JURSISTEKNI (Jurnal Sistem Informasi dan Teknologi Informasi) Vol 2, No. 1, Bulan Januari 2020: Hal. 10 - 18
ISSN. 2715-1875, E-ISSN. 2715-1883

PENYELESAIAN *TRAVELLING SALESMAN PROBLEM* MENGGUNAKAN ALGORITMA GENETIKA

[1)] Rizki Rino Pratama, [2)] Rintho Rante Rerung, [3)] Aditya Erfina
[1,2)]Teknik Informatika Universitas Nusa Putra
[3)]Sistem Informasi Universitas Nusa Putra
Jl. Raya Cibolang No.21Cisaat, Sukabumi
e-mail :[1)]rizki.pratama@nusaputra.ac.id[2)] rrantererung@gmail.com,[3)]aditya.erfina@nusaputra.ac.id

* Korespondensi: e-mail: rizki.pratama@nusaputra.ac.id

ABSTRAK

Optimasi pemilihan rute merupakan masalah yang banyak dibahas pada penelitian ilmu komputer. Penghitungan rute tercepat memegang peranan penting karena harus tepat waktu dan semua pelanggan dapat dilayani. Traveling salesman problem (TSP) bertujuan untuk meminimalkan jarak. Pencarian solusi untuk permasalahannya adalah dengan mengkombinasikan solusi-solusi (kromosom) untuk menghasilkan solusi baru dengan menggunakan operator genetika (seleksi, crossover dan mutasi). Untuk mencari solusi terbaik digunakan beberapa kombinasi probabilitas crossover dan mutasi serta ukuran populasi dan ukuran generasi. Dari hasil pengujian kombinasi probabilitas crossover yang terbaik adalah 0,4 dan mutasi adalah 0,6 sedangkan untuk ukuran generasi optimal adalah 2000. Dari nilai-nilai parameter ini didapatkan solusi yang memungkinkan untuk melayani semua nodes.

***Kata Kunci**: Traveling salesman problem, algoritma genetika, rute terbaik.*

ABSTRACT

Optimization of route selection is a problem that is widely discussed in computer science research. The calculation of the fastest route plays an important role because it must be on time and all customers can be served. Traveling salesman problem (TSP) aims to minimize distance. The search for a solution to the problem is to combine solutions (chromosomes) to produce new solutions using genetic operators (selection, crossover and mutation). To find the best solution, several combinations of crossover and mutation probabilities are used as well as population size and generation size. From the test results the best combination of crossover probability is 0.4 and mutation is 0.6 while for optimal generation size is 2000. From the values of these parameters we get a solution that allows to serve all nodes.

***Keywords:** Traveling salesman problem, genetic algorithm, the best route.*

I. PENDAHULUAN

Salah satu contoh pencarian rute tercepat yaitu pemilihan rute yang dipilih sopir pengirim barang untuk sampai pada tujuan dengan tepat waktu. Setiap daerah tujuan pengiriman tersebut harus dikunjungi satu kali, kemudian kembali lagi ke tempat awal. Permasalahan tersebut dikenal sebagai *Travelling Salesman Problem* (TSP) [1].

Beberapa penelitian sebelumnya untuk menyelesaikan TSP telah diajukan seperti "Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya" [2] dengan menggunakan *graph*, bisa dilakukan pencarian rute terpendek dikota Surabaya. "Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner" [3] membahas tentang jarak terpendek dari tujuan wisatawan yang ingin melakukan wisata kuliner dengan menggunakan algoritma genetika. "*Vehicie Routing Optimization Probiem with Time-windows and its Solution by Genetic Aigorithm*" oleh [4] meneliti tentang suatu optimasi masalah VRP dengan *time windows* menggunakan algoritma genetika. Pada penelitian ini digunakan algoritma genetika dengan harapan akan diperoleh optimasi rute perjalanan yaitu kondisi dimana terjadi kombinasi terbaik untuk jalur yang akan dilalui dan waktu perjalanan yang cepat, serta semua pelanggan dapat terlayani.

