

OPTIMASI RUTE PENGANTARAN PAKET MENGGUNAKAN METODE *GENETIC ALGORITHM* (GA)

Hani Zulfia Zahro¹⁾, Febriana Santi Wahyuni²⁾

^{1,2)}Teknik Informatika, Institut Teknologi Nasional Malang
hani.zulfia@lecturer.itn..ac.id

Abstrak, Meningkatnya jumlah pengguna *smartphone* pada saat ini, sejalan dengan jumlah pengakses toko *online* melalui web atau aplikasi. Beberapa kemudahan yang ditawarkan oleh jenis pembelanjaan *online* menyebabkan perubahan perilaku konsumen dalam berbelanja, dari jenis pembelanjaan konvensional ke online. Hal ini akan meningkatkan jumlah pengantaran dari penyedia jasa pengantaran paket ke konsumen atau pembeli. Permasalahan TSP (*Travelling Salesperson Problem*) adalah penentuan rute terbaik dari tempat asal ke tempat tujuan dan kembali ke tempat asal. Pada penentuan rute pengantaran suatu paket, tanpa harus mendatangi tempat yang sama lebih dari satu kali, jika hanya 10 tempat mudah untuk diselesaikan. Jika banyak data, rute yang harus dilewati semakin banyak dan waktu yang dibutuhkan juga semakin lama. Masalah TSP merupakan permasalahan *Nondeterministic Polynomial-Hard (NP-Hard)*. Salah satu teknik optimasi dapat untuk masalah TSP adalah *Genetic Algorithm* merupakan metode yang mengadaptasi teori evolusi Darwin yaitu genetika dan seleksi. Proses dari metode *Genetic Algorithm* meliputi inisialisasi populasi awal, *crossover*, mutasi, perhitungan nilai *Fitness*, evaluasi, seleksi hingga di peroleh nilai populasi yang baru. Data yang digunakan berupa data sekunder, dengan 6 kota dan 7 populasi dalam satu generasi. Dari implementasi diperoleh optimasi hasil rute terpendek dengan nilai fitness terkecil yaitu $255 A \rightarrow C \rightarrow B \rightarrow E \rightarrow F \rightarrow D \rightarrow A$

Kata Kunci : Genetic Algorithm, Travelling Salesperson Problem (TSP), Pengantaran Paket

PENDAHULUAN

Saat ini jasa pengantaran paket berkembang dengan pesat, sehingga berdampak pada ketatnya persaingan. Sehingga timbul permasalahan dalam menentukan rute pengantaran paket yang akan ditempuh dengan semua kota terlewati, dengan biaya yang minimum, ketepatan waktu pengiriman.

Persoalan *Travelling Salesperson Problem* (TSP) merupakan salah satu persoalan optimasi kombinatorial; jika diberikan sejumlah kota (atau tempat) dan biaya perjalanan dari satu kota ke kota lain. Deskripsi persoalannya adalah bagaimana menemukan rute perjalanan paling murah dari suatu kota dan mengunjungi semua kota lainnya, masing-masing kota hanya dikunjungi satu kali, dan harus kembali ke kota asal keberangkatan. Kombinasi dari semua rute perjalanan yang ada adalah faktorial dari jumlah kota. Biaya perjalanan bisa berupa jarak, waktu, bahan bakar, kenyamanan, dan sebagainya (Fitra *et al*, 2005). TSP dikenal sebagai permasalahan yang bersifat *Nondeterministic Polynomial-Hard (NP-Hard)*. Hal inilah yang menyebabkan penyelesaian secara eksak sulit dilakukan (Puspitorini, 2009)

Metode-metode dalam penyelesaian masalah optimasi untuk solusi TSP salah satunya adalah *Genetic Algorithm* (GA). GA pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). Konsep GA pada dasarnya menggunakan hukum seleksi alam dan teori evolusi alam oleh Charles Darwin GA adalah algoritma pencarian *heuristic* yang berdasarkan atas mekanisme evolusi biologis. (Kusumawati, 2007).

Berdasarkan latar belakang tersebut, maka penelitian ini bertujuan untuk mengetahui efektifitas antara metode GA dalam penentuan rute pengantaran paket.

