

# REKAYASA PERANGKAT LUNAK UNTUK PEMETAAN JALUR KABEL LISTRIK DENGAN MENGGUNAKAN ALGORITMA GENETIKA

Taufik Rachman<sup>(1)</sup>, Saiful Bukhori<sup>(2)</sup>

**Abstract:** Electric cable line mapping as Non-deterministic Polynomial-time Complete (NPC) is difficult to solve using ordinary linear algorithm. In solving the problem there isn't a node encounter twice and the route must shortest one, so that it's needed efficiently material. Depend on this problem in this research used Traveling Salesman Problem (TSP). The result showed that using genetic algorithm one of the methods that can be used to determine electric cable line that has high complexity, so gives optimum electric cable design, and using time and cost efficient.

**Keywords:** Non-Deterministic Polynomial Time Complete, Traveling Salesman Problem, Genetic Algorithm, Efficient

Tingginya permintaan saluran listrik menyebabkan sulitnya pemetaan pengawatan listrik pada setiap pemohon, baik itu rumah pribadi, tempat-tempat bisnis ataupun tempat sosial. Salah satu dari kesulitan pemetaan pengawatan ini, disebabkan banyaknya titik atau *node* yang harus dilewati, apabila disetarakan dengan permasalahan *Traveling Salesman Problem (TSP)* merupakan permasalahan yang bersifat *Nondeterministic Polynomialtime Complete (NPC)* yang sulit untuk diselesaikan dengan menggunakan algoritma *linear* biasa.

Permasalahan yang dihadapi pada persoalan pemetaan pengawatan listrik ini, hampir sama dengan permasalahan yang dipakai untuk menyelesaikan permasalahan *TSP*, yaitu setiap *node* tidak

boleh dikunjungi lebih dari satu kali dan lintasan yang ditempuh harus merupakan lintasan yang paling minimum atau terpendek, sehingga material terutama kabel yang dibutuhkan merupakan jumlah kabel yang paling efisien. Dengan mengacu dari permasalahan yang hampir sama tersebut, maka pada penelitian ini digunakan metoda yang dipakai pada penyelesaian untuk memperoleh solusi yang terbaik pada *TSP*, tentu saja dengan mempertimbangkan kasus teknis yang dihadapi pada pemasangan kabel listrik, sebagai contoh medan yang dihadapi dan pertimbangan-pertimbangan lain yang tentu saja harus tetap dipertimbangan oleh seorang pakar untuk memperoleh hasil yang optimal.

---

<sup>1)</sup> Taufik Rachman, S.Kom., Jurusan Teknik Informatika, STT STIKMA Internasional

<sup>2)</sup> Saiful Bukhori, ST, MKom, Jurusan Teknik Komputer, Universitas Jember

Permasalahan yang bersifat NPC seperti TSP atau cara menentukan jalur listrik untuk melewati *node* yang diinginkan, ini merupakan permasalahan yang tidak mempunyai penyelesaian yang lebih baik dari pada mencoba semua kemungkinan solusi yang ada. Hanya saja sayangnya, apabila cara tersebut dilakukan, maka akan membutuhkan waktu komputasi yang tidak sedikit untuk menyelesaikan permasalahan ini. Waktu komputasi akan bertambah seiring dengan bertambahnya suatu faktor dalam permasalahan yang bersifat NPC. Dalam permasalahan menentukan jalur listrik ini, waktu komputasi akan bertambah sesuai dengan pertambahan jumlah *node*, dikarenakan semakin banyak kemungkinan lintasan yang harus diperiksa untuk mencari lintasan minimum. Dengan demikian untuk jumlah *node* yang besar, maka sebuah algoritma *heuristic* dibutuhkan untuk menyelesaikan permasalahan ini.

Keberadaan algoritma genetika sebagai sebuah metoda *adaptive* adalah salah satu *alternative* untuk menyelesaikan permasalahan yang bersifat NPC tersebut. Dengan latar belakang ini, maka peneliti berusaha meneliti dengan merancang dan membuat perangkat lunak untuk menentukan jalur kabel listrik dengan menggunakan algoritma gene-tika.

