

PENGARUH PENGATURAN INDIVIDU PROSES CROSSOVER DAN MUTASI ALGORITME GENETIKA PADA KASUS TRAVELLING SALESMAN PROBLEM

Sean Coonery Sumarta

Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar

Alamat e-mail: sean.c.sumarta@gmail.com

ABSTRACT

One disadvantage in the genetic algorithm is the number of iterations in the search for the optimum solution. This caused long computation time. This study aimed to obtain the best method to determine the individual parent crossover operator and mutation Genetic Algorithm in the case of the Traveling Salesman Problem (TSP). the results obtained in this research with a set of individuals in the process of crossover and mutation can reduce the number of iterations. So as computing time can be even faster.

Keywords: Travelling Salesmen Problem, Genetic Algorithm, Crossover, Mutattion.

1. PENDAHULUAN

Salah satu permasalahan dalam bidang transportasi darat adalah mencari suatu rute perjalanan terpendek yang dapat di tempuh dari titik awal keberangkatan menuju titik akhir (tujuan), dengan harapan biaya perjalanan yang dikeluarkan dan waktu tempuh perjalanan seminimum mungkin. Masalah seperti ini dikategorikan dalam suatu permasalahan kombinatorial yang lebih dikenal dengan *Traveling Salesman Problem* (TSP). Terdapat beberapa pendekatan yang dapat digunakan dalam mencari solusi optimum dalam menjawab masalah TSP, diantaranya Algoritma Genetika, Pendekatan secara Stokastik, Neural Network dan pemrograman Linier [1].

Algoritma Genetika adalah algoritma optimasi modern berdasarkan evolusi biologis, yang meminjam hukum evolusi biologis yaitu untuk mencapai kelangsungan hidup yang paling cocok melalui pembiakan, warisan, diferensiasi dan persaingan, untuk mendekati solusi optimal secara bertahap. Algoritma Genetika adalah salah satu algoritma matang untuk memecahkan masalah *Nondeterministic Polynomial* (NP) saat ini, yang memiliki aplikasi luas dalam teori optimasi yang modern[2]. Algoritma ini memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya.

“Hanya individu-individu yang kuat yang mampu bertahan”. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Proses perkembangbiakan ini didasarkan pada analogi struktur genetik dan perilaku kromosom dalam populasi individu[3].

Algoritma genetica menggunakan teknik dan proses yang terinspirasi dari evolusi biologi untuk memecahkan masalah optimasi yang kompleks, terdapat beberapa seleksi diantaranya *natural selection, crossover and mutation* diterapkan untuk mendapatkan nilai baru dan menemukan solusi yang optimal. Algoritma genetica biasa digunakan untuk menyelesaikan berbagai permasalahan kompleks pada bidang fisika, biologi, ekonomi, sosiologi.

Beberapa penelitian yang dilakukan sebelumnya mengenai operator *crossover* dan mutasi cukup banyak diantaranya adalah referensi [4] yang melakukan evaluasi terhadap pasangan *crossover* dan mutasi yang memiliki nilai fitness yang terbaik serta penelitian yang dilakukan dalam referensi [5] yang menetapkan operator terbaik dalam tiap-tiap proses pada algoritma genetica dengan memasang tiap-tiap metode pada masing-masing operator algoritma genetica dan mengevaluasinya.

Algoritma Genetika merupakan salah satu metode efektif untuk menyelesaikan *Traveling Salesman Problem*. Dalam prosesnya, Algoritma Genetika

menghasilkan biaya evaluasi yang cukup besar. Biaya evaluasi yang dimaksudkan adalah proses komputasi yang lama. Proses komputasi yang lama disebabkan oleh proses iterasi yang panjang dan tahap-tahap pada algoritma genetika dalam mencari solusi optimal[6].

Salah satu penyebab proses iterasi yang panjang adalah ruang populasi dari individu *parent* yang besar. Dengan ruang populasi *parent* yang besar maka jumlah kemungkinan *parent* akan semakin banyak. Dampak dari kemungkinan *parent* yang besar menyebabkan anak hasil perkawinan akan sulit menuju perbaikan individu. Hal ini menyebabkan iterasi perhitungan untuk mendapatkan individu dengan *fitness* terbaik akan semakin panjang sehingga waktu komputasi akan semakin lama.

Penelitian ini fokus mendapatkan metode yang tepat dalam memilih individu *parent* pada operator *Crossover* dan Mutasi Algoritma Genetika pada kasus *Traveling Salesman Problem*.