METODE

Travelling Salesperson Problem (TSP)

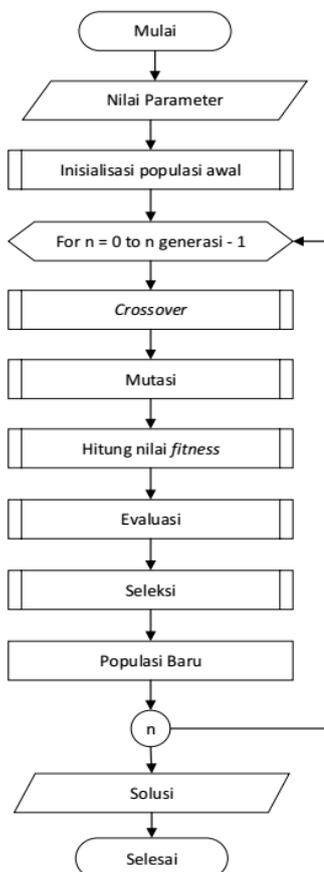
TSP adalah bagaimana menentukan rute optimal perjalanan *salesman* yang harus melalui semua kota tujuan tepat satu kali dan harus kembali ke kota awal” (Godaire dan Parmenter, 1997). Masalah tersebut dapat dimodelkan ke dalam graf berbobot dengan setiap kota tujuan digambarkan sebagai titik dan ruas jalan (rusuk) sebagai sisi berbobot mewakili panjang ruas jalan antara dua kota.

Masalah perjalanan *salesman* tersebut dalam graf adalah mencari lintasan tertutup minimum yang memuat semua titik dalam suatu graf berbobot tersebut.

TSP ini dikenal sebagai permasalahan yang bersifat *Nondeterministic Polynomial-Hard (NP-Hard)*. Hal inilah yang menyebabkan penyelesaian secara eksak sulit dilakukan. (Puspitorini, 2009)

Genetic Algorithm (GA)

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih "alamiah". Salah satu aplikasi algoritma genetika adalah pada permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi (Kusumawati, 2007)



Gambar 1 Alur Proses *Genetic Algorithm*

Proses Algoritma genetika ini dimulai dengan mengambil populasi awal dan

menerapkan operasi-operasi genetika untuk membentuk generasi berikutnya. Setiap individu dalam populasi dianggap sebagai sebuah kromosom yang dapat mewakili sebuah solusi yang layak dari masalah yang dikaji. Individu yang mempunyai kecocokan yang tinggi memiliki kesempatan untuk direproduksi dengan cara menukarkan informasi genetiknya melalui *crossover* dengan individu lain tersebut. Setelah *crossover* melakukan mutasi gen untuk merubah beberapa sifat gen tertentu secara non produksi. Keturunan yang dihasilkan dapat mengganti individu-individu yang memiliki fungsi kecocokan rendah (*steady state approach*). Siklus *reproduksi-crossover-mutasi-evaluasi* ini berulang sampai ditemukannya sebuah solusi yang memuaskan (*global optimum solution*). (Kusumadewi 2003) Adapun alur proses metode GA ditunjukkan pada Gambar 1.

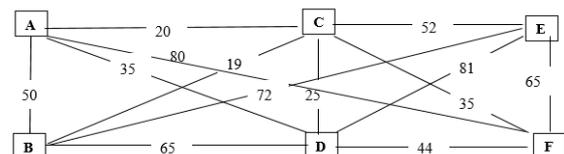
HASIL DAN PEMBAHASAN

Implementasi *Genetic Algorithm*

Tahapan proses implementasi algoritma genetika meliputi :

1. Inisialisasi

Tahapan pertama adalah inisialisasi kromosom, terdapat 7 populasi dalam suatu generasi. Ada 6 kota yang akan menjadi gen dalam kromosom kota selain kota asal.



