

# IMPLEMENTASI ALGORITMA KOLONI SEMUT PADA PROSES PENCARIAN JALUR TERPENDEK JALAN PROTOKOL DI KOTA YOGYAKARTA

**Bambang Yuwono, Agus Sasmito Aribowo, Siswanto Budi Wardoyo**

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta

Jl. Babarsari no 2 Tambakbayan 55281 Yogyakarta Telp (0274)-485323

Email : [mambambang@yahoo.com](mailto:mambambang@yahoo.com), [esbewe\\_mail@yahoo.com](mailto:esbewe_mail@yahoo.com), [doing4nothing@yahoo.com](mailto:doing4nothing@yahoo.com)

## Abstrak

*Sistem Informasi Geografi (SIG) adalah suatu sistem yang men-capture, mengecek, mengintegrasikan, memanipulasi, menganalisa dan menampilkan data yang secara spasial (keruangan) mereferensikan kepada kondisi bumi. Teknologi SIG mengintegrasikan operasi-operasi umum database, seperti query dan analisa statistik, dengan kemampuan visualisasi dan analisa yang unik yang dimiliki oleh pemetaan. Kemampuan inilah yang membedakan SIG dengan sistem informasi lainnya, yang membuatnya menjadi berguna unruk berbagai kalangan untuk menjelaskan kejadian, merencanakan strategi dan memprediksi apa yang akan terjadi. Sistem ini juga mampu memanipulasi, menganalisis, menampilkan, dan memodelkan data georeferensi untuk pemecahan suatu masalah. Salah satu permasalahan informasi keruangan adalah menentukan proses pencarian jalur terpendek jalan protokol di suatu wilayah tertentu. Metode pemodelan dalam proses pencarian jalur terpendek jalur protokol di kota Yogyakarta, dapat menampilkan informasi tiap jalurnya yang bertujuan untuk mencari jalur terpendek diantara kombinasi jalur yang diberikan serta melakukan pencarian (search) terhadap lokasi yang diinputkan oleh user. Parameter untuk menentukan jalur terpendek adalah jarak tempuh (kilometer) dari lokasi yang satu ke lokasi lainnya. Dalam pembuatan SIG ini, metode yang digunakan adalah metode waterfall, dan dibuat dengan software ArcView dan bahasa pemrograman Avenue. Setelah menyelesaikan skripsi mengenai implementasi algoritma koloni semut pada proses pencarian jalur terpendek jalan protokol di kota Yogyakarta, maka didapat kesimpulan telah dibangun sistem hasil penelitian implementasi algoritma koloni semut pada proses pencarian jalur terpendek di kota Yogyakarta, sehingga memudahkan pengguna dalam menentukan jalur terpendek saat mereka melakukan perjalanan*

*Keywords : Algoritma koloni semut, Jalur Terpendek, protokol*

## 1. PENDAHULUAN

Peta merupakan suatu media yang menyediakan informasi atau petunjuk tentang suatu lokasi baik itu lokasi jalan ataupun lokasi obyek-obyek tertentu yang pada umumnya memiliki perbandingan skala. Secara umum, pencarian jalur terpendek dapat dibagi menjadi dua metode yaitu metode konvensional dan metode heuristik. Dimana pencarian jalur terpendek ini berfungsi untuk mempercepat proses pencarian suatu tempat, menghemat waktu dan biaya. Metode heuristik terdiri dari beberapa macam algoritma yang biasa digunakan. Salah satunya adalah algoritma semut (*Ant Colony, Antco*). *Antco* diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996).

*Antco* diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996). Secara alamiah koloni semut mampu menentukan 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 dilalui. Semakin banyak semut yang melalui lintasan maka akan semakin jelas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali, dan sebaliknya, lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut. Mengingat prinsip algoritma yang didasarkan pada perilaku koloni semut dalam menemukan jarak perjalanan paling pendek tersebut, *Antco* sangat tepat digunakan untuk diterapkan dalam penyelesaian masalah optimasi, salah satunya adalah untuk menentukan jalur terpendek.

Berdasarkan latar belakang tersebut pada penelitian ini telah dapat mengimplementasikan algoritma koloni semut pada proses pencarian jalur terpendek jalan protokol di kota Yogyakarta.

