

Optimasi *Routing* pada *Metropolitan Mesh Network* Menggunakan *Adaptive Mutation Genetic Algorithm*

Merinda Lestandy¹, Sholeh Hadi Pramono², Muhammad Aswin³

Abstract—In dynamic and wide networks, such as Metropolitan Mesh Network (MMN), routing becomes very complex because a packet can be blocked before it reaches its destination. In addition, users can also log in or log out from network topology. Therefore, a good routing algorithm, which is able to reduce time in network update process or when there is an error in the network, are required. Routing problems can be represented as the shortest path problem to facilitate completion. In this paper, a routing algorithm optimization using Adaptive Mutation Genetic Algorithm (AMGA) on MMN is presented by determining a probability of 0.000005782 at the beginning, with crossover probability of 0.000847, to reduce or avoid premature convergence.

Intisari—Pada jaringan dinamis dan sangat besar seperti *Metropolitan Mesh Network* (MMN), *routing* menjadi sangat kompleks karena banyak potensi terhalangnya suatu paket di tengah jalan, sebelum mencapai tujuannya. Selain itu, pengguna pun dapat masuk dan keluar dari topologi jaringan. Oleh karena itu, dibutuhkan algoritme *routing* yang baik dan mampu menekan waktu dalam *update* jaringan atau jika terjadi kesalahan dalam jaringan. Permasalahan *routing* dapat direpresentasikan sebagai masalah jalur terpendek untuk memudahkan penyelesaiannya. Pada makalah ini dihasilkan bahwa *Adaptive Mutation Genetic Algorithm* (AMGA) mampu mengoptimalkan *routing* pada MMN dengan menentukan probabilitas mutasi di awal sebesar 0,000005782 dan probabilitas *crossover* sebesar 0,000847, sehingga mampu mengurangi atau menghindari adanya *premature convergence*.

Kata Kunci— *Metropolitan Mesh Network* (MMN), Optimasi *Routing*, *Adaptive Mutation Genetic Algorithm* (AMGA).

I. PENDAHULUAN

Semenjak komputer menjadi lebih kecil, lebih cepat, dan lebih mudah dioperasikan, kebutuhan pengguna akan peningkatan *bandwidth* semakin besar. Dengan adanya kemampuan komunikasi tambahan, perangkat lunak, dan perangkat keras yang lebih baik, pengguna aplikasi dapat mengisi semua *bandwidth* yang tersedia. Peningkatan *bandwidth* menghasilkan aplikasi yang membutuhkan sumber komunikasi yang lebih besar [1].

Penggunaan dan perkembangan jaringan komputer saat ini begitu pesat. Seiring dengan perkembangan tersebut,

kebutuhan pengguna akan kualitas jaringan semakin meningkat. Kualitas yang dimaksud adalah jaringan komputer yang terbebas dari masalah seperti pengiriman data yang lambat, koneksi yang tidak stabil, dan sebagainya, sehingga secara tidak langsung dapat mengurangi produktifitas kerja. Bahkan, dari tahun 2000 hingga 2017, perkembangan internet di dunia mencapai 923,9% dengan 3.696.238.430 pengguna dari data terakhir yang diperoleh [2]. Penyedia layanan sambungan internet yang lebih dikenal *Internet Service Providers* (ISP) mencoba memenuhi peningkatan permintaan lalu lintas tersebut dengan teknologi baru dan meningkatkan pemanfaatan sumber daya yang ada. *Routing* paket data dalam hal ini dapat memengaruhi kemampuan sumber daya jaringan [3].

Routing merupakan suatu proses pencarian jalur dalam suatu jaringan dari *node* sumber ke *node* tujuan yang digunakan sebagai jalur data. *Routing* memiliki peran fundamental dalam internet. *Routing* merupakan fungsi yang bertanggung jawab membawa data melewati sekumpulan jalur dalam jaringan dengan cara memilih jalur terbaik untuk dilewati data.

Dalam jaringan dinamis dan sangat besar seperti MMN, *routing* menjadi sangat kompleks karena banyak potensi dalam pertengahan perjalanan suatu paket dapat terhalang sebelum mencapai tujuannya. Selain itu, pengguna pun dapat masuk dan keluar dari topologi logika jaringan. Oleh karenanya dibutuhkan algoritme *routing* yang baik dan mampu menekan waktu dalam *update* jaringan ataupun jika terjadi kesalahan dalam jaringan.

Algoritme Genetika (*Genetic Algorithm*/GA) adalah algoritme pencarian heuristik yang didasarkan atas mekanisme evolusi biologis [4]. GA pertama kali dikembangkan oleh John Holland pada tahun 1975. GA melakukan penggabungan berbagai pilihan solusi terbaik secara acak dalam suatu kumpulan untuk mendapatkan solusi terbaik berikutnya, yaitu suatu kondisi yang memaksimalkan kecocokan atau disebut generasi. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya secara berulang. Algoritme ini diharapkan dapat menyimulasikan proses evolusioner.