2. TINJAUAN PUSTAKA

2.1 Traveling Salesman Problem

Permasalahan matematika tentang TSP dikemukakan pada tahun 1800 oleh matematikawan Irlandia William Rowan Hamilton dan matematikawan Inggris Thomas Penyngton. Diskusi mengenai awal studi dari Hamilton dan Kirkman dapat ditemukan di *Graph Theory* tahun 1736-1936 oleh N. L. Biggs, E. K.L. Loyd dan R. J. Wilson di Oxford, tahun 1976. [7]. Bentuk umum dari TSP pertama dipelajari oleh para matematikawan mulai tahun 1930. Diawali oleh Karl Menger di Vienna dan Harvard. Setelah itu permasalahan TSP dipublikasikan oleh Hassler dan Merrill Flood di Princeton. Penelitian secara detail dari hubungan Menger dan Whitney [8].

TSP merupakan suatu permasalahan optimasi dengan tujuan menemukan jalur perjalanan dengan biaya evaluasi minimum. Biaya evaluasi yang dimaksud dapat berupa jarak, waktu, bahan bakar, kenyamanan dan sebagainya. Dalam TSP, *salesmen* harus mengunjungi setiap kota yang ada tepat satu kali dan kembali ke kota asal.

Pada kasus TSP terdapat 2 jenis kasus TSP yaitu TSP Asimetris dan TSP Simetris.

Perbedaan 2 jenis kasus TSP dapat dijelaskan sebagai berikut :

1. TSP Asimetris

Pada TSP Asimetris, biaya dari *node* 1 ke *node* 2 tidak sama dengan biaya *node* 2 ke *node* 1 yang dinyatakan dalam Persamaan 2.1 sebagai berikut:

$$n_{12} \neq n_{21} \quad (1)$$

Jumlah jalur yang mungkin adalah permutasi dari jumlah *node* dibagi dengan jumlah *node*. Hal ini dapat dipahami secara siklus, sebuah jalur dengan urutan $n_{123} = n_{231} = n_{312}$, tetapi jalur $n_{123} \neq n_{321}$. Jadi apabila terdapat 10 *node*, maka jalur yang mungkin untuk TSP Asimetris dapat dihitung dengan menggunakan Persamaan 2.2:

$$P_{10}^{10} = \frac{10!}{10} = 362.880 \text{ jalur} \quad (2)$$

2. TSP Simetris

Pada TSP Simetris, biaya dari *node* 1 ke *node* 2 sama dengan biaya dari *node* 2 ke *node* 1 yang dinyatakan dalam Persamaan 2.3.

$$n_{12} = n_{21} \quad (3)$$

Jumlah jalur yang mungkin adalah permutasi dari jumlah *node* dibagi dengan 2 kali jumlah *node*. Hal ini dapat dipahami secara siklus, sebuah jalur dengan urutan $n_{123} = n_{231} = n_{312} = n_{321}$. Jadi apabila terdapat 10 *node*, maka jalur yang mungkin untuk TSP Simetris dapat dihitung dengan menggunakan Persamaan 2.4 sebagai berikut:

$$p_{10}^{10} = \frac{10!}{2 \times 10} = 181.440 \text{ jalur} \quad (4)$$

Pada TSP hasil yang ingin didapatkan adalah menemukan jalur terpendek yang saling terhubung dari *n* kota. Tiap kota hanya boleh dikunjungi sekali. Jarak antar kota *i* dan *j* dihitung dengan Persamaan *Euclidean*. Persamaan *Euclidean* dalam bentuk matematika dapat dilihat pada Persamaan 2.5 sebagai berikut:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

Dengan:

d_{ij} = jarak kota i dengan kota j
 x_i, y_i = koordinat (x,y) kota i
 x_j, y_j = koordinat (x,y) kota j

TSP dapat direpresentasikan ke dalam permasalahan graf. Tiap kota diwakili oleh *node* dalam koordinat *Cartesian*. Tiap *node* mempunyai koordinat (x,y) dan semua *node* tersebut saling terhubung satu dengan lainnya dengan jarak masing-masing *node* dapat dihitung dari koordinat-koordinat *node* masing-masing. Jika ada *node* yang tidak terhubung, maka *node* tersebut dihubungkan dan diberi nilai jarak yang nantinya bisa digunakan sebagai solusi optimal.

