

Penerapan Algoritma Genetika *Traveling Salesman Problem with Time Window*: Studi Kasus Rute Antar Jemput Laundry

Dwi Aries Suprayogi¹, Wayan F. Mahmudy²

Program Studi Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya
Jl. Veteran, Malang 65145

E-mail: ¹dwiariess@yahoo.co.id, ²wayan_firdaus@yahoo.com

Masuk: 20 Juni 2014; Direvisi: 18 Juli 2014; Diterima: 21 Juli 2014

Abstract. Routing optimization is a widely discussed problem in the area of computer science research. An example of the problem is laundry pickup service. In this case, each customer has reserved time to receive their order. Unlike the conventional traveling salesman problem (TSP) which goal is to choose the shortest path, in this case the precise arrival time each customer must be considered. The best Solution to the problem is achieved by combining solutions (chromosomes) to produce new solutions by using genetic operators (selection, crossover and mutation). A set of experiments is performed to find the best combination of parameters such as crossover probability, mutation probability, population size, and number of generation. From the test results the best combination crossover probability is 0,4 and mutation probability is 0,6. The optimum number of generation is 2000. The parameter can be used to obtain solution for serve all customer with different time window.

Keywords: Traveling salesman problem, time windows, genetic algorithm, optimum route, laundry pickup service.

Abstrak. Optimasi pemilihan rute merupakan masalah yang banyak dibahas pada penelitian ilmu komputer. Antar jemput laundry dengan pelanggan yang memiliki waktu khusus untuk menerima barang adalah salah satu contoh kasus pemilihan rute. Penghitungan rute tercepat memegang peranan penting karena harus tepat waktu dan semua pelanggan dapat dilayani. Berbeda dengan traveling salesman problem (TSP) konvensional yang bertujuan untuk meminimalkan jarak, kasus ini juga harus dipertimbangkan waktu ketersediaan setiap pelanggan. 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 pelanggan dengan time window masing - masing.

Kata Kunci: Traveling salesman problem, time windows, algoritma genetika, rute terbaik, antar jemput laundry.

1. Pendahuluan

Usaha laundry adalah usaha yang bergerak dibidang jasa cuci dan setrika. Keberadaan jasa cuci mencuci dan setrika sudah menjadi bagian dari kebutuhan hidup manusia. Berkembangnya bisnis laundry kiloan menjadikan persaingan di sektor ini menjadi semakin ketat. Untuk menjaga agar usaha ini tidak sepi oleh pelanggan setiap penyedia jasa laundry memiliki ciri khas dan cara promosi masing-masing, seperti menyediakan jasa antar jemput cucian (Anonymous, 2013). Pelanggan laundry ada yang mempunyai waktu tertentu untuk mengambil dan menerima cucian mereka. Melihat dari keadaan ini maka parameter waktu yang akan dihabiskan di jalan dapat diperkirakan rute mana yang akan diambil oleh sopir antar jemput cucian agar bisa datang di tempat sesuai dengan waktu yang di inginkan pelanggan. Untuk

menyelesaikan masalah ini maka dibuatlah suatu sistem yang memperhitungkan jarak antar pelanggan dengan melalui rute tercepat menggunakan konsep *Vehicle Routing Problem with Time Windows* (VRPTW) yang merupakan sebutan bagi VRP dengan kendala tambahan berupa adanya *time windows* pada masing-masing pelanggan (Gambardella, 1999). Waktu ketersediaan pada masing-masing pelanggan dapat berbeda satu sama lain dan dinyatakan dalam selang waktu berupa batas waktu awal sampai akhir pelayanan pada pelanggan tersebut.

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). salah satu bentuk pengembangan TSP yang lebih rumit yang melibatkan dua variabel atau lebih adalah TSP-TW yaitu pencarian rute optimal yang mempertimbangkan waktu total waktu perjalanan, dan waktu ketersediaan pelanggan (Gambardella, 1999).