Apabila dipelajari lebih mendalam tentang algoritma genetika, maka akan didapatkan bahwa bagian terpenting dari algoritma genetika adalah pindah silang (*crossover*) mutasi, seleksi (penghapusan populasi) dan evaluasi *fitness*. Konvergensi dini adalah kesulitan yang banyak dijumpai dalam penerapan algoritma genetika, juga dalam sebagian besar algoritma pencarian. Dalam penyelesaian permasalahan *combinatorial* seperti penentuan jalur kabel listrik atau-

pun TSP, konvergensi dini terjadi apabila pindah silang yang dilakukan menjadikan daerah konvergensi berada pada daerah optimum lokal. Dari permasalahan tersebut diatas, maka untuk dapat mewujudkan perangkat lunak dengan tujuan seperti yang diharapkan, maka secara garis besar ada tiga permasalahan yang harus dipecahkan yaitu: (1) pengkodean sebuah lintasan yang sesuai, sehingga penyelesaian mengarah pada hasil yang tepat, (2) perancangan *operator* genetika yang dapat diaplikasikan untuk menjaga konsistensi *building blocks* dan mencegah ketidaktepatan, dan (3) pencegahan konvergensi dini.

Tujuan penelitian ini, adalah: (1) penyelesaian permasalahan pemetaan jalur kabel listrik dengan menggunakan *Genetic Algorithm (GA)* dengan penanganan terhadap permasalahan yang biasa muncul dalam penerapan *GA* seperti lokal optimum atau dominasi individu-individu solusi pada generasi tertentu, (2) dengan menggunakan *GA* sebagai algoritma *heuristic*, maka diharapkan pada penelitian ini dapat dilihat apakah algoritma ini mampu menyelesaikan permasalahan pemetaan jalur kabel listrik dengan kompleksitas yang tinggi dengan jumlah titik yang banyak. Penanganan terhadap konvergensi dini yang dilakukan dalam penelitian ini, bertujuan untuk menghindari adanya lokal optimum pada penyelesaian yang dilakukan, dengan demikian dapat diuji apakah algoritma ini mampu memberikan solusi terhadap permasalahan yang diselesaikan, sehingga dapat menjadi *alternative* penyelesaian yang lebih tepat dan lebih optimal.

Persoalan TSP melibatkan seorang *traveling salesman* (penjaja dagangan) yang harus melakukan



kunjungan pada setiap kota yang menjadi bagiannya untuk menjajakan produknya. Rangkaian kota-kota yang dia kunjungi dinamakan lintasan, di mana dalam lintasan tersebut terdapat batasan, yaitu tidak boleh ada lebih dari satu kota yang sama. Dengan kata lain, dalam mengunjungi kota-kota penjaja tidak boleh singgah pada suatu kota lebih dari satu kali. Dan pada akhir perjalanan dia harus kembali ke kota tempat dia memulai perjalanan. Apabila diambil sebuah contoh, misalnya terdapat empat kota A, B, C dan D. Lintasan yang ditempuhnya adalah dari kota B ke kota A ke kota D dan ke kota C. Setelah sampai di kota C, maka *salesman* tersebut harus kembali ke kota B. Penyelesaian dan persoalan ini adalah nilai optimum dari rute yang paling murah, yaitu perjalanan dengan jarak paling pendek atau mempunyai total harga minimum, apabila diterapkan pada pemetaan jalur listrik, maka kabel yang paling pendek akan tetapi memiliki resiko teknis yang paling kecil.

