

Pencarian Solusi TSP (*Travelling Salesman Problem*) Menggunakan Algoritma Genetik

Teddy Rachmayadi

Teknik Informatika Institut Teknologi Bandung
Ganeca 10 Bandung
if16079@students.if.itb.ac.id

ABSTRAK

Algoritma genetik merupakan algoritma pencarian solusi yang didasarkan pada pemodelan suatu populasi. Pada algoritma genetik, ruang pencarian solusi dimodelkan sebagai kumpulan individu dalam populasi tersebut. Kumpulan individu tersebut merupakan kumpulan yang nilainya terbatas. Solusi yang dicari menggunakan algoritma genetik dimodelkan seperti sebuah kromosom pada makhluk hidup. Pada algoritma genetik, terdapat istilah-istilah penting yaitu gen, kromosom, dan rekombinasi. Gen merupakan pemodelan entitas pada persoalan. Dalam TSP, gen memodelkan sebuah kota yang harus didatangi. Kromosom merupakan sekumpulan gen yang menjadi kandidat solusi pencarian. Pada TSP, kromosom memodelkan tur kota yang dilalui. Lalu, rekombinasi merupakan metode untuk mendapatkan nilai yang dicari. Rekombinasi merupakan suatu proses memodifikasi gen yang terdapat pada suatu kromosom. Proses rekombinasi ini dilakukan ketika ada dua buah individu yang bereproduksi. Rekombinasi dapat berupa persilangan dan mutasi. Pada algoritma genetik, proses reproduksi dilakukan secara konkuren. Hal ini merupakan pemodelan dari peristiwa reproduksi yang terjadi dalam suatu populasi. Pencarian yang dilakukan pada algoritma genetik dihentikan jika ditemukan solusi optimal, tidak ditemukan solusi yang lebih baik, atau waktu yang dialokasikan telah digunakan sepenuhnya.

Kata kunci: algoritma genetik, TSP.

1. PENDAHULUAN

Algoritma genetik merupakan pemodelan komputasi yang terinspirasi oleh evolusi. Algoritma ini digunakan untuk proses optimasi. Algoritma genetik menggunakan istilah gen dalam menyimpan informasi. Gen merupakan struktur paling sederhana pada algoritma genetik. Pada algoritma ini, solusi optimal direpresentasikan sebagai untaian gen. Untaian gen tersebut disimpan dalam struktur

data bernama kromosom. Proses pencarian solusi dilakukan dengan cara melakukan operasi terhadap kromosom. Operasi yang dilakukan berupa rekombinasi kromosom. Operasi rekombinasi dapat dilakukan dengan dua cara. Cara pertama ialah persilangan. Lalu, cara kedua ialah mutasi. Cara persilangan dilakukan dengan cara menyilangkan sebagian gen orang tua pada kromosom anak.

tabel 1. Ilustrasi rekombinasi berupa persilangan

Nomor gen	Orang tua 1	Orang tua 2	Anak 1	Anak 2
1	A	A	A	A
2	B	B	B	B
3	C	C	C	C
4	D	D	D	D
5	E	E	E	E
6	F	F	F	F
7	G	G	G	G

tabel 2. Ilustrasi rekombinasi berupa mutasi

Nomor gen	Orang tua 1	Orang tua 2	Anak 1	Anak 2
1	A	A	A	A
2	B	B	B	B
3	C	C	C	C
4	D	D	D	d
5	E	E	E	E
6	F	F	F	F
7	G	G	G	G

Pada tabel1 dan tabel2, diperlihatkan tiga buah individu. Pada tabel1, orang tua 1 memiliki kromosom *ABCDEFGHI*. Sementara itu, orang tua 2 memiliki kromosom *ABCDEF*. Dari proses perkawinan, diperoleh dua anak dengan kromosom hasil persilangan kedua orang tuanya. Pada tabel1, kromosom anak mewarisi sebagian dari kromosom tiap orang tuanya. Pada tabel2, kromosom anak yang dihasilkan tidak hanya

memiliki gen hasil persilangan, melainkan juga memiliki gen yang bermutasi. Gen yang bermutasi pada perkawinan kedua ialah gen nomor 4.

