

ALGORITMA GENETIKA DENGAN PENDEKATAN MODEL PULAU PADA PERMASALAHAN *TRAVELLING SALESMAN*

Hardy

STMIK Mikroskil

Jl. Thamrin No. 112, 124, 140 Medan 20212

hardy@mikroskil.ac.id

Abstrak

Algoritma genetika telah banyak digunakan untuk menyelesaikan permasalahan-permasalahan susah (*hard problem*) atau permasalahan NP-Complete karena sifatnya yang heuristik dan mampu menghasilkan solusi yang optimal dalam waktu yang relatif cepat. Performansi dari algoritma genetika dalam menyelesaikan permasalahan tersebut dilihat dari seberapa cepat algoritma genetika mencapai solusi yang optimal atau seberapa luas penjelajahan algoritma genetika dalam ruang solusi yang luas (tingkat diversitas yang tinggi).

Algoritma genetika dengan pendekatan model pulau merupakan pendekatan paralel dimana beberapa instan algoritma genetika dijalankan secara bersamaan untuk menjelajahi ruang solusi mencari solusi optimal. Dengan menggunakan model pulau, algoritma genetika bisa dikembangkan lebih lanjut dalam hal peningkatan diversitas individu dalam populasi. Diversitas yang tinggi bisa meningkatkan peluang algoritma genetika untuk mendapatkan solusi yang paling optimal atau *global optimum*.

Di dalam penelitian ini digunakan studi kasus permasalahan *travelling salesman* untuk membandingkan algoritma genetika dengan konfigurasi jumlah pulau dan metode pertukaran informasi antar pulau yang berbeda. Kontribusi dari penelitian ini adalah untuk menghasilkan korelasi antara jumlah pulau yang digunakan dengan performansi algoritma genetika.

Kata kunci : *algoritma genetika, travelling salesman problem, model pulau, paralel*

1. Pendahuluan

Permasalahan *travelling salesman* merupakan permasalahan *NP-Complete* yang berarti tidak memiliki algoritma dengan solusi yang optimal. Di dalam permasalahan *travelling salesman*, sebuah solusi optimal dibutuhkan dimana seorang *salesman* harus mengelilingi n kota yang setiap kota hanya boleh ditempuh satu kali saja. Untuk menyelesaikan permasalahan tersebut diperlukan algoritma heuristik dikarenakan tingginya ruang solusi yang ada. Salah satu dari algoritma heuristik adalah algoritma genetika.

Algoritma genetika diciptakan oleh John Holland pada tahun 1960-an dengan tujuan untuk mempelajari dan mensimulasikan fenomena adaptasi di alam ke dalam komputer [1]. Sekarang, algoritma genetika yang digunakan dalam komputasi evolusi sangat berbeda dengan algoritma genetika yang diciptakan oleh John Holland pada masanya. Algoritma genetika sekarang digunakan untuk memecahkan permasalahan komputasional yang melibatkan pencarian solusi di ruang solusi yang sangat besar.

Di dalam penelitian ini, algoritma genetika dengan menggunakan model pulau diusulkan untuk menyelesaikan permasalahan *travelling salesman*. Algoritma genetika pada umumnya memiliki tingkat diversitas yang rendah sehingga untuk menutupi kelemahan tersebut digunakanlah model pulau. Model pulau merupakan model paralel dimana beberapa

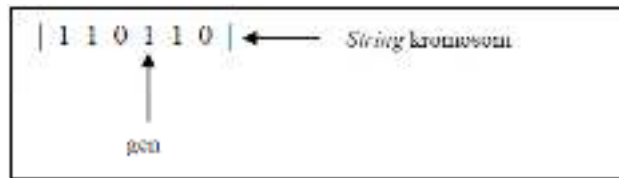
instan algoritma genetika dijalankan sekaligus dan saling menukar informasi setelah melewati sekian *epoch* secara periodik.

Kontribusi dari penelitian ini adalah pencarian korelasi antara jumlah pulau optimal serta metode pertukaran informasi optimal dengan performansi algoritma genetika melalui serangkaian pengujian yang melibatkan konfigurasi yang berbeda.