### Batasan Masalah

Adapun batasan pada penelitian ini sebagai berikut :

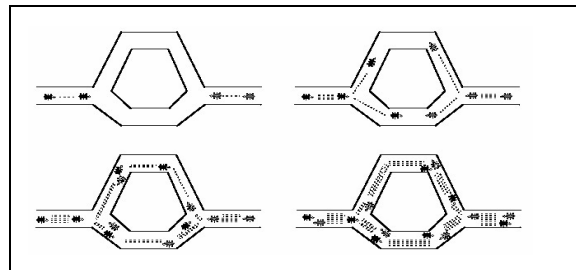
1. Jalan-jalan yang digunakan adalah jalan-jalan utama (jalan-jalan besar pada umumnya) dan tidak termasuk gang atau lorong.
2. Objek yang digunakan untuk mencari rute jalur terpendek adalah objek wisata di kota Yogyakarta dan hanya terdiri 20 lokasi sebagai contoh tempat wisata di kota Yogyakarta atau hanya objek dalam peta.

3. Dalam proses pencarian jalur terpendek ditentukan terlebih dahulu data asal dan data tujuan, sehingga proses pencarian dan penentuan jarak terjadi pada sistem.
4. Program ini tidak melihat atau menghiraukan suatu kondisi seperti jalur atau jalan yang sedang rusak atau kepadatan lalu lintas sehingga terjadi pengalihan jalur ke jalur lainnya. Semua jalur dianggap dalam keadaan normal.
5. Program ini tidak menghiraukan rambu-rambu lalu lintas ataupun *traffic light*.
6. Keterbatasan perubahan pada gambar peta Yogyakarta. Misalnya bila ada perubahan database program pada lokasi jalan ataupun objek wisata, maka gambar peta harus diganti dengan gambar peta lainnya yang telah di *update*.
7. Sistem Informasi Geografi ini, menampilkan visualisasi hasil pencarian jalur terpendek dengan menggunakan algoritma koloni semut.

## 2. TINJAUAN PUSTAKA

### Konsep Dasar Algoritma Koloni Semut

Koloni semut merupakan algoritma yang bersifat heuristik untuk menyelesaikan masalah optimasi. Algoritma ini diinspirasi oleh lingkungan koloni semut pada saat mencari makanan. Semut dapat mencari makanan. Semut dapat mencari lintasan terpendek dari suatu sumber menuju sarangnya, tanpa harus melihatnya secara langsung. Karena terinspirasi dari semut asli dinamakan algoritma koloni semut. Semut-semut mempunyai penyelesaian yang unik dan sangat maju, yaitu menggunakan jejak *pheromone* pada suatu jalur untuk berkomunikasi dan membangun solusi, semakin banyak jejak *pheromone* ditinggalkan, maka jalur tersebut akan diikuti oleh semut lain.



Gambar 1 Perjalanan semut menemukan sumber makanan

Dalam algoritma semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek, yaitu:

#### Langkah 1 :

a. Inisialisasi harga parameter-parameter algoritma.

Parameter-parameter yang di inisialisasi adalah :

1. Intensitas jejak semut antar kota dan perubahannya ( $\tau_{ij}$ )
2. Banyak kota ( $n$ ) termasuk  $x$  dan  $y$  (koordinat) atau  $d_{ij}$  (jarak antar kota)
3. Penentuan kota berangkat dan kota tujuan
4. Tetapan siklus-semut ( $Q$ )
5. Tetapan pengendali intensitas jejak semut ( $\alpha$ )
6. Tetapan pengendali visibilitas ( $\beta$ )
7. Visibilitas antar kota =  $1/d_{ij}$  ( $\eta_{ij}$ )
8. Jumlah semut ( $m$ )
9. Tetapan penguapan jejak semut ( $\rho$ )
10. Jumlah siklus maksimum ( $NC_{max}$ ) bersifat tetap selama algoritma dijalankan, sedangkan  $\tau_{ij}$  akan selalu diperbaharui harganya pada setiap siklus algoritma mulai dari siklus pertama ( $NC=1$ ) sampai tercapai jumlah siklus maksimum ( $NC=NC_{max}$ ) atau sampai terjadi konvergensi.

b. Inisialisasi kota pertama setiap semut.

Setelah inisialisasi  $\tau_{ij}$  dilakukan, kemudian  $m$  semut ditempatkan pada kota pertama yang telah ditentukan.

#### Langkah 2 :

Pengisian kota pertama ke dalam *tabu list*. Hasil inisialisasi kota pertama semut pada langkah 1 harus diisikan sebagai elemen pertama *tabu list*. Hasil dari langkah ini adalah terisinya elemen pertama *tabu list* setiap semut dengan indeks kota pertama.

