

# PEMANFAATAN METODE HEURISTIK DALAM PENCARIAN JALUR TERPENDEK DENGAN ALGORITMA SEMUT DAN ALGORITMA GENETIKA

Aprizal  
Program Studi Sistem Informasi  
Email : [aphrydpn@yahoo.com](mailto:aphrydpn@yahoo.com)

## ABSTRAK

*Tanpa program komputer hanyalah menjadi sebuah kotak yang tak berguna. Secara umum, pencarian jalur terpendek dapat dibagi menjadi dua metode yaitu metode konvensional dan heuristik.*

*Pemanfaatan metode heuristik yang diharapkan dapat menyelesaikan masalah pencarian jalur terpendek dengan hasil yang lebih variatif dan dengan waktu perhitungan yang lebih singkat. Pada metode konvensional logika yang dipakai hanya dengan membandingkan jarak masing-masing node dan kemudian mencari jarak yang terpendek. Namun, kelemahan metode konvensional pada keakuratan hasil yang didapatkan serta tingkat kesalahan yang dihasilkan pada perhitungan. Hal tersebut tidak akan menjadi masalah jika data yang dibutuhkan hanya sedikit, sebaliknya maka akan menyebabkan peningkatan tingkat kesalahan perhitungan dan penurunan keakuratan.*

*Pemanfaatan teknologi informasi pada pencarian jalur terpendek menghasilkan suatu hasil atau keluaran yang akurat dan tepat, untuk pilihan perjalanan seseorang dengan mempertimbangkan beberapa parameter yang lain. Untuk kasus yang berbeda algoritma akan memberikan hasil yang berbeda, tidak dapat dipastikan bahwa algoritma semut atau genetik yang terbaik. Secara konsep algoritma, metode konvensional lebih mudah untuk dipahami tetapi, hasil yang diperoleh dari metode heuristik lebih variatif. Dengan metode heuristik, waktu perhitungan yang diperlukan lebih cepat 30% dibandingkan dengan menggunakan metode konvensional.*

**Kata kunci:** Pencarian jalur terpendek, Heuristik, Algoritma Semut, Algoritma Genetika

## I. PENDAHULUAN

Untuk menggunakan atau memfungsikan sebuah komputer maka harus terdapat program yang terdistribusi di dalamnya, tanpa program komputer hanyalah menjadi sebuah kotak yang tak berguna. Program yang terdapat pada komputer sangat bervariasi dan setiap program pasti menggunakan algoritma. Algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintahnya dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apapun dengan catatan untuk setiap masalah memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma.

Dalam kehidupan, sering dilakukan 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. Hasil penentuan jalur terpendek akan menjadi pertimbangan dalam pengambilan keputusan untuk menunjukkan jalur yang akan ditempuh dan yang didapatkan juga membutuhkan kecepatan dan keakuratan dengan bantuan komputer.

Secara umum, pencarian jalur terpendek dapat dibagi menjadi dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional cenderung lebih mudah dipahami

dari metode heuristik, tetapi jika dibandingkan, hasil yang diperoleh dari metode heuristik lebih variatif dan waktu perhitungan yang diperlukan lebih singkat.

### a. Rumusan Masalah

Seringkali penyelesaian masalah jalur terpendek masih menggunakan metode konvensional bahkan menggunakan perhitungan manual. Pemanfaatan metode heuristik masih sangat jarang digunakan, sehingga dapat dirumuskan sebuah masalah yaitu dengan pemanfaatan metode heuristik yang diharapkan nantinya dapat menyelesaikan masalah pencarian jalur terpendek dengan hasil yang lebih variatif dan dengan waktu perhitungan yang lebih singkat.

### b. Batasan Masalah

Dari latar belakang dan rumusan masalah yang telah dijelaskan, penelitian dibatasi pada dua jenis algoritma yang digunakan dalam metode heuristik, yaitu algoritma genetika (*Genetic Algorithm*, GA) dan algoritma semut (*Ant Colony Algorithm*, Antco).