2.2 Algoritma Genetika

Algoritma Genetika (*Genetic Algorithms*) adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Pada dasarnya, evolusi dilakukan karena individu harus menyesuaikan diri dengan perubahan lingkungan yang terjadi seiring perkembangan waktu. Penyesuaian diri dilakukan dengan melakukan perubahan gen melalui proses perkembang-biakan

Terdapat empat kondisi yang sangat mempengaruhi proses evaluasi [9], adalah :

- Kemampuan individu untuk melakukan reproduksi.
- Keberadaan populasi individu yang melakukan reproduksi.
- Keberagaman individu dalam suatu populasi.
- Perbedaan kemampuan untuk bertahan.

Individu yang lebih kuat akan memiliki kemampuan bertahan dan reproduksi yang lebih tinggi dibandingkan dengan individu yang lebih lemah. Pada kurun waktu tertentu, populasi akan lebih banyak memuat individu yang lebih kuat.

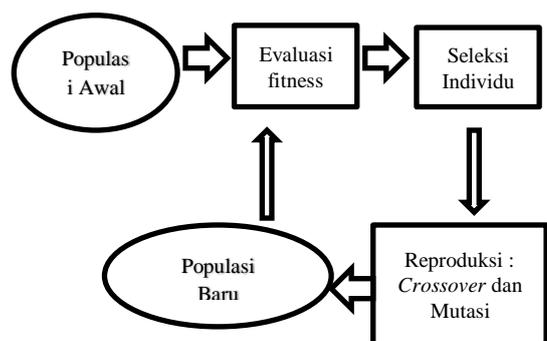
Algoritma genetika pertama kali dikenalkan oleh Jhon Holland dari Universitas Michigan pada tahun 1975[13]. Jhon Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi atau penyesuaian diri, alami maupun buatan, dapat diformasikan dalam terminologi genetika. Sejak pertama kali dikenalkan, Algoritma genetika telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma Genetika banyak digunakan pada masalah praktis yang

berfokus pada pencarian parameter-parameter optimal. Hal ini membuat banyak orang mengira bahwa Algoritma Genetika hanya digunakan untuk masalah optimasi. Namun pada kenyataannya, Algoritma Genetika juga memiliki kinerja yang bagus untuk menyelesaikan masalah-masalah selain optimasi.

Algoritma genetika yang dibuat Holland merupakan sebuah metode untuk memisahkan satu populasi kromosom (terdiri dari bit-bit 1 dan 0) ke baru dengan menggunakan "seleksi alam" dan operator genetik seperti *crossover* dan *mutation*. *Crossover* menukar bagian kecil dari dua kromosom, *mutation* mengganti secara acak nilai gen di beberapa lokasi pada kromosom. Dasar teori inilah yang menjadi dasar kebanyakan program yang menggunakan algoritma genetika pada saat ini.

Algoritma genetika digunakan untuk mendapatkan solusi yang tepat untuk masalah optimasi dari satu variabel atau multi variabel. Berbeda dengan teknik pencarian konvensional, algoritma genetika bermula dari himpunan solusi yang dihasilkan acak, himpunan ini disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan fungsi evaluasi. Setelah beberapa generasi maka algoritma genetika akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal[10].

Siklus dari algoritma genetika menurut [10], dapat dilihat pada gambar 1 sebagai berikut:



Gambar 1. Siklus algoritma Genetika.

3. METODOLOGI PENELITIAN

Setelah melakukan tinjauan dan studi pustaka maka pada penelitian ini metode-metode tiap bagian dari algoritma genetika menggunakan metode sebagai berikut:

3.1 Proses Encode

Pada proses Encode penelitian menggunakan metode *Permutation Encoding*. Setiap kota diberi nomor sesuai dengan jumlah kota yang akan ditempuh, misalnya:

Kota A = 1

Kota B = 2

Kota C = 3

Kota D = 4

Angka 1,2,3 dan 4 merupakan gen yang akan membentuk kromosom dalam hal ini urutan kota misalnya [1 2 3 4] atau [2 4 3 1].

3.2 Proses Perhitungan Fitness

Pada proses perhitungan fitness menggunakan rumus *Euclidean Distance* yaitu menghitung jarak total dari suatu urutan kota dengan model sebagai berikut:

$$\sum_{n=1}^{gen} \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2}$$

Dimana

Gen = Jumlah kota.