**Langkah 3 :**

Penyusunan jalur kunjungan setiap semut ke setiap kota. Koloni semut yang sudah terdistribusi ke kota pertama akan mulai melakukan perjalanan dari kota pertama sebagai kota asal dan salah satu kota-kota lainnya sebagai kota tujuan. Kemudian dari kota kedua, masing-masing koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari kota-kota yang tidak terdapat pada  $tabu_k$  sebagai kota tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus hingga mencapai kota yang telah ditentukan. Jika  $s$  menyatakan indeks urutan kunjungan, kota asal dinyatakan sebagai  $tabu_k(s)$  dan kota-kota lainnya dinyatakan sebagai  $\{N-tabu_k\}$ , maka untuk menentukan kota tujuan digunakan persamaan probabilitas kota untuk dikunjungi sebagai berikut,

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k' \in \{N-tabu_k\}} [\tau_{ik'}]^\alpha \cdot [\eta_{ik'}]^\beta} \text{ untuk } j \in \{N-tabu_k\}$$

$$p_{ij}^k = 0, \text{ untuk } j \text{ lainnya}$$

dengan  $i$  sebagai indeks kota asal dan  $j$  sebagai indeks kota tujuan.

**Langkah 4 :**

a. Perhitungan panjang jalur setiap semut.

Perhitungan panjang jalur tertutup (*length closed tour*) atau  $L_k$  setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan dilakukan berdasarkan  $tabu_k$  masing-masing dengan persamaan berikut:

$$L_k = d_{tabu_k(n), tabu_k(1)} + \sum_{s=1}^{n-1} d_{tabu_k(s), tabu_k(s+1)}$$

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

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

b. Pencarian rute terpendek.

Setelah  $L_k$  setiap semut dihitung, akan diperoleh harga minimal panjang jalur tertutup setiap siklus atau  $L_{minNC}$  dan harga minimal panjang jalur tertutup secara keseluruhan adalah atau  $L_{min}$ .

c. 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 perubahannya adalah:

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k$$

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  $tabu_k$

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

**Langkah 5 :**

a. Perhitungan harga intensitas jejak kaki semut antar kota untuk siklus selanjutnya.

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 :

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta \tau_{ij}$$

b. Atur ulang 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.

**Langkah 6 :**

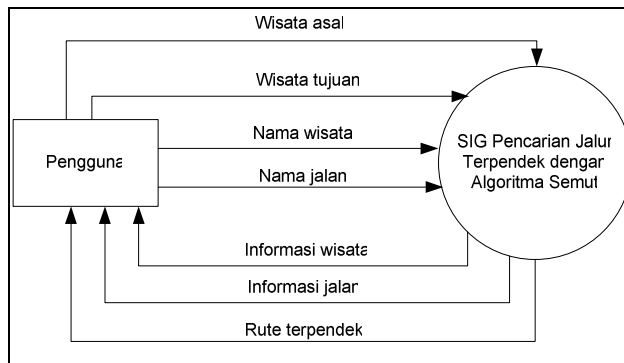
Pengosongan *tabu list*, dan ulangi langkah dua jika diperlukan. *Tabu list* perlu dikosongkan untuk diisi lagi dengan urutan kota yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum

terjadi konvergensi. Algoritma diulang lagi dari langkah dua dengan harga parameter intensitas jejak kaki semut antar kota yang sudah diperbaharui.

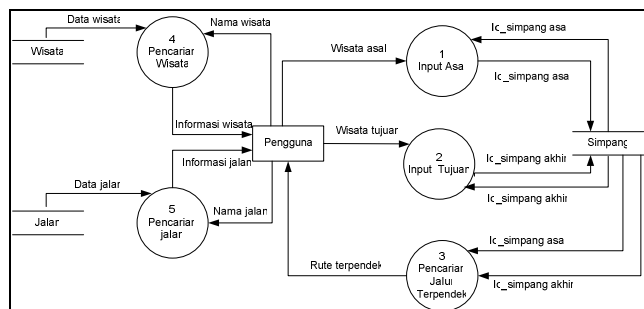
**3. METODE PENELITIAN**

Dalam menghitung penentuan jalur terpendek di kota Yogyakarta menggunakan algoritma koloni semut, dibutuhkan beberapa parameter dan data jalan. Parameter tersebut antara lain titik asal, titik tujuan, intensitas jejak semut antar titik ( $\tau_{ij}$ ), banyak titik (n), jarak antar titik ( $d_{ij}$ ), tetapan siklus semut (Q), tetapan pengendali intensitas jejak semut ( $\alpha$ ), tetapan pengendali visibilitas ( $\beta$ ), jumlah semut (m), tetapan penguapan jejak semut ( $\rho$ ), dan jumlah siklus maksimum ( $NC_{max}$ ).

