

PENYELESAIAN MASALAH POHON RENTANG MINIMUM DENGAN ALGORITME GENETIKA YANG DISEDERHANAKAN SUATU ADOPSI OPERATOR DARI SISTEM KEKEBALAN BUATAN

Zainudin Zukhri

*Teknik Informatika, Fakultas Teknologi Industri Universitas Islam Indonesia
Jl. Kaliurang Km 14.5 Yogyakarta
Email : zainudin@uii.ac.id*

Abstrak. Masalah pohon rentang minimum adalah bagaimana menemukan busur-busur pembentuk pohon dalam suatu jaringan yang menghubungkan semua simpul dengan jumlah bobot busur sekecil-kecilnya. Masalah ini merupakan bentuk pemodelan masalah praktis, seperti minimasi kabel instalasi listrik, minimasi kabel serat optik jaringan telekomunikasi, minimasi pipa instalasi air bersih dan sebagainya. Masalah ini memegang peranan penting dalam kehidupan nyata, sehingga usaha untuk memperbaiki kualitas penyelesaiannya merupakan kajian yang dilakukan, baik bersifat konseptual maupun aplikatif. Penelitian yang diusulkan merupakan penelitian konseptual yang bertujuan untuk memperbaiki metode penyelesaian masalah pohon rentang minimum dengan Algoritme Genetika. Gagasan perbaikan metode didapat dari adanya kesamaan prinsip algoritme ini dengan Sistem Kekebalan Buatan, yang berupa penyederhanaan langkah-langkah dalam Algoritme Genetika. Penyederhanaan yang diusulkan adalah pengabaian operator penyilangan sebagaimana yang berlaku pada Sistem Kekebalan buatan. Selain itu, penyederhanaan juga dilakukan dari aspek representasi kromosom. Usulan telah diujikan dengan menggunakan data pengujian penelitian lain yang sudah diketahui solusinya. Hasil penelitian menunjukkan bahwa penyederhanaan Algoritme Genetika yang diusulkan tidak mengurangi tingkat kualitas solusi yang dihasilkan. Hal ini berarti adanya perbaikan kinerja algoritme dari aspek waktu komputasi dan memori komputer yang dibutuhkan untuk proses komputasi. Penelitian ini dapat dikembangkan lagi di masa yang akan datang, baik dengan data pengujian yang lebih kompleks maupun dengan penyederhanaan penanganan parameter algoritma.

Kata kunci: Algoritme Genetika, penyederhanaan, pohon rentang minimum.

1. Pendahuluan

Masalah pohon rentang minimum (PRM) adalah salah satu masalah dalam jaringan yang terdiri dari banyak simpul yang dihubungkan oleh busur-busur berbobot. Yang harus ditemukan dalam masalah ini adalah busur-busur pembentuk pohon yang menghubungkan semua simpul dengan jumlah bobot busur penghubung sekecil-kecilnya[1]. Masalah ini merupakan bentuk pemodelan masalah praktis, seperti penghematan kabel penghubung dalam instalasi listrik, penghematan kabel fiber optik dalam jaringan telekomunikasi, minimasi panjang pipa dalam jaringan air bersih dan sebagainya.

Mengingat masalah ini memegang peranan penting dalam kehidupan nyata, maka upaya penyempurnaan pendekatan untuk menyelesaikannya merupakan kajian yang banyak dilakukan, seperti penelitian Dewi, Riska, Hidayat dan Hadiyanto [2, 3, 4, 5]. Penelitian Dewi dan Riska sama sekali tidak berusaha untuk menghasilkan memperbaiki kinerja metode penyelesaian karena hanya bersifat komparasi dari beberapa metode [2, 3], sedangkan penelitian Hidayat memang telah berusaha menyempurnakan bentuk representasi kromosom yang digunakan oleh peneliti sebelumnya dalam Gen dan Cheng [6] dan berhasil menawarkan solusi dengan penerapan Algoritme Genetika (AG) dengan representasi kromosom yang lebih mudah[4], tetapi penggunaan representasi kromosom dalam penelitian ini masih bisa disederhanakan. Adapun penelitian Hadiyanto hanya sekedar penerapan dari Algoritme Prim untuk menemukan kemungkinan pembentukan pohon alternatif dengan mempertimbangkan kontur geografis dan tidak ditujukan untuk meningkatkan kinerja dari metode yang digunakan [5].