II. TINJUAN PUSTAKA

2.1 *Travelling Salesman Problem*

Travelling Salesman Problem (TSP) adalah pencarian rute terpendek atau jarak minimum oleh seorang salesman dari suatu kota ke n-kota tepat satu kali dan kembali ke kota awal keberangkatan. TSP dapat diterapkan pada graph komplit berbobot yang memiliki total bobot sisi minimum, dimana bobot pada sisi adalah jarak. Rute TSP ini memuat semua titik pada graph tersebut tepat satu kali. Banyak algoritma telah dikembangkan dalam menyelesaikan permasalahan TSP, namun ada beberapa algoritma yang dirasa kurang dalam hal performasinya [1].

2.2 Algoritma Genetika

Algoritma generika pertama kali dikembangkan pada tahun 1975 oleh Jhon Hollan dari Unversitas Michigan [5]. Algoritma genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi yang dikemukakan oleh Charles Darwin. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. "Hanya individu-individu yang kuat yang mampu bertahan". Algoritma genetika mungkin tidak selalu mencapai hasil yang terbaik, tetapi seringkali memecahkan masalah dengan cukup baik. Algoritma genetika merepresentasikan suatu solusi permasalahan sebagai kromosom. Terdapat beberapa aspek penting dalam algoritma genetika antara lain defenisi fungsi *fitness*, defenisi dan implementasi representasi genetika, defenisi dan implementasi operasi genetika. Ketiga aspek diatas sangat mendukung kinerja algoritma genetika [5]. Jumlah populasi solusi yang besar adalah keunggulan algoritma genetika.

2.2.1 Nilai Fitness

Fitness adalah nilai yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan criteria atau tujuan (obyektif) permasalahan yang ingin dicapai [6]. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Fungsi tujuan untuk sistem optimasi menggunakan Algoritma genetika dapat ditunjukkan pada persamaan Persamaan 1 dan Persamaan 2 dibawah ini [6].

Nilai *fitness* = $1/f_x$ ……(1)

Dimana,

f_X □ □(c_{ij}) □ □p_i ……..(2)

Keterangan:

c_{ij} adalah waktu tempuh dari titik i ke titik j.

p_i merupakan penalti jika pelanggan dilayani diluar jadwal.

2.2.2 Crossover

Crossover adalah mekanisme yang dimiliki algoritma genetika dengan menggabungkan dua kromosom sehingga menghasilkan anak kromosom yang mewarisi ciri-ciri dasar dari parent *crossover* bekerja membangkitkan *offspring* baru dengan mengganti sebagian informasi dari parents [6]. Pada penelitian ini digunakan metode *crossover* PMX dikarenakan dengan metode ini bisa mencegah adanya gen ganda pada suatu individu. Langkah langkah metode ini adalah [7]:

1. Tentukan dua posisi pada kromosom dengan aturan acak. *Substring* yang berada dalam dua posisi ini dinamakan daerah pemetaan.
2. Tukar dua *substring* antar induk untuk menghasilkan anak.
3. Tentukan hubungan pemetaan di antara dua daerah pemetaan.
4. Tentukan kromosom keturunan mengacu pada hubungan pemetaan.

2.2.3 Mutasi

Proses mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk mengganti gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal [8]. Sehingga bisa disimpulkan bahwa mutasi akan meningkatkan variasi populasi. penelitian ini digunakan *reciprocal exchange* mutasi dengan memilih dua posisi secara random, kemudian menukar kedua posisi tersebut. Penggunaan metode ini dikarenakan sangat mudah dan sederhana untuk diimplementasikan dan hasil dari proses mutasi tidak akan terdapat gen yang sama pada anaknya.

2.2.4 Seleksi

Proses seleksi adalah proses untuk menyaring calon generasi yang baru. Induk yang baik akan mampu untuk menghasilkan anak yang baik. Semakin tinggi nilai *fitness* dari suatu individu maka semakin besar kemungkinannya untuk terpilih [3]. Kemampuan algoritma genetika untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan ke populasi. Penekanan selektif dapat diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikutnya.

Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik. Cara lain menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sejumlah populasi yang akan disimpan untuk generasi selanjutnya. Metode untuk seleksi yang sering digunakan antara lain adalah seleksi roda roulet (*roulette wheel selection*), seleksi ranking (*rank selection*), elitis dan seleksi turnamen (*tournament selection*).

2.2.4.1 Metode Seleksi *Roulette Wheel*

Metode seleksi *roulette wheel* merupakan metode yang paling sederhana serta paling banyak digunakan, dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini, orangtua dipilih berdasarkan nilai *fitness*nya, semakin baik nilai *fitness*nya maka semakin besar kemungkinannya untuk terpilih. Diandaikan semua kromosom diletakkan pada sebuah roda roulet, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*nya [9].