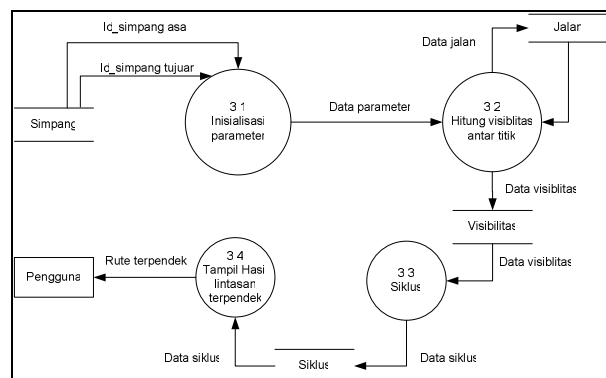
Titik asal dan tujuan ditentukan oleh pengguna sehingga sistem dapat memberikan hasil jalur terpendek sesuai dengan keinginan pengguna. Setelah menerima masukan asal dan akhir, sistem melakukan inisialisasi semua parameter yang dibutuhkan oleh algoritma koloni semut dan memprosesnya ke dalam siklus-siklus penghitungan sehingga dihasilkan jalur terpendek antara titik asal dan titik tujuan. Hasil penghitungan tersebut divisualisasikan sistem ke dalam peta. Berikut diagram DFD Level 0, Level 1 dan Level 2 proses 3



Gambar 2 DFD Level 0



Gambar 3 DFD Level 1



Gambar 4 DFD Level 2 Proses 3

**PESPEC**

**PSPEC 3.1 : Inisialisasi parameter**

Input :

Id simpang asal dan id simpang tujuan : Data Input

Output :

Data parameter : Data Output

Body :

Dalam proses inialisasi parameter inputan awalnya adalah id simpang asal dan id simpang tujuan, nilai kedua simpang ini diambil dari hasil pencocokan atau pengambilan dari tabel simpang yang telah di proses pada DFD level 1. dalam proses inialisasi parameter tersebut kedua simpang akan diinisialisasikan bersama dengan parameter-parameter yang akan digunakan untuk menghitung proses perhitungan visibilitas. Hasil proses tersebut adalah data parameter.

### PSPEC 3.2 : Hitung visibilitas antar titik

Input :

Data parameter dan data jalan : Data Input

Output :

Data jalan dan data visibilitas : Data Output

Body :

Dalam proses hitung visibilitas antar titik sebagai inputanya adalah data parameter yang merupakan hasil dari proses inialisasi parameter. Kemudian semua titik akan dihitung visibilitas antar simpangnya, sehingga dari proses hitung visibilitas antar titik akan menghasilkan data jalan yang akan disimpan dan dicocokkan dengan table jalan. Setelah data jalan tersebut sama dengan table jalan, maka data jalan yang berupa nilai simpang dan jarak akan kembali diproses pada proses hitung visibilitas antar titik untuk menghasilkan data visibilitas yang akan disimpan di table visibilitas.

### PSPEC 3.3 : Siklus

Input :

Data visibilitas : Data Input

Output :

Data siklus : Data Output

Body :

Dengan data visibilitas, koloni semut akan melakukan siklus perjalanan. Jumlah siklus bersifat tetap selama algoritma di jalankan, siklus akan dilakukan sampai tercapai jumlah siklus maksimum atau sampai terjadi konvergen, yang hasil akhirnya akan disimpan di tabel siklus.

### PSPEC 3.4 : Tampil hasil lintasan terpendek

Input :

Data siklus : Data Input

Output :

Rute terpendek : Data Output

Body :

Nilai data siklus akan digunakan untuk menentukan jalur kunjungan setiap semut ke setiap tujuan. Untuk menentukan tempat tujuan dilakukan perhitungan *pheromone*. Koloni semut yang sudah terdistribusi ke tujuan pertama akan mulai melakukan perjalanan dari tujuan pertama sebagai simpang asal dan salah satu simpang lainnya menjadi simpang tujuan. Perjalanan koloni semut berlangsung terus menerus hingga mencapai tujuan akhir. Seluruh jalur yang dilalui semua semut akan disimpan di tabu list, kemudian akan dipilih jalur yang terpendek, dan akan ditampilkan melalui visualisasi pada peta dan ditampilkan data jalan dan rute terpendek yang dilalui, sehingga dapat memudahkan pengguna untuk menentukan jalur terpendek wisata mana yang akan dituju.

