

## PERANCANGAN ALGORITMA GENETIKA UNTUK MENENTUKAN JALUR TERPENDEK

Fajar Saptono<sup>1</sup>, Taufiq Hidayat<sup>2</sup>

Laboratorium Pemrograman dan Informatika Teori

Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia

Jln. Kaliurang KM 14,5 Yogyakarta 55501, Telp. (0274) 895287 ext: 134, Faks. (0274) 895007

e-mail: [guitarist\\_keren@yahoo.com](mailto:guitarist_keren@yahoo.com)<sup>1</sup>, [ach\\_fiqrin@yahoo.com](mailto:ach_fiqrin@yahoo.com)<sup>2</sup>

### ABSTRAKSI

Algoritma genetika merupakan salah satu metode penyelesaian optimasi yang dikenal mampu menghasilkan nilai optimum. Makalah ini menerapkan perancangan algoritma genetika pada kasus Shortest Path Problem, dimana jalur terpendek dapat dilalui tanpa harus kembali ke titik awal seperti halnya kasus Travelling Salesman Problem Dengan menggunakan contoh data jarak antar kota yang telah diketahui dan representasi graf, algoritma genetika dapat memberi jalur optimum sesuai dengan yang diharapkan..

**Kata kunci:** Algoritma Genetika, Shortest Path Problem, Jalur Terpendek

### 1. LATAR BELAKANG

Sebuah perjalanan terkadang membutuhkan jalur/rute yang terpendek. Biasanya jalur terpendek tersebut didapatkan dengan cara menghitung waktu yang ditempuh, ataupun berdasarkan jarak dari kota asal ke kota tujuan. Semakin banyak alternatif jalur ke kota tujuan, semakin rumit cara untuk menghitung jalur terpendek. Untuk itu diperlukan sebuah mekanisme yang handal untuk dapat menentukan jalur terpendek dari kota sumber ke kota tujuan. Penerapan metode AI dalam perhitungan jalur terpendek merupakan salah satu solusi untuk dapat menyelesaikan masalah dengan jalur yang banyak dan rumit.

Algoritma Genetika (*Genetic Algorithm*, GA) merupakan salah satu cabang dari AI. Penemu algoritma genetika, John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. GA juga sering digunakan pada penyelesaian masalah optimasi, seperti pada kasus *Travelling Salesman Problem* (TSP), *Minimum Spanning Tree* (MST), dan Masalah Jalur Terpendek (*Shortest Path Problem*). Diharapkan penggunaan algoritma genetika pada masalah jalur terpendek menghasilkan suatu perhitungan yang akurat.

### 2. RUMUSAN MASALAH

Berdasarkan latar belakang di atas dapat dirumuskan permasalahan yang akan diselesaikan yaitu bagaimana merancang algoritma genetika untuk menyelesaikan permasalahan jalur terpendek.

### 3. TUJUAN

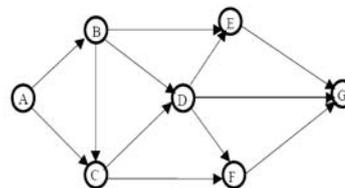
Tujuan yang diharapkan dari penelitian ini adalah membuat suatu rancangan algoritma genetika sehingga dapat menyelesaikan permasalahan jalur terpendek.

### 4. DASAR TEORI

#### 4.1 Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*) [11]. Suatu graf  $G$  terdiri dari dua himpunan yaitu himpunan  $V$  (simpul) dan himpunan  $E$  (busur).

Busur dapat menunjukkan hubungan (relasi) sembarang seperti rute penerbangan, jalan raya, sambungan telepon, ikatan kimia, dan lain-lain. Notasi graf:  $G(V, E)$  artinya graf  $G$  memiliki simpul  $V$  dan busur  $E$ .



Gambar 1. Contoh graf ABCDEFG

Menurut arah dan bobotnya, graf dibagi menjadi empat bagian, yaitu:

- Graf berarah dan berbobot: tiap busur mempunyai anak panah dan bobot.
- Graf tidak berarah dan berbobot: tiap busur tidak mempunyai anak panah tetapi mempunyai bobot.
- Graf berarah dan tidak berbobot: tiap busur mempunyai anak panah yang tidak berbobot.
- Graf tidak berarah dan tidak berbobot: tiap busur tidak mempunyai anak panah dan tidak berbobot.