2.2.4.2 Metode Seleksi Elitis

Metode Seleksi Elitis memilih individu-individu untuk dipakai pada generasi selanjutnya didasarkan pada urutan nilai *fitness*. Semakin tinggi nilai *fitness*nya maka individu tersebut akan dipertahankan [9]. Proses seleksi dilakukan dengan mengurutkan semua kromosom pada satu generasi lalu diambil sebanyak ukuran populasi yang diinginkan.

III. METODOLOGI PENELITIAN

Terdapat perlakuan data yang digunakan pada penelitian ini yaitu perlakuan dengan generasi terbanyak dan populasi terbanyak. Data jarak antar pelanggan didapatkan dari *google*. Contoh data jarak antar pelanggan bisa dilihat pada Gambar 1.

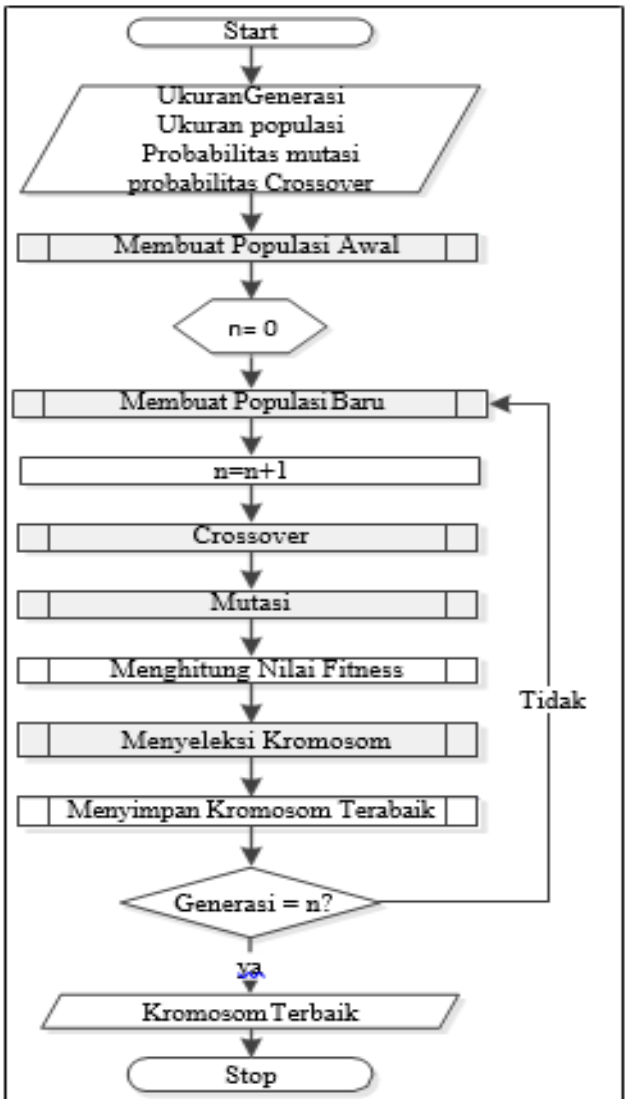
```
[[ -1. 467. 974.  12. 978. 987. 978. 876. 123. 145.]
 [467.  -1. 356.  97. 345. 234. 532. 567. 987. 759.]
 [974. 356.  -1. 567. 876. 987. 789. 234. 345. 478.]
 [ 12.  97. 567.  -1. 678. 754.  54. 876.  76. 489.]
 [978. 345. 876. 678.  -1. 123. 987.  65. 756. 458.]
 [987. 234. 987. 754. 123.  -1.  34. 987. 946. 787.]
 [978. 532. 789.  54. 987.  34.  -1. 456. 386. 864.]
 [876. 567. 234. 876.  65. 987. 456.  -1. 487. 456.]
 [123. 987. 345.  76. 756. 946. 386. 487.  -1. 986.]
 [145. 759. 478. 489. 458. 787. 864. 456. 986.  -1.]]
```