### c. Tujuan Penelitian

Penelitian bertujuan menyelesaikan masalah rute menggunakan metode heuristik, khususnya

algoritma genetika dan algoritma semut, mencoba mengimplementasikan dengan sebuah kasus

#### d. Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian adalah:

1. Menawarkan penyelesaian yang lebih mudah dalam perhitungan (sesuai dengan tujuan algoritma heuristik) untuk pencarian jalur terpendek
2. Dapat diaplikasikan menjadi sebuah perangkat lunak

## 2. LANDASAN TEORI

### a. Pencarian jalur terpendek

Secara umum penyelesaian masalah pencarian jalur terpendek dapat dilakukan menggunakan dengan dua buah metode, yaitu metode algoritma konvensional dan metode heuristik. Metode algoritma konvensional diterapkan dengan cara perhitungan matematis seperti biasa, sedangkan metode heuristik diterapkan dengan perhitungan kecerdasan buatan, dengan menentukan basis pengetahuan dan perhitungannya.

#### a. Metode konvensional

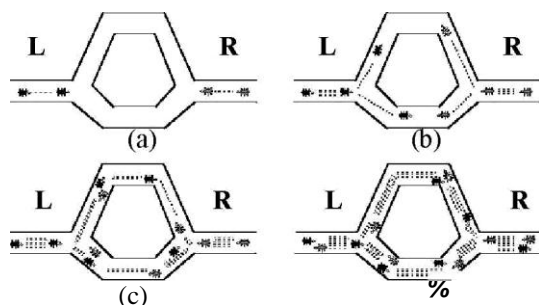
Metode konvensional berupa algoritma yang menggunakan perhitungan matematis biasa. Ada beberapa metode konvensional yang biasa digunakan untuk melakukan pencarian jalur terpendek, diantaranya algoritma Dijkstra, algoritma Floyd-Warshall, dan algoritma Bellman-Ford

#### b. Metode heuristic

Adalah sub bidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian dan penentuan jalur terpendek. Ada beberapa algoritma pada metode heuristik yang biasa digunakan dalam pencarian jalur terpendek. Namun dalam penelitian dibatasi hanya membahas dua macam algoritma yaitu algoritma semut dan algoritma genetika.

### b. Algoritma semut

Algoritma Semut diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996). Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan.



Gambar 1. Perjalanan semut menemukan sumber makanan.

sederhana, dan mempelajari lebih dalam tentang cabang dari ilmu kecerdasan buatan.

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 semakin jelas bekas jejak kakinya. Hal ini menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut melalui lintasan tersebut.

Gambar 1.a menunjukkan perjalanan semut dalam menemukan jalur terpendek dari sarang ke sumber makanan, terdapat dua kelompok semut yang melakukan perjalanan. Kelompok semut L berangkat dari arah kiri ke kanan dan kelompok semut R berangkat dari kanan ke kiri. Kedua kelompok berangkat dari titik yang sama dan dalam posisi pengambilan keputusan jalan sebelah mana yang akan diambil. Kelompok L membagi dua kelompok lagi. Sebagian melalui jalan atas dan sebagian melalui jalan bawah. Hal ini juga berlaku pada kelompok R. Gambar 1.b dan Gambar 1.c menunjukkan bahwa kelompok semut berjalan pada kecepatan yang sama dengan meninggalkan *feromon* atau jejak kaki di jalan yang telah dilalui. *Feromon* yang ditinggalkan oleh kumpulan semut yang melalui jalan atas telah mengalami banyak penguapan karena semut yang melalui jalan atas berjumlah lebih sedikit dari pada jalan yang di bawah. Hal ini disebabkan jarak yang ditempuh lebih panjang daripada jalan bawah. Sedangkan *feromon* yang berada di jalan bawah penguapannya cenderung lebih lama. Karena semut yang melalui jalan bawah lebih banyak daripada semut yang melalui jalan atas. Gambar 1.d menunjukkan bahwa semut-semut yang lain pada akhirnya memutuskan untuk melewati jalan bawah karena *feromon* yang ditinggalkan masih banyak. Sedangkan *feromon* pada jalan atas sudah banyak menguap sehingga semut-semut tidak memilih jalan atas tersebut. Semakin banyak semut yang melalui jalan maka semakin banyak semut yang mengikutinya, semakin sedikit semut yang melalui jalan, maka *feromon* yang ditinggalkan semakin berkurang. Dari sinilah kemudian terpilihlah jalur terpendek antara sarang dan 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 ( $T_{ij}$ )

(koordinat) atau dij (jarak antar kota)

2. Banyak kota ( $n$ ) termasuk  $x$  dan  $y$

3. Tetapan siklus-semut (Q)
  4. Tetapan pengendali intensitas jejak semut (a)
  5. Tetapan pengendali visibilitas (P)
  6. Visibilitas antar kota =  $1/d_{ij}$  (rijj)
  7. Banyak semut (m)
  8. Tetapan penguapan jejak semut (p)
  9. Jumlah siklus maksimum (NCmax) bersifat tetap selama algoritma dijalankan, sedangkan xij 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  $x_{ij}$  dilakukan, kemudian m semut ditempatkan pada kota pertama tertentu secara acak.