Suatu graf dapat direpresentasikan ke beberapa bentuk. Representasi graf dapat digunakan untuk mengimplementasikan graf tersebut ke dalam bentuk tertentu, sehingga dapat digunakan pada berbagai kasus yang berbeda. Representasi graf yang sering digunakan diantaranya:

- Matriks Kedekatan (*Adjacency Matrix*)

Untuk suatu graf dengan jumlah simpul sebanyak  $n$ , maka matriks kedekatan mempunyai ukuran  $n$

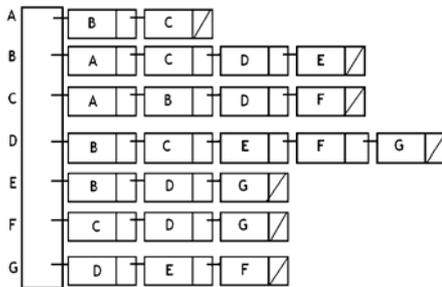
$x \times n$  ( $n$  baris dan  $n$  kolom). Matriks kedekatan untuk graf ABCDEFG dapat dilihat pada tabel 1.

**Tabel 1.** Matriks kedekatan graf ABCDEFG

	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	1	0	1	1	1	0	0
C	1	1	0	1	0	1	0
D	0	1	1	0	1	1	1
E	0	1	0	1	0	0	1
F	0	0	1	1	0	0	1
G	0	0	0	1	1	1	0

b. Senarai Kedekatan (*Adjacency List*)

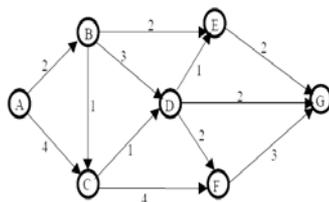
Pada simpul  $x$  dapat dianggap sebagai suatu senarai yang terdiri dari simpul pada graf yang berdekatan dengan  $x$ . Senarai kedekatan untuk graf ABCDEFG dapat dilihat pada gambar 2.



**Gambar 2.** Senarai kedekatan graf ABCDEFG

**4.2 Permasalahan Jalur Terpendek (*Shortest Path Problem*)**

Jalur terpendek adalah suatu jaringan pengarah perjalanan dimana seseorang pengarah jalan ingin menentukan jalur terpendek antara dua kota, berdasarkan beberapa jalur alternatif yang tersedia, dimana titik tujuan hanya satu. Gambar 3 menunjukkan suatu graf ABCDEFG.



**Gambar 3.** Graf berbobot ABCDEFG

Pada gambar 3, misalkan kita dari kota A ingin menuju Kota G. Untuk menuju kota G, dapat dipilih beberapa jalur yang tersedia:

- A → B → C → D → E → G
- A → B → C → D → F → G
- A → B → C → D → G
- A → B → C → F → G
- A → B → D → E → G
- A → B → D → F → G
- A → B → D → G

- A → B → E → G
- A → C → D → E → G
- A → C → D → F → G
- A → C → D → G
- A → C → F → G

Berdasarkan data di atas, dapat dihitung jalur terpendek dengan mencari jarak antara jalur-jalur tersebut. Apabila jarak antar jalur belum diketahui, jarak dapat dihitung berdasarkan koordinat kota-kota tersebut. Setelah didapatkan hasil jarak antar kota, jalur terpendek dapat dihitung menggunakan metode yang ada.

**4.3 Algoritma Genetika**

Algoritma genetika (*Genetic Algorithm, GA*) 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 [2]

Satu siklus iterasi algoritma genetika (sering disebut sebagai generasi) terdapat dua proses, yakni proses seleksi dan rekombinasi. Proses seleksi adalah proses evaluasi kualitas setiap string didalam populasi untuk memperoleh peringkat calon solusi. Berdasarkan hasil evaluasi, dipilih string-string yang akan mengalami proses rekombinasi. Proses pemilihan biasanya dilakukan secara acak, string dengan kualitas yang lebih baik akan memiliki peluang lebih besar untuk terpilih sebagai calon-calon string generasi berikutnya.