2. Kajian Pustaka

2.1 Algoritma Genetika

Algoritma genetika merupakan metode pencarian yang disesuaikan dengan proses genetika dari organisme-organisme biologi yang berdasarkan teori Charles Darwin [2]. Algoritma tersebut terinspirasi dari mekanisme seleksi alam, dimana individu yang lebih kuat kemungkinan akan menjadi pemenang dalam lingkungan yang kompetitif dan solusi yang optimal dapat diperoleh dan diwakili oleh pemenang akhir dari permainan genetika. Di dalam algoritma genetika tersedia solusi yang diterapkan pada sebuah populasi individu-individu yang masing-masing mewakili solusi yang mungkin. Setiap solusi yang mungkin disebut dengan kromosom, yang ditunjukkan dengan sekumpulan simbol dalam bentuk string dengan panjang tertentu dan biasanya dari rangkaian alphabet biner (0,1). Setiap bits dari kromosom tersebut dinamakan dengan gen [1]. Ilustrasi dari gen dan kromosom ditunjukkan oleh **Gambar 1**.



Gambar 1. Rangkaian gen yang membentuk kromosom dan berperan sebagai solusi dari populasi algoritma genetika.

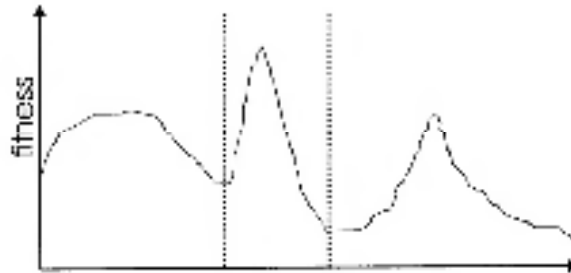
Selama berjalannya siklus (lihat Gambar 2) dari algoritma genetika, kromosom dari populasi yang tersedia dibentuk secara acak. Siklus operasional genetika akan berhasil apabila kromosom yang disebut orang tua dipilih melalui rutinitas penyeleksian yang spesifik. Gen dari orang tua digabung untuk menghasilkan keturunan yang merupakan generasi baru. Dari proses evaluasi yang dilakukan secara berulang, diharapkan kromosom yang lebih baik (sehat) akan menghasilkan keturunan sehat yang lebih banyak sehingga meningkatkan kualitas keseluruhan dari generasi berikutnya.



Gambar 2. Diagram alir dari siklus algoritma genetika

2.2 Modul Pulau Algoritma Genetika

Dalam teori Darwin mengenai evolusi [2], diterangkan bahwa dalam satu populasi bisa terdapat diversitas biologis dimana setiap kelompok-kelompok individu di dalam populasi berbeda-beda sesuai dengan perbedaan mekanisme pertumbuhan dan regional mereka. Dalam teori komputasional evolusi, diversitas tersebut bisa diartikan sebagai "permasalahan multimodal"[3] yaitu sebuah permasalahan dimana di dalam populasi solusi terdapat beberapa kelompok-kelompok solusi yang memiliki optima lokal tersendiri. Gambar 3 menggambarkan representasi dari ruang solusi permasalahan multimodal yang memiliki tiga optima lokal. Konsep multimodal lazim ditemukan di berbagai permasalahan yang membutuhkan algoritma genetika untuk menyelesaikannya.

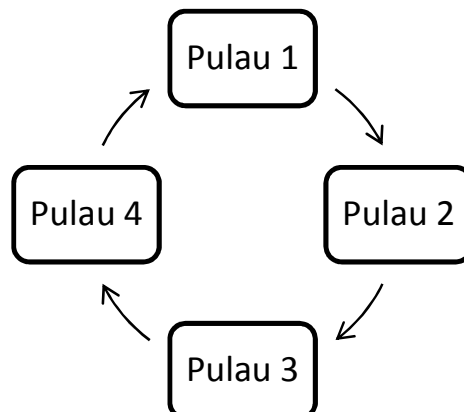


Gambar 3. Tiga optima lokal di dalam sebuah ruang solusi permasalahan multimodal.

