

REPRESENTASI JALUR (*PATH*) PADA *TRAVELING SALESMAN PROBLEM* UNTUK MENENTUKAN JARAK TERPENDEK MENGUNAKAN ALGORITMA GENETIKA

Yurika Permanasari, Ravi A. Salim

Jurusan Matematika, UNISBA, Jalan Tamansari No 1, Bandung, 40116, Indonesia
yurika@yahoo.com

Jurusan Matematika, UNISBA, Jalan Tamansari No 1, Bandung, 40116, Indonesia
ravisalim@yahoo.com

Abstrak. Algoritma Genetika merupakan algoritma pencarian yang berdasarkan kepada mekanisme seleksi alam dan genetika alam. Algoritma ini merupakan salah satu alat bantu untuk memudahkan pekerjaan atau menyelesaikan suatu kasus yang dihadapi dengan tidak melibatkan seluruh populasi. Pada kasus *Traveling Salesman Problem*, Algoritma Genetika merepresentasikan jalur kota yang dilalui sebagai kromosom yang terdiri dari gen-gen yang merupakan kota-kota yang harus dilalui. Dengan menentukan peluang crossover dan mutasi setiap kromosom mengalami regenerasi dengan harapan sebagian besar kromosom yang baik dapat diperiksa sebagai calon jalur terbaik. Dengan menggunakan *Order Crossover*(OX) pada Representasi *Path*, kasus *Traveling Salesman Problem* menghasilkan nilai pendekatan optimum yang sama baik untuk data 50% maupun 100% dari total populasi. Ini menunjukkan bahwa dengan menggunakan Algoritma Genetika, sebagian sampel sudah mewakili dari seluruh solusi yang ada.

Kata Kunci : Algoritma Genetika; *Travelling Salesman Problem*; *Order Crossover*

1. Latar Belakang

Travelling Salesman Problem merupakan permasalahan optimasi kombinatorial yang umumnya banyak dijumpai pada aplikasi matematika dan komputer. Permasalahan yang terdapat pada *Travelling Salesman Problem* adalah bagaimana menentukan rute perjalanan yang paling optimum dari satu kota dengan mengunjungi semua kota lainnya. Masing-masing kota hanya dikunjungi satu kali dan harus kembali ke kota asal tersebut.

Banyaknya kemungkinan untuk mencari nilai optimum dipengaruhi oleh banyaknya kota yang harus dilalui maupun kriteria-kriteria yang menjadi asumsi optimasi. Oleh karena itu algoritma optimasi *Traveling Salesman Problem* menjadi rumit dan tak terhingga waktu komputasinya.

Untuk menghindari hal ini, diperkenalkan penggunaan Algoritma Genetika, yaitu salah satu algoritma pencarian yang bekerja berdasarkan mekanisme dari seleksi alam dan genetika.

2. Traveling Salesman Problem (TSP)

Jika diberikan sejumlah kota (tempat) dan biaya perjalanan (*travelling cost*) dari satu kota ke kota lain, *Travelling Salesman Problem* merupakan salah satu masalah optimasi kombinatorial.. Tujuannya adalah bagaimana menemukan rute perjalanan paling murah dari suatu kota dan mengunjungi semua kota lainnya, dimana masing-masing kota dikunjungi hanya satu kali, dan harus kembali ke kota asal tersebut. Dalam masalah ini kombinasi dari semua rute perjalanan yang ada adalah faktorial dari jumlah kota. Biaya perjalanan bisa berupa jarak, waktu, bahan bakar, kenyamanan, dan sebagainya. Ruang pencarian untuk *TSP* adalah kumpulan dari permutasi-permutasi dari n kota. Permutasi dari n kota merupakan perjalanan lengkap (*complete*) dari n kota. Solusi optimal adalah permutasi yang menghasilkan biaya minimum dari suatu perjalanan. Ukuran dari ruang pencarian adalah $n!$.