Proses rekombinasi meliputi proses genetika untuk memperoleh string baru dari pertukaran karakter dari calon-calon string yang didapat pada tahap seleksi. String-string pada generasi baru dihasilkan dengan menggunakan operasi genetik secara acak pada calon string yang terpilih pada tahap seleksi. Proses rekombinasi akan menghasilkan string-string baru yang berbeda dibandingkan induknya dan dengan demikian diperoleh domain pencarian yang baru [9].

Cara kerja algoritma genetika sangat sederhana, hanya mencakup proses penduplikasian string-string dan pertukaran bagian-bagian dari string. Meskipun cukup sederhana, tetapi mempunyai kemampuan untuk menyelesaikan persoalan optimasi. Kemampuan ini didukung oleh tiga operator genetik yaitu reproduksi, rekombinasi dan mutasi. Pada reproduksi terjadi proses penduplikasian string berdasarkan nilai fungsi objektifnya. Nilai objektif ini dapat dilihat sebagai suatu keuntungan yang ingin dicapai atau dimaksimalkan. Sementara proses pertukaran bagian-bagian string dilakukan oleh operator rekombinasi dan mutasi.

Di samping ketiga operator dasar (reproduksi, rekombinasi, dan mutasi), parameter-parameter genetik (jumlah populasi, maksimum generasi, probabilitas rekombinasi, probabilitas mutasi, dan lain-lain), serta asumsi-asumsi yang digunakan dalam pemodelannya juga mempunyai peran penting.

Pada algoritma genetika terdapat beberapa proses yaitu:

a. Proses Pengkodean (*Encoding*)

Pada proses pengkodean, gen dapat direpresentasikan dalam bentuk string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika [5].

Ada beberapa macam teknik pengkodean yang dapat dilakukan dalam algoritma genetika, diantaranya pengkodean biner (*binary encoding*), pengkodean permutasi (*permutation encoding*), pengkodean nilai (*value encoding*) dan pengkodean pohon (*tree encoding*) [6].

b. Proses Seleksi

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 [5]: Seleksi dengan Roda Roulette (*Roulette Wheel Selection*), seleksi berdasarkan Ranking Fitness (*Rank-based Fitness*), seleksi *Stochastic Universal Sampling*, seleksi Lokal (*Local Selection*), seleksi dengan Pemotongan (*Truncation Selection*) dan seleksi dengan Turnamen (*Tournament Selection*).

c. Proses Rekombinasi

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*.

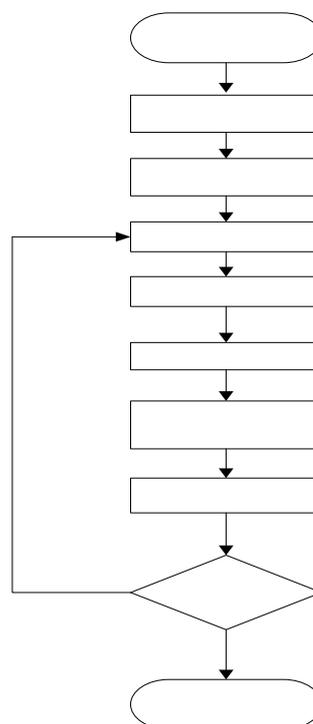
Ada beberapa macam proses rekombinasi yang ada pada algoritma genetika, diantaranya [5]: rekombinasi diskret, rekombinasi menengah, rekombinasi garis, rekombinasi satu titik, rekombinasi banyak titik, rekombinasi seragam, rekombinasi dengan permutasi,

d. Proses Mutasi

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 akan kehilangan kemiripan dari induknya dan algoritma juga akan kehilangan kemampuan untuk belajar dari *history* pencarian [5].

Ada dua macam proses mutasi yang ada pada algoritma genetika, diantaranya mutasi bilangan real dan mutasi biner.