## 4. HASIL DAN PEMBAHASAN

Proses implementasi dari perancangan sistem yang dilakukan pada bab sebelumnya akan dijelaskan pada bab ini. Implementasi bertujuan untuk menerjemahkan keperluan perangkat lunak. Tahap implementasi ini merupakan tahapan lanjutan dari tahap perancangan yang sudah dilakukan.

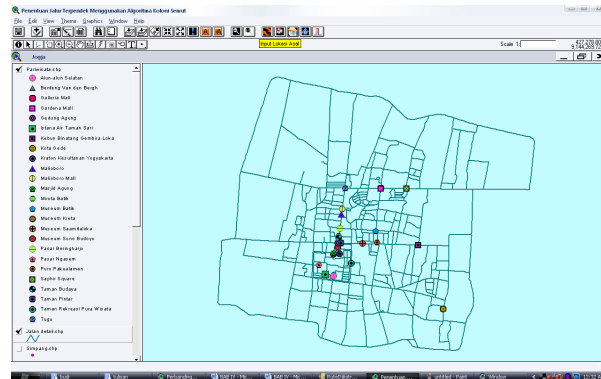
### Tampilan Awal dan Menu Utama



Gambar 5 Tampilan Utama

Dialog Pembuka (gambar 5) merupakan tampilan awal sistem. Tombol Masuk digunakan untuk memasuki aplikasi dan tombol Keluar digunakan untuk keluar dari aplikasi.

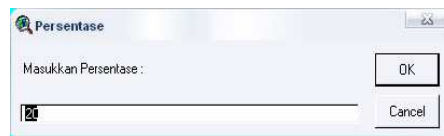
### Tampilan Awal



Gambar 6 Tampilan Peta

Setelah tombol Masuk pada dialog pembuka ditekan, sistem akan menampilkan peta kota Yogyakarta yang terdiri dari theme Pariwisata, Jalan, Simpang dan Kecamatan di Yogyakarta. Selain peta, gambar 6 juga menunjukkan tombol-tombol yang digunakan dalam proses sistem seperti Persentase Buffer, Tampil Label Jalan, Tampil Label Pariwisata, Pencarian Jalan, Pencarian Pariwisata, Penentuan Pariwisata Asal, Penentuan Pariwisata Tujuan, Proses Koloni Semut, Refresh, dan Keluar dari aplikasi.

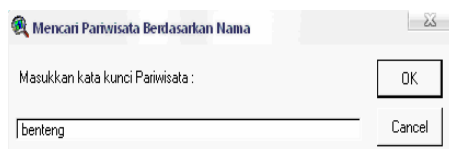
### Dialog Persentase Buffer



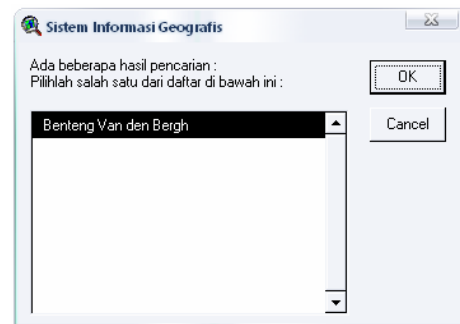
Gambar 7 Tampilan Dialog Input Persentase Buffer

Dialog Persentase Buffer (gambar 11) merupakan inputan awal nilai persen yang akan digunakan untuk menghitung area perbesaran buffer dalam menentukan simpang yang terpilih. Nilai Persentase ini sangat berpengaruh pada hasil pencarian jalur terpendek dalam proses pencarian jalur menggunakan algoritma Koloni Semut.

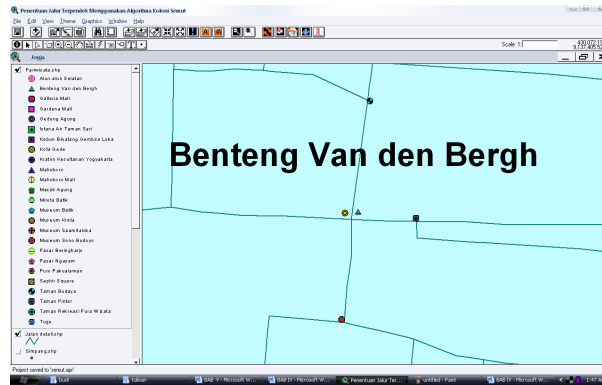
### Fasilitas Pencarian Pariwisata dan Jalan