Representasi jalur (*path*) digunakan sebagai representasi kromosom pada *TSP* untuk merepresentasikan sebuah perjalanan dengan n kota. Representasi ini merupakan representasi

yang paling natural, sebab hanya mendaftarkan urutan kota-kota yang akan dikunjungi. Kota-kota yang harus dikunjungi diwakili oleh bilangan-bilangan 1, 2, 3, ..., n . Sebagai contoh suatu perjalanan :

$$5 - 1 - 7 - 8 - 9 - 4 - 6 - 2 - 3$$

Dapat ditulis secara sederhana seperti di bawah ini :

$$(5\ 1\ 7\ 8\ 9\ 4\ 6\ 2\ 3)$$

3. Algoritma Genetika

Suatu sistem Algoritma Genetika terdiri dari beberapa proses utama yaitu seleksi, *crossover* (pindah silang) dan mutasi. Secara umum langkah-langkah Algoritma Genetika adalah sebagai berikut :

- a. Menentukan ukuran populasi (N) dan bentuk populasi awal secara acak.
- b. Menentukan peluang dari operator genetika (*crossover* dan mutasi).
- c. Melakukan reproduksi sesuai nilai fitness dari setiap string.
- d. Melakukan *crossover* dan mutasi.

Keuntungan dari penggunaan Algoritma Genetika dalam *Travelling Salesman Problem* sangat jelas terlihat pada kemudahan implementasi dan kemampuannya untuk menemukan solusi yang ‘cukup bagus’[1] atau *near optimum* (cukup optimum)[2] dan tidak melibatkan seluruh sampel di dalam proses, sehingga menghemat waktu proses dan menghindari komputasi menjadi tak terhingga.

Algoritma Genetika merupakan suatu algoritma pencarian yang bekerja berdasarkan mekanisme dari seleksi alam dan genetika alam. Algoritma ini dikembangkan oleh John Holland beserta kolega dan mahasiswanya di Universitas Michigan[3].

Pada genetika, kromosom terdiri dari susunan gen-gen. Tiap gen mempunyai nilai atau sifat tertentu. Pada Algoritma Genetika, kromosom bersesuaian dengan *string* yang dibentuk dari beberapa karakter. Setiap karakter ini mempunyai posisi (*locus*) dan mengandung nilai tertentu (*allele*). Satu atau beberapa string akan bergabung membentuk struktur (*genotip*), dan bila struktur tersebut didekodekan akan diperoleh satu titik yang merupakan salah satu alternatif solusi (*fenotip*).

Dalam Algoritma Genetika, suatu populasi terdiri dari string-string yang mempunyai nilai fitness tertentu. Tiap-tiap string dalam satu populasi mewakili satu solusi dalam domain solusi. Nilai fitness tiap string diperoleh dari fungsi obyektif dari permasalahan yang dihadapi. String-string yang mempunyai nilai fitness yang tinggi, biasanya akan bertahan dan berlanjut pada generasi selanjutnya. Sebaliknya string-string dengan nilai fitness yang kecil biasanya tidak dapat bertahan[1]. Pencarian solusi dilakukan secara iterative terhadap suatu populasi untuk menghasilkan populasi baru. Dalam satu siklus iterasi (yang disebut generasi), terdapat proses seleksi dan rekombinasi.

Proses seleksi dilakukan dengan mengevaluasi setiap string dalam populasi berdasarkan nilai *fitness*-nya untuk mendapatkan peringkat dari setiap string. Berdasarkan hasil evaluasi ini, selanjutnya dipilih string-string yang akan mengalami rekombinasi. Ini biasanya dilakukan secara acak terhadap string-string dengan nilai fitness yang lebih baik untuk dipilih menjadi calon anggota populasi yang baru.