Beberapa penelitian sebelumnya untuk menyelesaikan VRPTW telah diajukan seperti “Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya” (Purwanto, 2005) dengan menggunakan *graph*, bisa dilakukan pencarian rute terpendek dikota Surabaya. “Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner” (Widodo, 2010) membahas tentang jarak terpendek dari tujuan wisatawan yang ingin melakukan wisata kuliner dengan menggunakan algoritma genetika. “*Vehicle Routing Optimization Problem with Time-windows and its Solution by Genetic Algorithm*” oleh (Chen, 2013) 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.

2. Tinjauan Pustaka

2.1. *Vehicle Routing Problem with Time Windows*

Vehicle Routing Problem adalah suatu masalah pencaian jalur yang akan dilalui dengan tujuan mencari rute yang paling cepat atau pendek. *Vehicle Routing Problem with Time Windows* (VRPTW) yang merupakan sebutan bagi VRP dengan kendala tambahan berupa adanya *time windows* pada masing-masing pelanggan. Waktu ketersediaan pada masing-masing pelanggan dapat berbeda dan dinyatakan dalam selang waktu berupa batas waktu awal sampai akhir pelayanan pada pelanggan tersebut (Gambardella, 1999).

2.2. Algoritma Genetika

Algoritma generika pertama kali dikembangkan pada tahun 1975 oleh Jhon Hollan dari Universitas Michigan (Nugraha, 2008). 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 definisi fungsi *fitness*, definisi dan implementasi representasi genetika, definisi dan implementasi operasi genetika. Ketiga aspek diatas sangat mendukung kinerja algoritma genetika (Nugraha, 2008). 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 (Mahmudy, 2014). Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Fungsi tujuan untuk sistem optimasi rute antar jemput

laundry menggunakan Algoritma genetika dapat ditunjukkan pada persamaan Persamaan 1 dan Persamaan 2 dibawah ini (Mahmudy, 2013a).

$$\text{NilaiFitness} = \frac{1}{f_x} \quad (1)$$

dimana:

$$f_x = \sum (c_{ij}) + \sum p_i \quad (2)$$

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 (Mahmudy, 2013a). Pada penelitian ini digunakan metode *crossover* PMX dikarenakan dengan metode ini bisa mencegah adanya gen ganda pada suatu individu. Langkah – langkah metode ini adalah (Azmi, 2011): (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 (Zukhri, 2004). 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 (Widodo, 2010). 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 roulette (*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

fitnessnya maka semakin besar kemungkinannya untuk terpilih. Diandaikan semua kromosom diletakkan pada sebuah roda roulette, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitnessnya* (Wati, 2011).

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 *fitnessnya* maka individu tersebut akan dipertahankan (Wati, 2011). Proses seleksi dilakukan dengan mengurutkan semua kromosom pada satu generasi lalu diambil sebanyak ukuran populasi yang diinginkan.

3. Metodologi Penelitian

Terdapat dua jenis data antar pelanggan yang digunakan pada penelitian ini yaitu data pelanggan yang berjumlah 20 pelanggan untuk dataset 1 dan 30 pelanggan untuk dataset 2. Data jarak antar pelanggan didapatkan dari *google maps* dan data waktu ketersediaan pelanggan dibangkitkan secara acak dari pukul 07.00 sampai pukul 16.00. Contoh data jarak antar pelanggan bisa dilihat pada Tabel 1 dan untuk contoh data waktu ketersediaan pelanggan bisa dilihat pada Tabel 2.