**Langkah 2:**

Penyisiran kota pertama ke dalam *tabu list*. Hasil inisialisasi kota pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen pertama *tabu list*. Hasil dari langkah ini adalah terisinya elemen pertama *tabu list* setiap semut dengan indeks kota tertentu, yang berarti bahwa setiap  $tabu_k(l)$  bisa berisi indeks kota antara 1 sampai n sebagaimana hasil inisialisasi pada langkah 1.

**Langkah 3:**

Penyusunan rate kunjungan setiap semut ke setiap kota. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap kota, akan mulai melakukan perjalanan dari kota pertama masing-masing 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 sampai semua kota satu persatu dikunjungi atau telah menempati  $tabu_k$ . 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:

$$untak_{je} \{N-tabu_k\}$$

$$k \in \{N-tabu_k\}$$

dan  $p_j = 0$ , untuk j lainnya

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

**Langkah 4:**

- a. Perhitungan panjang rute setiap semut.  
Perhitungan panjang rute tertutup (*length closed tour*) atau  $L_k$  setiap semut dilakukan setelah satu siklus diselesaikan oleh semua  $d$  semut. Perhitungan dilakukan<sup>2</sup> berdasarkan  $tabu_k$  masing-masing dengan persamaan berikut:

$$L_k = \sum_{s=1}^{n-1} d_{i_s, i_{s+1}}$$

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 didapat harga minimal panjang rute tertutup setiap siklus atau  $L_{minNC}$  dan harga minimal panjang rute tertutup secara keseluruhan adalah  $L_{min}$ .

- 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 perubahan ini adalah:

$$\Delta \tau_{ij} = \tau_{ij} - \rho \tau_{ij} + \frac{Q}{L_k}$$

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

$$\Delta \tau_{ij} = \frac{Q}{L_k}$$

untuk (i,j) e kota asal

dan kota tujuan dalam  $tabu_k$

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

**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:

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

- 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 2 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 2 dengan harga parameter intensitas jejak kaki semut antar kota yang sudah diperbaharui.

### **c. Algoritma genetika**

Algoritma genetika adalah algoritma pencarian yang didasarkan atas mekanisme seleksi alami dan evolusi biologis. Algoritma genetika mengkombinasikan antara deretan struktur dengan pertukaran informasi acak ke bentuk algoritma pencarian dengan beberapa perubahan bakat pada manusia. Pada setiap generasi, himpunan baru dari deretan individu dibuat berdasarkan kecocokan pada generasi sebelumnya (Goldberg, 1989).

Dalam algoritma genetika, diperlukan beberapa proses untuk menentukan jalur terpendek, yaitu:

#### **a. Proses Pengkodean (Encoding)**

Adalah salah satu proses yang sulit dalam algoritma genetika. Hal ini disebabkan Karena proses pengkodean untuk setiap permasalahan berbeda-beda karena tidak semua teknik pengkodean cocok untuk setiap permasalahan. Proses pengkodean ini menghasilkan suatu deretan yang kemudian disebut kromosom. Kromosom terdiri dari sekumpulan bit yang dikenal sebagai gen.

Ada beberapa macam teknik pengkodean yang dapat dilakukan dalam algoritma genetika (Lukas, 2005), diantaranya pengkodean biner (*binary encoding*), pengkodean permutasi (*permutation encoding*), pengkodean nilai (*value encoding*) dan pengkodean pohon (*tree encoding*).

#### **b. Proses Seleksi**

Adalah proses untuk menentukan individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana keturunan terbentuk dari individu-individu terpilih tersebut. Langkah pertama yang dilakukan dalam seleksi adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai obyektif dirinya sendiri terhadap nilai obyektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* kemudian akan digunakan pada tahap seleksi berikutnya.