Proses rekombinasi meliputi proses genetika untuk memperoleh populasi baru dari string-string yang diperoleh pada tahap seleksi. Anggota populasi baru diperoleh dengan menerapkan operator genetika (*crossover* dan mutasi) secara acak pada string-string yang terpilih dalam

tahap seleksi. Pada tahap rekombinasi ini akan diperoleh string-string baru yang berbeda dengan string-string induk dengan demikian akan diperoleh pencarian yang baru.

Cara mengkodekan permasalahan dalam kromosom merupakan suatu hal yang penting dalam Algoritma Genetika. Makalah ini menggunakan skema pengkodean *Permutasi encoding* yaitu pengkodean dimana setiap gen bernilai salah satu dari bilangan bulat dalam interval $[1, N]$ untuk suatu N .

Parameter genetika yang berguna dalam pengendalian operator-operator genetika adalah : Ukuran populasi, Probabilitas *Crossover*, Probabilitas Mutasi.

Pemilihan nilai-nilai parameter sangat berpengaruh terhadap kinerja Algoritma Genetika. Adapun operator dasar yang sering dipergunakan, yaitu Reproduksi, *Crossover*, dan Mutasi.

4. Reproduksi

Reproduksi adalah proses dimana string individu di-copy berdasarkan nilai fungsi objektifnya, yaitu fungsi *fitness*. Meng-copy string berdasarkan nilai *fitness*-nya berarti bahwa string dengan nilai lebih tinggi memiliki kemungkinan lebih tinggi pula untuk menyumbang satu atau lebih keturunan ke dalam generasi selanjutnya. Operator reproduksi ini dilakukan pada proses seleksi. Pada proses seleksi, yang diperlukan hanyalah nilai *fitness* dari setiap string. Semakin besar nilai *fitness* dari suatu string, semakin besar pula kemungkinan dari string tersebut untuk terpilih menjadi calon anggota bagi generasi selanjutnya. Perangkat lunak yang dipergunakan pada operator genetika ini adalah metode piringan rolet (*roulette wheel*).

Untuk *Travelling Salesman Problem*, masalahnya adalah meminimalkan total biaya (masalah minimasi seperti yang telah diuraikan di atas). Oleh karena itu nilai *fitness* yang bisa digunakan adalah 1 dibagi total biaya. Dalam hal ini yang dimaksud total biaya pada suatu jalur perjalanan *salesman* adalah jumlah jarak kartesian antara satu kota dengan kota yang lainnya secara melingkar pada jalur tersebut. Jarak kartesian antara kota A dan kota B dihitung dengan rumus :

$$\|A - B\| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

Berikut ini langkah-langkah melakukan proses seleksi dengan piringan rolet :

- Hitung nilai *fitness eval* (v_i) untuk setiap kromosom v_i ($i=1, \dots, uk_pop$).
- Cari total *fitness* dari populasi.

$$F = \sum_{i=1}^{uk_pop} eval(v_i). \quad (2.5)$$

- Hitung nilai probabilitas (p_i) untuk setiap kromosom v_i ($i=1, \dots, uk_pop$).

$$p_i = eval(v_i)/F \quad (2.6)$$

- Hitung nilai probabilitas kumulatif q_i untuk setiap kromosom v_i ($i=1, \dots, uk_pop$).

$$q_i = \sum_{j=1}^i p_j \quad (2.7)$$

Proses pemilihan kromosom berdasarkan pemutaran piringan rolet sebanyak jumlah kromosom (uk_pop). Berikut ini pemilihan kromosom untuk populasi yang baru: .

- Bangkitkan nilai acak r dengan nilai antara $[0..1]$.
- Jika $r < q_i$, maka pilih kromosom ke-1 (v_i); sebaliknya pilih kromosom ke- i (v_i) dimana $2 \leq i \leq uk_pop$ dan $q_{i-1} < r \leq q_i$.