Tabel 1. Contoh data jarak antar pelanggan

Tujuan	0	1	2	3	4	5
0	0	7	13	12	8	9
1	7	0	8	6	6	6
2	13	8	0	8	13	12
3	12	6	8	0	10	12
4	8	6	13	10	0	15
5	9	6	12	12	15	0

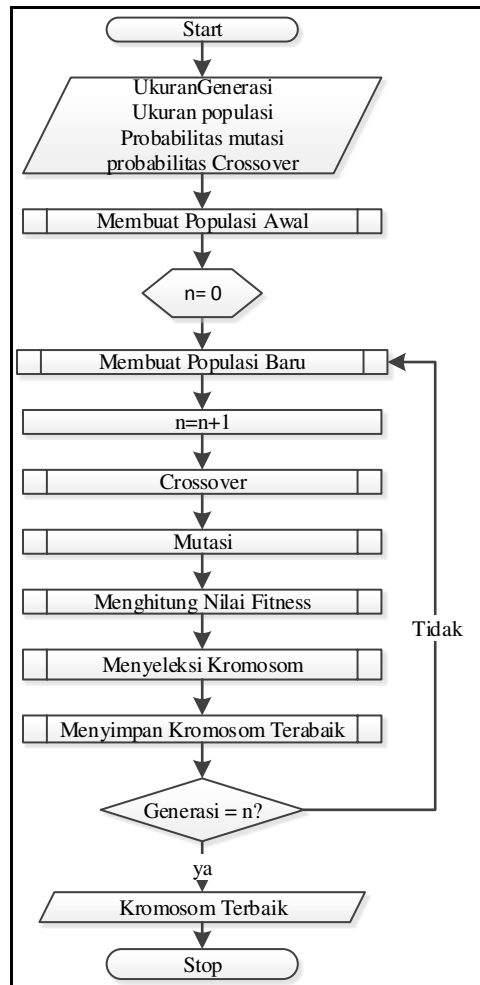
Tabel 2. Contoh data waktu ketersediaan pelanggan

Tujuan	Ketersediaan Waktu
0	08.00-16.00
1	08.00-14.00
2	14.00-16.00
3	12.00-14.00
4	09.00-13.00
5	08.00-09.00

Proses pencarian rute tercepat dengan algoritma genetika adalah seperti pada *flowchart* Gambar1. 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 *fitnessnya* 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 algoritma genetika adalah menampilkan kromosom yang memiliki nilai *fitness* tertinggi dari semua generasi.



Gambar1. Flowchart Program

Menurut (Mahmudy, 2013a) tidak ada metode pasti untuk menentukan parameter genetika yang digunakan seperti probabilitas *crossover*, probabilitas mutasi, ukuran populasi dan ukuran generasi. Penelitian ini dilakukan beberapa rangkaian sekenario uji coba. Sekenario uji coba yang dilakukan antara lain adalah: (1) Uji coba membandingkan metode seleksi *roulette wheel* dan metode seleksi elitis. (2) Uji coba untuk menentuka ukuran generasi yang optimal. (3) Uji coba untuk menentukanukuran populasi yang optimal. (4). Uji coba untuk mencari kombinasi probabilitas mutasi dan probabilitas *crossover* yang optimal.