Dari definisi *TSP* tersebut, maka dapat disimpulkan bahwa terdapat dua batasan dalam persoalan ini, yaitu batasan pertama adalah setiap kota tidak boleh dikunjungi lebih dari satu kali dan pada akhir perjalanan harus kembali ke kota asal, dengan kata lain lintasan akan berbentuk siklik. Batasan yang kedua, bahwa lintasan yang ditempuh adalah lintasan yang paling minimum atau terpendek. Data yang diketahui dalam permasalahan ini adalah jumlah kota yang harus dikunjungi beserta jarak antar kotanya. Penyelesaian persoalan ini, dengan pendekatan langsung adalah dengan menghitung semua kemungkinan rute yang ada, kemudian dipilih satu rute yang terpendek. Jika ada  $n$  kota yang harus dikunjungi maka ada  $n!/(2n)$  rute yang harus diselidiki. Dengan

cara ini, jumlah waktu komputasi yang diperlukan meningkat seiring dengan bertambahnya ukuran dari persoalan, yaitu jumlah kota. Sebagai contoh, untuk 15 kota akan didapat kurang lebih  $4,4 \times 10^{10}$  rute yang mencerminkan banyaknya rute yang harus diselidiki untuk mendapatkan rute yang optimal, sehingga sulit diselesaikan dengan bantuan perhitungan manual. Dengan kondisi *TSP* ini, maka dapat dikategorikan sebagai permasalahan yang bersifat *NPC*, yang tidak dapat diselesaikan dengan sebuah algoritma *linear standard* jika kompleksitas permasalahan besar. *TSP* akan dapat diselesaikan dengan program komputer berkecepatan tinggi, jika jumlah *node* banyak, misalnya terdapat 50 *node* maka dengan menggunakan *graph* yang besar sebuah algoritma *heuristic* akan selalu dibutuhkan.

Penerapan algoritma genetika untuk menyelesaikan berbagai permasalahan termasuk *TSP* pada prinsipnya tidak jauh berbeda dengan struktur algoritma genetika secara umum. Akan tetapi, ada beberapa hal yang perlu di perhatikan, yang menunjukkan adanya perbedaan antara algoritma genetika konvensional dengan algoritma genetika pada penyelesaian berbagai permasalahan pada saat ini termasuk *TSP* yang merupakan permasalahan kombinatorial. Beberapa hal yang perlu diperhatikan untuk mendapatkan suatu perancangan penyelesaian permasalahan yang sesuai dengan tujuan yang diharapkan, adalah: (1) pengkodean sebuah lintasan yang sesuai, sehingga penyelesaian mengarah pada hasil yang tepat, (2) perancangan *operator* genetika yang dapat diaplikasikan untuk menjaga konsistensi *building blocks* dan mencegah ketidaktepatan, dan (3) pencegahan konvergensi dini.

Pengkodean sebagai langkah awal penggunaan algoritma genetika, dalam penyelesaian pemetaan jalur kabel listrik dengan menggunakan algoritma genetika, juga dilakukan pengkodean terlebih dahulu. Representasi bilangan *biner* tidak sesuai jika diterapkan pada permasalahan yang memiliki sifat kombinatorial. Selama beberapa *decade* terakhir, telah ditemukan beberapa skema representasi yang sesuai untuk permasalahan kombinatorial. Diantaranya adalah, representasi permutasi dan representasi kunci random.

Representasi permutasi merupakan representasi yang paling alami untuk lintasan yang diterapkan permasalahan *TSP*, di mana titik-titik merupakan bagian yang harus dilalui. Wilayah pencarian untuk representasi ini adalah himpunan permutasi dari titik-titik berupa simbol abjad, sebagai contoh sebuah lintasan dari titik-titik berikut ini:

3 – 2 – 5 – 4 – 7 – 1 – 6 – 9 – 8

dari representasi di atas, dapat diuraikan bahwa, lintasan berangkat dari titik 3 dilanjutkan ke titik 2, seterusnya titik 5, 4, 7, 1, 6, 9 dan diakhiri pada titik 8 yang untuk selanjutnya kembali ke titik 3. Representasi ini sering juga disebut dengan representasi *path* atau representasi *order*.