Ada dua mekanisme yang digunakan untuk menemukan solusi terbaik dalam permasalahan multimodal [3]. Ketiga mekanisme tersebut adalah sebagai berikut.

1. Menjalankan algoritma genetika beberapa kali dan menyimpan solusi yang terbaik. Cara ini sederhana tetapi bisa saja tidak cukup efektif dalam menjelajahi ruang solusi yang luas.
2. Menjalankan beberapa algoritma genetika secara paralel dan secara periodik membiarkan masing-masing instan algoritma genetika untuk berbagi informasi diantara mereka. Mekanisme ini adalah mekanisme yang digunakan dalam penelitian di jurnal ini.

Ide utama dari pendekatan komputasi evolusi paralel [4] adalah dengan menjalankan beberapa populasi sekaligus secara paralel dengan struktur komunikasi tertentu. Struktur komunikasi tersebut bisa berupa sebuah cincin atau struktur apapun yang mungkin untuk diimplementasi. Gambar 4 menunjukkan struktur komunikasi cincin antara 4 buah pulau.



Gambar 4. Empat buah instan algoritma genetika/pulau yang saling berbagi informasi secara periodik dengan menggunakan struktur komunikasi cincin.

3. Metode Penelitian

Dalam penelitian ini, langkah-langkah yang ditempuh adalah sebagai berikut.

1. Mengumpulkan data sampel dari kasus *travelling salesman* yang disimpan di dalam XML.
2. Membentuk populasi dari data sampel tersebut.
3. Membentuk konfigurasi yang berbeda dengan menggunakan populasi yang ada.
4. Menjalankan algoritma genetika dengan menggunakan konfigurasi yang berbeda. Untuk pertukaran individu dari setiap algoritma genetika bisa menggunakan satu dari dua mekanisme yang berbeda yaitu menukar individu terburuk dengan individu terbaik (*Best to Worst*) dan secara acak (*Random*).
5. Menguji hasil dari setiap konfigurasi.

Parameter standar dari algoritma genetika yang digunakan dalam penelitian ini adalah sebagai berikut.

Parameter	Value
Ukuran populasi	500
# dari kota	20
Jumlah generasi	10000
Persentasi Mutasi	3%
Epoch	1000
Operator Mutasi	Insertion
Operator Kombinasi	Standar

4. Hasil dan Pembahasan

Berikut adalah hasil dari berbagai pengujian dengan menggunakan jumlah pulau yang berbeda. Setiap pulau dijalankan dengan menggunakan instan algoritma genetika yang berbeda dimana setiap 1000 epoch terjadi pertukaran informasi antara setiap pulau.

Dari setiap pengujian terdapat enam indikator yaitu *Worst* (*Random* dan *Best to Worst*), *Best* (*Random* dan *Best to Worst*), dan *Average* (*Random* dan *Best to Worst*). *Worst* menandakan solusi dengan fitness terburuk dalam populasi, *Best* menandakan solusi dengan fitness terbaik dalam populasi, sedangkan *Average* menandakan rata-rata fitness dalam populasi. *Random* dan *Best to Worst* adalah dua jenis mekanisme pertukaran individu yang berbeda.