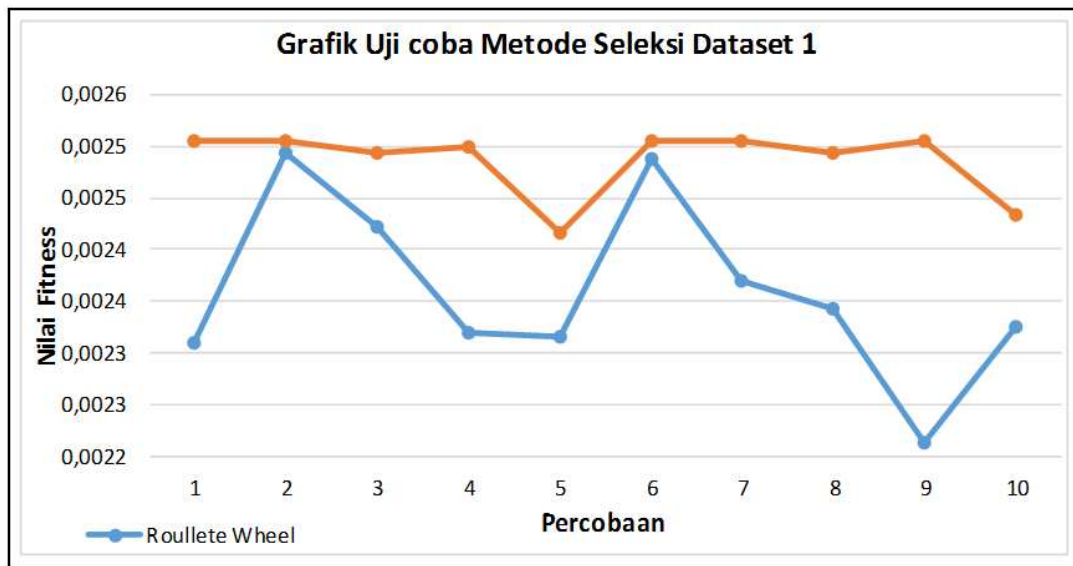
5. Hasil dan Pembahasan

5.1. Hasil dan Analisa Uji Coba Perbandingan Metode Seleksi *Roulette wheel* dan Elitis

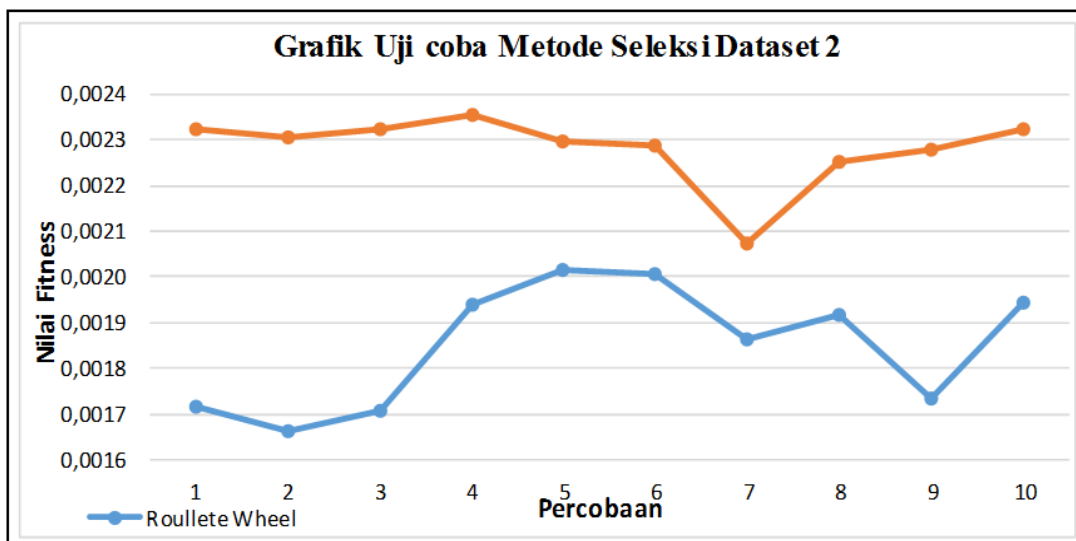
Uji coba yang pertama dilakukan pengujian metode seleksi elitis dan metode seleksi *roulette wheel*. Setiap metode seleksi dilakukan 10 kali percobaan dengan ukuran populasi adalah 40 populasi, ukuran generasi adalah 1000 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,5. Hasil uji coba bisa dilihat pada Gambar 2 untuk dataset 1, dan Gambar 3 untuk dataset 2.