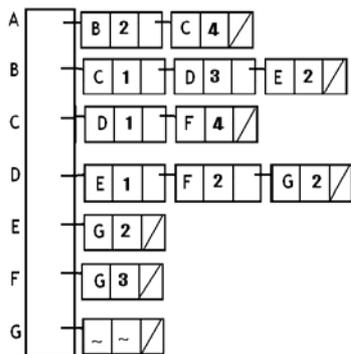
Berikut adalah diagram alir untuk algoritma genetika:



Gambar 4. Diagram alir algoritma genetika

## 5. RANCANGAN ALGORITMA GENETIKA

Pada graf berbobot ABCDEG diatas, dapat dibuat representasi menggunakan senarai kedekatan, pada gambar 4.



**Gambar 5.** Senarai kedekatan graf berbobot ABCDEFG

Apabila digunakan parameter sebagai berikut:

- Ukuran Populasi = 10
- Peluang Crossover = 0.500
- Peluang Mutasi = 0.100
- Peluang Pelestarian = 0.100
- Maksimum Generasi = 50
- Panjang Kromosom = 7

Maka perancangan algoritma genetika adalah: GENERASI 1

- a. Menentukan Populasi awal dengan nilai fitnessnya sejumlah ukuran populasi. Tabel populasi awal terdapat pada tabel 2.

**Tabel 2.** Populasi Awal

	<i>Kromosom</i>	<i>Fitness</i>
1	ACFGBDE	0.091
2	ABDFGCE	0.100
3	ABCDEGF	0.143
4	ACDFGBE	0.100
5	ABCDGEF	0.167
6	ABEGCDF	0.167
7	ACDGBEF	0.143
8	ABDGCEF	0.143
9	ABCFGDE	0.100
10	ABCDFGE	0.111

- b. Menentukan nilai probabilitas fitness sejumlah ukuran populasi. Nilai probabilitas fitness untuk generasi pertama terdapat pada Tabel 3.

**Tabel 3.** Probabilitas Fitness

	<i>pk</i>	<i>qk</i>
1	0.072	0.072
2	0.079	0.151
3	0.113	0.264
4	0.079	0.343
5	0.132	0.475
6	0.132	0.607
7	0.113	0.720
8	0.113	0.833
9	0.079	0.912
10	0.088	1.000

- c. Membangkitkan bilangan acak sejumlah ukuran populasi untuk seleksi. Bilangan acak untuk seleksi pada generasi pertama terdapat pada Tabel 4.

**Tabel 4.** Bilangan Acak untuk Seleksi

	<i>Bilangan</i>
1	0.400
2	0.419
3	0.019
4	0.803
5	0.814
6	0.659
7	0.138
8	0.624
9	0.207
10	0.944

- d. Didapatkan hasil dari seleksi, yaitu pencocokan nilai jangkauan bilangan acak untuk seleksi dengan  $q_k$ . Hasil seleksi dengan nilai fitness terdapat pada Tabel 5.

**Tabel 5.** Hasil Seleksi

	<i>Kromosom</i>	<i>Fitness</i>	<i>Ke-</i>
1	ABCDGEF	0.091	5
2	ABCDGEF	0.091	5
3	ACFGBDE	0.091	1
4	ABDGCEF	0.143	8
5	ABDGCEF	0.143	8
6	ACDGBEF	0.143	7
7	ABDFGCE	0.100	2
8	ACDGBEF	0.143	7
9	ABCDEGF	0.143	3
10	ABCDFGE	0.111	10

- e. Membangkitkan bilangan acak sejumlah ukuran populasi untuk rekombinasi. Bilangan acak untuk rekombinasi pada generasi pertama terdapat pada Tabel 6.

**Tabel 6.** Bilangan Acak untuk Seleksi

	<i>Bilangan</i>
1	0.124
2	0.687
3	0.073
4	0.235
5	0.746
6	0.794
7	0.705
8	0.710
9	0.723
10	0.480

- f. Mencari induk yang akan direkombinasi. Induk yang akan direkombinasi pada generasi pertama terdapat pada Tabel 7.

**Tabel 7.** Induk yang akan direkombinasi

	<b>Kromosom</b>	<b>Fitness</b>	<b>Ke-</b>
<b>1</b>	<b>ACFGBDE</b>	0.091	1
<b>2</b>	<b>ABCDEGF</b>	0.143	3
<b>3</b>	<b>ACDFGBE</b>	0.100	4
<b>4</b>	<b>ABCDFGE</b>	0.111	10

- g. Membangkitkan bilangan acak sejumlah ukuran populasi untuk mutasi. Bilangan acak untuk mutasi pada generasi pertama terdapat pada Tabel 8.

