

PENYELESAIAN *TRAVELING SALESMAN PROBLEM* DENGAN METODE ALGORITMA GENETIKA (*UNIFORM CROSSOVER*)

Ekra Sanggala¹⁾, Shylvia Khamdani²⁾

¹D4 Logistik Bisnis, Politeknik Pos Indonesia

email: ekrasanggala@mail.ru

²D4 logistik Bisnis, Politeknik Pos Indonesia

email: shylviakhamdani06@gmail.com

Abstrak

Rute yang optimal dibutuhkan untuk menunjang aktivitas kita sehari-hari agar lebih efektif dan efisien. Untuk mendapatkan rute yang optimal tersebut dapat digunakan suatu metode yaitu algoritma genetika. Algoritma Genetik merupakan proses pencarian dengan pendekatan heuristik yang dapat digunakan pada berbagai macam permasalahan optimasi. Pada proses algoritma genetik ini terdiri dari beberapa langkah, yaitu menentukan parameter, inisialisasi, crossover, mutation, genotype-phenotype mapping, fitness, selection, dan termination. Salah satu permasalahan optimasi yang dapat diselesaikan dengan metode algoritma genetika ini ada TSP (Traveling Salesman Problem). Pada kasus yang akan diselesaikan menggunakan data 10 kota dan dalam algoritma digunakan uniform crossover. Dalam penyelesaian kasus TSP dengan algoritma genetik ini dapat diselesaikan dengan bantuan aplikasi microsoft excel untuk mempermudah pengerjaan.

Kata Kunci: *algoritma genetika, TSP (Traveling Salesman Problem), Optimasi, Uniform Crossover*

1. PENDAHULUAN

Transportasi merupakan aktivitas yang setiap hari terjadi disekitar kita. Untuk menunjang aktivitas yang efektif dan efisien tentu diperlukannya transportasi yang baik, tentu itu dapat terjadi dengan rute yang optimal untuk dilalui. Dalam penentuan rute yang optimal tersebut dapat ditentukan dengan sebuah metode salah satunya yaitu algoritma genetika.

Algoritma Genetika merupakan proses pencarian dengan pendekatan heuristik yang dapat digunakan pada berbagai macam permasalahan optimasi [2]. Proses kerja Algoritma genetika menirukan proses evolusi yang berdasarkan pada mekanisme seleksi alam dan genetik alam. Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi.

solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness* [4]. Secara umum algoritma genetik terdiri dari beberapa proses yaitu [2]:

1. *Setting Parameters*
2. *initialization*

3. *crossover*
4. *mutation*
5. *Ghenotype-phenotype mapping,*
6. *fitness*
7. *selection,* dan
8. *termination.*

Salah satu proses penting dalam algoritma genetika adalah *crossover*. Terdapat beberapa jenis *crossover*, seperti *one point crossover*, *uniform crossover*, dan *two points crossover*. Pada pembahasan kali ini akan digunakan *uniform crossover*. *Uniform crossover* dikenal sebagai salah satu *crossover* yang paling efektif untuk proses algoritma genetika [5]. Pada *uniform crossover* ini dilakukan recombination secara selektif. Didalam proses *uniform crossover* menjelaskan bahwa 0 merupakan kemungkinan gen pada anak diturunkan dari gen orang tua pertama, dan 1 merupakan kemungkinan gen pada anak diturunkan dari orang tua kedua [6].

Metode algoritma genetika dapat digunakan untuk berbagai permasalahan optimasi karena termasuk kedalam metode metaheuristik. Salah satu permasalahan yang dapat diselesaikan deng metode algoritma genetik adalah TSP (*Traveling Salaseman Problem*). TSP merupakan suatu permasalahan dalam mencari rute terpendek, dimana

pada TSP ini mengunjungi tepat satu kali suatu kota dan kembali ke kota awal [1].

Dalam jurnal kali ini penulis akan melakukan pembahasan terhadap bagaimana penerapan metode algoritma genetika (*uniform crossover*) terhadap permasalahan TSP (*Traveling Salesman Problem*). Dalam kasus TSP ini penulis akan menggunakan 10 kota sebagai permasalahan. Dalam penyelesaian kasus TSP dengan metode algoritma genetika ini penulis menggunakan *Microsoft Excel* untuk mempermudah proses penyelesaian.