2. METODE

Secara umum, algoritma genetik diimplementasikan dengan cara membuat suatu populasi dengan berbagai kromosom. Setelah populasi terbentuk, dilakukan evaluasi terhadap sejumlah individu dengan menggunakan fungsi kelayakan. Individu-individu yang lolos evaluasi merupakan individu-individu yang diharapkan dapat memberikan solusi optimasi. Individu-individu tersebut kemudian dikawinkan dengan harapan memperoleh solusi optimasi yang lebih baik. Proses pemilihan individu-individu tersebut dilakukan berulang-ulang untuk setiap generasi hingga diperoleh hasil yang optimal atau hasil yang dianggap optimal. Secara umum, algoritma genetik ialah sebagai berikut:

1. Memilih populasi awal
2. Menghitung nilai kelayakan dari setiap individu
3. *Repeat*
 - 3.1. Memilih individu-individu yang terbaik berdasarkan fungsi kelayakan untuk dilakukan reproduksi
 - 3.2. Menghasilkan generasi baru melalui reproduksi disertai operasi genetik berupa persilangan dan mutasi
 - 3.3. Menghitung nilai kelayakan generasi baru
 - 3.4. Mengganti individu-individu yang memiliki nilai kelayakan yang paling rendah dengan generasi baru yang nilai kelayakannya lebih baik
4. *Until* terminasi. (terminasi dilakukan jika ditemukan solusi optimal, tidak ditemukan solusi yang lebih baik, atau waktu yang dialokasikan telah digunakan sepenuhnya)

Pada algoritma genetik terdapat istilah fungsi kelayakan. Fungsi kelayakan merupakan fungsi yang menentukan nilai kelayakan suatu individu untuk dijadikan kandidat pencarian solusi. Penentuan fungsi kelayakan ini bergantung pada permasalahan yang dihadapi. Dalam algoritma genetik, penentuan nilai kelayakan suatu individu dilakukan secara konkuren. Hal ini dilakukan untuk mengefektifkan waktu pada proses pembobotan setiap individu. Selain itu, proses konkuren tersebut juga merupakan pemodelan suatu populasi. Yaitu adanya sejumlah peristiwa reproduksi pada saat yang bersamaan. Adanya sifat konkuren pada operasi pembobotan menjadikan algoritma genetik menggunakan waktu yang relatif sedikit dalam mencari solusi.

2.1 Kromosom individu

Pada persoalan TSP kali ini, setiap individu merepresentasikan jalur yang mungkin dilalui. Pada kasus ini, jalur tersebut disimpan dalam sebuah tabel. Pada tabel tersebut, disimpan kota yang akan dituju dan kota yang telah dituju. Hal ini dimaksudkan agar setiap kota dapat dikunjungi dan tidak ada kota yang tidak dikunjungi. Berikut ialah contoh instansiasi masalah:

Kromosom t1 = ABCDEFGH

tabel 3. Instansiasi penyimpanan kromosom individu

Kota	Kota tujuan	Kota yang telah dituju
A	B	H
B	C	A
C	D	B
D	E	C
E	F	D
F	G	E
G	H	F
H	A	G

Dengan cara ini, kromosom menyimpan daftar kota menyerupai cincin. Sehingga, solusi berupa sirkuit Hamilton dapat direpresentasikan dengan baik oleh kromosom suatu individu.

2.2 Rekombinasi

Operasi persilangan dalam TSP akan lebih rumit. Hal ini disebabkan kromosom anak harus tetap valid. Yaitu pada kromosom tersebut, setiap kota dikunjungi hanya satu kali. Persilangan akan mengambil setiap jalur yang ada dari kedua orang tua dan menempatkannya pada kedua anak. Lalu, untuk anak pertama, akan dilakukan pengambilan berselang-seling antara mengambil jalur yang muncul pada orang tua 2 kemudian orang tua 1. Sementara itu, untuk anak kedua, dilakukan pengambilan berselang-seling pada orang tua 1 kemudian orang tua 2. Jika dihasilkan anak dengan kromosom yang tidak valid, Maka, gen yang mengakibatkan kromosom tidak valid akan diganti dengan gen yang valid secara acak.

Selain persilangan, digunakan juga proses mutasi. Proses ini dilakukan agar diperoleh individu yang memiliki gen yang unik. Sehingga, anak yang dihasilkan melalui proses persilangan memiliki variasi yang tinggi.

2.3 Fungsi kelayakan

Pada persoalan TSP, algoritma genetik menggunakan fungsi kelayakan sebagai fungsi yang menghasilkan nilai jarak tempuh suatu tur. Hal ini disebabkan fungsi kelayakan akan digunakan untuk mengeliminasi sejumlah individu yang bukan kandidat solusi. Dengan demikian,

individu yang memiliki nilai kelayakan yang paling besar akan digantikan oleh individu hasil perkawinan yang nilai kelayakannya lebih kecil.