Gambar 1. Matriks jarak antar nodes

Proses pencarian rute tercepat dengan algoritma genetika adalah seperti pada *flowchart* Gambar 2. Proses pertama adalah dengan menginisialisasikan parameter awal yaitu: (1) Memasukkan tujuan dan waktu ketersediaan untuk masing-masing tujuan. (2) Ukuran individu pada setiap populasi. (3) Ukuran generasi. (4) Probabilitas *crossover*. (5) Probabilitas mutasi. Setelah menginisialisasikan parameter awal proses selanjutnya adalah membangkitkan populasi awal dengan panjang kromosom adalah banyaknya tujuan yang akan dituju, banyaknya populasi sesuai dengan jumlah individu yang telah diinisialisasi sebelumnya. Setelah mendapatkan populasi awal langkah selanjutnya adalah reproduksi dengan cara melakukan *crossover* dan mutasi. Pada proses *crossover* dan mutasi populasi diambil untuk dijadikan sebagai calon induk. Pemilihan induk dilakukan secara random untuk menghasilkan anak sebanyak probabilitas *crossover* dan mutasi.

Proses selanjutnya adalah dengan menghitung nilai *fitness* semua kromosom dari semua proses pada generasi ini. Proses perhitungan nilai *fitness* dengan menggunakan Persamaan 1 dan Persamaan 2. setelah semua kromosom dihitung nilai *fitness*nya dilanjutkan dengan menyeleksi kromosom untuk diproses pada generasi selanjutnya dan

menyimpan kromosom dengan nilai *fitness* terbaik. Pemilihan kromosom terbaik dilakukan dengan cara membandingkan nilai *fitness* terbaik pada setiap generasi. Proses seleksi melibatkan seluruh kromosom dari generasi awal dan kromosom hasil dari proses *crossover* dan mutasi. Hasil akhir dari algortima genetika adalah menampilkan kromosom yang memiliki nilai *fitness* tertinggi dari semua generasi

