kota (dij)
   - Kota berangkat dan kota tujuan
   - Tetapan siklus-semut (Q)
   - Tetapan pengendali intensitas jejak semut (α), nilai $\alpha \geq 0$
   - Tetapan pengendali visibilitas (β), nilai $\beta \geq 0$
   - Visibilitas antar kota = 1/dij (ηij)
   - Banyak semut, jumlah semut yang melakukan pencarian (m)
   - Tetapan penguapan jejak semut (ρ) , nilai ρ harus > 0 dan < 1 untuk mencegah jejak *feromon* yang tak terhingga.
   - Jumlah siklus maksimum (NCmax) bersifat tetap selama algoritma dijalankan, sedangkan τij akan selalu diperbaharui harganya pada setiap siklus algoritma mulai dari siklus pertama (NC=1) sampai tercapai jumlah siklus maksimum (NC=NCmax) atau sampai terjadi konvergensi.

b. Inisialisasi kota pertama setiap semut. Setelah inisialisasi τij dilakukan, kemudian m semut ditempatkan pada kota pertama tertentu secara acak.
   h menempati *tabuk*. Jika *s* menyatakan indeks urutan kunjungan, kota asal dinyatakan sebagai *tabuk* (*s*) dan kota-kota lainnya dinyatakan sebagai {*N-tabuk*}.

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in \{N-tabu_k\}} [\tau_{ik}]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \quad \text{untuk } j \in \{N\text{-}tabu_k\} \quad \ldots\ldots (1)$$

dan

$$P_{ij}^{k} = 0, \text{ untuk } j \text{ lainnya} \quad \ldots\ldots (2)$$

- Perhitungan panjang rute setiap semut. Perhitungan panjang rute tertutup (*length closed tour*) atau Lk setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini dilakukan berdasarkan *tabuk* masing-masing.

$$L_k = dtabu_k(n), tabu_k(1) + \sum_{s=1}^{n-1} d\, tabu_k(s), tabu_k(s+1) \quad \ldots\ldots (3)$$

  Dengan d_{ij} adalah jarak antara kota i ke kota j yang dihitung berdasarkan persamaan :

$$d_{ij} = \sqrt{(xi - xj)^2 - (yi - yj)^2} \quad \ldots (4)$$

- Pencarian rute terpendek.
  Setelah Lk setiap semut dihitung, akan didapat harga minimal panjang

rute tertutup setiap siklus atau LminNC dan harga minimal panjang rute tertutup secara keseluruhan adalah Lmin.

- Perhitungan perubahan *harga intensitas jejak kaki semut antar kota.*

  Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar kota yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar kota. Persamaan perubahan ini adalah :

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad \text{.....(5)}$$

  Dengan $\Delta\tau_{ij}^{k}$ adalah perubahan harga intensitas jejak kaki semut antar kota setiap semut yang dihitung berdasarkan persamaan :

$$\Delta\tau_{ij}^{k}=\frac{Q}{L_k}$$, untuk (i,j) $\in$ kota asal dan kota tujuan dalam $tabu_k$

.....(6)

$\Delta\tau_{ij}^{k}$ = 0, untuk i,j lainnya.....(7)

a. Perhitungan *harga intensitas jejak kaki semut antar kota* untuk siklus berikutnya. Harga intensitas jejak kaki semut antar kota pada semua lintasan antar kota ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewati. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antar kota untuk siklus selanjutnya dihitung dengan persamaan :

$\tau_{ij}=\rho.\tau_{ij}+\Delta\tau_{ij}$ .....(8)

b. *Reset harga perubahan intensitas jejak kaki semut antar kota.* Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar kota perlu diatur kembali agar memiliki nilai sama dengan nol.

2.2 Teori Graf

2.2.1 Definisi Graf

Menurut Teddy Marcus Zakaria & Agus Prijono (2006:115), graf adalah kumpulan simpul (*vertices* atau *nodes*) yang dihubungkan satu sama lain melalui sisi atau busur (*edges*).

- Perbedaan graf dengan pohon, pada graf mungkin terjadi siklus (*cycle*) dan graf dapat terdiri lebih dari satu sambungan.
- Aplikasi graf

  Contoh : hubungan antara kota-kota dalam suatu negara, dimana simpul mewakili kota dan busur mewakili jalan yang menghubungkan antara 2 kota. Graf G didefinisikan sebagai pasangan himpunan (V, E), dalam hal ini :

  Graf $G = (V, E)$, yang dalam hal ini:

  V = himpunan yang terbatas dan tidak kosong dari simpul-simpul.

  $= \{ v_1 , v_2 , \dots , v_n \}$

  E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul

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

  atau dapat ditulis : G = (V , E) artinya graf G memiliki V simpul dan E busur. Simpul-simpul pada graf dapat merupakan obyek sembarang seperti kota, nama anak, dan sebagainya. Busur dapat menunjukkan hubungan (*relasi*), sembarang seperti rute penerbangan, jalan raya, sambungan telepon, dan lain-lain.