Terdapat kelemahan besar di dalam GA, yaitu ketika terjebak pada *local optima*, kinerja GA terus memburuk dan tidak ada cara untuk melepaskan diri [5]. Oleh karena itu, pada GA sering terjadi *premature convergence* atau konvergen pada solusi *local optima* yang disebabkan ruang pencarian yang terbatas [6]. Memperlebar ruang pencarian GA mampu menghindari *premature convergence*. Salah satu cara untuk memperlebar ruang pencarian GA adalah dengan melakukan perubahan pada mutasi dengan teknik *adaptive mutation* pada GA. Teknik *adaptive mutation* merupakan mutasi yang didasarkan ukuran ruang pencarian.

¹ Mahasiswa, Program Studi Magister Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Jalan M.T. Haryono 167 Malang 65145, INDONESIA (e-mail: lestandy_merinda@yahoo.com)

^{2, 3} Dosen, Program Studi Magister Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Jalan M.T. Haryono 167 Malang 65145, INDONESIA (e-mail: sholehpramono@gmail.com, maswin@ub.ac.id)

Berdasarkan uraian di atas, maka dalam makalah ini digunakan AMGA untuk menyelesaikan masalah jalur terpendek dalam MMN, sehingga mampu mengoptimalkan lalu lintas data, kecepatan transfer data, dan meningkatkan kinerja jaringan komputer.

II. ROUTING PROTOKOL MENGGUNAKAN GA

Banyak penelitian yang dikembangkan dalam beberapa tahun terakhir untuk mengoptimalkan kelemahan GA dengan menggunakan *adaptive mutation* dan menyelesaikan masalah *routing* pada jaringan. Adapun penelitian-penelitian yang relevan sebagai acuan penelitian dapat diuraikan sebagai berikut.

Sebelumnya, GA digunakan untuk desain *routing protocol Wireless Mesh Network (WMN)*. *Routing protocol* dianalisis dengan menerakan nilai evaluasi servis *node*, memanfaatkan tingkat kehilangan data, jumlah *hop bandwidth*, dan status lalu lintas yang diterapkan pada GA. Adapun hasil yang telah dilakukan adalah GA mampu menemukan solusi optimal dalam membangun jalur *routing protocol WMN* yang dinamis [7].

Penggunaan GA juga diterapkan untuk *Quality of Service (QoS) multicast routing protocol*. Parameter QoS yang digunakan antara lain tunda (*delay*), biaya (*cost*), dan sebagainya, untuk menemukan rute baru. Dari hasil simulasi, disimpulkan bahwa GA memiliki rasio keberhasilan pencarian dan perhitungan waktu yang lebih baik dibandingkan dengan algoritme konvensional karena menghubungkan *node* awal untuk setiap *node* tujuan dengan menggunakan tiga jalur minimum [8].

Selain itu, GA mampu menilai pelaksanaan *protocol multipath distance vector protocol* untuk mengefisienkan *routing* pada *Mobile Adhoc Networks (MANETs)*. Kinerja GA dibandingkan dengan *On-demand Multipath Distance Vector (AOMDV)*. Simulasi dilakukan dengan menggunakan kuantitas *node* yang berbeda (25, 50, 75, dan 100) dan ukuran jaringan adalah 1.200 m². Jangkauan komunikasi *node* adalah 250 m dan jenis lalu lintas *bit rate* konstan. Terlihat bahwa metode GA meningkat pada tabel *Packet Delivery Ratio* sebesar 0,19%, 2,42%, 0,72%, dan 3,08% jika dibandingkan dengan metode AOMDV dengan 25, 50, 75, dan 100 jumlah *node*. Pada tabel *Average End to End Delay in second*, metode GA menurun sebesar 11,86%, 20,16%, 33,87%, dan 15,71% jika dibandingkan dengan metode AOMDV dengan 25, 50, 75, dan 100 jumlah *node*. Selanjutnya, metode GA berkurang pada tabel *Average Number of hops to sink* sebesar 10,09%, 0,86%, 10,72%, dan 14,62% jika dibandingkan dengan metode AOMDV dengan 25, 50, 75, dan 100 jumlah *node* [9].