Dalam penelitian akan diusulkan adanya perbaikan metode penyelesaian masalah PRM. Usulan metode akan diuji melalui eksperimen yang ditujukan untuk mengetahui sejauhmana pengaruh penyederhanaan yang diusulkan terhadap peningkatan kinerja AG. Bentuk perbaikan yang diusulkan adalah penyederhanaan operator dalam AG dengan mengadopsi operator dalam SKB.

Kajian awal yang dilakukan adalah dengan memperhatikan kesamaan antara AG dan SKB. Dalam AG dikenal 3 (tiga) operator yang paling sering digunakan, yaitu operator seleksi, operator penyilangan dan operator mutasi [7]. Sementara itu dalam SKB dikenal adanya operator seleksi, operator kloning dan operator mutasi [8]. Jika tidak diperhatikan secara seksama, maka operator dalam kedua algoritme terlihat hampir sama satu dengan yang lainnya. Padahal jika dilakukan perbandingan operator kloning dan operator seleksi, ternyata terdapat kemiripan, yaitu operator kloning dapat dikatakan termasuk dalam operator seleksi [9]. Dengan memasukkan operator kloning sebagai salah satu metode seleksi, maka dapat dikatakan bahwa dalam SKB hanya terdapat dua operator, yaitu operator seleksi dan operator mutasi, sehingga yang membedakan antara AG dan SKB adalah operator penyilangan. Berdasarkan pertimbangan bahwa SKB dapat digunakan tanpa adanya operator penyilangan, maka seharusnya akan dapat berlaku hal yang sama jika operator penyilangan ini diabaikan saja dalam AG. Analogi inilah landasan usulan untuk menghilangkan operator penyilangan dalam AG.

Adapun hipotesis dalam penelitian ini adalah AG yang disederhanakan akan mempunyai kinerja yang sama dengan AG sebelum dilakukan penyederhanaan. Untuk memudahkan penyebutan, selanjutnya akan digunakan istilah AG murni untuk algoritme sebelum disederhanakan atau disingkat dengan GA_0 dan AG yang disederhanakan untuk algoritme yang diusulkan atau disingkat dengan GA_1 .

2. Metode Penelitian

Untuk menyelesaikan penelitian ini dilakukan tahapan penelitian sebagai berikut:

- a. Melakukan kajian komparatif antara AG dan SKB.
- b. Mendesain model penyederhanaan dalam AG untuk menyelesaikan masalah PRM.
- c. Mengimplementasikan desain model dalam bentuk perangkat lunak simulasi.
- d. Melakukan pengujian experimental terhadap perangkat lunak simulasi yang telah dikembangkan dan melakukan analisis hasil.
- e. Melakukan studi komparasi pengaruh model yang diusulkan terhadap kinerja AG. Dalam studi komparasi ini digunakan pengujian statistik untuk menganalisis data sampel yang didapat dari eksperimen sebagaimana direkomendasikan oleh [10].

2.1. Struktur Algoritme Genetika

Pada prinsipnya struktur dasar GA_0 yang akan diuji sama dengan struktur dasar AG sebagaimana dibahas oleh Zuhri [7]. Adapun penyederhanaan yang diterapkan pada GA_0 secara prosedural adalah diabaikannya proses manipulasi kromosom dalam operator genetika, yaitu proses penyilangan. Pengabaian operator penyilangan ini diadopsi dari tidak adanya proses manipulasi representasi masalah dalam SKB. Di samping itu, dalam pengujian algoritme ini pengguna memang diperbolehkan untuk memilih probabilitas penyilangan sebesar 0% [11]. Dengan kata lain, GA_1 dapat dikatakan diuji dengan probabilitas penyilangan sebesar 0% secara permanen.