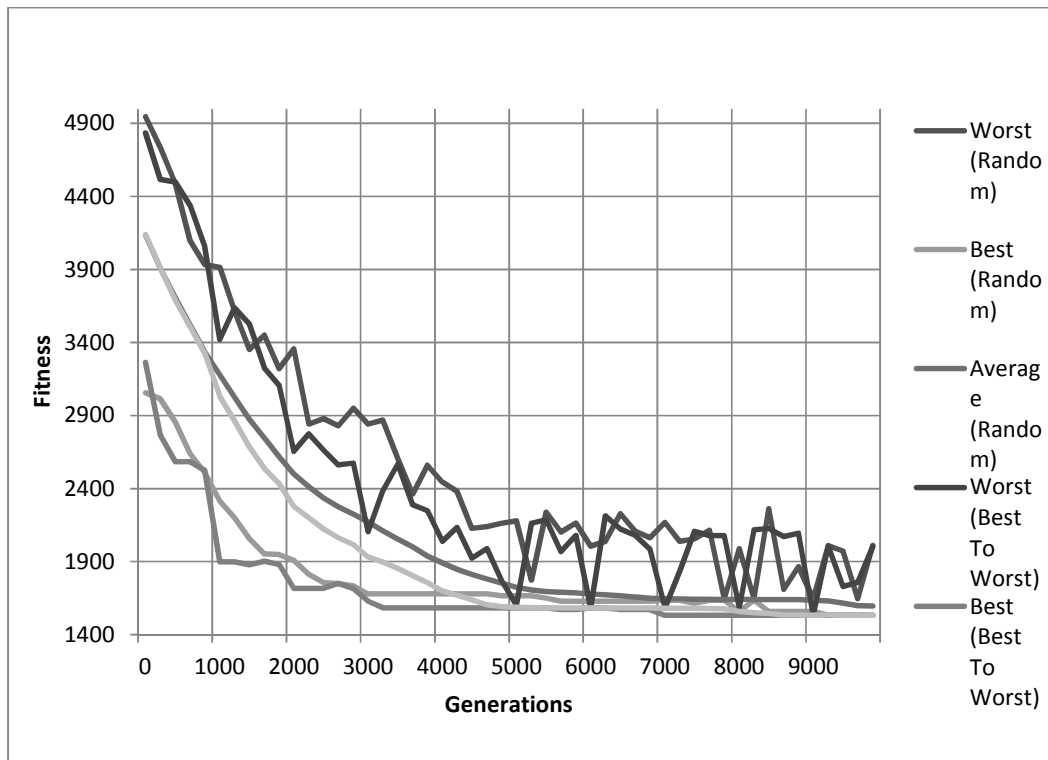
4.1 Komparasi performansi algoritma genetika dengan menggunakan dua pulau

Dalam pengujian berikut, dua instan dari algoritma genetik dijalankan dengan menggunakan model dua pulau dimana setiap pulau menggunakan mekanisme pertukaran individu yang berbeda. Instan pertama menukar individu dengan menggunakan pemilihan acak sedangkan instan lainnya menukar individu terburuk dari pulau pertama dengan individu terbaik dari pulau kedua. Gambar 5 memperlihatkan hasil pengujian dengan menggunakan dua buah pulau.

Dari hasil pengujian didapat metode *best to worst*, selisih jarak antara *worst*, *best*, dan *average* semakin mengecil dan mulai menyatu menjadi satu garis lebih cepat daripada metode acak. Hal tersebut berarti metode *best to worst* memiliki diversitas yang lebih kecil jika

dijalankan dalam generasi yang tinggi. Akan tetapi, metode tersebut bisa mencapai nilai fitness terbaik dalam waktu yang lebih cepat dari metode acak.

Dari hasil analisis tersebut didapatkan kesimpulan bahwa untuk jumlah generasi yang tinggi dan ruang solusi yang besar, metode *best to worst* lebih buruk daripada metode acak karena diversitasnya kecil dimana bisa memperkecil peluang untuk penjelajahan ruang solusi lebih lanjut. Akan tetapi untuk jumlah generasi dan ruang solusi yang kecil, metode *best to worst* lebih bagus karena bisa mencapai solusi terbaik dengan lebih cepat.