Dalam menghitung jarak antara dua kota, digunakan rumus jarak antara dua buah titik. Definisi rumus tersebut secara matematis ialah sebagai berikut

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

3. PENGUJIAN

Pada kali ini, akan dilakukan pengujian terhadap implementasi algoritma genetik dalam menyelesaikan TSP. Parameter yang akan digunakan ialah jumlah kota, posisi kota, dan waktu penyelesaian. Pada pengujian ini digunakan spesifikasi komputer sebagai berikut

tabel 4. Spesifikasi komputer pengujian

Processor	Intel Pentium 4 Hyperthread 2,8 GHz
RAM	1 GB DDR1 PC3200
OS	WinXP Profesional Service Pack 2
Program simulasi TSP	Travelling Salesman Problem (TSP) Genetic Algorithm (GA) by Michael LaLena

Pengujian dilakukan dua kali untuk setiap kasus. Pengujian pertama dilakukan untuk memperoleh solusi optimal atau solusi yang dianggap optimal. Pengujian kedua dilakukan untuk mendapatkan waktu pencarian solusi. Pencarian waktu solusi dilakukan dengan menunggu program 10 detik setelah diperoleh solusi yang sama dengan pengujian pertama. Dengan demikian, waktu penyelesaian didefinisikan sebagai berikut:

$$t = t_p - 10 \quad (2)$$

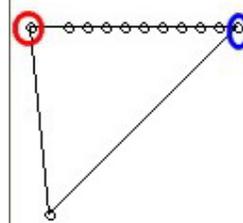
t ialah waktu penyelesaian

t_p ialah waktu percobaan

tabel 5. Pengujian pertama algoritma genetik

Kasus	1	
Jumlah kota	12	
Posisi kota		
Nomor kota	Absis (posisi pixel)	Ordinat (posisi pixel)
1	10	100
2	20	200
3	30	100
4	40	100
5	50	100
6	60	100
7	70	100

8	80	100
9	90	100
10	100	100
11	110	100
12	120	100
Perkiraan waktu penyelesaian	t < 1 detik	

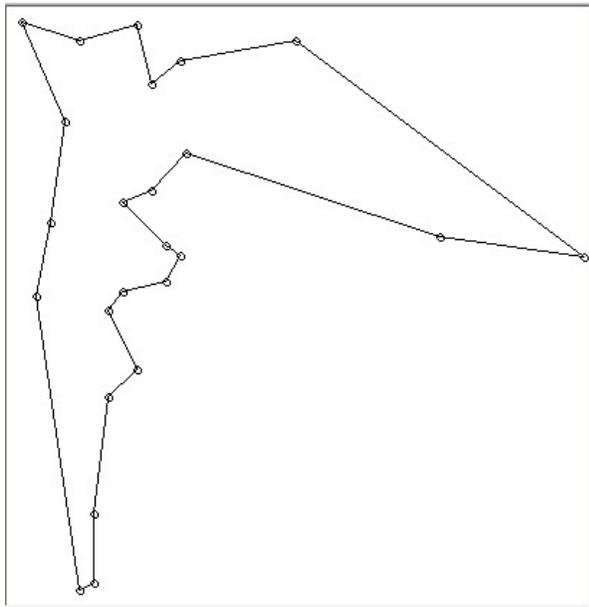


gambar 1. Hasil pengujian pertama. lingkaran merah menunjukkan posisi kota nomor 1 dan lingkaran biru menunjukkan posisi kota nomor 12

tabel 6. Pengujian kedua algoritma genetik

Kasus	2	
Jumlah kota	24	
Posisi kota		
Nomor kota	Absis (posisi pixel)	Ordinat (posisi pixel)
1	10	10
2	20	200
3	30	149
4	40	79
5	50	23
6	60	399
7	70	270
8	80	197
9	90	12
10	100	127
11	110	165
12	120	37
13	100	53
14	200	23
15	300	159
16	400	173
17	50	404
18	60	351
19	70	210
20	80	135
21	90	251
22	120	172
23	110	190
24	124	101
Perkiraan waktu	1.9	

penyelesaian

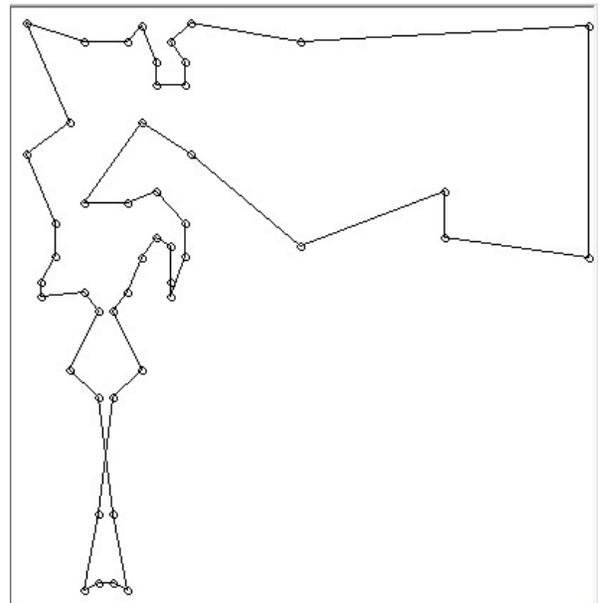


gambar 2. Hasil pengujian kedua.