2.2. Representasi kromosom

Penelitian ini mengusulkan representasi kromosom berbentuk himpunan bilangan yang digunakan sebagai pemilih urutan simpul berikutnya dalam proses pembentukan pohon. Setiap gen dalam kromosom merupakan bilangan yang dapat bernilai pada interval 0 sampai 100. Bilangan yang merupakan gen dalam kromosom tidak secara langsung merupakan indeks dari simpul yang dipilih, tetapi merupakan kode dari indeks pasangan simpul yang visibel untuk membentuk pohon. Representasi kromosom dalam penelitian ini diperlihatkan pada Gambar 1.

g_1	g_2	g_3	.	.	.	.	g_n
-------	-------	-------	---	---	---	---	-------

Dengan

- g_i adalah bilangan yang dapat bernilai 0 sampai 100
- n adalah cacah simpul dalam jaringan

Gambar 1. Representasi Kromosom yang Diusulkan

Secara singkat pendekodean kromosom menjadi solusi atau prosedur pembentukan pohon dimulai dari pembentukan himpunan busur visibel untuk membentuk pohon yang dilanjutkan dengan memilih satu di antara busur-busur visibel tersebut berdasarkan bilangan pada gen yang bersesuaian. Mengingat pada pemilihan yang pertama semua busur dalam jaringan dapat dipilih, maka khususnya pada pemilihan pertama ini yang dipilih adalah simpul jaringan. Hal ini dimaksudkan untuk penghematan proses, karena cacah semua busur pembentuk jaringan pasti lebih banyak dari pada banyak simpul. Hanya gen yang pertama ini saja yang digunakan untuk memilih simpul, sedangkan gen selanjutnya digunakan untuk memilih busur. Dengan demikian sangat beralasan jika cacah gen dalam kromosom tidak sebanyak cacah busur, tetapi sebanyak cacah simpul dalam jaringan.

Dengan prinsip gen-gen adalah bilangan pemilih busur, maka proses pembentukan busur sebenarnya dilakukan secara bertahap. Pada setiap tahapan dibutuhkan proses untuk menyisihkan busur-busur yang tidak visibel karena busur ini jika terpilih akan membentuk siklus (kecuali untuk tahapan pertama). Untuk memudahkan proses pembentukan himpunan busur yang visibel, maka busur visibel dibatasi pada busur yang salah satu simpulnya sudah membentuk pohon sementara.

2.3. Operator Algoritme Genetika

Operator AG berfungsi untuk mendapatkan kualitas solusi dari titik-titik pencarian yang disajikan oleh populasi kromosom (yaitu operator evaluasi) dan untuk memanipulasi populasi agar berubah nilainya (yaitu operator seleksi, operator penyilangan dan operator mutasi). Untuk operator yang digunakan untuk perubahan nilai populasi, terdapat beberapa variasi metode yang dapat dipilih untuk diterapkan dalam penelitian ini [12, 13, 14]. Meskipun demikian, mengingat eksperimen yang akan dilakukan dalam penelitian ini adalah untuk mengetahui pengaruh penyederhanaan operator GA_0 , maka sembarang metode untuk setiap operator dapat dipilih, dengan syarat untuk kedua algoritme yang diuji digunakan metode yang sama. Adapun metode yang akan digunakan diperlihatkan pada Tabel 1.

2.4. Parameter Algoritme Genetika

Parameter AG yang akan digunakan dalam eksperimen diperlihatkan pada Tabel 2. Sebagaimana halnya pemilihan metode yang digunakan dalam operator algoritme, penentuan nilai parameter algoritme dilakukan secara sembarang tetapi masih mempertimbangkan nilai-nilai yang biasa digunakan dalam pengujian AG dengan syarat untuk kedua algoritme yang diuji digunakan parameter algoritme yang sama.

2.5. Metode Pengukuran Kualitas Solusi

Definisi kualitas solusi yang dihasilkan dalam penelitian ini dilakukan dengan mempertimbangkan hal-hal sebagai berikut:

- Konsep kualitas solusi harus dapat diberlakukan secara umum untuk semua data pengujian dan harus dapat mendukung pengujian hipotesis yang sudah ditentukan.
- Posisi nilai penyelesaian masalah PRM yang mungkin dalam sumbu koordinat kartesian. Dalam hal ini solusi PRM selalu positif. Solusi yang dihasilkan oleh algoritme pencarian untuk selanjutnya disebut Solusi.
- Algoritme pencarian dikatakan dapat menemukan solusi yang paling berkualitas jika dapat menemukan solusi yang sudah diketahui. Solusi yang sudah diketahui untuk selanjutnya disebut Solusi Minimum.
- Solusi yang paling rendah kualitasnya adalah kandidat solusi dalam proses komputasi yang nilainya paling besar. Nilai ini untuk selanjutnya disebut Solusi Maksimum.