Gambar 8 Tampilan Form Pencarian Pariwisata



Gambar 9 Tampilan Dialog Pencarian Pariwisata

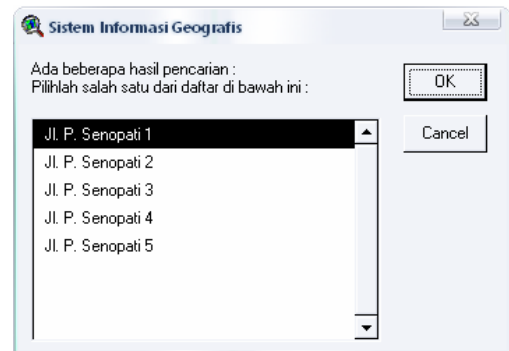


Gambar 10 Tampilan Hasil Pencarian Pariwisata

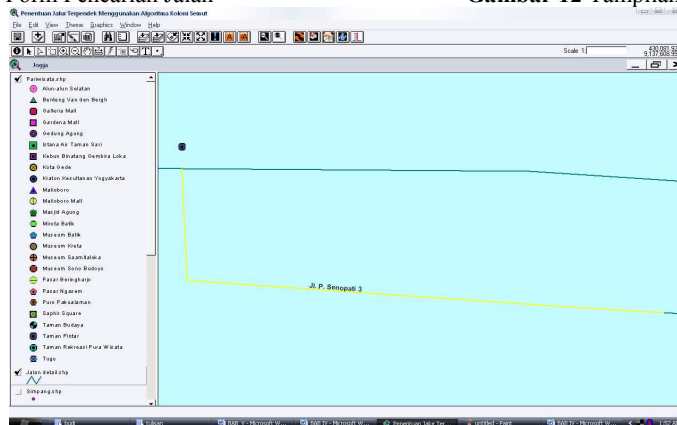
### Fasilitas Pencarian Jalan



Gambar 11 Tampilan Form Pencarian Jalan

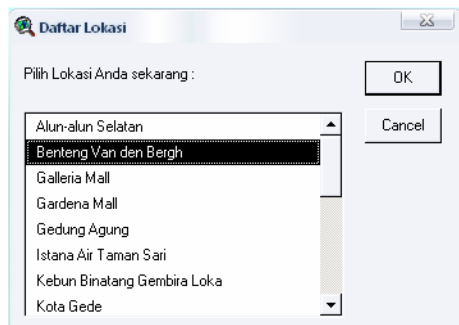


Gambar 12 Tampilan Dialog Pencarian Jalan

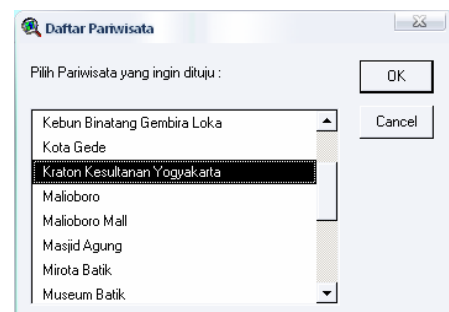


Gambar 13 Tampilan Hasil Pencarian Jalan

Sebelum memulai proses pencarian jalur, perlu diinputkan terlebih dahulu pariwisata asal (gambar 14) yang merupakan awal dari jalur dan pariwisata tujuan (gambar 15) yang merupakan akhir dari jalur yang dituju.



Gambar 14 Tampilan Dialog Input Pariwisata Asal



Gambar 15 Tampilan Dialog Input Pariwisata Tujuan

Gambar 15 merupakan tampilan dialog yang digunakan untuk menginputkan lokasi tujuan, maka setelah tujuan dipilih, kemudian klik OK .

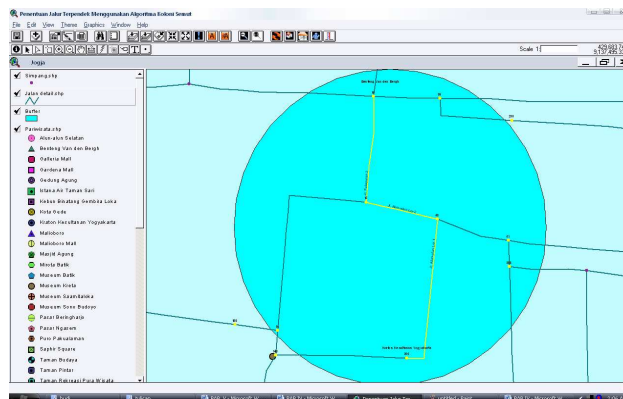
### Proses Pencarian Jalur Terpendek

Setelah lokasi asal dan lokasi tujuan diinputkan, sistem akan menyimpan id\_simpang masing-masing lokasi sebagai variabel simpang awal dan simpang akhir jalur. Id\_simpang untuk simpang awal beserta nama lokasi disimpan ke dalam tabel Asal, sedangkan id\_simpang untuk simpang akhir beserta nama lokasinya disimpan ke dalam tabel Tujuan. Kedua tabel tersebut berguna untuk menampung data nama lokasi masing-masing simpang. Misalkan akan dilakukan pencarian jalur terpendek dari Benteng Van den Bergh menuju Kraton Kasultanan Yogyakarta. Maka, tabel asal dan tujuan yang terbentuk terlihat pada gambar 14.