Peningkatan *routing* pada *Wireless Sensor Network (WSN)* dengan *Hybrid GA* dilakukan dengan membangun kolam gen (*gene pool*) untuk desain operator genetika untuk menghindari *premature convergence*. Dengan menggunakan MATLAB, simulasi WSN ditetapkan. Jumlah *node* sensor adalah 210 dan radius komunikasi adalah 20. Secara acak didistribusikan di wilayah 100 × 100 area, *node* sensor awal yang merata antara 200 J dan 400 J, paket data yang dikumpulkan oleh *node* sensor adalah sebesar 512 *byte*. Setiap *node* mengirimkan

paket sekitar 0,2 J dan menerima paket sekitar 0,01 J. Siklus hidup jaringan adalah tujuannya ketika jumlah *node* dalam jaringan di bawah 85%. Hasil penelitian tersebut adalah *Hybrid GA* mampu menyeimbangkan penggunaan energi dan memperpanjang siklus hidup jaringan, sehingga meningkatkan efisiensi jaringan WSN [10]. Namun, GA pada *routing protocol* dalam topologi *spanning tree* untuk WSN mampu mengurangi penyebaran nilai-nilai energi pada *node*, sehingga jaringan mampu menghemat waktu dengan lebih baik [11].

Teknik *adaptive mutation* kemudian diusulkan untuk meningkatkan kinerja GA pada fungsi *De-Jong Cost Function*. Teknik yang diusulkan ditujukan untuk meningkatkan konvergensi. Hasil tes menunjukkan bahwa penggunaan teknik *adaptive mutation* memiliki konvergensi solusi lebih dekat dengan solusi optimal dibandingkan dengan teknik mutasi statis lain, seperti mutasi Gaussian dan teknik mutasi seragam. Dari hasil percobaan, *adaptive mutation* yang diusulkan pada GA terbukti mendekati solusi optimal dibandingkan dengan GA biasa, yaitu *De-Jong Cost Function* yang minimal untuk beberapa variasi [12].

III. ADAPTIVE MUTATION GENETIC ALGORITHM

Pada GA, probabilitas mutasi, P_m , bernilai tetap, sehingga mengakibatkan masalah optimasi dari efisiensi GA kurang maksimal. Saat ini, operator *adaptive mutation* umumnya digunakan di GA untuk memecahkan masalah optimasi, terutama untuk masalah dunia nyata. *Adaptive mutation* bertujuan untuk mempertahankan keragaman genetik dalam populasi dan mencegah GA dari *premature convergence* [6].

Nilai rasio mutasi pada *adaptive mutation* diberikan di awal. Setelah itu, dihitung nilai *progress* dengan persamaan berikut.

$$progress_i(t) = \sum_{j=1}^{M_i} (\max [f(p_j^i(t)), f(c_j^i(t))] - f(p_j^i(t))) \quad (1)$$

dengan $progress_i(t)$ adalah nilai kemajuan atau *progress* dari operator i pada generasi t , f adalah *fitness* dari individu, $p_j^i(t)$ adalah induk dan *offspring* (keturunan) yang dihasilkan oleh operator mutasi i pada generasi t , dan M_i mewakili jumlah individu yang dipilih operator mutasi i untuk bermutasi. Rasio mutasi operator i diperbarui sesuai dengan nilai *progress* rata-rata pada generasi t , sesuai dengan persamaan berikut.

$$p_i(t+1) = \frac{progress_i(t)}{\sum_{j=1}^N progress_j(t)} (P_{mutation} - N * \delta) + \delta \quad (2)$$

dengan $p_i(t)$ adalah nilai *progress* operator i pada generasi t , N adalah jumlah total operator mutasi, $\delta = 0.01$ adalah rasio mutasi minimum untuk setiap operator mutasi dan $P_{mutation}$ merupakan probabilitas mutasi awal.

Berikut ini adalah algoritme yang membedakan GA dengan AMGA. Kerangka umum GA ditunjukkan pada Algoritme 1

sedangkan kerangka GA dengan operator mutasi adaptif ditunjukkan pada Algoritme 2. Algoritme 2 berbeda dari standar GA (Algoritme 1) dalam beberapa langkah, seperti Langkah 4, 10, dan 11.

Algoritme 1 Kerangka umum Algoritme Genetika [6]

- 1: **Input:** A problem instance
- 2: $t := 0$;
- 3: Randomly generate an initial population $P(t)$;
- 4: Evaluate the fitness of each individual of $P(t)$;
- 5: **while** $t < max_gen$ **do**
- 6: Selection $P(t)$;
- 7: Crossover $P(t)$;
- 8: Mutation $P(t)$;
- 9: Evaluation $P(t)$;
- 10: $t := t + 1$;
- 11: **end while**
- 12: **Output:** A (sub-optimal) solution

Algoritme 2 Algoritme Genetika dengan *adaptive mutation* [6]