Gambar 5. Komparasi dengan dua pulau

4.2 Komparasi performansi algoritma genetika dengan menggunakan empat pulau

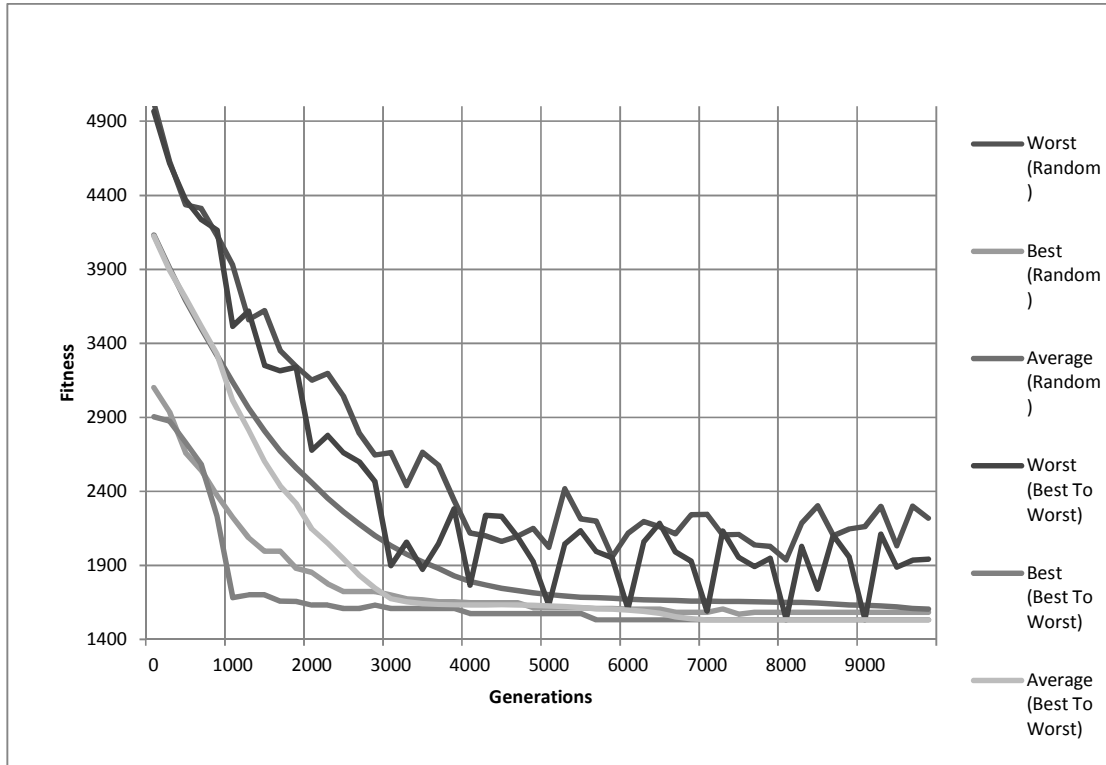
Dalam pengujian berikut, empat instan dari algoritma genetik dijalankan dengan menggunakan model empat pulau dimana setiap pulau menggunakan mekanisme pertukaran individu yang berbeda. Gambar 6 memperlihatkan hasil pengujian dengan menggunakan empat buah pulau.

Dari hasil pengujian didapat hasil yang sama dengan pengujian dengan menggunakan model dua pulau sehingga bisa disimpulkan juga bahwa jumlah generasi yang tinggi dan ruang solusi yang besar, metode *best to worst* lebih buruk daripada metode acak karena diversitasnya kecil dimana bisa memperkecil peluang untuk penjelajahan ruang solusi lebih lanjut. Akan tetapi untuk jumlah generasi dan ruang solusi yang kecil, metode *best to worst* lebih bagus karena bisa mencapai solusi terbaik dengan lebih cepat.

4.3 Komparasi performansi algoritma genetika dengan menggunakan enam pulau

Dalam pengujian berikut, enam instan dari algoritma genetik dijalankan dengan menggunakan model enam pulau dimana setiap pulau menggunakan mekanisme pertukaran individu yang berbeda. Gambar 7 memperlihatkan hasil pengujian dengan menggunakan enam buah pulau.

Dari hasil pengujian didapat hasil yang sama dengan pengujian dengan menggunakan model dua pulau sehingga bisa disimpulkan juga bahwa jumlah generasi yang tinggi dan ruang solusi yang besar, metode *best to worst* lebih buruk daripada metode acak karena diversitasnya kecil dimana bisa memperkecil peluang untuk penjelajahan ruang solusi lebih lanjut. Akan tetapi untuk jumlah generasi dan ruang solusi yang kecil, metode *best to worst* lebih bagus karena bisa mencapai solusi terbaik dengan lebih cepat.