X_n = koordinat titik x kota n

Y_n = koordinat titik y kota n

3.3 Proses Seleksi

Proses seleksi menggunakan metode *roulette wheel*. Prosedur dari *roulette wheel* sebagai berikut:

1. Menghitung nilai fitness (f_k) dari tiap individu.
2. Menghitung probabilitas seleksi (P_k) setiap individu dimana

$$P_k = \frac{f_k}{\sum f_k}$$

3. Mengacak sebuah bilangan [0,1].
4. Memilih individu dimana bilangan random tersebut sebagai induk.
5. Lakukan langkah 3 dan 4 sebanyak jumlah individu dalam populasi.

3.4 Proses Crossover.

Pada proses crossover peneliti menggunakan metode heuristic crossover. Dengan langkah sebagai berikut:

1. Memilih 2 induk (bapak dan ibu) sebagai basis kota yang akan dikunjungi.
2. Hasilkan sebuah kota random sebagai titik awal untuk anak.
3. Untuk kota selanjutnya memilih dan mengevaluasi kota pada bapak dan kota pada ibu sesuai urutan kota. Pilih yang mempunyai jarak terdekat dengan kota sebelumnya.
4. Kota yang dipilih menjadi titik awal pada kota selanjutnya.
5. Lakukan langkah 3 untuk kota yang akan dicari sampai anak lengkap.
6. Bila kota yang dipilih sudah dikunjungi hasilkan kota random.

Untuk induk yang akan dikawin-silangkan dalam penelitian ini dilakukan beberapa variasi. Variasi yang dilakukan dengan metode sebagai berikut :

- **Metode 1** : Induk diambil secara acak dari kumpulan populasi induk yang dihasilkan pada proses seleksi.
- **Metode 2** : salah satu induk ditetapkan sebagai induk terbaik dan akan dikawin silangkan dengan individu induk lain.

3.5 Proses Mutasi

Pada proses mutasi penelitian ini menggunakan metode insertion dengan langkah sebagai berikut:

1. Menetapkan sebuah angka probabilitas mutasi P_b .
2. Menghasilkan sebuah angka random [0,1].
3. Jika angka random $< P_b$ maka individu tidak di mutasi dan bila angka random $> P_b$ maka individu akan dimutasi.
4. Hasilkan 2 angka random sebagai titik yang akan dipertukarkan.
5. Pertukaran posisi 2 gen yang berada pada titik tersebut.

Untuk proses mutasi juga dilakukan beberapa variasi untuk proses mutasi. Variasi yang dilakukan dengan metode :

- **Metode 1** : semua anak hasil perkawinan di mutasi
- **Metode 2** : anak yang akan di mutasi adalah individu yang terbaik dalam populasi anak hasil perkawinan.

4. HASIL DAN PEMBAHASAN

Pengujian dari Algoritma genetika dengan pemilihan individu pada proses crossover dan mutasi dengan menggunakan parameter sebagai berikut :

1. Jumlah iterasi = 1000 iterasi
2. Koefisien mutasi = 0,9
3. Data kota = berlin52
4. Ukuran populasi = 1000 individu
5. Jumlah pengujian tiap metode = 10 kali

Pengujian dilakukan dengan mengamati jumlah iterasi untuk mendapat nilai optimal pada tiap metode yang diterapkan pada proses crossover dan mutasi dan menghitung nilai error fitness dengan menggunakan rumus sebagai berikut :

$$Error = \frac{Nilai\ 1 - Nilai\ 2}{Nilai\ 2} \times 100\% \quad (9)$$

Dengan :

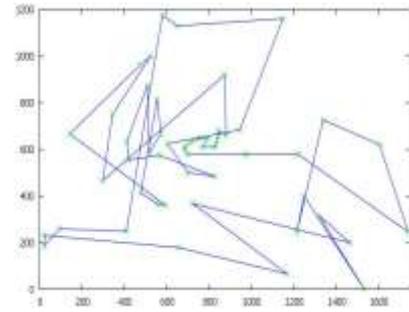
Nilai 1 = Nilai Fitness Terbaik

Nilai 2 = Nilai Optimal

Hasil pengujian algoritma genetika dengan *crossover* menggunakan metode 1 dan mutasi menggunakan metode 1 dapat dilihat pada tabel 1 dan jalur terbaik yang terbentuk dengan 10 kali pengulangan dapat dilihat pada gambar 2. Dari tabel 1 dapat dilihat rata – rata presentase error = 104,7037 dari jarak optimal kasus berlin52 dan pada gambar 2 jalur yang dihasilkan memiliki banyak perpotongan.