**Tabel 8.** Bilangan Acak untuk Mutasi

	<b>Bilangan</b>
<b>1</b>	0.170
<b>2</b>	0.025
<b>3</b>	0.240
<b>4</b>	0.004
<b>5</b>	0.478
<b>6</b>	0.024
<b>7</b>	0.824
<b>8</b>	0.173
<b>9</b>	0.582
<b>10</b>	0.429

- h. Kromosom yang kena mutasi adalah: 2,4,6. Kromosom yang terkena mutasi terdapat pada tabel 9.

**Tabel 9.** Kromosom yang terkena mutasi

	<b>Kromosom</b>	<b>Fitness</b>	<b>Ke-</b>
<b>1</b>	<b>ABDFGCE</b>	0.100	2
<b>2</b>	<b>ACDFGBE</b>	0.100	4
<b>3</b>	<b>ABEGCDF</b>	0.167	6

Sehingga didapatkan populasi baru untuk generasi selanjutnya, berdasarkan proses seleksi, rekombinasi dan mutasi.

Iterasi akan berulang sampai seterusnya sampai generasi ke-50 (maksimum generasi), sehingga didapatkan jalur terpendek antara graf ABCDEFG adalah jalur A-B-E-G dengan bobot 6.

## 6. KESIMPULAN

Berdasarkan perancangan algoritma genetika yang ada, maka dapat diambil beberapa kesimpulan yaitu:

- Untuk kasus Shortest Path Problem, kromosom dirancang dengan menggunakan prinsip pengacakan, dimana untuk perhitungan dilakukan hanya untuk jalur yang dilalui dari kota sumber dan kota tujuan, dengan sisa kromosom (kota yang tidak dilalui) berfungsi sebagai pelengkap agar kromosom tidak terpotong.
- Tingkat persentase optimasi algoritma genetika ditentukan oleh maksimum generasi. Semakin

banyak generasi yang diproses, maka semakin tinggi tingkat optimasi suatu kasus.

- Dengan tidak mengharuskan melalui seluruh jalur, hasil optimasi kasus Shortest Path dapat lebih menghasilkan tingkat optimasi yang lebih tinggi dibanding kasus Travelling Salesman Problem dimana seluruh jalur harus dilalui.

## PUSTAKA

- Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computing*. Heidelberg: Springer. 2003.
- Goldberg, D. E. *Genetic Algorithms in Search, Optimization & Machine Learning*. New York: Addison-Wesley. 1989.
- Kruse, R. L., Leung, B. P. & Tondo, C. L. *Data Structures and Program Design in C*. New Delhi: Prentice-Hall Inc. 1999.
- Kusumadewi, S. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu. 2003.
- Kusumadewi, S. & Purnomo, H. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu. 2005.
- Lukas, S., Anwar, T. & Yuliani, W. Penerapan Algoritma Genetika untuk Travelling Salesman Problem dengan Menggunakan Metode Order Crossover dan Insertion Mutation, *Seminar Nasional Aplikasi Teknologi Informasi*, I 1-I 5. 2005.
- Mutakhir, I., Saptono, F. & Hasanah, N. Implementasi Metode Heuristik pada Teknologi Informasi untuk Menentukan Jarak Terpendek. Makalah disampaikan pada *Lomba Karya Tulis Ilmiah Universitas Islam Indonesia*. Yogyakarta. 2007.
- Rayward, V. J., Osman, I. H., Reeves, C. R. & Smith, G. D. *Modern Heuristic Search Methods*. John Wiley & Sons: England. 1996.
- Saputro, N. & Dirgagautama, E. Penerapan Algoritma Genetik pada Permainan Catur Jawa. *Integral*, Vol 9, No.1, 17-26. 2004.
- Tettamanzi, A. & Tomassini, M. *Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems*. Heidelberg: Springer. 2001.
- Zakaria, T. M. & Prijono, A. *Konsep dan Implementasi Struktur Data*. Bandung: Informatika. 2006.