Dengan mempertimbangkan beberapa hal di atas, maka kualitas solusi didefinisikan sebagai jarak pergerakan proses pencarian dari titik terjauh dari solusi menuju ke solusi yang ditemukan dibagi dengan jarak pergerakan proses pencarian terbesar atau seperti ditunjukkan oleh persamaan (1)

$$\text{Akurasi} = \frac{\text{Solusi Maksimum} - \text{Solusi yang dihasilkan}}{\text{Solusi Maksimum} - \text{Solusi Minimum}} \cdot 100\% \quad (1)$$

Tabel 1. Operator Algoritme Genetika yang Digunakan

Operator	Metode
Seleksi	Kombinasi Roda Rolet dan Elitisme
Penyilangan	Penyilangan satu-titik potong
Mutasi	Konversi nilai dalam interval yang diijinkan

Tabel 2. Parameter Algoritma Genetika yang Digunakan

Parameter Algoritma	Nilai
Probabilitas Penyilangan	50%
Probabilitas Mutasi	1%
Ukuran Populasi	20
Banyak Generasi	500

2.6. Metode Pengujian Hipotesis

Dengan mempertimbangkan konsep hipotesis yang sudah ditentukan pada bab sebelumnya dan konsep pengukuran kinerja pada bagian sebelumnya, maka hipotesis yang diuji berdasarkan data statistik adalah kualitas solusi yang dihasilkan oleh GA₁ diduga sama dengan kualitas solusi yang dihasilkan GA₀ yang dinotasikan sebagai persamaan (2) dan persamaan (3).

$$H_0: \mu_1 = \mu_2 \quad (2)$$

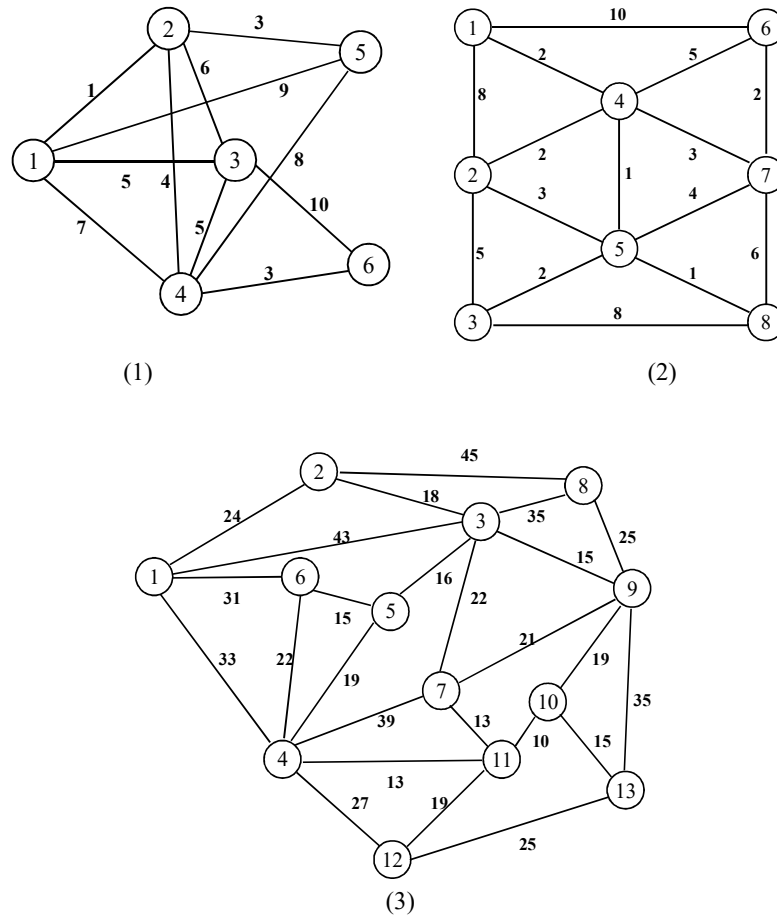
$$H_1: \mu_1 \neq \mu_2 \quad (3)$$

Berdasarkan bentuk kedua hipotesis sebagaimana tersebut di atas, maka metode pengujian hipotesis yang akan diterapkan adalah uji t dua-sampel. Dalam pengujian ini hipotesis nol (H_0) gagal ditolak jika nilai t yang dihitung berdasarkan data statistik (t statistik) berada pada interval t kritis pada kedua sisi dan sebaliknya jika nilai t statistik berada di luar interval t kritis pada kedua sisi, maka hipotesis alternatif (H_1) akan diterima. Untuk memudahkan perhitungan yang dibutuhkan dalam kedua uji t di atas, maka dalam penelitian ini digunakan fitur analisis data yang terdapat di dalam perangkat lunak Microsoft Excel[15].