- 1: $t := 0$;
- 2: Randomly initialize a population of individuals;
- 3: Evaluate the fitness of each individual;
- 4: Initialize the mutation ratio for each mutation operator equally;
- 5: **repeat**
- 6: Assign various mutation operators according to their mutation ratios;
- 7: Employ the crossover operator to create offspring;
- 8: Mutate each offspring according to one of the four mutation operators;
- 9: Evaluate the fitness of each offspring;
- 10: Calculate the progress value of each operator according to Eq. (4).
- 11: Update the mutation ratios of mutation operators according to their progress value Eq. (5);
- 12: Select best individuals according to their fitness to form the next population;
- 13: $t := t + 1$;
- 14: **until** the stop condition is satisfied

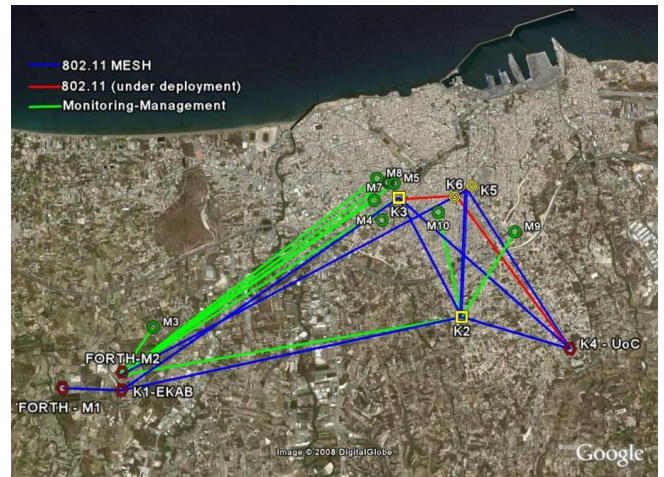
IV. METODOLOGI

A. Sumber Data

Gbr. 1 merupakan sumber data dalam makalah ini, yang mengacu pada MMN dengan luas area sekitar 60 km² dan berisi 15 node di kota Heraklion, Crete, sebagai model kajian makalah. Topologi jaringan *mesh* yang digunakan pada makalah ini ditunjukkan pada Gbr. 1.

B. Definisi Operasional

Konsep yang diangkat dalam makalah ini adalah optimasi *routing*. Optimasi merupakan pencarian konfigurasi optimum, yaitu mencari nilai minimum dan nilai maksimum. Dalam *routing*, tujuan optimasi adalah untuk mencapai kualitas pengiriman data dari satu titik ke titik lainnya melalui perantara berupa sebuah perangkat keras yang bernama *router*.



Gbr. 1 Metropolitan Mesh Network di kota Heraklion, Crete [2].

TABEL I
INFORMASI LAYANAN NODE

Variabel	Definisi Operasional
Traffic Volume	Jumlah data yang melewati <i>node</i> .
Relay Traffic Volume	Jumlah lalu lintas yang dikirim setelah mengirim data dari <i>node</i> tetangga.
Packet Error Rate	Tingkat kesalahan saluran <i>link</i> .

TABEL II
INFORMASI PESAN BROADCAST GATEWAY

Variabel	Definisi Variabel
Hop Count	Jumlah <i>hop</i> jalur yang ditransfer pada <i>gateway</i> .
Path Bandwidth	<i>Bandwidth</i> maksimum dari jalur ke <i>gateway</i> .
Loop Info	Informasi di jalur <i>route</i> , bukan jarak terdekat

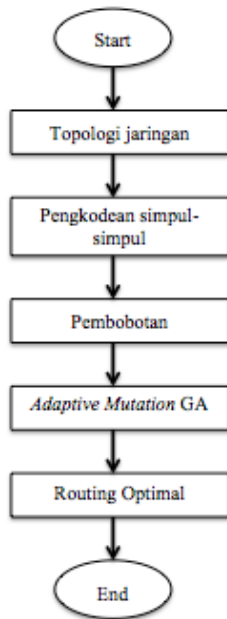
TABEL III
ALGORITME OPTIMASI ROUTING

Variabel	Definisi Operasional
Adaptive Mutation GA	Teknik proses mutasi secara adaptif pada GA dengan memberikan nilai rasio mutasi di awal kemudian dihitung <i>value progress</i> -nya.

Variabel-variabel yang memengaruhi optimasi *routing* antara lain adalah sebagai berikut.

1) *Elemen-elemen Layanan Node*: Setiap *node mesh* di MMN mentransmisikan data ke *router* dan klien tetangga. Data melewati lebih dari satu *hop* dalam jangkauan transmisi. *Node mesh* mengirimkan data melalui *gateway* untuk mengirimkan data di luar *mesh*. Berbagai pesan dilewatkan pada proses ini. Informasi yang ditunjukkan pada Tabel I digunakan sebagai tingkat kerugian keseluruhan lalu lintas dengan menggunakan informasi dari penggunaan layanan pada *node mesh*. *Gateway* pada Tabel II digunakan sebagai elemen evaluasi layanan. Hal ini menggunakan elemen informasi untuk pesan *broadcast gateway* yang menginformasikan *node* tetangga dan informasi yang dibuat dari aplikasinya.