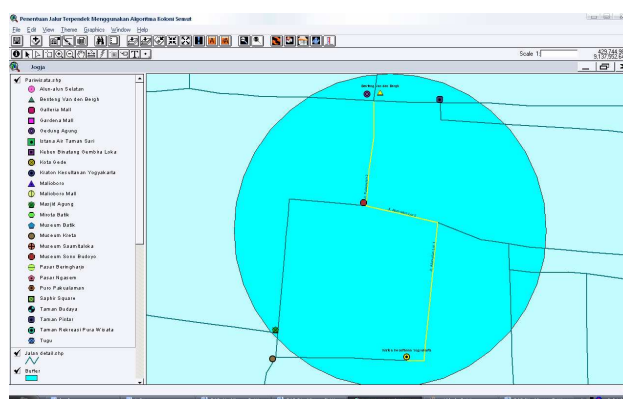
| ASAL                  |        | TUJUAN                       |       |
|-----------------------|--------|------------------------------|-------|
| Nmasal                | Noasal | Tuj                          | Notuj |
| Benteng Van den Bergh | 16     | Kraton Kesultanan Yogyakarta | 206   |

Gambar 16 Tampilan Tabel Asal dan Tujuan

Dari koordinat simpang tengah, dapat dihitung jari-jari dan diameter lingkaran *buffer*. Diameter lingkaran kemudian ditambahkan dengan persentase yang diinputkan. Persentase ini sangat menentukan keakuratan pencarian jalur karena menentukan area simpang yang dipilih. Semakin besar area simpang, semakin besar pula peluang mendapatkan jalur. Namun, jika simpang yang terpilih semakin besar, berpengaruh terhadap waktu pencarian jalur. Hasil *buffer* dengan simpang-simpang terpilihnya dapat dilihat pada gambar 17.



Gambar 17 Tampilan Buffer dan Simpang terpilih



Gambar 18 Tampilan Hasil Pencarian Jalur Terpendek Koloni Semut

```

Route - Notepad
File Edit Format View Help
HASIL PENCARIAN RUTE TERPENDEK KOLONI SEMUT

Dari: Benteng van den Bergh ke kraton kesultanan Yogyakarta
Jalan yang ditempuh :
=> Jl. Pekapalan 2 => Jl. Alun-alun Lor 2 => Jl. Alun-alun Lor 1

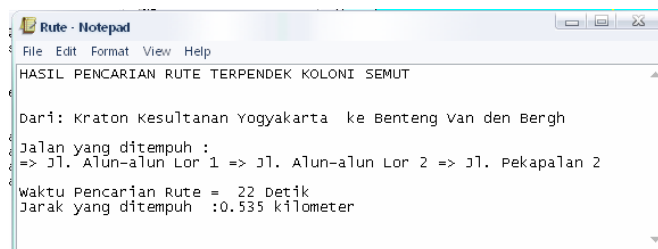
waktu Pencarian Rute = 55 detik
Jarak yang ditempuh :0.535 kilometer
    
```



**Gambar 19** Tampilan Rincian Jalan yang Ditempuh Jalur Terpendek Benteng Van den Bergh ke Kraton Yogyakarta

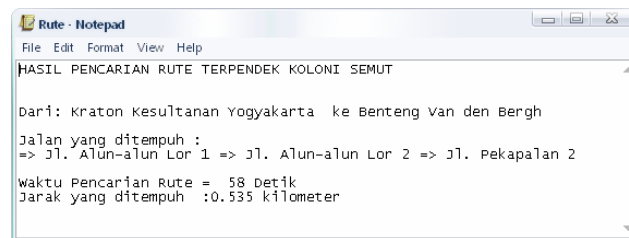
Simpang-simpang terpilih oleh *buffer* ditampilkan ke dalam tabel temp yang kemudian diproses untuk pencarian jalur terpendek. Dari setiap simpang di field no di tabel temp akan dipasangkan dengan simpang yang lainnya. Pasangan simpang tersebut akan dicocokkan di tabel jalan sehingga didapatkan pasangan simpang yang sesuai dengan pasangan simpang untuk nama jalan yang ada di tabel jalan. Selain itu, dicek juga arah jalan. Jika jalan yang dipilih ternyata satu arah, maka dicek apakah jalur dapat dilanjutkan atau tidak sesuai dengan arahnya.