Grafik menunjukkan bahwa dengan 10 kali percobaan dengan dataset1 nilai *fitness* yang dihasilkan metode seleksi *roulette wheel* selalu berada dibawah nilai *fitness* percobaan dengan menggunakan metode seleksi elitis. Pada Gambar 2 bisa dilihat metode seleksi elitis cenderung menghasilkan nilai *fitness* yang stabil dibandingkan dengan metode seleksi *roulette wheel*. Tidak jauh berbeda dengan 10 kali percobaan pada dataset2 pada Gambar 3 nilai *fitness* yang dihasilkan metode seleksi *roulette wheel* selalu berada dibawah nilai *fitness* percobaan dengan

menggunakan metode seleksi elitis. Pada Gambar 3 bisa dilihat metode seleksi elitis cenderung menghasilkan nilai *fitness* yang stabil dibandingkan dengan metode seleksi *roulette wheel*.



Gambar 2. Grafik Perbandingan Metode Seleksi dataset 1



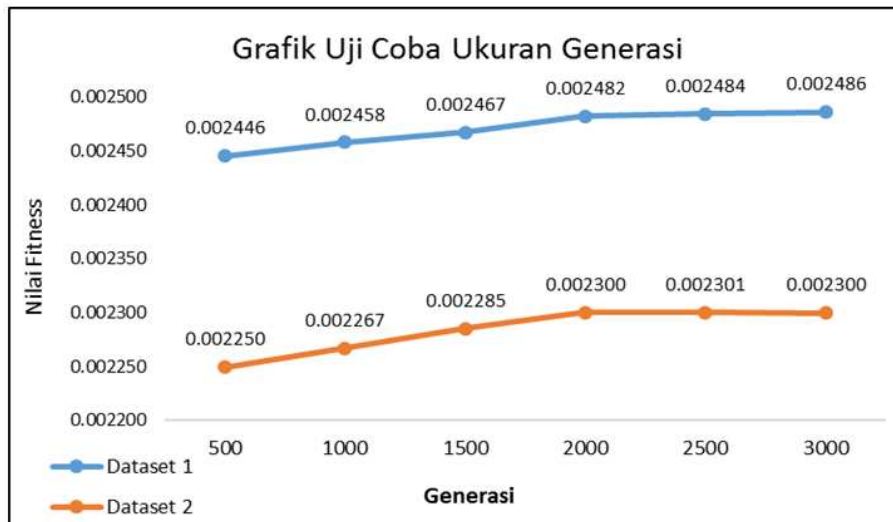
Gambar 3. Grafik Perbandingan Metode Seleksi dataset 2

Uji coba menggunakan dua jenis dataset yang berbeda bisa disimpulkan bahwa seleksi dengan menggunakan metode elitis lebih sesuai untuk masalah *traveling salesman problem* dengan *time windows* karena pada metode seleksi elitis semua individu yang bagus langsung dipilih menjadi indukan yang akan di proses pada generasi selanjutnya, sedangkan dengan metode *roulette wheel* individu yang belum bagus masih punya kesempatan untuk menjadi induk untuk generasi selanjutnya, sehingga nilai *fitness* yang dihasilkan pada setiap percobaan cenderung tidak stabil dan tidak optimal.

4.2. Hasil dan Analisa Uji Coba Ukuran Generasi

Uji coba yang kedua dilakukan pencarian ukuran generasi yang optimal. Setiap ukuran generasi dilakukan 20 kali percobaan dengan ukuran populasi adalah 40 populasi, ukuran generasi adalah kelipatan 500 mulai dari 500 sampai 3000 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,3. Metode seleksi yang digunakan adalah elitis.

Pada Gambar 4 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 with time windows* adalah 2000 generasi, dikarenakan dengan ukuran generasi 2000 sampai 3000 generasi, nilai *fitness* yang dihasilkan tidak berbeda jauh dan cenderung membentuk garis lurus.