5. Crossover

Setelah proses seleksi dengan operator reproduksi selesai, akan dilakukan proses rekombinasi terhadap individu baru dengan menggunakan operator *crossover* atau pindah silang, yaitu proses memadukan sifat-sifat dari 2 kromosom orang tua yang menghasilkan 2 keturunan yang mirip dengan menukar segmen-segmen yang cocok diantara 2 kromosom orang tua. *Crossover* dapat berakibat buruk jika ukuran populasinya sangat kecil. Dalam suatu populasi yang sangat kecil, suatu kromosom dengan gen-gen yang baik akan sangat cepat menyebar menjadi kromosom-kromosom lain yang tercampur dengan gen-gen yang kurang baik. Untuk mengatasi masalah ini digunakan suatu aturan bahwa *crossover* hanya bisa dilakukan dengan suatu probabilitas tertentu, yaitu parameter probabilitas crossover (p_c). Artinya, *crossover* bisa dilakukan hanya jika suatu bilangan random $[0,1]$ yang dibangkitkan kurang dari p_c yang ditentukan. Pada umumnya p_c di-set mendekati 1. Untuk lebih jelasnya, berikut langkah-langkah proses rekombinasi dengan operator *crossover*:

- Bangkitkan nilai acak r dengan nilai antara $[0..1]$;
- Jika $r < p_c$, pilih kromosom yang diberikan untuk *crossover* lalu lakukan proses rekombinasi.

Setelah diambil 2 buah kromosom secara acak, selanjutnya dibangkitkan bilangan acak pos dengan nilai antara $[1..m-1]$ (m merupakan panjang jumlah bit-kromosom). Nilai pos sendiri merepresentasikan posisi titik penyilangan 2 kromosom ($b_1b_2... b_{pos}b_{pos+1}...b_m$) dan ($c_1c_2... c_{pos}c_{pos+1}...c_m$). setelah proses rekombinasi menjadi: ($b_1b_2... b_{pos}c_{pos+1}...c_m$) dan ($c_1c_2... c_{pos}b_{pos+1}...b_m$).

6. Mutasi

Selanjutnya akan dilakukan proses rekombinasi dengan operator mutasi. Mutasi adalah proses perubahan nilai sebuah bit pada string yang terjadi secara acak. Proses mutasi dilakukan dengan cara mengubah nilai sebuah bit dari 0 menjadi 1 atau sebaliknya. Proses mutasi ini memerlukan parameter probabilitas mutasi (p_m). Untuk semua gen yang ada, jika bilangan random yang dibangkitkan kurang dari p_m yang ditentukan, maka ubah gen tersebut menjadi nilai kebalikannya (seperti dijelaskan diatas, 0 diubah menjadi 1, dan 1 diubah menjadi 0). Biasanya p_m diset sebagai $1/n$, dimana n adalah jumlah gen dalam kromosom. Dengan p_m sebesar ini berarti mutasi hanya terjadi pada sekitar 1 gen saja. Pada Algoritma Genetika, nilai p_m adalah tetap selama evolusi. Untuk lebih jelasnya, berikut langkah-langkah proses rekombinasi dengan operator mutasi :

- Bangkitkan nilai acak r dengan nilai antara $[0..1]$;
- Jika $r < p_m$, lakukan mutasi pada bit yang dipilih.

Setelah semua operator berhasil dijalankan, maka populasi yang baru siap untuk evaluasi selanjutnya.

7. Order Crossover

Operator crossover ini merupakan penyilangan urutan. Misalkan p_1 dan p_2 adalah individu-individu yang hendak disilangkan.

$$P_1 = (1 2 3 \mid 4 5 6 7 \mid 8 9)$$

$$P_2 = (4 5 2 \mid 1 8 7 6 \mid 9 3)$$

Dua posisi pemotongan dipilih, dalam hal ini ditandai dengan “ | ”. Turunannya mula-mula mewarisi bagian tengah antara dua posisi pemotongan.

$$O_1 = (x x x \mid 4 5 6 7 \mid x x)$$

$$O_2 = (x x x \mid 1 8 7 6 \mid x x)$$

Untuk menangani O_1 , P_2 dinyatakan kembali mulai dari kode kota pertama setelah posisi pemotongan kedua.