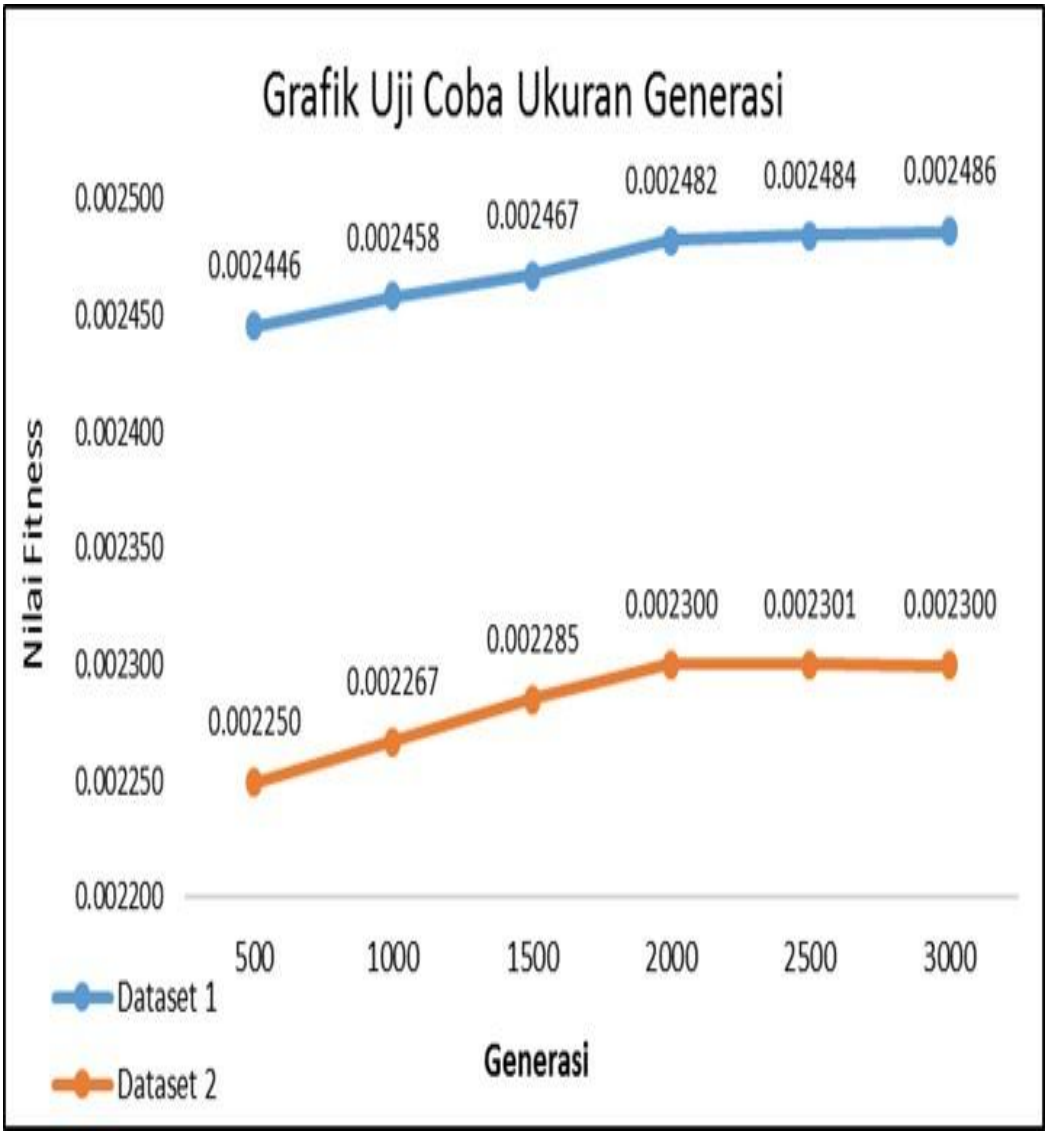


Gambar 2: Flowchart Program

IV. HASIL DAN PEMBAHASAN

Dilakukan pencarian ukuran generasi yang optimal. Setiap ukuran generasi dilakukan 6 kali percobaan dengan ukuran populasi adalah 50 populasi, ukuran generasi adalah kelipatan 500 mulai dari 500 sampai 3000 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,2. Metode seleksi yang digunakan adalah elitis.

Pada Gambar 3 bisa dilihat bahwa ukuran generasi berpengaruh terhadap hasil dari proses algoritma genetika. Nilai terendah terdapat pada generasi 500 yaitu ukuran generasi terendah pada percobaan ini dikarenakan algoritma genetika belum memproses secara optimal karena kurangnya generasi. Ukuran generasi yang optimal untuk *traveling salesman problem* adalah 2000 generasi, dikarenakan dengan ukuran generasi 2000 sampai 3000 generasi, nilai *fitness* yang dihasilkan tidak berbeda jauh dan cenderung membentuk garis lurus.