Representasi kunci-kunci random merupakan sebuah *genotype* dengan bilangan-bilangan acak dari (0, 1). Nilai ini digunakan sebagai kunci-kunci untuk mengkodekan penyelesaian. Sebagai contoh, sebuah kromosom untuk 9 titik bisa direpresentasikan sebagai berikut:

[0.23 0.82 0.45 0.74 0.87 0.11 0.56 0.69 0.78]

posisi  $i$  dalam *list* menunjukkan titik  $i$ . Nilai acak dalam posisi  $i$  menentukan urutan didatanginya titik  $i$  dalam lintasan. Dengan kunci-kunci random di atas, dapat

ditentukan bahwa 0.11 adalah titik yang paling kecil, sehingga titik ke-6 menempati urutan pertama dan 0.23 merupakan nilai terkecil kedua, sehingga kota ke-1 menempati urutan kedua dan seterusnya. Sehingga, dari kunci-kunci random di atas dapat ditentukan lintasan sebagai berikut:

6 – 1 – 3 – 7 – 8 – 4 – 9 – 2 – 5

Kunci-kunci random dapat memperkecil keturunan yang tidak mungkin dengan merepresentasikannya dalam cara yang sederhana.

Populasi awal sangat berpengaruh terhadap kinerja algoritma genetika. Apabila dalam populasi awal terdapat kromosom-kromosom yang sudah baik, algoritma genetika akan lebih cepat menemukan solusinya. Akan tetapi, adanya kromosom ini menyebabkan dominasi kromosom super, yang dapat menimbulkan pencarian terjebak pada *local optima*. Cara inisialisasi awal untuk menghasilkan seluruh kromosom untuk permasalahan kombinatorial ini adalah pembangkitan dengan cara acak. Populasi awal acak adalah dengan menggunakan permutasi Josephus, misalnya adalah titik dari 1 sampai 9, permutasi dari lintasan dapat dilakukan dengan menentukan titik awal dan selang, misalnya titik awal adalah 6 dan selang adalah 5, maka lintasan berangkat dari titik 6 selang 5 dari titik 6 adalah titik 2 (dengan asumsi titik 1 sampai dengan 9 membentuk *circular list*). Titik 2 dihapus dari *list*. Selang 5 kemudian adalah 7. Proses ini diulang hingga ada satu lintasan dalam *list*.

## METODE

Prosedur pra-seleksi

Langkah 1 Hitung nilai *fitness* bagi tiap kromosom

$$\text{eval}(v_i) = f(x), \quad i = 1, 2, 3, \dots, \text{size}$$



Langkah 2 Hitung total nilai *fitness*

$$F = \sum_{i=1}^{size} eval(v_i)$$

Langkah 3 Hitung probabilitas seleksi  $P_i$  untuk tiap kromosom

$$P_i = \frac{eval(v_i)}{F} \quad i = 1, 2, \dots, size$$

Langkah 4 Hitung probabilitas komulatif  $q_i$  untuk setiap kromosom

$$q_i = \sum_{j=1}^i p_j \quad i = 1, 2, 3, \dots, size$$

Prosedur seleksi

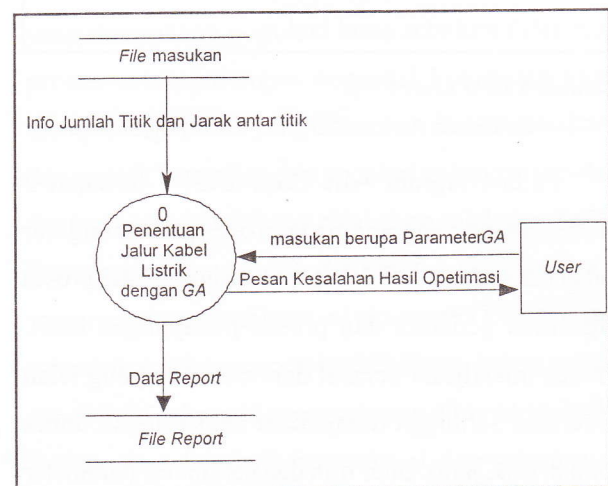
Langkah 1 Bangkitkan bilangan acak  $r$  antara 0 dan 1

Langkah 2 Jika  $r \leq q_i$ , maka pilih kromosom pertama ( $v_i$ ). Jika tidak, pilih kromosom  $v_i$  ( $2 \leq i \leq size$ ) sedemikian rupa sehingga  $q_{i-1} < r \leq q_i$ .