2. METODE PENELITIAN

Dalam penyelesaian kasus ini terdapat beberapa langkah yang dilakukan penulis, langkah ini sesuai dengan langkah pada metode algoritma genetika, yaitu:

1. *Setting Parameters*
2. *initialization*
3. *crossover*
4. *mutation*
5. *phenotype mapping*
6. *fitness*
7. *selection*
8. *termination*.

A. *Setting Parameters*

Langkah pertama yang dilakukan adalah menentukan parameter. Parameter yang dimaksud adalah:

1) *Crossover probability*

Crossover Probability menunjukkan seberapa besar kemungkinan persilangan atau perkawinan antar kedua orang tua yang terpilih untuk dapat menemukan solusi terbaru [3]. Nilai untuk *crossover probability* ditentukan secara *random*. Rentang nilai *crossover probability* adalah dari angka 0 sampai dengan 1.

2) *Mutation Probability*

Mutation probability adalah peluang terjadinya mutasi pada sebuah individu [3]. Nilai untuk *mutation probability* ditentukan secara *random*. Rentang nilai *mutation probability* adalah dari angka 0 sampai dengan 1.

3) *Generation*

Parameter generation menunjukkan berapa banyak generasi yang akan ada pada proses algoritma genetika [2]. Batasan untuk nilai parameter ini adalah bilangan positif.

4) *Individual*

Parameter *individual* menunjukkan berapa banyak individu yang akan hidup pada sebuah generasi [2]. Batasan untuk nilai parameter ini adalah bilangan positif.

5) *Survivor*.

Parameter *survivor* menunjukkan berapa banyak individu terbaik pada setiap generasi yang akan hidup pada generasi selanjutnya [2].

B. *Initialization*

Inisialisasi merupakan tahap pembentukan kandidat solusi yang disebut dengan kromosom. Inisialisasi ini dilakukan secara acak (*random*). Sebelum dilakukan proses inisialisasi harus diketahui bentuk representasi dari kromosom. Kromosom dapat berupa *string biner*, *integer*, *floating point*, abjad, dll.

C. *Crossover*

Crossover merupakan operator yang memungkinkan kombinasi dua material genetik dari dua solusi atau lebih. Sebelum adanya proses *crossover*, terlebih dahulu terdapat proses seleksi orang tua yang biasanya terdiri dari dua orang tua. Setelah seleksi pasangan, langkah selanjutnya adalah perkawinan. Berdasarkan perspektif biologi, pasangan dari spesies yang sama mengkombinasikan material genetik mereka dan menurunkan kepada keturunan baru [2].

Terdapat berbagai macam *crossover* seperti *one point crossover*, *uniform crossover*, dan *two points crossover*.

D. *Mutation*

Pada proses *mutation* terjadi perubahan suatu solusi dengan cara mengubah susunan gen individu [2]. Perubahan susunan gen pada proses mutasi ini terjadi secara acak [2]. Dari proses mutasi ini akan diketahui seberapa kuat suatu individu menerima gangguan.

E. *Genotype-phenotype Mapping*

Genotype merupakan susunan gen pada individu yang belum menjadi solusi, dengan kata lain *genotype* ini harus diterjemahkan kedalam bentuk *Phenotype* agar menjadi sebuah solusi [2]. Sebagai contoh, dalam penyelesaian TSP optimasi yang berkelanjutan, *genotype* merupakan solusi itu sendiri sehingga tidak perlu diterjemahkan. Tetapi banyak model proses evolusi lain yang menggunakan proses pemetaan ini [2].

F. *Fitness*

Dalam tahap perhitungan nilai *fitness*, *phenotype* dari sebuah solusi harus dievaluasi dalam fungsi *fitness*. Fungsi *fitness* adalah untuk memastikan kualitas dari solusi-solusi yang telah dihasilkan dalam algoritma genetika [2].

G. *Selection*

Untuk dapat menuju solusi yang optimal, keturunan terbaik harus diseleksi untuk menjadi orangtua pada populasi yang baru. Dari keturunan baru yang dihasilkan, yang terbaik akan terpilih agar mencapai kemajuan yang

optimal. Proses seleksi ini berdasarkan dari nilai *fitness* pada suatu populasi [4].