Tabel 1. Hasil pengujian dengan crossover menggunakan metode 1 dan mutasi 1

No	Jarak Terbaik	Iterasi Terbaik	Persentase
1	16169.8	478	114.3967
2	15957.6	491	111.5831
3	15031.3	456	99.30125
4	15638.7	942	107.3548
5	15413.1	683	104.3636
6	15937	532	111.31
7	15521	856	105.7942
8	13970.1	622	85.23071
9	15760.4	343	108.9684
10	14988.5	990	98.73376
Rata-rata	15438.75	639.3	104.7037

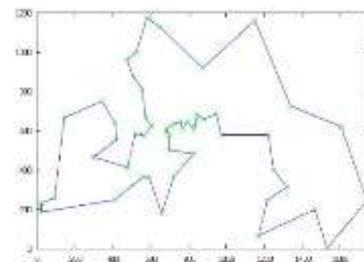


Gambar 2. Jalur terbaik dengan crossover menggunakan metode 1 dan mutasi menggunakan metode 1

Hasil pengujian algoritma genetika dengan *crossover* menggunakan metode 1 dan mutasi menggunakan metode 2 dapat dilihat pada tabel 2 dan jalur terbaik yang terbentuk dengan 10 kali pengulangan dapat dilihat pada gambar 3. Dari tabel 1 dapat dilihat rata – rata presentase error = 9,319 dari jarak optimal kasus berlin52 dan pada gambar 2 jalur yang dihasilkan sudah tidak menghasilkan perpotongan. Namun dalam pengujian ini rata-rata iterasi untuk mendapatkan hasil optimum adalah 791.3 iterasi.

Tabel 2. Hasil pengujian dengan crossover menggunakan metode 1 dan mutasi 2

Percobaan	Jarak Terbaik	Iterasi Terbaik	Persentase
1	8277.52	835	9.75232
2	8208.44	680	8.836383
3	8512.48	984	12.86767
4	8362.41	934	10.87788
5	8473.87	839	12.35574
6	8385.85	946	11.18868
7	8619.15	693	14.28202
8	7939.05	892	5.264519
9	7937.87	556	5.248873
10	7731.99	554	2.519093
Rata - rata	8244.863	791.3	9.319318

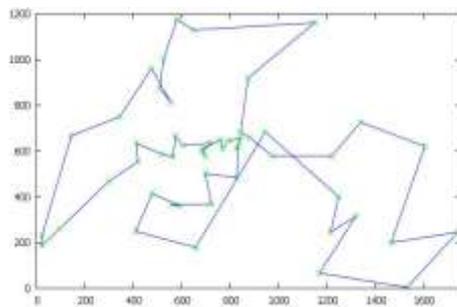


Gambar 3. Jalur terbaik dengan crossover menggunakan metode 1 dan mutasi 2

Hasil pengujian algoritma genetika dengan *crossover* menggunakan metode 2 dan mutasi menggunakan metode 1 dapat dilihat pada tabel 2 dan jalur terbaik yang terbentuk dengan 10 kali pengulangan dapat dilihat pada gambar 4. Dari tabel 3 dapat dilihat rata – rata presentase error = 36,40635 dari jarak optimal kasus berlin52 dan pada gambar 4 jalur yang dihasilkan memiliki perpotongan. Namun dalam pengujian ini rata-rata iterasi untuk mendapatkan hasil optimum turun menjadi 660,7 iterasi.

Tabel 3. Hasil pengujian dengan crossover menggunakan metode 1 dan mutasi menggunakan metode 2

Percobaan	Jarak Terbaik	Iterasi Terbaik	Persentase
1	10994.4	428	45.77566
2	10326.3	761	36.91726
3	9165.95	847	21.53209
4	10519.5	215	39.47892
5	9170.72	955	21.59533
6	10095.2	837	33.85309
7	10423.7	729	38.2087
8	10771.5	938	42.82021
9	11429.1	335	51.53938
10	9981.3	562	32.34288
Rata-rata	10287.77	660.7	36.40635



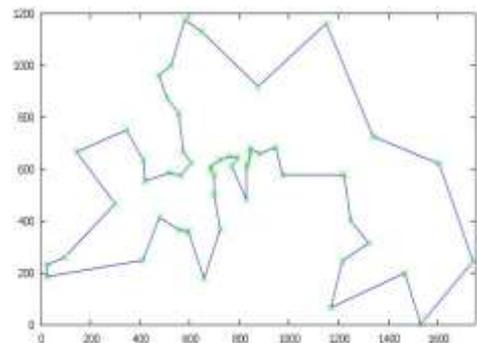
Gambar 4. Jalur terbaik dengan crossover menggunakan metode 2 dan mutasi menggunakan metode 1