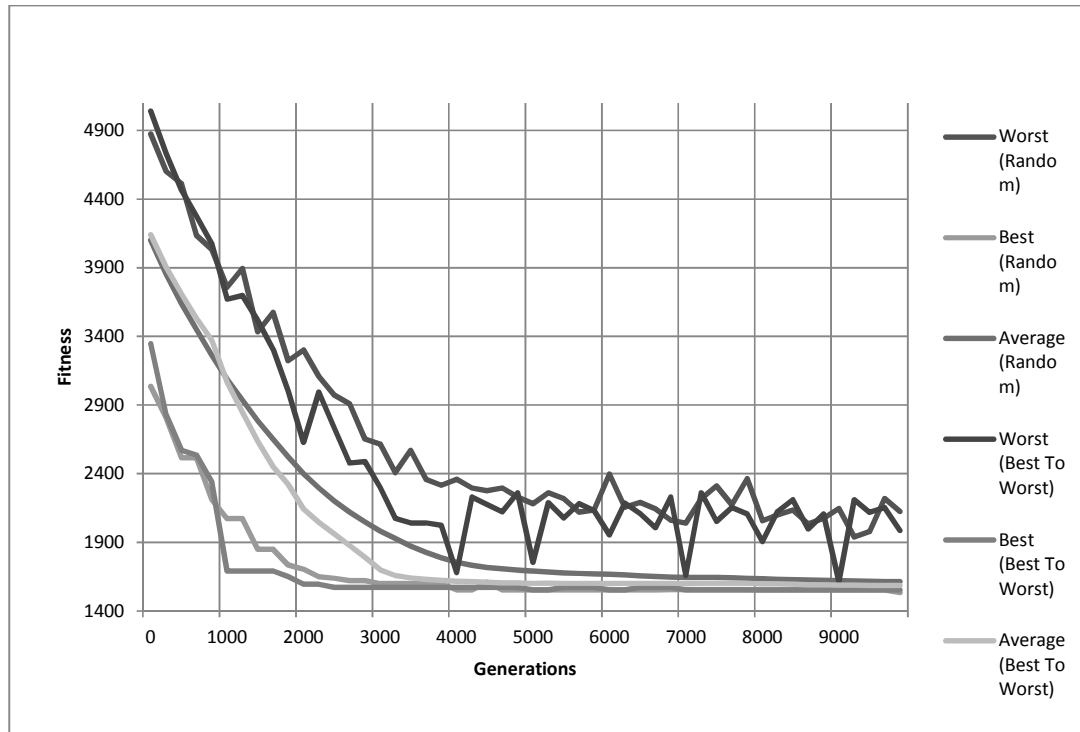
Gambar 6. Komparasi dengan empat buah pulau