2) *Algoritme Optimasi*: Tabel III merupakan algoritme optimasi yang digunakan untuk mengoptimalkan *routing* pada MMN.



Gbr. 2 Perancangan sistem optimasi routing Metropolitan Mesh Network.

3) *Perancangan Sistem*: Perancangan sistem digunakan untuk mendefinisikan optimasi routing MMN dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Gbr. 2 menunjukkan perancangan sistem optimasi MMN pada AMGA. Perancangan ini dimulai dengan menentukan topologi jaringan yaitu topologi mesh, kemudian dilakukan pengkodean simpul-simpul pada node untuk memudahkan dalam proses komputasi. Setelah dilakukan pengkodean, selanjutnya dilakukan pembobotan pada masing-masing node. Setelah proses tersebut, dilakukan proses routing dengan menggunakan AMGA. Perancangan sistem optimasi MMN ditunjukkan pada Gbr. 2.

V. HASIL DAN PEMBAHASAN

A. *Pengkodean Simpul*

Pengkodean simpul dilakukan dengan mengodekan simpul-simpul pada jaringan dengan bilangan bulat positif 1, 2, 3, ..., n, dengan n adalah banyaknya simpul pada jaringan. Tiap kode dari simpul dianggap sebagai gen pada kromosom, sehingga kromosom merupakan untaian kode-kode dari simpul pada jaringan yang tidak berulang dan merepresentasikan urutan atau jalur seperti ditunjukkan pada Tabel IV.

B. *Pembobotan*

Sebelum proses pengiriman data dilakukan, diperlukan proses pembobotan simpul. Pembobotan didapatkan dengan menggunakan metode layanan node berdasarkan elemen evaluasi yang didefinisikan pada elemen evaluasi layanan node. Nilai yang dihasilkan dari metode evaluasi ini digunakan untuk mengetahui fungsi fitness untuk evaluasi optimal terhadap operasi genetik.

Metode evaluasi dari node dihitung berdasarkan jumlah hop, bandwidth, traffic, dan Packet Error Rate Evaluation (PE).

TABEL IV
PENGKODEAN SIMPUL METROPOLITAN MESH NETWORK

Node	Kode Node/Simpul
M1	1
K1	2
K2	3
K3	4
K5	5
M9	6
M10	7
M2	8
K4	9
K6	10
M3	11
M4	12
M5	13
M7	14
M8	15

TABEL V
EVALUASI BOBOT

Elemen	Simbol	Bobot (0-1)
Hop count	α	0,1
Path bandwidth	β	0,2
Traffic (Own/Relay)	γ	0,4
Packet Error Rate	δ	0,3

$$Hop(H) = \alpha \times \frac{Next\ Hop\ Count}{Current\ Hop\ Count} \tag{3}$$

$$Bandwidth(B) = \beta \times \frac{Path\ Bandwidth}{Max\ Bandwidth} \tag{4}$$

$$Traffic(T) = \gamma \times \frac{|Traffic\ Volume - Relay\ Traffic\ Volume|}{Traffic\ Volume + Relay\ Traffic\ Volume} \tag{5}$$

$$Packet\ Error\ Rate\ Evaluation(PE) = \delta \times Packet\ Error\ Rate \tag{6}$$

Nilai evaluasi dari layanan node adalah sebagai berikut.

$$x = H + B + T + PE \tag{7}$$

Nilai bobot yang digunakan bernilai mulai dari 0 sampai dengan 1. Nilai bobot berjumlah 1. Berikut adalah nilai bobot, ditunjukkan pada Tabel V.

Nilai bobot masing-masing node dengan melakukan perhitungan di atas ditunjukkan pada Tabel VI.

C. *Inisialisasi Populasi*

Tahap ini bertujuan untuk membangkitkan sebuah populasi yang berisi sejumlah kromosom yang telah ditentukan banyaknya. Misalnya, banyaknya kromosom dalam populasi awal telah ditentukan sebanyak 10 kromosom, terlihat seperti pada Tabel VII, kromosom dibentuk secara acak dengan menetapkan gen pertama sebagai node sumber, dalam hal ini node 1.