Hasil pengujian algoritma genetika dengan *crossover* menggunakan metode 2 dan mutasi menggunakan metode 2 dapat dilihat pada tabel 4 dan jalur terbaik yang terbentuk dengan 10 kali pengulangan dapat dilihat pada gambar 5. Dari tabel 4 dapat dilihat rata – rata presentase error = 6,359692 dari jarak optimal kasus berlin52 dan pada gambar 5 jalur yang dihasilkan tidak

memiliki perpotongan. Dalam pengujian ini rata-rata iterasi untuk mendapatkan hasil optimum turun menjadi 41,5 iterasi.

Tabel 4. Hasil pengujian dengan crossover menggunakan metode 1 dan mutasi menggunakan metode 2

Percobaan	Jarak Terbaik	Iterasi Terbaik	Persentase
1	8543.93	29	13.28467
2	8109.63	32	7.526253
3	8152.46	45	8.094139
4	8046.78	50	6.69292
5	7598.44	57	0.748343
6	8129.49	39	7.789578
7	7910.84	34	4.89048
8	7774.73	39	3.085786
9	7962.67	50	5.577698
10	7987.51	40	5.907054
Rata-rata	8021.648	41.5	6.359692



Gambar 5. Jalur terbaik dengan crossover menggunakan metode 2 dan mutasi menggunakan metode 1

5. KESIMPULAN

Dari hasil dan pembahasan penyelesaian masalah TSP menggunakan Algoritma Genetika dengan pengaturan individu pada proses *crossover* dan mutasi dapat disimpulkan bahwa dengan pengaturan individu pada proses *crossover* dan mutasi dapat menurunkan jumlah iterasi sehingga dapat menghemat waktu komputasi. Dalam penelitian ini, metode pengaturan individu pada proses *crossover* adalah individu terbaik pada proses perhitungan sebelumnya merupakan salah satu induk dan di kawinkan dengan individu yang lain pada populasi induk dan pada proses mutasi metode yang memberikan hasil yang optimum adalah

individu yang mengalami mutasi adalah individu terbaik hasil dari proses crossover.

5. DAFTAR PUSTAKA

- [1] Rizal, J., “Optimasi Pada Traveling Salesman Problem (TSP) dengan Pendekatan Simulasi Annealing”, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Bengkulu, Indonesia, 2007.
- [2] Ma, Fangfang dan Li, 2009, “An algorithm in solving the TSP based on the improved Genetic Algorithm”, Fundamental Department, Shandong University of Science and Technology, Tai'an, China, The 1st International Conference on Information Science and Engineering (ICISE).
- [3] Sutojo, T. dkk., “ Kecerdasan Buatan”, Penerbit Andi, Yogyakarta, 2010.
- [4] Kusum dan Hadush, “Combined Mutation Operators of Genetic Algorithm for the Travelling Salesman problem”, International Journal of Combinatorial Optimization Problems and Informatics, Vol. 2, No.3, Sep-Dec 2011.
- [5] Yingying Yu, dkk., 2011, “A New Design of Genetic Algorithm for Solving TSP”, Transportation Management College Dalian Maritime University Dalian, China, Fourth International Joint Conference on Computational Sciences and Optimization.
- [6] Abramson, D., and Abela, 1992, A Parallel Genetic Algorithm For Solving The School Timetabling Problem, 15 Australian Computer Science Conference (ACS-15), 29 – 31 January, pp – 11
- [7] Bambrick, 1997, Lecturer Timetabling Using Genetic Algorithm, Thesis, University of Queensland.
- [8] Mahmudy, WF, Marian, RM & Luong, LHS 2014, 'Hybrid genetic algorithms for part type selection and machine loading problems with alternative production plans in flexible manufacturing system', ECTI Transactions on Computer and Information Technology (ECTI-CIT), vol. 8, no. 1, pp. 80-93.
- [9] Rahman Erama dan Retantyo Wardoyo, 2013, Modifikasi Algoritma Genetika Untuk Penyelesaian Permasalahan Penjadwalan Pelajaran Sekolah, Prosiding Seminar Nasional Ilmu Komputer 2013, Indonesia Computer and Instrumentation Support Society (indoCEIS), pp-47
- [10] Riska Hardini Purnamasari , Implementasi Algoritma Genetika Untuk Pencarian Rute Minimum Dalam Travelling Salesman Problem, Universitas Komputer Indonesia, 2009.