H. Termination

Termination merupakan kondisi dimana proses evolusi utama harus diberhentikan [2]. Berhentinya proses evolusi ini biasanya sudah ditentukan diawal dalam parameter generasi. Pada saat hasil mendekati optimum, kemajuan dari peningkatan fungsi *fitness* akan menurun secara signifikan, dan jika tidak ditemukan proses yang signifikan, proses evolusi dapat diberhentikan [2].

3. HASIL DAN PEMBAHASAN

Pada pembahasan kali ini penulis akan menyelesaikan salah satu kasus TSP dengan menggunakan metode algoritma genetika (*uniform crossover*). Pada kasus TSP ini terdapat 10 kota yang akan dikunjungi, dan masing-masing kota ini memiliki jarak tempuh yang berbeda. Berikut merupakan data 10 kota beserta jarak tempuhnya.

kota	0	1	2	3	4	5	6	7	8	9	10
0	0	5	17	10	8	12	7	8	9	14	8
1	5	0	20	5	6	9	2	3	14	10	7
2	17	20	0	24	16	16	19	23	13	22	26
3	10	5	24	0	9	11	5	2	19	9	7
4	8	6	16	9	0	4	4	8	14	7	13
5	12	9	16	11	4	0	7	10	17	5	16
6	7	2	19	5	4	7	0	4	15	8	9
7	8	3	23	2	8	10	4	0	17	9	6
8	9	14	13	19	14	17	15	17	0	22	17
9	14	10	22	9	7	5	8	9	22	0	16
10	8	7	26	7	13	16	9	6	17	16	0

Gambar 1. Data Jarak dari 10 Kota

A. Menentukan Parameter

Langkah pertama yang dilakukan pada metode algoritma genetika adalah menentukan parameter. Parameter yang digunakan dalam proses algoritma genetika yaitu, *crossover probability*, *mutation probability*, *generation*, *individual* dan *survivor*. Penentuan nilai parameter ini ditentukan secara *random*. Rentang nilai untuk *crossover probability*, *mutation probability* adalah 0 sampai dengan 1, sedangkan untuk *generation*, *individual* dan *survivor* adalah semua bilangan positif. Berikut merupakan parameter yang digunakan pada kasus TSP kali ini.

PARAMETERS	
crossover prob	0,5
mutation prob	0,5
generation	2
individual	5
survivor	1

Gambar 2. Parameter

B. Inisialisasi Rute

Inisialisasi merupakan tahap pembentukan kandidat solusi yang disebut dengan kromosom. Pada tahap inisialisasi ini, dilakukan pembentukan susunan kandidat solusi yang direpresentasikan sebagai susunan kromosom.

Pembentukan susunan gen ini dilakukan sebanyak lima kali, karena dalam satu generasi terdapat 5 individu seperti yang sudah ditetapkan pada parameter individu. Berikut merupakan susunan kromosom pada generasi 0.

generation 0	chromosom											
individu	1	2	3	4	5	6	7	8	9	10	11	12
vladimir	0	2	5	6	3	4	7	8	9	10	1	0
irina	0	2	5	8	3	4	9	10	6	7	1	0
maxim	0	3	5	9	1	7	8	2	10	4	6	0
elena	0	5	4	3	2	1	10	9	8	7	6	0
yuri	0	2	6	4	9	7	10	5	1	3	8	0

Gambar 3. Inisialisasi pada generasi 0

Pada setiap individu diberi nama sebagai identitas. Setiap individu memiliki 12 kromosom, karena sesuai dengan konsep TSP yaitu mengunjungi titik tepat satu kali yang diawali dari titik *start* (nol) dan kembali lagi ke titik *start*.

C. Menentukan nilai Length

Langkah selanjutnya adalah menentukan *length* (jarak) dari masing-masing individu. Cara untuk menghitung jarak adalah dengan menentukan jarak antar kota pada setiap susunan gen yang kemudian akan dijumlahkan secara keseluruhan. Berikut merupakan nilai *length* dari masing-masing individu.

generation 0	chromosom												length
individu	1	2	3	4	5	6	7	8	9	10	11	12	
vladimir	0	2	5	6	3	4	7	8	9	10	1	0	129
irina	0	2	5	8	3	4	9	10	6	7	1	0	122
maxim	0	3	5	9	1	7	8	2	10	4	6	0	119
elena	0	5	4	3	2	1	10	9	8	7	6	0	142
yuri	0	2	6	4	9	7	10	5	1	3	8	0	120

Gambar 4. Nilai length dari masing-masing individu

Pada gambar 4 dapat terlihat bahwa individu yang memiliki nilai *length* (jarak) paling kecil adalah individu Maxim. Ini menjadikan Maxim sebagai *survivor* yang akan hidup pada generasi selanjutnya.