Gambar 4. Grafik rata-rata nilai fitness tiap generasi dataset 1

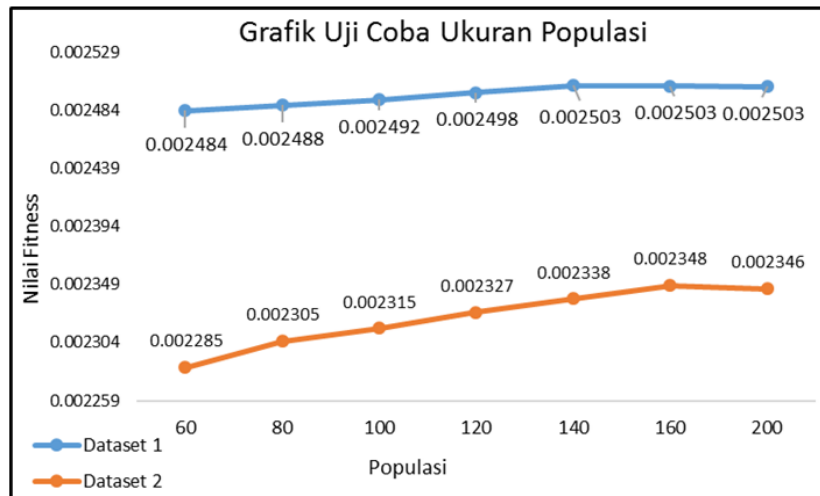
Terlalu banyak ukuran generasi belum tentu membuat algoritma genetika menjadi lebih optimal. Selain waktu proses yang menjadi lebih lama hasil nilai *fitness* yang dihasilkan juga belum tentu jauh lebih baik dari generasi yang lebih rendah. Ukuran generasi yang tinggi akan mengakibatkan proses evolusi 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.

5.3. Hasil dan Analisa Uji Coba Ukuran Populasi

Uji coba yang ketiga dilakukan pencarian ukuran populasi yang optimal. Setiap ukuran populasi dilakukan 20 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 5 bisa dilihat kenaikan signifikan rata-rata nilai *fitness* untuk 20 kali percobaan pada *dataset 1* 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 *TSPTW* pada dataset 1. Untuk dataset 2 bisa dilihat kenaikan signifikan rata-rata nilai *fitness* untuk 20 kali percobaan mulai dari ukuran populasi 60 sampai dengan populasi 160 populasi, namun pada ukuran populasi 160 sampai 200 populasi sudah tidak terjadi perubahan yang cukup signifikan dan pada grafik cenderung membentuk garis lurus. 160 populasi merupakan ukuran populasi yang optimal untuk masalah *traveling salesman problem with time windows* pada dataset 2. Dilihat dari perbedaan ukuran populasi

yang optimum dari hasil percobaan dengan dua jenis *dataset* yang berbeda maka bisa disimpulkan bahwa perbedaan *dataset* yang digunakan dapat mempengaruhi nilai *fitness* dan parameter genetika optimum yang dihasilkan oleh algoritma genetika.