TABEL VI
BOBOT NODE

Node	Kode	H	B	T	PE	x
M1	1	0,0067	0,0159	0,1123	0	0,1349
K1	2	0,0233	0,0457	0,0371	0	0,1061
K2	3	0,0300	0,0228	0,0284	0,28125	0,3624
K3	4	0,0233	0,0834	0,0378	0	0,1445
K5	5	0,0133	0,1109	0,0220	0	0,1462
M9	6	0,0033	0,0125	0,0182	0	0,0340
M10	7	0,0366	0,0125	0,0484	0	0,0642
M2	8	0,0300	0,1722	0,0321	0	0,2309
K4	9	0,0366	0,1144	0,0210	0	0,1654
K6	10	0,0300	0,0280	0,0242	0,2812	0,3367
M3	11	0,0033	0,0110	0,0429	0,2625	0,3231
M4	12	0,0067	0,0110	0,0585	0	0,0762
M5	13	0,0067	0,0111	0,0192	0	0,0336
M7	14	0,0033	0,2000	0,0042	0	0,2455
M8	15	0,0033	0,0273	0,0396	0	0,0675

TABEL VII
POPULASI AWAL YANG TERBENTUK

Kromosom	Representasi Kromosom
Kromosom 1	1-2-3-8-14-4-5-6-7-9-10-11-12-13-15
Kromosom 2	1-2-3-4-8-14-5-6-7-9-10-11-12-13-15
Kromosom 3	1-2-5-9-3-8-14-4-6-7-10-11-12-13-15
Kromosom 4	1-2-5-9-10-4-8-14-3-6-7-11-12-13-15
Kromosom 5	1-2-4-10-9-3-8-14-5-6-7-11-12-13-15
Kromosom 6	1-2-5-9-10-4-3-8-14-6-7-11-12-13-15
Kromosom 7	1-2-5-9-10-4-3-14-6-7-8-11-12-13-15
Kromosom 8	1-2-5-3-4-8-14-6-7-9-10-11-12-13-15
Kromosom 9	1-2-5-3-8-14-4-6-7-9-10-11-12-13-15
Kromosom 10	1-2-3-5-9-4-8-14-6-7-10-11-12-13-15

D. Evaluasi Fungsi Fitness

Fungsi *fitness* digunakan untuk menentukan seberapa baik individu yang direpresentasikan oleh suatu kromosom. Dalam makalah ini, permasalahan jalur terpendek yaitu mencari jarak terpendek dari 15 *node* dan 20 simpul/busur, diperoleh dengan menghitung nilai *fitness*-nya. Perhitungan *fitness* adalah sebagai berikut.

$$Fitness = \begin{cases} \frac{1}{\sum_{i=1}^{n-1} C_i(g_i, g_{i+1})} & ; \text{jalur valid} \\ 1000 & ; \text{jalur tidak valid} \end{cases} \quad (8)$$

dengan $C_i(g_i, g_{i+1})$ adalah *cost* antara gen g_i dan gen g_{i+1} dalam kromosom dari n gen (simpul).

Fitness dipengaruhi oleh jalur yang terkandung dalam kromosom bersangkutan. Jika terdapat kromosom yang memiliki jalur tidak benar, maka *fitness* akan bernilai 1000 dan hal tersebut berlaku sebaliknya, apabila jalur benar maka nilainya sama dengan fungsi *fitness* yang telah ditentukan. Dari kromosom pada Tabel VII kemudian didapatkan nilai *fitness*-nya dengan perhitungan sebagai berikut.

$$\begin{aligned} \text{Kromosom 1} &= 1/(0,1349+0,1061+0,3624+0,2309+0,2455) \\ &= 1/1,0798 \\ &= 0,9261 \end{aligned}$$

$$\begin{aligned} \text{Kromosom 2} &= 1/(0,1349+0,1061+0,3624+0,1445+0,2309+0,2455) \\ &= 1/1,2243 \\ &= 0,8168 \\ \text{Kromosom 3} &= 1/(0,1349+0,1061+0,1462+0,1654+0,3624+0,2309+0,2455) \\ &= 1/1,3914 \\ &= 0,7187 \\ \text{Kromosom 4} &= 1/(0,1349+0,1061+0,1462+0,1654+0,3367+0,1445+0,2309+0,2455) \\ &= 1/15102 \\ &= 0,6622 \\ \text{Kromosom 5} &= 1/(0,1349+0,1061+0,1445+0,3367+0,1654+0,3624+0,2309+0,2455) \\ &= 1/1,7264 \\ &= 0,5792 \\ \text{Kromosom 6} &= 1/(0,1349+0,1061+0,1462+0,1654+0,3367+0,1445+0,3624+0,2309+0,2455) \\ &= 1/1.8726 \\ &= 0,5340 \\ \text{Kromosom 7} &= 1/(0,1349+0,1061+0,1462+0,1654+0,3367+0,1445+0,3624+0,2455) \\ &= 1/1,6417 \\ &= 0,06091 \\ \text{Kromosom 8} &= 1/(0,1349+0,1061+0,1462+0,3624+0,1445+0,2309+0.2455) \\ &= 1/1,3705 \\ &= 0,7297 \\ \text{Kromosom 9} &= 1/(0,1349+0,1061+0,1462+0,3624+0,2309+0,2455) \\ &= 1/1,226 \\ &= 0,8157 \\ \text{Kromosom 10} &= 1/(0,1349+0,1061+0,3624+0,1462+0,1654+0,1445+0,2309+0,2455) \\ &= 1/1,5359 \\ &= 0,6511 \end{aligned}$$

Hasil perhitungan di atas disusun dalam Tabel VIII.