Ada beberapa macam proses seleksi yang ada pada algoritma genetika, diantaranya (Kusumadewi, 2005):

1. Seleksi dengan Roda Roulette (*Roulette Wheel Selection*), dengan memetakan individu-individu dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*-nya.
2. Seleksi berdasarkan Ranking Fitness (*Rank-based Fitness*), yaitu dengan cara mengurutkan populasi menurut nilai obyektifnya.
3. Seleksi Pengambilan Sampling Stochastic (*Stochastic Universal Sampling*), dengan memetakan individu-individu seperti halnya roda roulette, kemudian memberikan sejumlah *pointer* sebanyak individu yang ingin diseleksi pada garis tersebut.
4. Seleksi Lokal (*Local Selection*), seleksi yang dilakukan hanya pada konstrain tertentu.
5. Seleksi dengan Pemotongan (*Truncation Selection*), seleksi buatan yang biasanya digunakan oleh populasi yang jumlahnya sangat besar.
6. Seleksi dengan Turnamen (*Tournament Selection*), menetapkan suatu nilai turnamen untuk individu-individu yang dipilih secara acak dari suatu populasi.

#### **c. Proses Rekombinasi**

Adalah proses untuk menyilangkan dua kromosom sehingga membentuk kromosom baru yang harapannya lebih baik dari pada induknya. Rekombinasi dikenal juga dengan nama *crossover*. Tidak semua kromosom pada suatu populasi akan mengalami proses rekombinasi. Kemungkinan suatu kromosom mengalami proses rekombinasi didasarkan pada probabilitas *crossover* yang telah ditentukan terlebih dahulu. Probabilitas *crossover* menyatakan peluang suatu kromosom akan mengalami *crossover*.

Ada beberapa macam proses rekombinasi yang ada pada algoritma genetika, diantaranya (Kusumadewi, 2005):

1. Rekombinasi diskret, dengan menukar nilai variabel antar kromosom induk
2. Rekombinasi menengah, merupakan metode rekombinasi yang hanya digunakan untuk variabel *real* dan variabel yang bukan biner.
3. Rekombinasi garis, memiliki prinsip yang sama dengan rekombinasi menengah, dengan nilai alpha sama untuk semua variabel.
4. Penyilangan satu titik, dengan menukar variabel-variabel antar kromosom pada satu titik untuk menghasilkan anak.
5. Penyilangan banyak titik, dengan menukar variabel-variabel antar kromosom pada banyak titik untuk menghasilkan anak.
6. Penyilangan seragam, dengan membuat sebuah mask penyilangan sepanjang panjang kromosom secara acak.
7. Penyilangan dengan permutasi, dengan cara memilih sub-barisan suatu turnamen dari satu

induk dengan tetap menjaga urutan dan posisi sejumlah kota yang mungkin terhadap induk lainnya.

**d. Proses Mutasi**

Adalah proses penambahan nilai acak yang sangat kecil dengan probabilitas rendah pada variabel keturunan. Peluang mutasi didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak dievaluasi, tetapi bila peluang mutasi ini terlalu besar maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya dan algoritma juga akan kehilangan kemampuan untuk belajar dari *history* pencarian (Kusumadewi, 2005). Ada beberapa macam proses mutasi yang ada pada algoritma genetika, diantaranya:

1. Mutasi bilangan real, dengan mendefinisikan ukuran langkah mutasi, kecil atau besar.
2. Mutasi biner, dengan mengganti satu atau beberapa nilai gen dari kromosom.

Berikut adalah langkah algoritma genetika sederhana untuk pencarian jalur terpendek:

**Langkah 1**

Inisialisasi generasi awal. Generasi awal harus diinisialisasi kosong, sehingga belum ada generasi. Generasi = 0.

**Langkah 2**

Inisialisasi populasi awal, P (generasi) secara acak. Populasi yang ditentukan di inisialisasi secara acak.

**Langkah 3**

Evaluasi nilai *fitness* pada setiap individu dalam P (generasi). Nilai *fitness* adalah nilai yang menunjukkan kualitas suatu kromosom dalam populasi.