2.7. Data Pengujian

Data yang digunakan untuk pengujian kinerja algoritme yang diusulkan dalam penelitian ini adalah data-data yang digunakan dalam penelitian-penelitian terkait sebelumnya. Pertimbangan utama dalam penggunaan data pengujian ini berhubungan dengan konsep pengukuran kualitas sebagaimana telah dijelaskan sebelumnya, yaitu perlu digunakan data pengujian yang sudah diketahui solusi optimalnya. Berikut adalah 3 (tiga) data jaringan yang akan diujikan dalam penelitian ini beserta nilai optimal solusinya:

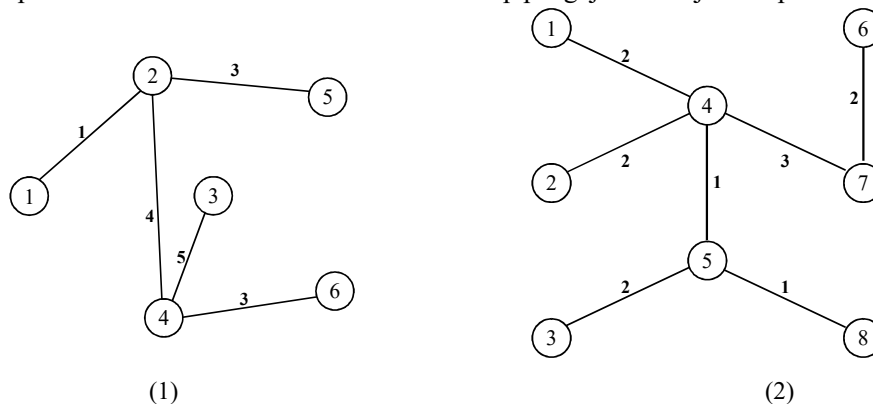
- Dua data jaringan yang telah diuji oleh Hidayat sebagaimana terlihat pada Gambar 2 (1) dengan nilai solusi optimal sebesar 16[4].
- Dua data jaringan yang telah diuji oleh Hidayat sebagaimana terlihat pada Gambar 2 (2) dengan nilai solusi optimal sebesar 13[4].
- Data jaringan Setiawan dan Wibowo sebagaimana terlihat pada Gambar 2 (3) dengan nilai solusi optimal sebesar 202[16].

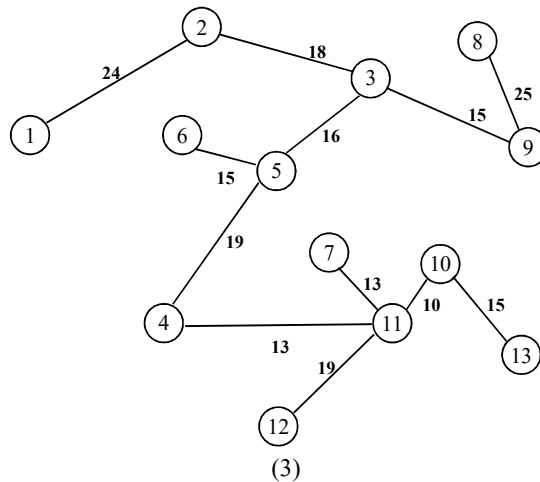


Gambar 2. Jaringan-Jaringan yang Diujikan^[4, 16]

3. Hasil Dan Pembahasan

Eksperimen yang sudah dilakukan untuk menguji GA_0 dan GA_1 ternyata berhasil menemukan solusi terbaik berbentuk PRM yang sama. Solusi terbaik tersebut didapat dari 7 (tujuh) pengujian untuk setiap data pengujian yang menghasilkan nilai solusi bervariasi, kecuali GA_0 pada pengujian data pertama. Eksperimen membuktikan bahwa kedua algoritme dapat menemukan solusi berturut-turut untuk ketiga data pengujian sebesar 16, 13 dan 202. PRM yang dihasilkan dari eksperimen ditunjukkan pada Gambar 3 dan nilai akurasi untuk setiap pengujian ditunjukkan pada Tabel 3.