D. Pemilihan Orang Tua

Dalam pemilihan orang tua dapat dilihat dari nilai *fitness* suatu individu, karena jika semakin besar nilai *fitness* maka semakin besar peluang untuk menjadi orangtua. Untuk mendapatkan nilai *fitness* dilakukan beberapa perhitungan seperti, *conversion*, *probability to be parent*, dan *accumulation*. Perhitungan ini mengacu pada nilai *length* masing-masing individu. Berikut merupakan rumus untuk mendapat nilai *conversion*, *probability to be parent*, dan *accumulation* :

- *Conversion* = nilai *length* terbesar - nilai *length* individu
- *Probability to be parent* = *conversion* individu / total *conversion*
- *Accumulation* = prob to be parents n + accumulation n-1

individual	length	conversio	prob. To be	accumulat	rank
vladimir	129	13	0,1666667	0,166667	4
irina	122	20	0,2564103	0,423077	3
maxim	119	23	0,2948718	0,717949	1
elena	142	0	0	0,717949	5
yuri	120	22	0,2820513	1	2
total	78				

Gambar 5. Nilai Fitness

Setelah diketahui nilai *fitness* maka akan dilakukan pemilihan orang tua. Pada pemilihan orang tua dipilih berdasarkan nilai *parents random* yang ditentukan secara acak oleh penulis dari rentang 0 sampai dengan 1. Individu yang terpilih adalah individu dengan nilai *parents random* lebih kecil atau sama dengan nilai *accumulation*. Seperti terlihat pada gambar dibawah untuk *father* 1 yang terpilih Irina karena $0,2 \leq 0,2564103$.

parents	paren	chromosome											individu	
		1	2	3	4	5	6	7	8	9	10	11		12
father 1	0,2	0	2	5	8	3	4	9	10	6	7	1	0	irina
mother 1	0,6	0	3	5	9	1	7	8	2	10	4	6	0	maxim
father 2	0,4	0	2	5	8	3	4	9	10	6	7	1	0	irina
mother 2	1	0	2	6	4	9	7	10	5	1	3	8	0	yuri

Gambar 6. Daftar Orangtua

F. Crossover

Proses selanjutnya adalah *crossover*. *Crossover* yaitu proses persilangan gen antar kedua orangtua sehingga menghasilkan keturunan baru. Untuk menentukan pasangan orangtua yang akan mengalami *crossover* yaitu dengan menentukan nilai *crossover* random terlebih dahulu. Pasangan orangtua yang mengalami *crossover* adalah pasangan yang memiliki nilai *crossover* random lebih kecil atau sama dengan *crossover probability*, dan jika sebaliknya maka tidak akan mengalami *crossover* dan anak yang dihasilkan 100% susunan gen sama dengan orang tua. Pada gambar terlihat bahwa pasangan orangtua pertama yang akan mengalami *crossover* karena nilai *crossover* random $0,5 \leq 0,5$ nilai *crossover probability*.

parents	parents	chromosome												individu	crossover	rand	crossover
		1	2	3	4	5	6	7	8	9	10	11	12				
father 1	0,2	0	2	5	8	3	4	9	10	6	7	1	0	irina			
mother 1	0,6	0	3	5	9	1	7	8	2	10	4	6	0	maxim	0,5	TRUE	
father 2	0,4	0	2	5	8	3	4	9	10	6	7	1	0	irina			
mother 2	1	0	2	6	4	9	7	10	5	1	3	8	0	yuri	0,8	FALSE	

Gambar 7. Proses penentuan crossover

Pada proses *crossover* kali ini menggunakan *uniform crossover*. *Uniform crossover* ini pertama kali diperkenalkan Syswerda, menggunakan rasio pencampuran yang tetap, dalam memilih bit dari salah satu orangtua yang dilakukan secara acak [2]. Seperti pada kasus yang dikerjakan kali ini.