$$P_2 = (9 3 4 5 2 1 8 7 6)$$

Kemudian kode-kode yang sudah dimiliki O_1 dibuang dari P_2 :

$$9 - 3 - 2 - 1 - 8$$

Barisan ini kemudian diletakkan setelah posisi pemotongan kedua dari bersambung ke posisi pemotongan pertama.

$$(2 1 8 \mid 4 5 6 7 \mid 9 3)$$

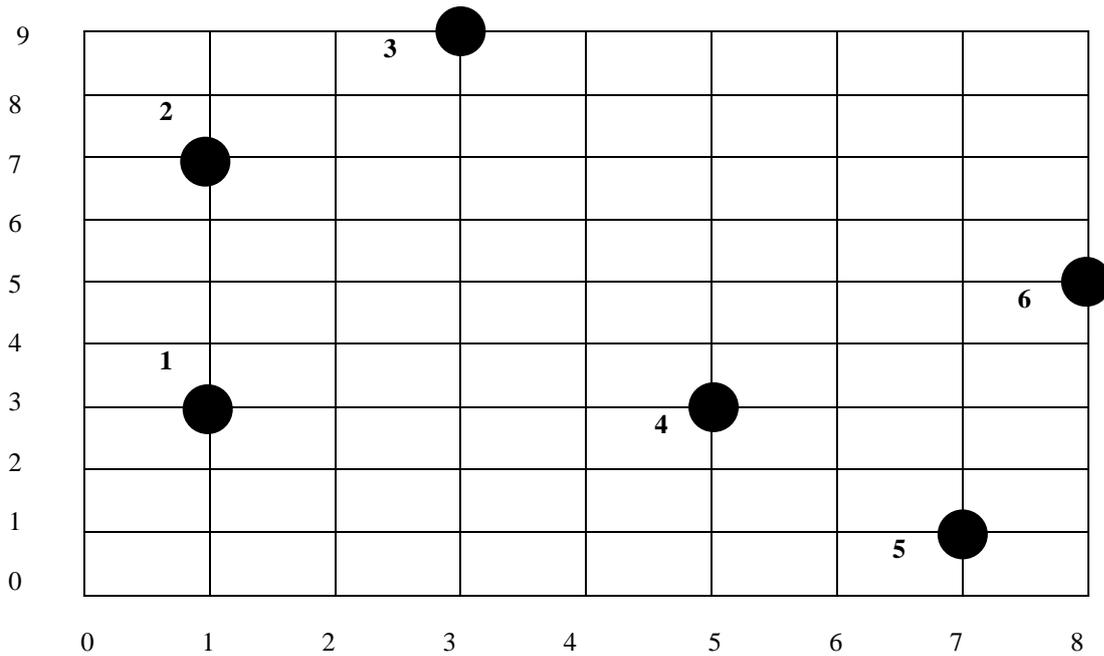
O_2 diperoleh dengan cara yang sama serupa dengan langkah-langkah yang dilewati untuk mendapatkan O_1

$$(3 4 5 \mid 1 8 7 6 \mid 9 2)$$

Probabilitas crossover yang diberikan adalah $(p_c) = 0.25$, sehingga dapat diasumsikan bahwa 25% kromosom dari ukuran populasi akan mengalami crossover.

8. Eksperimen

Misalkan diberikan suatu peta dua dimensi suatu kota beserta data jarak antar kotanya, seperti di bawah ini.