E. Seleksi

Setelah terbentuk populasi awal, selanjutnya hasil populasi tersebut diseleksi. Metode seleksi yang digunakan dalam program simulasi ini adalah *roulette wheel selection*. Metode ini digunakan untuk melakukan seleksi *roulette* terhadap populasi (*chroms*) dengan nilai *fitness* (*fitness*) dan mengembalikan sejumlah kromosom dengan jumlah yang diinginkan (*count*). Besarnya probabilitas sebuah kromosom ditentukan oleh nilai *fitness*-nya, Semakin kecil nilai *fitness* sebuah kromosom, semakin besar kemungkinan kromosom tersebut untuk terseleksi. Total *fitness*, Tf , dihitung menggunakan (9).

$$Tf = \sum F_k \quad k = 1,2,3, \text{popsize} \quad (9)$$

Dari persamaan di atas didapatkan hasil sebagai berikut.

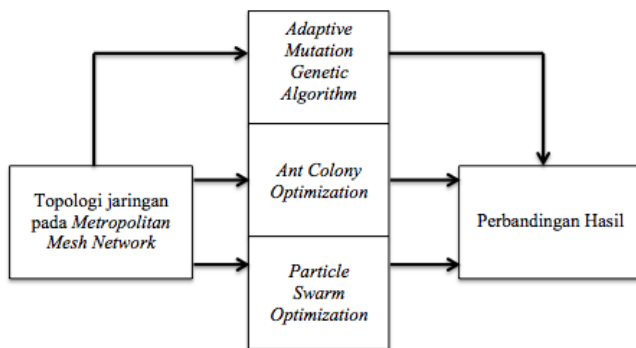
$$\begin{aligned} Tf &= 0,9261+0,8168+0,7187+0,6622+0,5792+0,5340 \\ &\quad +0,6091+0,7297+0,8157+0,6511 \\ &= 6,5036 \end{aligned}$$

TABEL VIII
FITNESS MASING-MASING KROMOSOM

Kromosom	Representasi Kromosom	Fitness
Kromosom 1	1-2-3-8-14-4-5-6-7-9-10-11-12-13-15	0,9261
Kromosom 2	1-2-3-4-8-14-5-6-7-9-10-11-12-13-15	0,8168
Kromosom 3	1-2-5-9-3-8-14-4-6-7-10-11-12-13-15	0,7187
Kromosom 4	1-2-5-9-10-4-8-14-3-6-7-11-12-13-15	0,6622
Kromosom 5	1-2-4-10-9-3-8-14-5-6-7-11-12-13-15	0,5792
Kromosom 6	1-2-5-9-10-4-3-8-14-6-7-11-12-13-15	0,5340
Kromosom 7	1-2-5-9-10-4-3-14-6-7-8-11-12-13-15	0,6091
Kromosom 8	1-2-5-3-4-8-14-6-7-9-10-11-12-13-15	0,7297
Kromosom 9	1-2-5-3-8-14-4-6-7-9-10-11-12-13-15	0,8157
Kromosom 10	1-2-3-5-9-4-8-14-6-7-10-11-12-13-15	0,6511

TABEL IX
HASIL PERBANDINGAN BERDASARKAN WAKTU ITERASI

Metode Pengujian		
AMGA	ACO	PSO
0,046556	1,447164	2,6800



Gbr. 3 Skema pengujian AMGA pada Metropolitan Mesh Network.

Dari total *fitness* kemudian ditentukan peluang individu masing-masing kromosom.

$$\begin{aligned}
 \text{Kromosom 1} &= 0,9261 \times \frac{100}{6,5036} = 14,2398056461 \\
 \text{Kromosom 2} &= 0,8168 \times \frac{100}{6,5036} = 12,5591979827 \\
 \text{Kromosom 3} &= 0,7187 \times \frac{100}{6,5036} = 11,0508026324 \\
 \text{Kromosom 4} &= 0,6622 \times \frac{100}{6,5036} = 10,1820530168 \\
 \text{Kromosom 5} &= 0,5792 \times \frac{100}{6,5036} = 8,90583676673 \\
 \text{Kromosom 6} &= 0,5340 \times \frac{100}{6,5036} = 8,2108370749 \\
 \text{Kromosom 7} &= 0,6091 \times \frac{100}{6,5036} = 9,3655821391 \\
 \text{Kromosom 8} &= 0,7297 \times \frac{100}{6,5036} = 11,2199397257 \\
 \text{Kromosom 9} &= 0,8157 \times \frac{100}{6,5036} = 12,5422842733 \\
 \text{Kromosom 10} &= 0,6511 \times \frac{100}{85,4212} = 10,0113783135
 \end{aligned}$$