4.4 Komparasi performansi algoritma genetika dengan menggunakan dua, empat, dan enam pulau

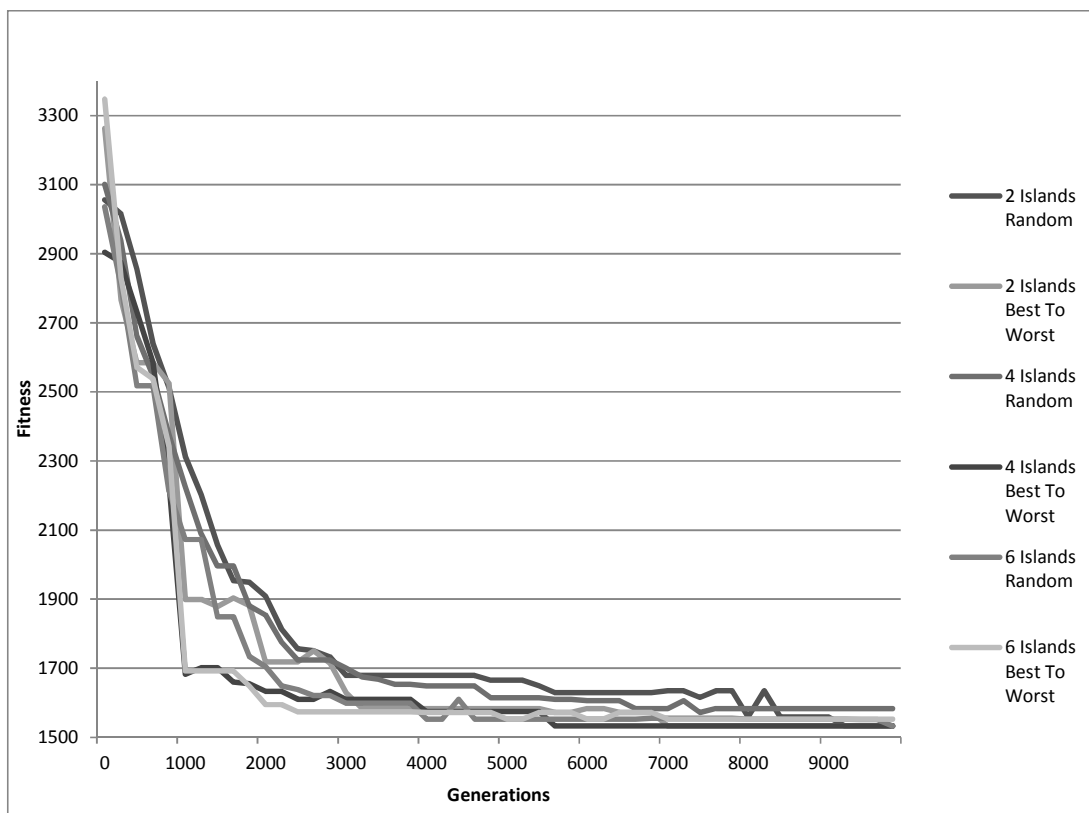
Dalam pengujian berikut, perbandingan performansi dari dua, empat, dan enam pulau sebelumnya dilakukan untuk mengetahui jumlah pulau optimal untuk permasalahan *travelling salesman*. Gambar 8 memperlihatkan hasil pengujian dengan menggunakan enam buah pulau.

Dalam Gambar 8 dapat dilihat bahwa metode *best to worst* mempunyai performansi yang lebih baik dari pada penggunaan metode acak. Selain itu, dapat juga dilihat bahwa semakin tinggi jumlah pulau, semakin cepat instan mencapai solusi terbaik.

Kesimpulan dari pengujian ini adalah peningkatan jumlah pulau bisa membantu algoritma genetika untuk mencapai solusi terbaik akan tetapi dibutuhkan generasi yang lebih tinggi. Menggunakan metode *best to worst* juga bisa membantu instan mencapai solusi terbaik walaupun bisa berdampak berkurangnya diversitas dan tidak baik untuk generasi yang tinggi.



Gambar 7. Komparasi dengan menggunakan enam pulau



Gambar 8. Komparasi performansi dengan menggunakan 2, 4, dan 6 pulau

5. Kesimpulan

Dari penelitian ini didapatkan kesimpulan-kesimpulan sebagai berikut.

1. Untuk jumlah generasi yang tinggi dan ruang solusi yang besar, metode *best to worst* lebih buruk daripada metode acak karena diversitasnya kecil dimana bisa memperkecil peluang untuk penjelajahan ruang solusi lebih lanjut. Akan tetapi untuk jumlah generasi dan ruang solusi yang kecil, metode *best to worst* lebih bagus karena bisa mencapai solusi terbaik dengan lebih cepat.
2. Peningkatan jumlah pulau bisa membantu algoritma genetika untuk mencapai solusi terbaik akan tetapi dibutuhkan generasi yang lebih tinggi. Menggunakan metode *best to worst* juga bisa membantu instan mencapai solusi terbaik walaupun bisa berdampak berkurangnya diversitas dan tidak baik untuk generasi yang tinggi.

Referensi

- [1] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts: MIT Press, 1998.
- [2] C. Darwin, *The Origin of Species*: John Murray, 1859.
- [3] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*: Springer, 2003.
- [4] J. P. Cohoon, *et al.*, "Punctuated equilibria: A parallel genetic algorithm," *Grefenstette*, pp. 148-154.