Gambar 1 : Peta dua dimensi untuk travelling salesman problem
 Koordinat: 1(1 3), 2(1 7), 3(3 9), 4(5 3), 5(7 1), 6(8 5)

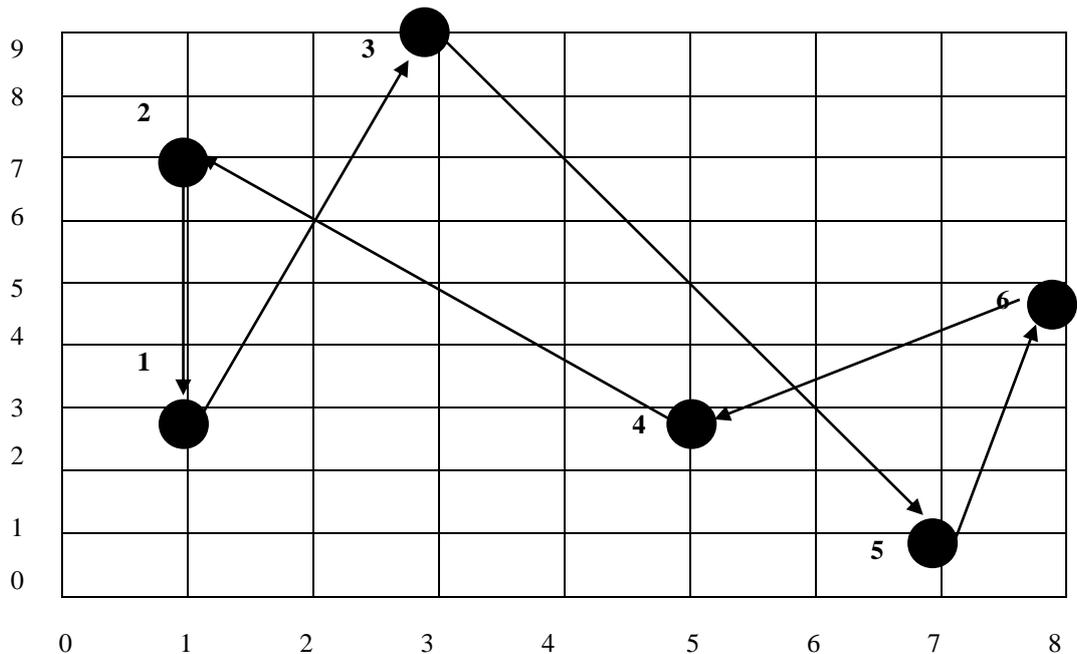
8.1. Data Jarak antar Kota

$$\begin{aligned}
 B(1, 2) = B(2, 1) &= \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} = \sqrt{(1-1)^2 + (3-7)^2} = 4 \\
 B(1, 3) = B(3, 1) &= \sqrt{(X_1 - X_3)^2 + (Y_1 - Y_3)^2} = \sqrt{(1-3)^2 + (3-9)^2} = 6,3246 \\
 B(1, 4) = B(4, 1) &= \sqrt{(X_1 - X_4)^2 + (Y_1 - Y_4)^2} = \sqrt{(1-5)^2 + (3-3)^2} = 4 \\
 B(1, 5) = B(5, 1) &= \sqrt{(X_1 - X_5)^2 + (Y_1 - Y_5)^2} = \sqrt{(1-7)^2 + (3-1)^2} = 6,3246 \\
 B(1, 6) = B(6, 1) &= \sqrt{(X_1 - X_6)^2 + (Y_1 - Y_6)^2} = \sqrt{(1-8)^2 + (3-5)^2} = 7,2801 \\
 B(2, 3) = B(3, 2) &= \sqrt{(X_2 - X_3)^2 + (Y_2 - Y_3)^2} = \sqrt{(1-3)^2 + (7-9)^2} = 2,8284 \\
 B(2, 4) = B(4, 2) &= \sqrt{(X_2 - X_4)^2 + (Y_2 - Y_4)^2} = \sqrt{(1-5)^2 + (7-3)^2} = 5,6569 \\
 B(2, 5) = B(5, 2) &= \sqrt{(X_2 - X_5)^2 + (Y_2 - Y_5)^2} = \sqrt{(1-7)^2 + (7-1)^2} = 8,4853 \\
 B(2, 6) = B(6, 2) &= \sqrt{(X_2 - X_6)^2 + (Y_2 - Y_6)^2} = \sqrt{(1-8)^2 + (7-5)^2} = 7,2801 \\
 B(3, 4) = B(4, 3) &= \sqrt{(X_3 - X_4)^2 + (Y_3 - Y_4)^2} = \sqrt{(3-5)^2 + (9-3)^2} = 6,3246 \\
 B(3, 5) = B(5, 3) &= \sqrt{(X_3 - X_5)^2 + (Y_3 - Y_5)^2} = \sqrt{(3-7)^2 + (9-1)^2} = 8,9443 \\
 B(3, 6) = B(6, 3) &= \sqrt{(X_3 - X_6)^2 + (Y_3 - Y_6)^2} = \sqrt{(3-8)^2 + (9-5)^2} = 7,2111 \\
 B(4, 5) = B(5, 4) &= \sqrt{(X_4 - X_5)^2 + (Y_4 - Y_5)^2} = \sqrt{(5-7)^2 + (3-1)^2} = 2,8284 \\
 B(4, 6) = B(6, 4) &= \sqrt{(X_4 - X_6)^2 + (Y_4 - Y_6)^2} = \sqrt{(5-8)^2 + (3-5)^2} = 3,6055 \\
 B(5, 6) = B(6, 5) &= \sqrt{(X_5 - X_6)^2 + (Y_5 - Y_6)^2} = \sqrt{(7-8)^2 + (1-5)^2} = 4,1231
 \end{aligned}$$