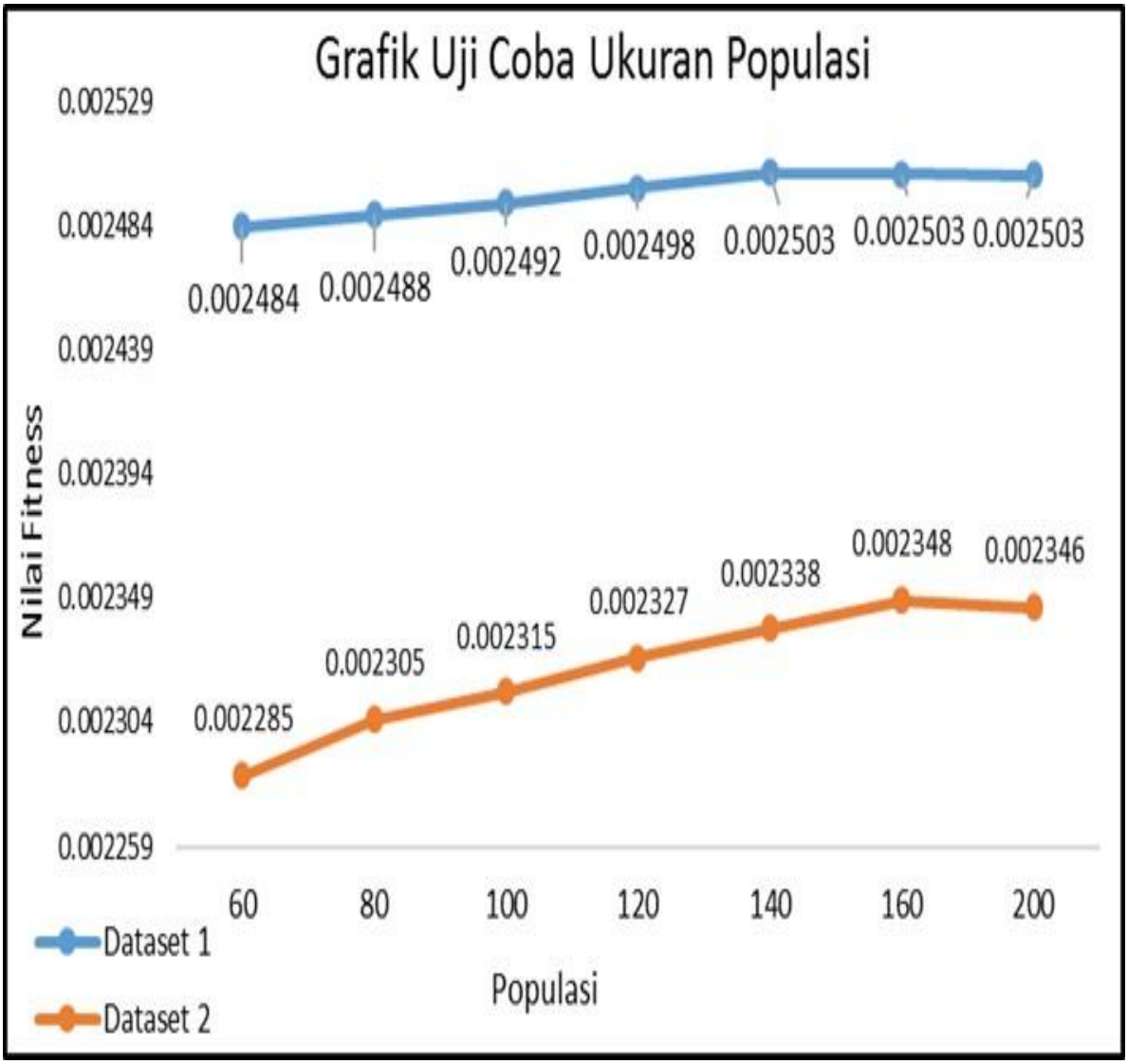


Gambar 3. Grafik rata-rata nilai fitnes tiap generasi dataset

Terlalu banyak ukuran generasi belum tentu membuat algoritma genetika menjadi lebih optimal. Selain waktu proses yang menjadi lebih lama hasil nilai *fitness* yang dihasilkan juga belum tentu jauh lebih baik dari generasi yang lebih rendah. Ukuran generasi yang tinggi akan mengakibatkan proses evolusi semakin sering dilakukan. Pada setiap satu generasi, akan dilakukan proses rekombinasi yang terdiri atas *crossover* dan mutasi. Sehingga semakin banyak generasi maka proses rekombinasi akan semakin sering dilakukan. Tentunya hal ini akan berpengaruh pula terhadap individu – individu baru yang dihasilkan. Semakin banyak melakukan proses *crossover* dan mutasi maka individu – individu baru yang dihasilkan akan semakin bervariasi dan memungkinkan pula bervariasinya nilai *fitness* yang dihasilkan. Dengan begitu akan memberikan peluang yang besar untuk mendapatkan nilai *fitness* yang baik.

Dilakukan pencarian ukuran populasi yang optimal. Setiap ukuran populasi dilakukan 7 kali percobaan. Ukuran populasi adalah 60 sampai 200 populasi dengan kelipatan 40 populasi, ukuran generasi adalah 1500 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,3. Metode seleksi yang digunakan adalah elitis.Hasil uji coba bisa dilihat pada Gambar 5.

Pada Gambar 4 bisa dilihat kenaikan signifikan rata-rata nilai *fitness* untuk 20 kali percobaan pada *dataset* mulai dari ukuran populasi 60 sampai dengan 140 populasi, namun untuk ukuran populasi 140 sampai 200 populasi sudah tidak terjadi perubahan yang cukup signifikan dan pada grafik cenderung membentuk garis lurus. 140 populasi merupakan ukuran populasi yang optimal untuk masalah *TSP* pada dataset.