Gambar 5. Grafik hasil uji coba ukuran populasi dataset 1

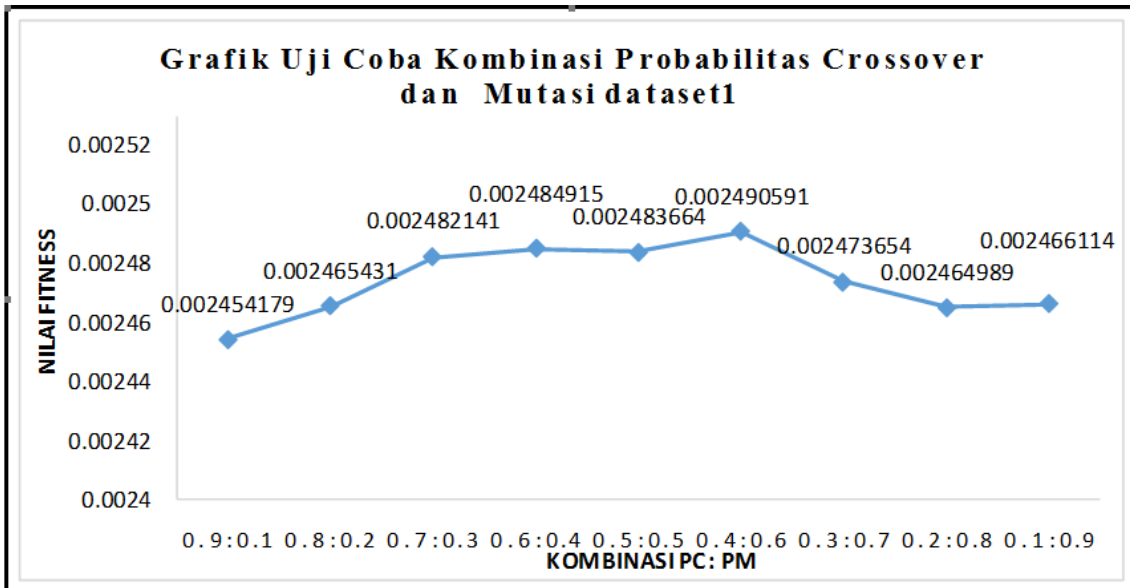
Semakin tinggi ukuran populasi maka berpengaruh pada rata-rata nilai *fitness* yang didapatkan namun semakin tinggi ukuran populasi juga berpengaruh pada waktu pemrosesan algoritma genetika akan semakin lama, sehingga menentukan berapa ukuran populasi yang optimal sangat diperlukan. Ukuran populasi yang optimal untuk setiap *dataset* dapat berbeda-beda seperti pada grafik Gambar 5. 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).

5.4. Hasil dan Analisa Uji Coba Kombinasi Probabilitas *Crossover* dan Mutasi

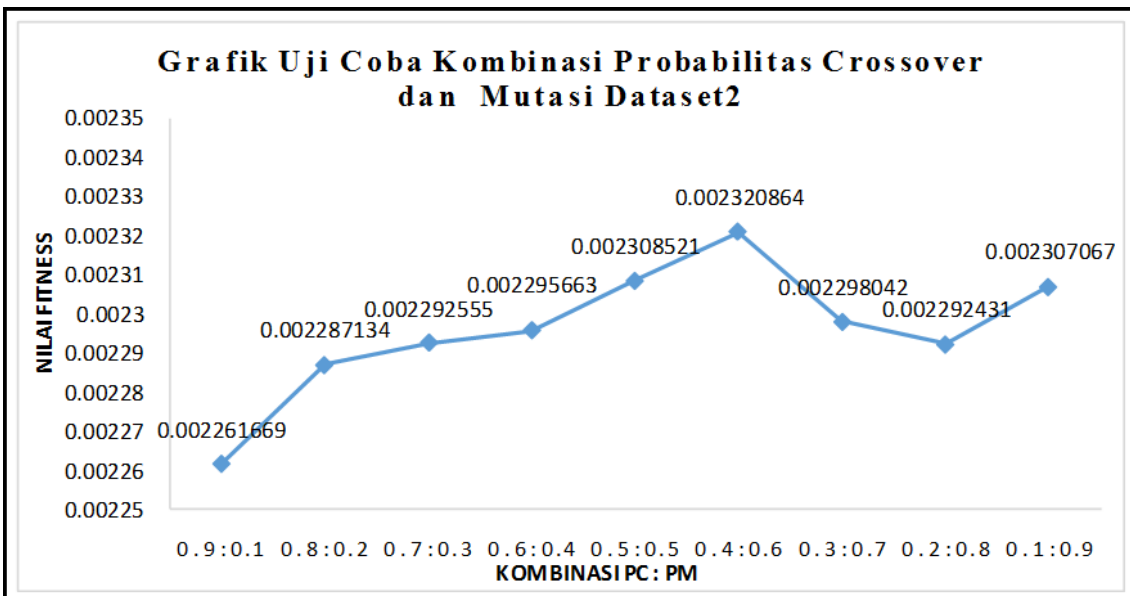
Uji coba yang keempat dilakukan pencarian kombinasi probabilitas *crossover* dan mutasi yang optimal. Penentuan kombinasi tingkat *crossover* dan mutasi sangat penting dilakukan untuk memperoleh solusi yang baik (mendekati optimum). Tingkat *crossover* yang terlalu besar (dan tingkat mutasi kecil) akan menghilangkan kesempatan algoritma genetika untuk mengeksplorasi daerah pencarian baru. Pada kondisi sebaliknya (tingkat *crossover* besar, tingkat mutasi kecil), algoritma genetika tidak mampu mengeksplorasi daerah optimum lokal (Mahmudy, 2012) (Mahmudy, 2013b).