Untuk menghindari dominasi individu tertentu pada proses algoritma genetika, maka digunakan beberapa mekanisme penyesuaian, yaitu: (1) *windowing*, merupakan mekanisme nilai *fitness* setiap kromosom dikurangi dengan nilai *fitness* yang terkecil, mekanisme ini memperbesar probabilitas seleksi dari kromosom yang paling kuat, akan tetapi menghilangkan kromosom yang paling lemah, (2) *exponential*, merupakan nilai *fitness* setiap kromosom ditambah dengan 1, kemudian dikuadratkan, mekanisme ini memperbesar probabilitas seleksi dari kromosom-kromosom yang relatif lemah, (3) *linear normalization* (normalisasi *linear*) bertujuan untuk mempertinggi kemampuan kromosom terbaik untuk reproduksi pada saat kromosom yang lemah justru memungkinkan untuk memproduksi keturunan. Untuk mekanisme *windowing* dan *eksponential*, mekanisme ini berlaku untuk pencarian nilai maksimum. Jika mekanisme ini diterapkan pada proses pemetaan jalur kabel listrik di mana yang dicari adalah

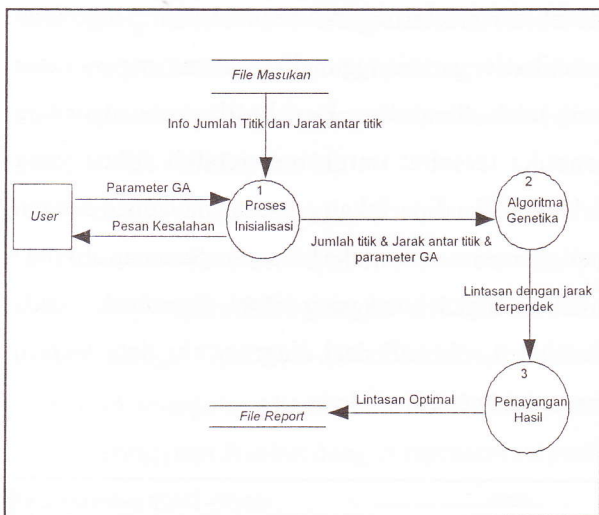
nilai minimum, yang dilakukan adalah memaksimalkan nilai *eval* yang didapatkan dalam masing-masing kromosom.

Secara garis besar, pemrograman perangkat lunak untuk pemetaan jalur kabel listrik dengan menggunakan algoritma genetika ini dibagi dalam tiga bagian besar yaitu: (a) proses inialisasi, (b) proses algoritma genetika, dan (c) proses penayangan hasil. Proses yang pertama adalah proses inialisasi dari masukan yang telah diberikan, sehingga didapatkan angka-angka untuk jumlah titik, jarak antar titik dan parameter-parameter yang dibutuhkan untuk proses algoritma genetika. Kemudian, dari hasil proses inialisasi tersebut digunakan untuk proses algoritma genetika dengan menggunakan parameter-parameter yang telah ditentukan. Dari hasil proses algoritma genetika tersebut untuk mengetahui solusi yang terbaik, maka diperlukan proses penayangan dengan menggunakan antarmuka keluaran. Susunan alir data untuk perangkat lunak yang dibuat, digambarkan pada Gambar 1, diawali dari diagram alir data *level 0* sampai dengan *level 3*.