parents	parents	chromosome												individu	crossover	rand	crossover
		1	2	3	4	5	6	7	8	9	10	11	12				
father 1	0,2	0	2	5	8	3	4	9	10	6	7	1	0	irina			
mother 1	0,6	0	3	5	9	1	7	8	2	10	4	6	0	maxim	0,5	TRUE	0 1 1 1 0 1 0 0 1 0
father 2	0,4	0	2	5	8	3	4	9	10	6	7	1	0	irina			
mother 2	1	0	2	6	4	9	7	10	5	1	3	8	0	yuri	0,8	FALSE	

Gambar 8. Posisi crossover

Pada Gambar diatas ditentukan terlebih dahulu posisi *crossover* untuk pasangan pertama. Dalam penentuannya

digunakan angka 0 dan 1. Berikut merupakan keterangannya:

- Anak 1 : angka 0 = gen ayah; angka 1 = gen ibu
- Anak 2 : angka 1 = gen ibu; angka 0 = gen ayah

Untuk yang diblok dengan warna oren merupakan gen untuk anak 1, dan yang berwarna biru gen untuk anak 2.

Berikut merupakan hasil dari *crossover* yang sudah diberi nama setiap individunya.

name	chromosome												name
	1	2	3	4	5	6	7	8	9	10	11	12	
Child 1	0	2	5	9	1	4	8	10	6	4	1	0	shylvia
Child 2	0	3	5	8	3	7	9	2	10	7	6	0	steven
child 1	0	2	5	8	3	4	9	10	6	7	1	0	severus
child 2	0	2	6	4	9	7	10	5	1	3	8	0	sonia

Gambar 9. Hasil Crossover

G. Pelengkapan Gen

Susunan gen yang diturunkan kepada anak masih belum lengkap, karena terdapat gen yang hilang (*lost gen*). Gen yang hilang ini harus dilengkapi agar susunan gen nya sempurna.

name	chromosome												lost gen
	1	2	3	4	5	6	7	8	9	10	11	12	
shylvia	0	2	5	9	1	4	8	10	6	4	1	0	0 3 dan 7
steven	0	3	5	8	3	7	9	2	10	7	6	0	0 1 dan 4
severus	0	2	5	8	3	4	9	10	6	7	1	0	
sonia	0	2	6	4	9	7	10	5	1	3	8	0	

Gambar 10. Lost Gen

Terlihat pada gambar 10, bahwa individu Shylvia dan Steven terdapat *lost gen*. Berarti untuk gen 1 dan 4 pada Shylvia akan ditukar dengan gen 3 dan 7 pada Steven agar susunan gen dari masing-masing individu lengkap. Berikut merupakan hasil pelengkapan gen dari individu Shylvia dan Steven.

name	chromosome												lost gen
	1	2	3	4	5	6	7	8	9	10	11	12	
shylvia	0	2	5	9	3	7	8	10	6	4	1	0	
steven	0	3	5	8	3	7	9	2	10	7	6	0	
severus	0	2	5	8	3	4	9	10	6	7	1	0	
sonia	0	2	6	4	9	7	10	5	1	3	8	0	

Gambar 11. Hasil Pelengkapan Gen

H. Mutasi

Mutasi merupakan perubahan susunan gen suatu individu, biasanya mutasi terjadi pada individu baru dari hasil *crossover*. Individu yang akan mengalami mutasi adalah individu yang memiliki nilai probabilitas mutasi lebih kecil atau sama dengan *mutation probability*. Nilai probabilitas *mutation* didapatkan dari hasil acak (*random*) oleh program ataupun manual. Setelah diketahui individu yang akan mengalami mutasi, maka ditentukan *position mutation*. *Position mutation* merupakan posisi perubahan gen, ini ditetapkan secara acak (*random*).

name	chromosome												prob. Mutasi	mutation	pos. Mutation
	1	2	3	4	5	6	7	8	9	10	11	12			
shylvia	0	2	5	9	3	7	8	10	6	4	1	0	0,2	TRUE	6 & 10
steven	0	3	5	8	3	7	9	2	10	7	6	0	0,6	FALSE	
severus	0	2	5	8	3	4	9	10	6	7	1	0	0,5	TRUE	3 & 9
sonia	0	2	6	4	9	7	10	5	1	3	8	0	0	FALSE	