**Langkah 4**

- a. Menambahkan generasi baru dengan persamaan: generasi = generasi+1
- b. Seleksi populasi tersebut untuk mendapatkan kandidat induk P' (generasi)
- c. Lakukan *crossover* pada P' (generasi).
- d. Lakukan mutasi pada P' (generasi).
- e. Lakukan evaluasi *fitness* setiap individu pada P' (generasi).
- f. Bentuk populasi baru, P (generasi) = {P(generasi1) yang bertahan, P' (generasi)}.

**3. METODE PENELITIAN**

Pada penelitian ini, terdapat beberapa metode pengumpulan data yang digunakan, yaitu: 1. Metode kepustakaan

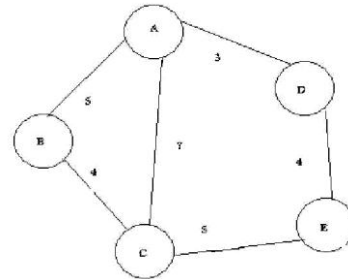
Metode pengumpulan data kepustakaan

dilakukan dengan mengumpulkan data-data dari sumber atau buku yang relevan terhadap penelitian. 2. Metode wawancara

Metode wawancara dilakukan dengan cara tatap muka dan menanyakan langsung kepada objek yang pernah melakukan penelitian sebelumnya.

**4. PEMBAHASAN**

**4.1 Pencarian jalur terpendek**



**Gambar 2.** Ilustrasi Jalur Terpendek dengan Jarak

Pada dasarnya permasalahan pencarian jalur terpendek antar kota merupakan pencarian jalur terpendek antar titik yang telah diketahui koordinatnya. Dengan mengetahui konsep pencarian jalur terpendek antar titik, untuk selanjutnya dapat diterapkan pada pencarian jalur terpendek pada berbagai kota yang ingin diketahui. Contoh kasus yang akan diambil adalah pencarian jalur terpendek antara titik A dan titik E.

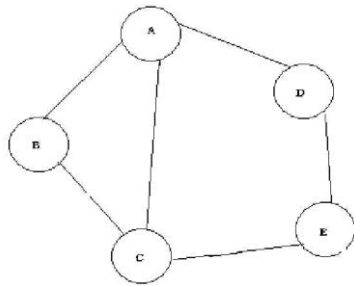
Terdapat dua jenis kasus yang bisa diturunkan dari gambar di atas. Kasus pertama adalah mengetahui jarak antar node yang ditunjukkan dengan garis penghubung antar titik.. Kasus yang kedua adalah dengan dengan mengetahui koordinat titik saja. Gambar 2 merupakan jenis kasus yang pertama yaitu dengan mengetahui jarak antar titik. Sedangkan gambar 3 merupakan jenis kasus yang kedua, yaitu dengan mengetahui titik koordinatnya saja. Untuk kasus pertama, penyelesaian cenderung lebih mudah karena jarak antar titik telah diketahui sebagai berikut:

**Tabel 1.** Tabel Jarak antar Titik

	A	B	C	D	E
A	0	5	7	3	-
B	5	0	4	-	-
C	7	4	0	-	5
D	3	-	-	0	4
E	-	-	5	4	0

Sedangkan untuk kasus kedua yang telah ditunjukkan pada Gambar 3.

Dari gambar 3 di atas, misalnya titik–titik yang telah ditentukan mempunyai koordinat sebagaimana tabel 2.



**Gambar 3.** Ilustrasi Jalur Terpendek tanpa Jarak

**Tabel 2.** Koordinat titik antar kota

	X	Y
A	20	50
B	10	25
C	18	10
D	50	40
E	55	15

Karena belum diketahui jarak antar titiknya dan hanya diketahui titik koordinat saja, Maka perhitungan dimulai dari penentuan jarak antar simpul titik dengan menggunakan titik yang diketahui. Titik yang diketahui menggunakan koordinat sumbu X dan Y. Langkah di atas merupakan langkah yang harus dilakukan untuk semua metode.