Setiap kombinasi probabilitas *crossover* dan mutasi dilakukan 20 kali percobaan. Ukuran populasi adalah 40 populasi, ukuran generasi adalah 1500 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,1 : 0,9 sampai 0,9 : 0,1. Metode seleksi yang digunakan adalah elitis. Hasil uji coba bisa dilihat pada grafik Gambar 6 untuk dataset 1, dan Gambar 7 untuk dataset 2.

Berdasarkan Gambar 6 dan Gambar 7 rata-rata nilai *fitness* terbesar terdapat pada kombinasi probabilitas *crossover* dan probabilitas mutasi 0,4:0,6 dengan rata-rata nilai *fitness* adalah 0,00249 untuk dataset 1, dan 0,00232 untuk dataset 2. Berdasarkan Gambar 6 dan Gambar 7 bisa dilihat rata-rata nilai *fitness* semakin ketengah maka grafik semakin memuncak dengan kombinasi probabilitas *crossover* 0,4 dan probabilitas mutasi sebagai puncak tertingginya. Berarti kombinasi probabilitas *crossover* dan probabilitas mutasi yang terbaik untuk menyelesaikan masalah *TSP-TW* adalah 0,4:0,6.



Gambar 6. Grafik hasil uji coba kombinasi probabilitas *crossover* dan mutasi dataset 1



Gambar 7. Grafik hasil uji coba kombinasi probabilitas *crossover* dan mutasi dataset 2

Perbedaan nilai probabilitas *crossover* dan mutasi mempengaruhi nilai *fitness* yang dihasilkan. Semakin besar nilai probabilitas *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.

6. 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 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 (Mahmudy, 2013b).

Referensi

- Anonimous. 2013. *Antar Jemput Laundry Kiloan*. “Dari Jasa Antar Jemput Secuter”, (Online), (<http://antarjemputsecuter.wordpress.com>, diakses 19 April 2014)
- 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*.
- Chen, T., & Zhou, G. 2013. “Vehicle Routing Optimization Problem With Time-Windows And Its Solution By Genetic Algorithm”. *Journal of Digital Information Management Volume 11 Number 2 April 2013*: ZhejiangGongshangUniversity.
- 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.
- Mahmudy, WF, Marian, RM & Luong, LHS 2012, 'Solving part type selection and loading problem in flexible manufacturing system using real coded genetic algorithms – Part II: optimization', *International Conference on Control, Automation and Robotics, Singapore*, 12-14 September 2012, World Academy of Science, Engineering and Technology, pp. 706-710,
- Mahmudy, F. W. 2013a.”*Algoritma Evolusi*”.2013a.Malang: Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- 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.
- 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.
- Nugraha, I. 2008.”*Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar*”. Institut Teknologi Bandung. Institut Teknologi Bandung.
- 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.
- 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.
- Widodo, AW & Mahmudy, WF 2010. 'Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner', *Kursor*, vol. 5, no. 4, pp. 205-211.
- Zukhri, Z. 2004. “Penyelesaian Masalah Penugasan dengan Algoritma Genetika”. *Seminar Nasional Aplikasi Teknologi Informasi*, Yogyakarta : Universitas Islam Indonesia.