Gambar 4: Grafik hasil uji coba ukuran populasi dataset

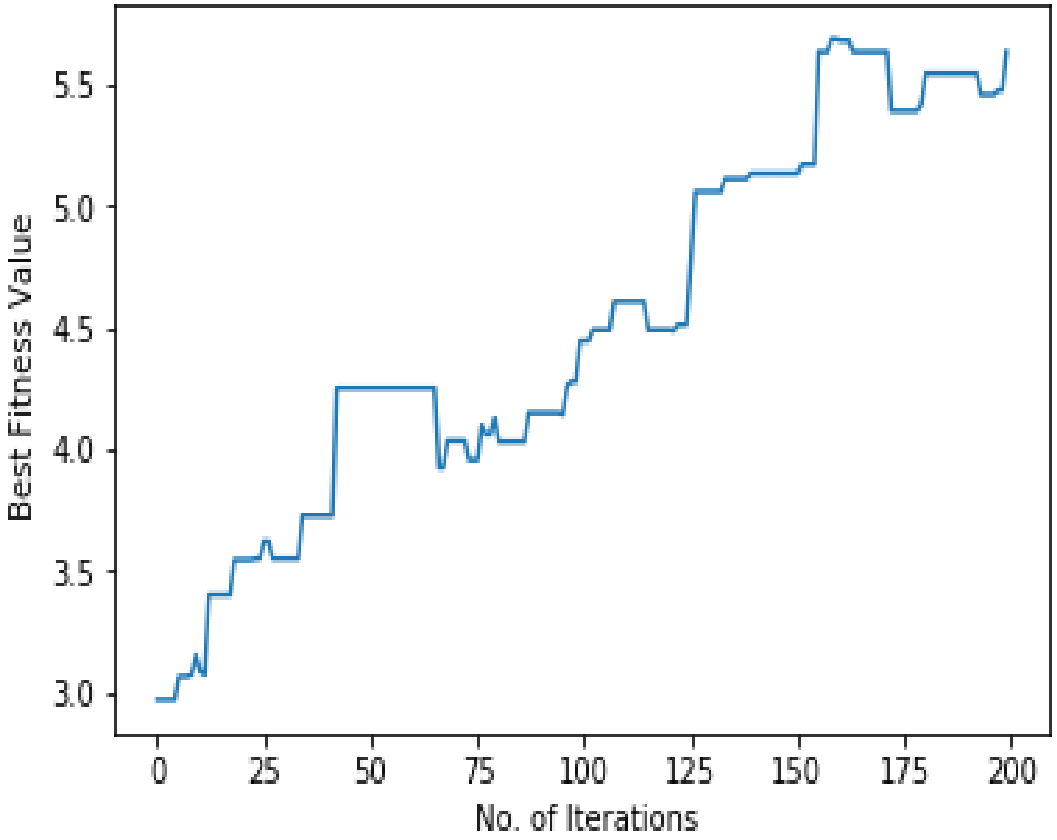
Semakin tinggi ukuran populasi maka berpengaruh pada rata-rata nilai *fitness* yang didapatkan namun semakin tinggi ukuran populasi juga berrpengaruh pada waktu pemrosesan algoritma genetika akan semakin lama, sehingga menentukan berapa ukuran populasi yang optimal sangat diperlukan. Ukuran populasi yang optimal untuk setiap *dataset* dapat berbeda- beda seperti pada grafik Gambar 4. Ukuran populasi yang optimal untuk *dataset* 1 adalah 140 populasi berbeda dengan *dataset* 2 yaitu 160 populasi, hal ini dapat disebabkan karena ukuran populasi yang kecil dimana variasi individu-individu di dalamnya sedikit. Individu yang akan terpilih sebagai kandidat *parent* akan memiliki variasi terbatas, dan anak yang dihasilkan bisa jadi memiliki sifat yang mirip dengan nilai *fitness* yang hampir dekat (tidak mengalami improvisasi).

Perbedaan nilai probabilitas *crossover* dan mutasi mempengaruhi nilai *fitness* yang dihasilkan. Semakin besar nilai probabiltas *crossover* dan probabilitas mutasi maka peluang individu yang mengalami proses *crossover* dan mutasi akan semakin besar dan akan semakin banyak individu – individu baru yang dihasilkan. Dengan demikian nilai *fitness* yang dihasilkan juga semakin bervariasi, sehingga peluang untuk mendapatkan individu dengan nilai *fitness* yang baik juga akan semakin besar.

```
Best Fitness Value at iteration 184 is 5.544
Best Fitness Value at iteration 185 is 5.544
Best Fitness Value at iteration 186 is 5.544
Best Fitness Value at iteration 187 is 5.544
Best Fitness Value at iteration 188 is 5.544
Best Fitness Value at iteration 189 is 5.544
Best Fitness Value at iteration 190 is 5.544
Best Fitness Value at iteration 191 is 5.544
Best Fitness Value at iteration 192 is 5.544
Best Fitness Value at iteration 193 is 5.544
Best Fitness Value at iteration 194 is 5.456692913385827
Best Fitness Value at iteration 195 is 5.456692913385827
Best Fitness Value at iteration 196 is 5.456692913385827
Best Fitness Value at iteration 197 is 5.456692913385827
Best Fitness Value at iteration 198 is 5.475169300225733
Best Fitness Value at iteration 199 is 5.475169300225733
Best Fitness Value at iteration 200 is 5.634146341463414

HIGHEST FITNESS OVERALL :5.686987104337632 AT ITERATION 159
```