Gambar 2. Ilustrasi jarak antar kota

$$K [1] = [B C E F D]$$

$$K [2] = [B D F E C]$$

$$K [3] = [D B E C F]$$

$$K [4] = [C F E B D]$$

$$K [5] = [F D B E C]$$

$$K [6] = [C D B E F]$$

$$K [7] = [D F E C B]$$

2. Evaluasi Kromosom

Menghitung nilai *fitness* dari tiap kromosom yang telah dibangkitkan

$$F[k] = (x_1 + x_2) + (x_2 + x_3) + ..$$

$$\begin{aligned} \text{Fitness [1]} &= AB+BC+CE+EF+FD+DA \\ &= 50+19+52+65+44+35 \\ &= 265 \end{aligned}$$

$$\begin{aligned} \text{Fitness [2]} &= AB+BD+DF+FE+EC+CA \\ &= 50+65+44+65+52+20 \end{aligned}$$

Fitness [3] = 296 = AD+DB+BE+EC+CF+FA = 35+65+72+52+35+80 = 339 Fitness [4] = AC+CF+FE+EB+BD+DA = 20+35+65+72+65+35 = 292 Fitness [5] = AF+FD+DB+BE+EC+CA = 80+44+65+72+52+20 = 333 Fitness [6] = AC+CD+DB+BE+EF+FA = 20+25+65+72+65+80 = 327 Fitness [7] = AD+DF+FE+EC+CB+BA = 35+44+65+52+19+50 = 265	$P_4 = \frac{0.003424658}{0.023361165} = 0.146596178$ $P_5 = \frac{0.003003003}{0.023361165} = 0.128546799$ $P_6 = \frac{0.003058104}{0.023361165} = 0.130905456$ $P_7 = \frac{0.003773585}{0.023361165} = 0.161532392$
--	---

Probabilitas nilai fitness diperoleh dari nilai fitness dibagi dengan jumlah fitness

$$Q_i = \frac{1}{fitness_{[i]}}$$

$$Q_1 = \frac{1}{265} = 0.003773585$$

$$Q_2 = \frac{1}{296} = 0.003378378$$

$$Q_3 = \frac{1}{339} = 0.002949853$$

$$Q_4 = \frac{1}{292} = 0.003424658$$

$$Q_5 = \frac{1}{333} = 0.003003003$$

$$Q_6 = \frac{1}{327} = 0.003058104$$

$$Q_7 = \frac{1}{265} = 0.003773585$$

$$\text{Total} = 0.023361165$$

Berdasarkan hasil probabilitas fitness di atas nilai kromosom ke-1 mempunyai *fitness* paling kecil sehingga probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari kromosom lainnya. Proses seleksi menggunakan *roulete-wheel*, maka terlebih dahulu mencari nilai probabilitas kumulatif

$$P_{[i]} = \frac{Q_{[i]}}{\text{Total}}$$

$$P_1 = \frac{0.003773585}{0.023361165} = 0.161532392$$

$$P_2 = \frac{0.003378378}{0.023361165} = 0.254818433$$

$$P_3 = \frac{0.002949853}{0.023361165} = 0.385723889$$

Dari probabilitas di atas dapat terlihat bahwa kromosom ke-1 mempunyai *fitness* paling kecil mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari kromosom lainnya. Untuk proses seleksi kita menggunakan *roulette-wheel*, untuk itu kita terlebih dahulu mencari nilai kumulatif dari probabilitasnya.

$$C[1] = 0.126271634$$

$$C[2] = 0.126271634 + 0.128546799 = 0.254818433$$

$$C[3] = 0.254818433 + 0.130905456 = 0.385723889$$

$$C[4] = 0.385723889 + 0.144615149 = 0.530339037$$

$$C[5] = 0.530339037 + 0.146596178 = 0.676935215$$

$$C[6] = 0.676935215 + 0.161532392 = 0.838467608$$

$$C[7] = 0.838467608 + 0.161532392 = 1$$

3. Crossover

Tahap berikutnya adalah *Crossover*, yaitu pindah silang yang dilakukan antara 2 kromosom untuk menghasilkan kromosom anak (*offspring*). Kromosom anak mewarisi sebagian sifat kromosom *parent*. Pada *Travelling Salesperson Problem* (TSP) dapat menggunakan *crossover* satu titik (*one point crossover*). Proses *crossover* dilakukan secara acak untuk menentukan posisi persilangan dengan cara menukarkan bagian kanan dari titik potong kedua *parent* kromosom untuk menghasilkan kromosom anak.