F. Pengujian

Pengujian dilakukan untuk mengetahui metode AMGA, layak digunakan untuk optimasi *routing* MMN atau tidak. Skema pengujian optimasi *routing* dilakukan dengan cara membandingkan metode yang diusulkan (AMGA) dengan

metode *Ant Colony Optimization* (ACO) dan *Particle Swarm Optimization* (PSO). Proses metode pembandingan ditunjukkan dalam Gbr. 3.

Berdasarkan Tabel IX, waktu iterasi berbeda-beda untuk masing-masing metode. Hal ini disebabkan proses iterasi untuk masing-masing metode berbeda pula. Berdasarkan proses iterasi, terdapat empat jenis metode, di antaranya AMGA, ACO, dan PSO. Ketiga metode tersebut mampu memberikan hasil *routing* yang optimal dengan bobot yang telah diberikan. Pada proses AMGA, waktu iterasi yang diperoleh sebesar 0,046556 detik dengan operasi mutasi ditentukan di awal yaitu sebesar 0,000005782 dan probabilitas *crossover* sebesar 0,000947. Sedangkan proses ACO dan PSO menghasilkan jalur dengan memunculkan semua *node* tanpa memikirkan jalur tersebut terhubung atau tidak.

VI. KESIMPULAN

Berdasarkan hasil pembahasan dan pengujian, disimpulkan bahwa AMGA mampu mengoptimalkan *routing* pada MMN dengan menentukan probabilitas mutasi di awal sebesar 0,000005782 dengan probabilitas *crossover* sebesar 0,000947, sehingga mampu mengurangi atau menghindari adanya *premature convergence*.

REFERENSI

- [1] Kessler, Gary C. & Train, David A. (1999). Metropolitan Area Networks: Concepts, Standards, and Services. Network : McGraw-Hill.
- [2] Miniwatss Marketing Group. (2017). <http://www.internetworldstats.com/stats.htm>, 12 Maret 2017, pk. 15.19.
- [3] Kumar, R. & Kumar, M. (2010). Exploring Genetic Algorithm for Shortest Path Optimization in Data Networks. *Global Journal of Computer Science and Technology*. Vol. 10 Issues 11 (Ver. 1.0), p 8-12.
- [4] Lin, X. H., Kwok Y. K., & Lau V. K. N. (2002). A genetic algorithm based approach to route selection and capacity flow assignment. *Computer Communications*. p 96-974.
- [5] Yun, Y. (2006). *Hybrid genetic algorithm with adaptive local search scheme*. Computer & Industrial Engineering 51:p.821-838.
- [6] Korejo, Imtiza Ali. (2010). Adaptive mutation Operators for Evolutionary Algorithms. Department of Computer Science for the degree of Doctor of Philosophy.
- [7] Yoon, Chang-Pyo & Ryou, Hwang-Bin. (2011). A Genetic Algorithm for the Routing Protocol of Wireless Mesh Networks. *Information Science and Applications (ICISA), 2011 International Conference on Jeju Island, South Korea*. IEEE. p.1-6.
- [8] Koyama, Akio, Toshiki Nishie, Jupei Arai, & Leonard Barolli. (2005). A New Quality of Service Multicast Routing Protocol Based on Genetic Algorithm. *Proceedings of the 2005 11th International Conference on Parallel and Distributed System (ICPADS'05)*. IEEE. p.1-6.
- [9] Seetaram, J., & Kumar, P Satish. (2016). An Energy Aware Genetic Algorithm Multipath Distance Vector Protocol for Efficient Routing. *Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on Chennai, India*. IEEE. p.1975-1980.
- [10] Guo, Lejiang & Tang, Qiang. (2010). An Improved Routing Protocol in WSN with Hybrid Genetic Algorithm. *Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, IEEE. p.289-292.
- [11] Apetroaei, Ioana., Ionut-Alexandru Oprea, Bogdan-Eugen Proca, & Laura Gheorghie. (2011). Genetic algorithms applies in routing protocols for wireless sensor networks. *Roedunet International Conference (RoEduNet)*. IEEE.
- [12] Rajakumar, B. R., & George, Dr. Aloysius. (2012). A New Adaptive Mutation Technique for Genetic Algorithm. *IEEE International Conference on Computational Intelligence and Computing Research*.