Gambar 12. Mutasi

Dari gambar diatas diketahui bahwa individu yang mengalami mutasi adalah Shylvia dan Severus, karena

nilai probabilitas mutasi $0,2 \leq 0,5$ nilai *mutation probability*. Kemudian ditentukan posisi mutasi dari masing-masing individu yaitu seperti pada gambar. Posisi tersebut merupakan posisi gen yang akan ditukar posisinya. Pada individu Shylvia posisi mutasinya adalah 6 dan 10, yang berarti pada gen ke 6 yaitu gen 7 akan ditukar posisi dengan gen ke 10 yaitu gen 4. Berikut merupakan hasil dari mutasi.

		chromosome											
name		1	2	3	4	5	6	7	8	9	10	11	12
shylvia		0	2	5	9	3	4	8	10	6	7	1	0
steven		0	1	5	8	3	4	9	2	10	7	6	0
severus		0	2	6	8	3	4	9	10	5	7	1	0
sonia		0	2	6	4	9	7	10	5	1	3	8	0

Gambar 13. Hasil Mutasi

I. Selection

Seleksi dilakukan untuk mengetahui individu mana saja yang akan bertahan dan hidup pada generasi selanjutnya. Pada generasi 0 individu yang bertahan dan akan terpilih untuk hidup di generasi satu adalah keempat anak dan satu *survivor*. Keempat anak adalah Shylvia, Steven, Severus, dan Sonia, untuk 1 *survivor* adalah Maxim. Berikut merupakan individu yang hidup di generasi selanjutnya.

GENERASI 1		chromosome												length
name		1	2	3	4	5	6	7	8	9	10	11	12	
maxim		0	3	5	9	1	7	8	2	10	4	6	0	119
shylvia		0	2	5	8	3	4	9	7	10	1	6	0	116
steven		0	3	5	9	4	10	8	2	6	7	1	0	107
severus		0	2	6	8	3	4	9	10	5	7	1	0	136
sonia		0	2	6	4	9	7	10	5	1	3	8	0	120

Gambar 14. Individu pada Generasi 1

J. Termination

Pada kasus yang dikerjakan, proses algoritma akan berhenti pada generasi ke 2. Berikut merupakan hasil akhir dari proses algoritma.

		chromosome												length
name		1	2	3	4	5	6	7	8	9	10	11	12	
isep		0	2	5	9	3	7	8	10	6	4	1	0	107
toni		0	2	5	1	3	4	8	10	6	4	9	0	121
jus		0	7	5	9	3	10	8	2	6	4	1	0	103
jeni		0	6	5	4	9	8	7	2	10	1	3	0	147

Gambar 15. Hasil Akhir Generasi 2

Dari gambar 15 dapat terlihat bahwa terdapat perubahan terhadap nilai jarak. Dimana nilai jarak terkecil yaitu 103 yang dimiliki individu Uus. Ini menunjukkan bahwa penyelesaian TSP dengan metode algoritma genetika menghasilkan nilai jarak yang lebih baik.

4. KESIMPULAN

Dari hasil penelitian ini dapat diketahui bahwa kasus TSP (*Traveling Salesman Problem*) dapat diselesaikan dengan mudah dan mendapatkan hasil yang baik menggunakan metode algoritma genetika yang dipadukan dengan *uniform crossover*. Dapat terlihat pada gambar 15 bahwa anak yang dihasilkan di generasi 2 memiliki nilai legth yang lebih kecil dari individu sebelumnya. Metode algoritma genetika juga akan cocok jika digunakan di

perusahaan dalam menyelesaikan kasus pendistribusian yang lain.

5. REFERENSI

[1] L. Hoffman, Karia. dkk. *Traveling Salesman Problem*. USA : Kluwer Academic Publishers. . 2001
 [2] Kramer,Oliver. *Genetic Algorithm Essentials*. Jerman: Springer. 2017.
 [3] Altoria, Mavida. A. 2007. *Implementasi Algoritma Genetik Untuk Masalah Penjadwalan Job-Shop*. Yogyakarta : Universitas Sanata Dharma.
 [4] Chandra W, Julian, dan Rofiyandi Aghitsni. *Sistem Informasi Penjadwalan Menggunakan Algoritma Genetika Pada Program Studi Sastra InggrisFakultas Sastra UNIKOM*. 2012. Bandung: UNIKOM
 [5] Semenkin, Eugene, dan Maria Semenkina. 2018. *Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator*. Rusia: Siberian State Aerospace University.
 [6] M. Spears, William, dan Kenneth A. De Jong. 1995. *An Analysis of Multi-point Crossover*. USA: George Mason University.