Dalam proses pencariannya, jika dilakukan pembalikan simpang asal dan simpang tujuan, terjadi proses siklus yang berbeda. Untuk pencarian jalur terpendek dari Benteng Van den Bergh ke Kraton Kasultanan Yogyakarta (gambar 19) diperlukan 3 siklus dan membutuhkan waktu pencarian 55 detik. Sedangkan, dari Kraton Kasultanan Yogyakarta ke Benteng Van den Bergh (gambar 20) hanya memerlukan 1 siklus dan hanya membutuhkan waktu pencarian 22 detik. Hal ini berpengaruh pada waktu pencarian jalurnya.



**Gambar 20** Tampilan Rincian Jalan yang Ditempuh Jalur Terpendek Kraton Yogyakarta ke Benteng Van den Bergh

Jumlah semut yang digunakan untuk pencarian pun berpengaruh kepada waktu dan hasil pencarian jalur. Semakin banyak jumlah semut membuat waktu pencarian menjadi semakin lama. Namun, semakin banyak jumlah semut, semakin tinggi tingkat keberhasilan penemuan pencarian jalur terpendek. Gambar 21 menunjukkan bahwa pencarian dari Kraton Yogyakarta ke Benteng Van den Bergh dengan jalur yang sama dan menggunakan 50 semut membutuhkan waktu yang lebih lama yaitu 58 detik.



**Gambar 21** Tampilan Rincian Jalan yang Ditempuh Jalur Terpendek dari Kraton Kasultanan Yogyakarta ke Benteng Van Den Bergh dengan 50 Semut

Pada siklus ke-2 dan ke-3, dilakukan pengurutan simpang karena urutan simpang mempengaruhi hasil probabilitas tiap simpang. Simpang yang terpilih terlebih dahulu memiliki nilai probabilitas paling kecil sehingga memberikan peluang paling besar simpang tersebut yang dipilih untuk dijadikan simpang tujuan berikutnya.

## 5. KESIMPULAN

Dari penelitian tentang pencarian jalur terpendek menggunakan algoritma Koloni Semut, dapat diambil kesimpulan sebagai berikut :

- Algoritma Koloni Semut dapat digunakan untuk melakukan pencarian jalur terpendek berdasarkan jarak jalan.
- Keberhasilan pencarian jalur terpendek bergantung pada jumlah semut. Semakin besar jumlah semut, semakin besar pula kemungkinan keberhasilan pencarian jalur terpendeknya dan hasilnya pun semakin akurat.
- Pengacakan urutan simpang sebagai dasar pencarian dapat dilakukan dengan mengurutkan simpang secara *ascending* dan *descending* pada perulangan siklusnya.

## Saran

- Buffer* dapat dikembangkan dengan beberapa model untuk mencari proses pencarian yang lebih cepat.
- Parameter pencarian jalur dapat ditambah, misalkan berdasarkan waktu, tingkat kepadatan lalu lintas, waktu keberangkatan, dan lain-lain.

- c. Dapat dikembangkan dengan menambah siklus dengan kondisi urutan simpang yang acak sehingga jalur terpendek yang dihasilkan lebih akurat.

## 6. DAFTAR PUSTAKA

- Dorigo, M dan Gambardella, L.M., 1996, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, Université Libre de Bruxelles Belgium  
<http://elesys.fsaintek.unair.ac.id/admin/makalah/algortihmgenetik.pdf> (accessed, 2 Februari 2009).  
<http://mail.informatika.org/~rinaldi/Matdis/2006-2007/Makalah/Makalah0607-93.pdf> (accessed, 2 Februari 2009).
- Fathansyah, 1999, *Basis Data, Informatika*, Bandung
- Fathansyah, 2004, *Basis Data*, Informatika, Bandung.
- Jogiyanto, H.M., 1999, *Analisis dan Desain Sistem Informasi*, Andi, Yogyakarta
- Kadir, Abdul., 2003, *Pengenalan Sistem Informasi*. Andi, Yogyakarta.
- Nuarsa, I Wayan., 2002, *Menganalisis Data Spasial dengan Arcview 3.3*, PT, Elex Media Komputindo, Jakarta
- Prahasta, Eddy., 2001, *Konsep-konsep Dasar Sistem Informasi Geografis*, Informatika, Bandung
- Pressman, S.R., 2002, *Rekayasa Perangkat Lunak*, Andi Offset, Yogyakarta