Proses *crossover* dilakukan secara acak dan berulang sampai diperoleh individu yang terbaik. Setelah dilakukan *crossover* yang menghasilkan *offspring* kemudian dilakukan kembali iterasi dan *crossover*. Peluang *crossover* (P_c) menunjukkan rasio dari anak yang akan dihasilkan dalam setiap generasi dengan ukuran populasi. Misalkan ukuran populasi 100 ($popsiz = 100$), sedangkan $P_c = 25\%$, berarti diharapkan ada 20 kromosom dari

100 kromosom yang ada pada populasi tersebut mengalami *crossover* (Kusumadewi, 2003)

4. Mutasi

Setelah dilakukan *crossover* tahap selanjutnya dilakukan mutasi. Tujuan dari mutasi ini adalah agar individu-individu yang ada dalam populasi semakin bervariasi. Mutasi berperan jika pada populasi awal banyak sedikit solusi yang terpilih. Sehingga dilakukan proses mutasi yang berguna untuk mempertahankan keanekaragaman individu meskipun tidak diketahui apa yang akan terjadi pada individu baru.

Mutasi yang digunakan adalah *swapping mutation*. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh Peluang Mutasi (Pm). Jika peluang mutasi terlalu kecil maka banyak gen yang mungkin berguna tidak dievaluasi. Namun apabila peluang mutasi terlalu besar maka terlalu banyak gangguan acak, sehingga akan kehilangan kemiripan dari induknya.

Panjang total gen = jumlah gen 1 kromosom * jumlah kromosom

$$= 5 * 6$$

$$= 30$$

Misal Pm = 20% maka peluang mutasi

$$= 0.2 * 30 = 6$$

$$K [1] = [C B E F D]$$

$$K [2] = [D B C E F]$$

$$K [3] = [B D E F C]$$

$$K [4] = [F C E D B]$$

$$K [5] = [D F C E B]$$

$$K [6] = [D C B E F]$$

$$K [7] = [D E F C B]$$

$$\text{Fitness [1]} = AC+CB+BE+EF+FD+DA$$

$$= 20+19+72+65+44+35$$

$$= 255$$

$$\text{Fitness [2]} = AD+DB+BC+CE+EF+FA$$

$$= 35+65+19+52+65+80$$

$$= 316$$

$$\text{Fitness [3]} = AB+BD+DE+EF+FC+CA$$

$$= 50+65+81+65+35+20$$

$$= 316$$

$$\text{Fitness [4]} = AF+FC+CE+ED+DB+BA$$

$$= 72+35+52+81+65+50$$

$$= 355$$

$$\text{Fitness [5]} = AD+DF+FC+CE+EB+BA$$

$$= 19+44+35+52+72+50$$

$$= 272$$

$$\text{Fitness [6]} = AD+DC+CB+BE+EF+FA$$

$$= 35+25+19+72+65+80$$

$$= 296$$

$$\text{Fitness [7]} = AD+DE+EF+FC+CB+BA$$

$$= 35+81+65+35+19+50$$

$$= 285$$

Rute yang dipilih adalah fitness terkecil yaitu 255 dengan rute $A \rightarrow C \rightarrow B \rightarrow E \rightarrow F \rightarrow D \rightarrow A$

KESIMPULAN

Dari hasil implementasi metode *Genetic Algorithm* diperoleh simpulan rute terpendek untuk masalah TSP tersebut adalah kromosom yang mempunyai nilai fitness terkecil. Dengan nilai fitness 255 dan rute yang harus dikunjungi adalah $A \rightarrow C \rightarrow B \rightarrow E \rightarrow F \rightarrow D \rightarrow A$.

DAFTAR PUSTAKA

- Fitrah, A., Zaky, A., Fitrasani, 2006, *Penerapan Algoritma Genetika pada Persoalan Pedagang Keliling (TSP)*, SekolahTinggi Elektro Dan Informatika ITB.
- Goldberg, David. (1999). *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: Usa-Print
- Haupt, R.L. and Haupt, S.E. (2004) *Practical Genetic Algorithms*. 2nd Edition, John Wiley & Sons, Inc., Hoboken.
- Kusumadewi, Sri. (2003). *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Puspitorini, S. (2009). *Penyelesaian Masalah Traveling Salesman Problem Dengan Jaringan Saraf Self Organizing*. Media Informatika, 6(1).
- Saddat Muhamad Reza, Suharsono Teguh Nurhadi. *Penentuan Optimalisasi TSP (Travelling Salesman Problem) Distribusi Barang Menggunakan Algoritma Genetika Di Buka Mata Adv*.