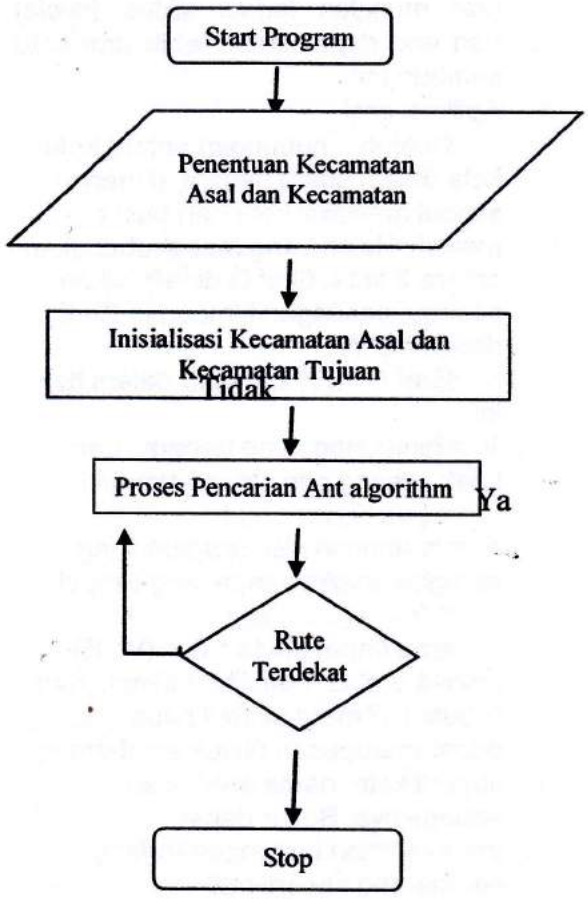
3. ANALISIS DAN PERANCANGAN SISTEM

3.1 Perancangan Perangkat Lunak

3.1.1 Metode Perancangan

Perancangan aplikasi yang dibangun menggunakan 4 (empat) model diagram UML (*Unified Modelling Language)* yaitu : *use case diagram*, *class diagram*, *activity diagram* dan *sequence diagram*.

3.1.2 *Flowchart*



Gambar 3.3 *Flowchart* jalannya program aplikasi pencarian rute terdekat antar kecamatan kota Tangerang untuk pengguna atau pengendara.

4. IMPLEMENTASI DAN PENGUJIAN SISTEM

Setelah melakukan analisis dan perancangan terhadap pengembangan aplikasi pencarian rute terdekat antar kecamatan kota Tangerang, tahapan selanjutnya adalah implementasi dan pengujian.

Implementasi dan simulasi pencarian jalur terpendek dikembangkan dengan menggunakan bahasa pemrograman *Java 2 Micro Edition* (J2ME) yang terdiri dari lingkungan implementasi, pengkodean dan antarmuka, serta hal-hal yang berhubungan dengan pengujian aplikasi.

4.1 Pengkodean

Kode sumber untuk program aplikasi pencarian rute terdekat tersebut dikembangkan dengan menggunakan bahasa pemrograman *Java 2 Micro Edition* (J2ME).

Ant.java

Class ini menjadi *class* utama (super kelas) dalam menciptakan keluaran berupa tampilan pilihan menu pada jendela menu utama. Pada saat di eksekusi, *class* ini merupakan *class* yang akan mengendalikan setiap aksi dari jendela utama.

4.2. AntAlgorithm.java

Ant Algorithm class yang berfungsi untuk keperluan algoritma semut yang digunakan pada Ant class sebagai main class.

4.3. Contoh Output

Contoh tampilan aplikasi sebagai berikut:

5. PENUTUP

5.1 Kesimpulan

Dari hasil pembahasan dan pembuatan program pencarian rute terdekat menggunakan algoritma semut (*Ant Algorithm*) ini dapat dilakukan. Pencarian rute terdekat menggunakan algoritma semut (*Ant Algorithm*) ini secara fungsional bekerja dengan baik sesuai kebutuhan yang telah didefinisikan pada tahap analisis sampai perancangan.

5.2 Saran

Teknologi yang dibahas dalam penulisan belum mencakup seluruh aspek yang diperlukan dalam penentuan jalur terpendek untuk pengendara menggunakan algoritma semut dalam sebuah aplikasi. Maka dari itu disarankan lebih lanjut :

1. Pengembangan pencarian rute terdekat menggunakan algoritma semut tidak hanya berdasarkan jarak terpendeknya saja tetapi juga berdasarkan lama perjalanan, waktu perjalanan yang paling efisien.
2. Mengembangkan aplikasi pencarian rute terdekat menggunakan teknologi yang mampu mengidentifikasi titik kemacetan serta titik yang tidak mengalami kemacetan.

DAFTAR PUSTAKA

1. Dorigo, M dan Gambardella, L.M. *Ant Colony System, A Cooperative Learning Approach to the Travelling Problem*: Universite Libre de Bruxelles, 1996.

2. Teddy Marcus Zakar Agus Prijono. *Konsep dan Implementasi, Struktur Data.* Bandung: Informatika, 2006.

(* Dosen Jurusan Teknik Informatika Fakultas Teknik Universitas Darma Persada.

(** Mahasiswa Jurusan Teknik Informatika Fakultas Teknik Universitas Darma Persada.