tabel 7. Pengujian ketiga algoritma genetik

Kasus	3	
Jumlah kota	48	
Posisi kota		
Nomor kota	Absis (posisi pixel)	Ordinat (posisi pixel)
1	10	10
2	20	200
3	30	149
4	40	79
5	50	23
6	60	399
7	70	270
8	80	197
9	90	12
10	100	127
11	110	165
12	120	37
13	100	53
14	200	23
15	300	159
16	400	173
17	50	404
18	60	351
19	70	210
20	80	135
21	90	251

22	120	172
23	110	190
24	124	101
25	10	101
26	20	190
27	30	172
28	40	251
29	50	135
30	60	210
31	70	351
32	80	404
33	90	173
34	100	159
35	110	23
36	120	53
37	100	37
38	200	165
39	300	127
40	400	12
41	50	197
42	60	270
43	70	399
44	80	23
45	90	79
46	120	149
47	110	200
48	124	10
Perkiraan waktu penyelesaian	15.77 detik	

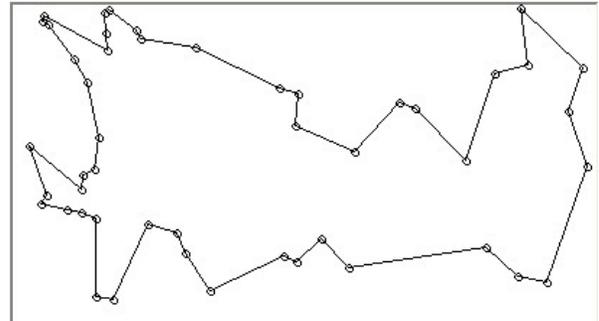


gambar 3. Hasil pengujian ketiga

tabel 8. Pengujian keempat algoritma genetik

Kasus	4	
Jumlah kota	48	
Posisi kota		
Nomor kota	Absis (posisi pixel)	Ordinat (posisi pixel)
1	351	2
2	66	3
3	63	5
4	21	7
5	20	11
6	24	13
7	85	17
8	64	19
9	88	23
10	126	29
11	65	31
12	42	37
13	356	41
14	394	43
15	333	47
16	51	53
17	184	57
18	197	61
19	267	67
20	278	71
21	384	73
22	195	83
23	59	91
24	11	97
25	236	101
26	313	107
27	397	111
28	56	113
29	48	117
30	47	127
31	23	131
32	19	137
33	37	141
34	47	143
35	57	147
36	93	151
37	113	157
38	213	161
39	327	167
40	119	171
41	187	173
42	196	177
43	232	181
44	349	187
45	369	191
46	136	197

47	57	201
48	69	203
Perkiraan waktu penyelesaian	13.80 detik	



gambar 4. Hasil pengujian keempat

4. KESIMPULAN

Dari empat buah percobaan tersebut, jumlah kota menentukan nilai perkiraan waktu pencarian solusi. Akan tetapi, untuk jumlah kota yang sama, perkiraan waktu pencarian solusi tidak mutlak sama. Hal tersebut dibedakan oleh jarak antara suatu kota dengan kota yang lain serta posisi setiap kota.

Algoritma genetik mengandalkan konkurensi yang dimilikinya. Hal tersebut disebabkan konkurensi dapat menghemat waktu pengerjaan jika proses komputasi dilakukan oleh lebih dari satu buah mesin komputasi (prosesor *multicore*). Dalam mencari solusi, algoritma genetik mengembangkan prinsip *greedy*. Prinsip *greedy* digunakan pada saat pengambilan individu-individu yang memiliki nilai kelayakan yang baik untuk kemudian dilakukan reproduksi dari individu-individu tersebut.

REFERENSI

- [1] Darrell Whitley, "A Genetic Algorithm Tutorial".
- [2] www.lalena.com, tanggal akses 13 Mei 2008.
- [3] Tom Holland, "Tackling the Travelling Salesman Problem with a Genetic Algorithm".
- [4] Buthainah Fahren Al-Dulaimi, and Hamza A. Ali, "Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA)", PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, 28, 2008, 296-302.