Penyelesaian kasus diatas dapat dilakukan dengan dua metode seperti yang telah dijelaskan pada bab II, yaitu metode konvensional dan metode heuristik. Pada metode konvensional, logika yang dipakai sangat sederhana. Yaitu hanya dengan membandingkan jarak masing-masing node dan kemudian mencari jarak yang terpendek. Namun, kelemahan metode konvensional ini adalah pada keakuratan hasil yang didapatkan serta tingkat kesalahan yang dihasilkan pada perhitungan. Hal tersebut tidak akan menjadi masalah jika data yang dibutuhkan hanya sedikit, sebaliknya maka akan menyebabkan peningkatan tingkat kesalahan perhitungan dan penurunan keakuratan. Penelitian ini akan membahas mengenai penyelesaian jalur terpendek menggunakan metode heuristik dengan algoritma semut dan algoritma genetika.

#### 4.2 Penyelesaian pencarian jalur terpendek

##### a. Algoritma semut

Berdasarkan contoh kasus yang ada, maka langkah yang harus dilakukan adalah: 1. Menginisialisasi parameter-parameter yang diperlukan, yaitu:

- $T_{ij}$  (intensitas jejak semut antar kota) dan perubahannya, parameter ini berfungsi menentukan jumlah intensitas jejak semut antar kota sehingga diketahui jalur terpendek yang dihasilkan.
- n (banyak kota) termasuk x dan y (koordinat) atau  $d_{ij}$  (jarak antar kota), pada contoh kasus

di atas, jumlah  $n = 5$  dan mempunyai koordinat atau jarak yang telah ditentukan.

- Q (tetapan siklus-semut), a (tetapan pengendali intensitas jejak semut), p (tetapan pengendali visibilitas),  $r^{\wedge}$  (visibilitas antar kota= $l/d_{ij}$ ), dan p (tetapan penguapan jejak semut), nilai dari parameter hams didefinisikan dahulu karena bersifat sebagai konstanta.
  - m (banyak semut), jumlah semut yang akan digunakan untuk menyelusuri jalur bisa bernilai sembarang tergantung oleh pengguna. Contoh kasus diatas menggunakan 100 semut.
  - $NC_{max}$  (jumlah siklus maksimum), adalah jumlah maksimum siklus yang ingin di jalankan, hingga menemukan hasil yang terbaik. Misal maksimum siklus 10 kali, maka perhitungan akan dilakukan maksimal 10 kali hingga menemukan hasil yang terpendek.
- Menempatkan kelompok semut tersebut pada kota pertama. Pemilihan kota pertama dilakukan secara acak.
  - Setelah menempatkan kota pertama dalam tabu list, dimulailah perjalanan semut-semut tersebut dari kota pertama menuju kota tujuan yang telah ditentukan berdasarkan persamaan probabilitas pada bagian dua. Jarak yang dicari adalah jarak dari kota A ke kota E, dengan mencari jalur terpendek dan hasil jalur yang didapatkan tidak harus melewati semua kota.
  - Menghitung panjang perjalanan dari masing-masing semut dan kemudian ditentukan jalur terpendek berdasarkan  $T_{ij}$  (harga intensitas jejak kaki semut).
  - Langkah 1 sampai langkah 4 akan diulang sebanyak  $N_{cmax}$  atau jika telah mengalami konvergen. Setiap dimulainya siklus baru, maka harga  $T_{ij}$  *di-reset* ulang bernilai sama dengan nol.

##### b. Algoritma genetika

Berdasarkan contoh kasus di atas, maka langkah yang harus dilakukan adalah: 1. Menginisialisasi parameter-parameter yang diperlukan, yaitu:

- Popsiz (ukuran populasi): ukuran populasi yang akan digunakan untuk perhitungan. Contoh pada kasus diatas ditentukan ukuran populasi sebanyak 10.
- pc (peluang crossover): peluang terjadinya rekombinasi atau persilangan pada suatu kromosom. Contoh pada kasus diatas ditentukan peluang *crossover* sebesar 0.500
- pm (peluang mutasi): persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Contoh pada kasus diatas ditentukan peluang mutasi sebesar 0.100

- d. kb (peluang pelestarian): peluang pelestarian dari populasi yang ada. Contoh pada kasus diatas ditentukan peluang pelestarian sebesar 0.100
- e. MaxGen (maksimum generasi): maksimum generasi yang akan ada pada perhitungan. Contoh pada kasus diatas ditentukan maksimum jumlah generasi sebanyak 50
- f. (panjang kromosom): panjang kromosom yang diinginkan. Contoh pada kasus diatas ditentukan panjang kromosom adalah 5

2. Melakukan seleksi sesuai dengan metode yang dibutuhkan.  
Misalkan pada kasus, dilakukan seleksi menggunakan metode Roda Roulette, maka langkah-langkah yang harus dikerjakan:

- a. Mencari nilai fitness relatif dan fitness kumulatif. Misalkan Total Fitness = 12,5720.