Gambar 5. Hasil Terbaik TSP dengan GA



Gambar 6. Hasil Terbaik TSP dengan GA

V. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah menggunakan ukuran generasi, ukuran populasi, probabilitas *crossover* dan mutasi dan metode seleksi yang tepat algoritma genetika dapat diimplementasikan untuk menyelesaikan permasalahan antar jemput *laundry*. Metode seleksi Elitis lebih baik dan lebih stabil dari pada metode seleksi *roulette wheel*. Ukuran generasi yang optimal adalah 2000 generasi dengan Probabilitas *crossover* yang optimal adalah 0,4 dan probabilitas mutasi yang optimal adalah adalah 0,6. Perbedaan *dataset*.yang digunakan dapat mempengaruhi nilai optimal parameter algoritma genetika. Dari nilai-nilai parameter ini didapatkan solusi yang memungkinkan untuk melayani semua pelanggan dengan *time window* masing – masing,

Untuk pengembangan penelitian selanjutnya dapat dikembangkan untuk menyelesaikan permasalahan pencarian rute perjalanan tercepat dengan menambahkan faktor seperti data kepadatan jalan dan data lampu merah pada jalan yang dilalui yang dapat mempengaruhi waktu tempuh perjalanan serta diambil secara *real time*. Pada kasus seperti ini diperlukan metode yang lebih baik untuk menjamin bahwa solusi yang mendekati optimum bisa tercapai. *Hibridisasi* algoritma genetika dengan metode lain merupakan satu pilihan yang terbukti efektif pada beberapa kasus kompleks [6].

DAFTAR PUSTAKA

[1] Gambardella, L. M., Taillard, E., & Agazzi, G. 1999. "A Multiple Ant Colony System For Vehicle Routing Problems With Time Windows". New Ideas in Optimization :McGraw- Hill, London.

[2] Purwanto, Y., Purwitasari, D., & Wibowo, W. A. 2005. "Implementasi Dan Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya". Jurnal Penelitian dan Pengembangan Telekomunikasi (Edisi 10).Surabaya:ITS.

[3] Widodo, AW & Mahmudy, WF 2010. 'Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner', Kursor, vol. 5, no. 4, pp. 205-211.

[4] Chen, T., & Zhou, G. 2013. "Vehicie Routing Optimization Probiem With Time-Windows And Its Soiution By Genetic Aigorithm". Journal of Digital Information Management Volume 11 Number 2 April 2013: Zhejiang Gongshang University.

[5] Nugraha, I. 2008."Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar". Institut Teknologi Bandung. Institut Teknologi Bandung.

[6] Mahmudy, WF, Marian, RM & Luong, LHS. 2013b, 'Real coded genetic algorithms for solving flexible job-shop scheduling problem – Part II: optimization', Advanced Materials Research, vol. 701, pp. 364-369.

[7] Azmi, N., Jamaran, I., Arkeman, Y., & Mangunwidjaya, D.2011. "Penjadwalan Pesanan Menggunakan Algoritma Genetika Untuk Tipe Produksi Hybrid And Flexible Flowshop Pada Industri Kemasan Karton". Jurnal Teknik Industri Vol 1 No 7.

[8] Zukhri, Z. 2004. "Penyelesaian Masalah Penugasan dengan Algoritma Genetika". Seminar Nasional Aplikasi Teknologi Informasi, Yogyakarta : Universitas Islam Indonesia.

[9] Wati, A. W. 2011. "Penerapan Algoritma Genetika Dalam Optimasi Model Dan Simulasi Dari Suatu Sistem". Jurnal Keilmuan Tehnik Industri (Edisi 1 Nomor 2), Jakarta : Universitas Trisakti.