Gambar 1 Diagram Alir Data *Level 0*

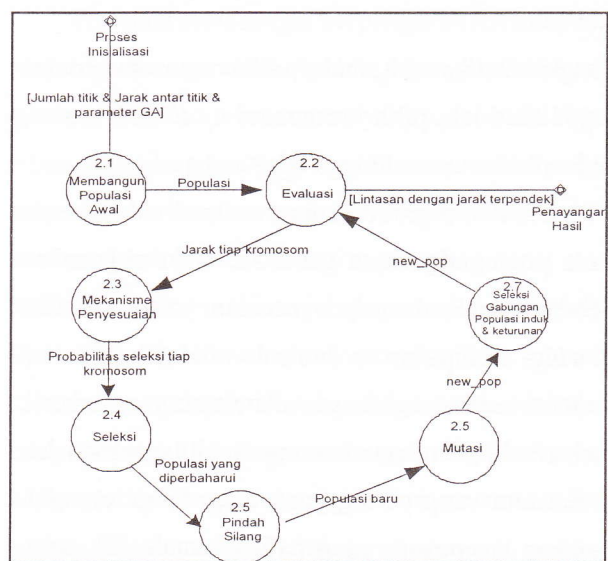
Proses pemetaan jalur kabel listrik dengan GA ini memerlukan dua macam data masukan yaitu data dari pemakai yang berupa parameter GA, data tentang informasi jumlah titik dan jarak antar titik. Hasil proses pemetaan jalur kabel listrik dengan GA tersebut terdapat dua keluaran, yaitu pesan kesalahan kepada user bila terjadi kesalahan pada hasil optimasi, akan tetapi bila tidak terjadi kesalahan akan disimpan pada file report. Detail dari proses pemetaan jalur kabel listrik dengan GA dapat ditunjukkan seperti dalam gambar DFD level 1 pada Gambar 2.



Gambar 2 DFD Level 1

Pada Diagram Alir Data level 1 terdapat 3 proses utama yang ada pada proses pemetaan jalur kabel listrik dengan GA yaitu proses inisialisasi, proses algoritma genetika dan proses penayangan hasil. Proses inisialisasi berasal dari masukan yang telah diberikan sehingga didapatkan angka-angka untuk jumlah titik, jarak antar titik dan parameter-parameter yang dibutuhkan untuk proses algoritma genetika.

Kemudian dari hasil proses inisialisasi tersebut digunakan untuk proses algoritma genetika dengan menggunakan parameter-parameter yang telah ditentukan. Dari hasil proses algoritma genetika tersebut untuk mengetahui solusi yang terbaik, maka diperlukan proses penayangan dengan menggunakan antarmuka keluaran. Detail dari proses algoritma genetika dapat ditunjukkan seperti dalam gambar DFD level 2 pada Gambar 3



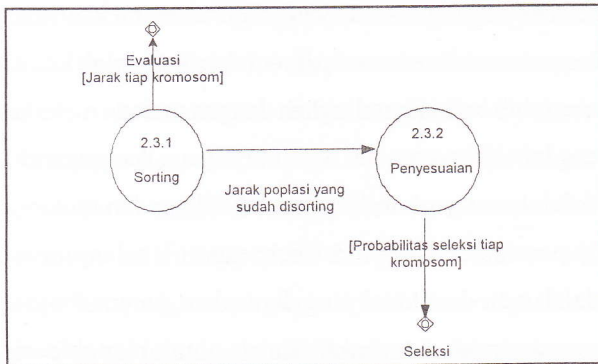
Gambar 3 DFD Level 2 Proses Algoritma Genetika

Pada Diagram Alir Data level 2 untuk proses algoritma genetika terdapat 7 proses sesuai dengan struktur algoritma genetika secara umum yaitu: (a) proses membangun populasi awal, (b) proses evaluasi, (c) proses mekanisme penyesuaian, (d) proses seleksi, (e) proses pindah silang, (f) proses mutasi, dan (g) proses seleksi gabungan populasi induk dan keturunan

Proses mekanisme penyesuaian terdapat sub proses yang detail dari proses mekanisme penye-



suaiian tersebut dapat ditunjukkan seperti dalam gambar DFD *level 3* pada Gambar 4



Gambar 4 DFD *Level 3* Proses Mekanisme Penyesuaian

Hirarki modul yang diimplementasikan pada penelitian ini, mewakili data-data yang masuk dan keluar sesuai dengan diagram alir data pada Gambar 4 terdiri atas: (a) data berbentuk *text* dari *file \*.txt*, (b) jumlah titik, (c) jarak antar titik, (d) parameter jumlah individu dalam populasi, (e) parameter probabilitas pindah silang, (f) parameter probabilitas mutasi, (g) populasi, baik populasi lama ataupun populasi yang sudah diperbaharui, (h) jarak tiap kromosom, (i) probabilitas seleksi setiap kromosom, (j) jarak terpendek, dan (k) kromosom terbaik/lintasan terpendek.