Maka dapat dicari fitness relatif (pk) dari tiap-tiap kromosom:

- 1.  $p_1 = F_1 / \text{Total Fitness} = 0,902 / 12,5720 = 0,071$ .
- 2.  $p_2 = F_2 / \text{Total Fitness} = 0,823 / 12,5720 = 0,065$ .
- 3.  $p_3 = F_3 / \text{Total Fitness} = 0,627 / 12,5720 = 0,049$ .

dan seterusnya sampai sebanyak ukuran populasi. Fitness kumulatif (qk) dapat dicari sebagai berikut:

- 1.  $q_1 = p_1 = 0,071$ .
- 2.  $q_2 = q_1 + p_2 = 0,071 + 0,065 = 0,136$ .
- 3.  $q_3 = q_2 + p_3 = 0,136 + 0,049$  dan seterusnya sampai sebanyak ukuran populasi.

- b. Membangkitkan bilangan acak sebanyak ukuran populasi = 10. Bilangan acak berukuran 0-1.

3. Melakukan persilangan sesuai dengan metode yang dibutuhkan.

- a. Membangkitkan bilangan acak antara 0-1 sebanyak ukuran populasi, yaitu 10 buah untuk memilih kromosom mana saja yang akan dilakukan persilangan.
- b. Memilih bilangan acak yang kurang dari peluang crossover, sehingga kromosomnya dapat disilangkan.

- 4. Melakukan mutasi sesuai dengan metode yang dibutuhkan.
  - a. Menghitung jumlah bit yang ada pada populasi, dengan rumus  $\text{Popsiz} * L = 10 * 6 = 60$ .
  - b. Membangkitkan bilangan acak antara 0-1 sebanyak jumlah bit (360 buah).
  - c. Kromosom yang terkecil daripada peluang mutasi, akan terkena mutasi.
  - d. Populasi akhir setelah dilakukan mutasi akan dijadikan sebagai populasi awal untuk generasi berikutnya.
  - e. Ulangi langkah a sampai d sebanyak ukuran maksimum gen = 50.
  - f. Temukan nilai fitness terbaik

## 5. KESIMPULAN

- a. Pemanfaatan teknologi informasi pada pencarian jalur terpendek menghasilkan suatu hasil atau keluaran yang akurat dan tepat, untuk pilihan perjalanan seseorang dengan mempertimbangkan beberapa parameter yang lain.
- b. Untuk kasus yang berbeda algoritma akan memberikan hasil yang berbeda, tidak dapat dipastikan bahwa algoritma semut atau genetic yang terbaik.
- c. Secara konsep algoritma, metode konvensional lebih mudah untuk dipahami. Namun, hasil yang diperoleh dari metode heuristik lebih variatif.
- d. Dengan metode heuristik, waktu perhitungan yang diperlukan lebih cepat 30% dibandingkan dengan menggunakan metode konvensional.

## 6. SARAN

- a. Diharapkan ada penelitian lebih lanjut untuk mengetahui efisiensi dari pencarian jalur terpendek menggunakan Metode heuristik.
- b. Diharapkan adanya penelitian yang dapat membandingkan antar metode heuristik yang lain.

## DAFTAR PUSTAKA

- 1) Goldberg, D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*, New York: Addison-Wesley, 1989
- 2) Kusumadewi, S., *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu, 2003
- 3) Kusumadewi, S., dan H., Purnomo, *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristi*, Yogyakarta: Graha Ilmu, 2005
- 4) Lukas, dkk., “Penerapan Algoritma Genetika untuk Travelling Salesman Problem dengan Menggunakan Metode Order Crossover dan Insertion Mutation”, Seminar Nasional Aplikasi Teknologi Informasi, hlm: 1 s/d 5, 2005
- 5) Efendi, R., “Penerapan algoritma semut untuk pemecahan masalah spanning tree pada kasus pemasangan jaringan kabel telepon”. *Tugas Akhir*, Jurusan Teknik Informatika, Universitas Islam Indonesia, 2003.