Gambar 3. Pohon Rentang Minimum yang Ditemukan dari Pengujian

Tabel 3. Hasil Pengujian

Data Pertama			Data Kedua			Data Ketiga		
No	Akurasio	Akurasi	No	Akurasio	Akurasi	No	Akurasio	Akurasi
1	100,00	100,00	1	100,00	100,00	1	86,15	100,00
2	100,00	100,00	2	100,00	92,31	2	94,62	95,38
3	100,00	95,45	3	88,46	88,46	3	90,00	87,69
4	100,00	100,00	4	88,46	100,00	4	93,08	100,00
5	100,00	100,00	5	100,00	100,00	5	80,00	93,08
6	100,00	100,00	6	100,00	100,00	6	90,00	100,00
7	100,00	100,00	7	100,00	100,00	7	100,00	95,38

Untuk pengujian menggunakan data pertama terdapat perbedaan rerata kinerja dari kedua algoritme sebesar 100% dan 99,35%. Hasil uji t dua-sampel menghasilkan perbandingan nilai t statistik dan nilai t kritis pada kedua sisi yang menunjukkan $t_{\text{statistik}}$ berada pada interval kedua sisi t_{kritis} ($t_{\text{kritis}} < t_{\text{statistik}} < t_{\text{kritis}}$), sehingga $H_0: \mu_1 = \mu_2$ diterima. Jadi untuk data pertama perbedaan rerata kinerja sampel tidak cukup signifikan untuk membuktikan perbedaan kinerja kedua algoritme. Untuk pengujian menggunakan data kedua terdapat perbedaan rerata kinerja dari kedua algoritme sebesar 96,70% dan 97,25%. Hasil uji t dua-sampel menghasilkan perbandingan nilai t statistik dan nilai t kritis pada kedua sisi yang juga menunjukkan $t_{\text{statistik}}$ berada pada interval kedua sisi t_{kritis} ($-t_{\text{kritis}} < t_{\text{statistik}} < t_{\text{kritis}}$), sehingga $H_0: \mu_1 = \mu_2$ juga diterima. Jadi untuk data kedua perbedaan rerata kinerja sampel tidak cukup signifikan untuk membuktikan perbedaan kinerja kedua algoritme. Untuk pengujian menggunakan data ketiga terdapat perbedaan rerata kinerja dari kedua algoritme sebesar 90,55% berbanding 95,93%. Hasil uji t dua-sampel menghasilkan perbandingan nilai t statistik dan nilai t kritis pada kedua sisi untuk pengujian dengan data ketiga, ternyata juga menunjukkan $t_{\text{statistik}}$ berada pada interval kedua sisi t_{kritis} ($t_{\text{kritis}} < t_{\text{statistik}} < t_{\text{kritis}}$), sehingga $H_0: \mu_1 = \mu_2$ juga diterima. Jadi untuk data ketiga perbedaan rerata kinerja sampel juga tidak cukup signifikan untuk membuktikan perbedaan kinerja kedua algoritme. Ringkasan secara keseluruhan hasil uji t dua-sampel diperlihatkan pada Tabel 4. Berdasarkan tabel ini dapat diketahui dengan jelas bahwa semua perbedaan rerata nilai solusi yang dihasilkan dalam eksperimen, ternyata tidak cukup signifikan untuk membuktikan bahwa terdapat perbedaan kinerja dari kedua algoritme yang diujikan.

Tabel 4. Ringkasan Hasil Uji t Dua Sampel

No	Uji t dua sampel pada α sebesar 0,01		Hasil uji t dua sampel
	Nilai t statistik	Nilai t kritis 2-sisi	
1	1,00000	3,707428	$H_0: \mu_1 = \mu_2$ diterima
2	-0,19612	3,054539	$H_0: \mu_1 = \mu_2$ diterima
3	-181481	3,105807	$H_0: \mu_1 = \mu_2$ diterima

Dengan demikian dapat disimpulkan bahwa hasil uji t dua sampel yang menunjukkan bahwa penyederhanaan AG dengan menghilangkan operator penyilangan terbukti tidak mengurangi kinerjanya.

4. Simpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan dapat disimpulkan bahwa penyederhanaan AG tidak mengurangi kualitas solusi yang dihasilkan, khususnya untuk menyelesaikan masalah PRM. Algoritme belum diujikan untuk data dengan tingkat kerumitan yang lebih tinggi, sehingga terdapat kemungkinan kinerja algoritme menjadi menurun pada karakteristik data masukan yang mempunyai tingkat kerumitan lebih dari data ketiga dalam pengujian (cacah simpul lebih dari 20 dan atau cacah busur lebih dari 50). Penyederhanaan AG dengan mengabaikan operator penyilangan juga membawa dampak kemudahan bagi pengguna pada waktu penentuan parameter algoritme, yaitu eksekusi AG tanpa parameter probabilitas penyilangan. Pada masa yang akan datang akan lebih sederhana lagi, jika AG dapat dijalankan tanpa parameter probabilitas mutasi.

Daftar Pustaka

- [1]. Sedgewick, R. dan Wayne, K, 2016, Algorithms, <http://algs4.cs.princeton.edu/home/>, diakses pada tgl 13 Juni 2016.
- [2]. Dewi, Y.A.S., 2014, Perbandingan Algoritma Prim dan Kruskal Untuk Menyelesaikan Masalah Minimum Spanning Tree. Laporan Penelitian, tidak dipublikasikan. Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer, AMIKOM.
- [3]. Riska., 2014, *Perbandingan Algoritma Kruskal Dengan Algoritma Genetika Dalam Penyelesaian Masalah Minimum Spanning Tree (MST)*. Skripsi, tidak dipublikasikan. Bengkulu: Fakultas Teknik, Universitas Bengkulu.
- [4]. Hidayat, T., *Algoritma Genetik untuk Pemecahan Persoalan Minimum Spanning Tree*. Proc. Semnas Aplikasi Sistem Cerdas dalam Rekayasa dan Bisnis, FTI UII Juli 2000.
- [5]. Hadiyanto, 2012, Penentuan Pohon Rentang Minimum Pada Distribusi Jaringan Listrik Berdasarkan Kondisi Geografis Suatu Wilayah Dengan Algoritma Prim Studi Kasus: Jaringan Listrik Distribusi Primer Kota Samarinda. Foristek 2(1), 153-164.
- [6]. Gen, M. dan Cheng, R, 1997, *Genetic Algorithms and Engineering Optimization*. Canada: John Wiley dan Sons. Inc.
- [7]. Zukhri, Z., 2014, *Algoritma Genetika, Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Penerbit Andi Yogyakarta.
- [8]. Botu, S.S., & Barai, S.V., *Structural Optimization Using Artificial Immune System*. Proc. Blucher Mechanical Engineering Sao Paulo 2014.
- [9]. Vandering, E.W, 2016, Lecture Note: Genetic Algorithms and Genetic Programming, http://docdb.fnal.gov/FOCUS/DocDB/0001/000159/003/gp_lecture_1.pdf, diakses pada tgl 13 Juni 2016.
- [10]. Bartz-Beielstein, T., Chiarandini, M., Paquete, L. & Preuss, M., 2010, *Experimental Methods for the Analysis of Optimization Algorithms*. New York: Springer.
- [11]. Obitko, M., 1998, Genetic Algorithms, <http://www.obitko.com/tutorials/genetic-algorithms/index.php> diakses pada tgl 5 Nopember 2016.
- [12]. Razali, N.M. dan Geraghty, J., Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. Proc. the World Congress on Engineering 2011. London: Juli 2011.
- [13]. Soni, N. dan Kumar, T., 2014, Study of Various Crossover Operators in Genetic Algorithms. International Journal of Computer Science and Information Technologies 5 (6), 7235-7238.

- [14]. Hassanat, A.B.A., Alkafaween, E., Al-Nawaiseh, N.A., Abbadi, M.A., Alkasassbeh, M., dan Alhasanat, M.B., 2016, Enhancing Genetic Algorithms using Multi Mutations: Experimental Results on the Travelling Salesman Problem. *International Journal of Computer Science and Information Security* 14(7), 785-801.
- [15]. Zaiontz, C., 2013, Real Statistics Using Excel, <http://www.real-statistics.com/> diakses pada tgl 1 Nopember 2016.
- [16]. Setiawan, S. & Wibowo, A.M., 1999, Struktur Data Graf: Representasi & Algoritma-algoritma, <http://aren.cs.ui.ac.id/sda/archive/1998/handout/handout20.html> diakses pada tgl 31 Oktober 2016.