Dari data alir diagram dan hirarki modul tersebut di atas, maka dapat dijelaskan proses-prosesnya sebagai berikut: (1) membangun populasi awal, di mana pada proses ini dilakukan dengan dipandu bilangan acak murni untuk kromosom pertama dan kromosom selanjutnya dengan permutasi josephus, (2) pada bagian proses evaluasi ada dua sub proses yaitu proses menghitung jarak lintasan dari populasi yang telah dihasilkan dan proses selanjutnya adalah mencatat lintasan terpendek, sub proses yang kedua

tidak berpengaruh terhadap proses selanjutnya (setelah proses evaluasi), dari sub proses ini iterasi algoritma genetika akan berhenti apabila iterasi telah dilakukan sebanyak sejumlah generasi, (3) pada bagian mekanisme penyesuaian juga terdiri dari dua sub proses yaitu proses persiapan untuk penyesuaian berupa proses *sorting* dan kemudian dilanjutkan dengan proses penyesuaian, hal ini dilakukan untuk semua metode, baik metode *windowing*, eksponensial maupun normalisasi *linear* seperti yang dijelaskan pada bab sebelumnya, (4) proses seleksi dilakukan dengan seleksi campuran antara stokastik dan elitism dari populasi lama hingga didapatkan populasi baru, (5) proses pindah silang dilakukan sebanyak  $P_c * size$ , dengan salah satu metode pindah silang untuk permasalahan kombinatorial *size* adalah jumlah populasi dalam setiap generasi, (6) proses mutasi dilakukan sebanyak  $P_m * size * L$ , di mana *L* adalah jumlah titik dalam lintasan (jumlah gen dalam setiap kromosom) dari proses pindah silang dan mutasi dihasilkan populasi keturunan, (7) proses seleksi gabungan populasi induk dan populasi keturunan dilakukan dengan menggabungkan populasi induk yang merupakan populasi lama sebelum dilakukan proses seleksi dengan populasi keturunan yang merupakan populasi yang dihasilkan dari pindah silang dan mutasi, kemudian dari populasi gabungan tersebut dengan kapasitas  $2 * size$  dilakukan seleksi terhadap kromosom-kromosom terbaik untuk populasi baru, yang mana populasi baru ini ukurannya kembali lagi sebanyak *size* kromosom, apabila iterasi belum selesai atau belum sebanyak generasi yang diinginkan, maka proses akan kembali ke evaluasi dan seterusnya, dengan populasi yang dievaluasi merupakan populasi

baru yang didapatkan pada proses ini, (8) proses penayangan hasil akan dilakukan bila iterasi telah dilakukan sebanyak jumlah generasi dan proses algoritma genetika akan berhenti, proses penayangan ini dilakukan dengan menayangkan hasil optimasi beberapa jarak terpendek dan lintasan terpendek yang didapatkan.

### HASIL DAN PEMBAHASAN

Unjuk kerja algoritma sangat dipengaruhi oleh parameter yang digunakan dan nilai-nilai yang dimasukkan terhadap parameter tersebut. Dari hasil uji coba dapat diketahui bahwa parameter-parameter yang berpengaruh terhadap kinerja algoritma genetika dalam penyelesaian permasalahan pemetaan jalur kabel listrik dengan menggunakan algoritma genetika agar didapatkan hasil yang optimal dengan terhindar dari kasus konvergensi dini adalah: (1) ukuran populasi; parameter ini adalah parameter yang pertama kali berpengaruh terhadap kinerja algoritma genetika, untuk jumlah titik yang besar, maka ukuran populasi untuk ruang pencarian harus diberikan dengan cukup, akan tetapi permasalahan dengan jumlah titik yang relatif kecil, maka ukuran populasi cukup diberikan dengan ukuran yang kecil, dari tabel tersebut dapat dilihat bahwa jumlah titik sebesar 7 cukup dengan ukuran populasi 11, maka sudah didapatkan nilai yang optimum, untuk jumlah titik 10 ternyata dengan ukuran populasi 30 menunjukkan hasil yang lebih meningkat dibandingkan dengan jumlah populasi 15 akan tetapi jika ukuran populasi terlalu besar, akan percuma karena mengakibatkan pemborosan waktu, hal ini dapat dilihat dalam uji coba untuk 15 kota dengan ukuran populasi yang diperbesar,