Setelah melalui tahapan-tahapan algoritma genetika untuk mencari nilai optimum menggunakan software aplikasi MATLAB 07, didapat Jalur terbaik :

4 2 1 3 5 6

Panjang jalur terbaik : 32.6543 unit kartesian.
Sehingga dapat digambarkan :



Gambar 1 Jalur Optimum

9. Kesimpulan

Algoritma *Traveling Salesman Problem* melibatkan banyak parameter yang berpengaruh pada kinerjanya mencari solusi yang sebaik mungkin. Simulasi untuk mencari solusi menggunakan Algoritma Genetika dalam program MATLAB menunjukkan bahwa analisis algoritma tersebut cukup baik dan mendekati optimal. Hal ini terlihat dari 3 kali *running program*, Panjang Jalur Terbaiknya tidak memiliki nilai yang jauh berbeda. Untuk ukuran populasi yang diambil dari 50% maupun 100% dari total populasi Panjang Jalur Terbaik kedua populasi tersebut adalah 25.9901.

Implementasi Algoritma Genetika dalam MATLAB menjadi mudah dipahami, karena di dalam MATLAB telah tersedia banyak fungsi untuk operasi-operasi matriks.

Algoritma Genetika tidak melibatkan seluruh populasi di dalam proses, sehingga menghemat waktu proses dan menghindari komputasi menjadi tak terhingga. Dengan demikian Algoritma Genetika baik untuk digunakan pada masalah-masalah yang melibatkan banyak populasi.

Daftar Pustaka

- [1] Anton, Howard, 1994. *Aljabar Linier Elementer*, Jakarta : Erlangga.
- [2] Goldberg, David E., (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Massachusetts: Addison-wesley Publishing Company, Inc.
- [3] Johnsonbaugh, Richard., (1998). *Matematika Diskrit edisi Bahasa Indonesia*. Jakarta : Pearson Education Asia Ptc. Ltd dan PT. Prenhallindo.
- [4] Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs, Third, Revised and Extended Edition*. Berlin, Heidelberg, New York : Springer-Verlag.
- [5] Suyanto. (2005). *Algoritma Genetika dalam MATLAB*. Yogyakarta : Andi.
- [6] Saputro, Nico., (2003). *Pengenalan Huruf dengan Memakai Algoritma Genetik*. Integral, Vol 8 No. 2, Oktober 2003.
- [7] Syamsudin, Aries., (2004). *Pengenalan Algoritma Genetik*. (*IlmuKomputer.com*).