tetap menghasilkan nilai optimum pada wilayah yang sama dengan jumlah populasi dibawahnya, (2) parameter kedua yang menentukan adalah metoda mekanisme penyesuaian yang digunakan, mekanisme *linear normalization* relatif memberikan unjuk kerja yang lebih baik dibandingkan dengan metoda metoda yang lain, (3) parameter berikutnya yang berpengaruh adalah besar probabilitas pindah silang dan mutasi, (4) parameter yang turut berpengaruh selanjutnya adalah metode mutasi yang digunakan, karena fungsi *operator* mutasi terbukti adalah untuk menghindari konvergensi dini, dari hasil uji coba, maka dapat ditentukan bahwa metode inversion merupakan metode yang menghasilkan nilai paling optimum.

Parameter-parameter yang lain seperti ukuran generasi, asalkan ukuran populasi yang diberikan cukup, parameter ini tidak banyak berpengaruh, akan tetapi tidak menutup kemungkinan suatu saat dibutuhkan ukuran generasi yang cukup, sedangkan untuk metode pindah silang secara umum semua metode yang ada memberikan kinerja yang sama pada algoritma genetika, sedangkan untuk jenis seleksi campuran relatif memberikan hasil yang lebih baik.

### SIMPULAN

Penentuan jalur kabel listrik dengan menggunakan algoritma genetika merupakan salah satu metode yang dapat digunakan, terutama untuk penentuan jalur kabel listrik yang memiliki kompleksitas yang tinggi, sehingga diharapkan akan menghasilkan desain jalur listrik yang optimum dengan penggunaan waktu dan biaya secara efisiensi.

Unjuk kerja algoritma genetika untuk penyelesaian penentuan jalur kabel listrik dengan meng-



gunakan algoritma genetika sangat dipengaruhi oleh parameter-parameter yang digunakan dan nilai-nilai dari parameter yang digunakan. Pengaruh parameter-parameter adalah: (1) pada percobaan dengan jumlah titik yang kecil, maka daerah pencarian cukup dengan ukuran populasi yang kecil, akan tetapi untuk jumlah titik yang besar, ukuran populasi yang kecil sangat berpengaruh terhadap ruang pencarian, artinya nilai optimum sulit didapatkan dengan kondisi ruang pencarian yang sempit seperti ini, dalam algoritma genetika, hal ini sulit diatasi dengan menambah ukuran populasi, (2) normalisasi *linear* merupakan mekanisme penyesuaian yang paling sesuai untuk menghindari terjadinya dominasi individu-individu tertentu pada ruang pencarian, untuk menghindari konvergensi dini, maka harus dipilih metode mutasi yang tepat untuk permasalahan yang akan diselesaikan, (3) metode seleksi campuran merupakan metode seleksi yang paling tepat untuk penyelesaian kasus penentuan

jalur kabel listrik dengan menggunakan algoritma genetika ini.

Apabila algoritma genetika dioptimalkan, artinya pemilihan parameter dan nilai dari parameter diberikan secara tepat untuk eksploitasi ruang pencarian dan eksploitasi nilai optimum, maka algoritma genetika akan memberikan unjuk kerja yang lebih bagus dalam hal mendapatkan nilai optimum yang diinginkan. Hal ini dapat dibuktikan dengan melakukan uji coba seperti pada pembahasan terdahulu.

#### DAFTARRUJUKAN

- Gen, M., and, Runwei, C. 1997. *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons, Inc.
- Lawrence, D. 1991. *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold.
- Michalewicz, Z. 1996. *Genetic Algorithms + Data Structure = Evolution Programs*. New York: Springer-Verlag Berlin Heidelberg.
- Rich, E. 1991. *Artificial Intelligence*. New Jersey: McGraw-Hill, Inc.