

# PENERAPAN ALGORITMA EA-SHORT PADA PROTOKOL ROUTING AOMDV UNTUK MENEMUKAN RUTE HANDAL BERBASIS ENERGI DI JARINGAN MANET

(Application of EA-SHORT Algorithm on AOMDV Routing Protocol to Find Reliable Energy Based Routes in Manet Networks)

B. Nurwahyu Hairani, Andy Hidayat Jatmika\*, Fitri Bimantoro  
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA  
Email: baiqhaira22@gmail.com, [andy, bimo]@unram.ac.id

## Abstract

Mobile Ad Hoc Network (MANET) is a stand-alone wireless network, consisting of several nodes that can move in all directions freely. The routing protocol used as the object of this study is AOMDV. This research carries out an energy calculation process at a node that can be used as a new energy-based route by utilizing topology changes. Energy efficiency in the routing protocol can be done using the Energy-Aware SHORT (EA-SHORT) method. The main purpose of the EA-SHORT algorithm is to distribute the traffic load of each node on MANET fairly, by extending the life of the node and network by routing packets through nodes that have enough remaining power and avoiding nodes that have low power. AOMDV performance will be compared with AOMDV which has been modified with EA-SHORT measured from the specified parameter values. Analysis results show that the application of the EA-SHORT algorithm in the efficiency of route search succeeded in improving performance. The results of throughput on EA-AOMDV increased by 13.904% for an area of 500x500 m<sup>2</sup> and 13.905% at 1000x1000 m<sup>2</sup>. The packet delivery ratio increased by 0.91% and 2.273%. Average end-to-end delay decreased by 20.482% and 18.734%.

**Keywords:** MANET, AOMDV, Energy aware SHORT, EA-AOMDV, Routing protocol

\*Penulis Korespondensi

## 1. PENDAHULUAN

Mobile Ad Hoc Network (MANET) ialah jaringan nirkabel yang berdiri sendiri, terdiri dari sejumlah *node* yang mampu bergerak ke segala arah secara bebas. Akibat pergerakan yang bebas tersebut topologi jaringan MANET cenderung sering berubah [1]. *Node-node* pada jaringan MANET terdiri dari *node* sumber sebagai pengirim dan *node* tujuan sebagai penerima, dan antar *node* pengirim dan penerima disebut *node* tengah yang berfungsi sebagai perantara kedua *node* tersebut. Masing-masing *node* pada jaringan MANET memiliki level yang sama, artinya semua *node* berperan sebagai *router* [2].

Dalam hal komunikasi antar *node*, perlu adanya proses *routing*. Proses ini biasanya dikerjakan oleh suatu protokol *routing* agar *node-node* mampu saling berkiriman data. Pada penelitian ini objek yang digunakan adalah protokol *routing Ad Hoc On-Demand Multipath Distance Vektor* (AOMDV). AOMDV merupakan protokol *routing* yang bersifat reaktif AOMDV memiliki mekanisme *route discovery* dan *route maintenance* [2].

Pencarian rute pada protokol *routing* AOMDV dilakukan dengan cara memperhitungkan jarak berdasarkan jumlah hop. Apabila pada *routing table* sudah tidak ada rute cadangan lagi, maka AOMDV akan melakukan pencarian rute kembali dari awal. Pada AOMDV adanya perpindahan rute menyebabkan banyak terjadi paket data drop dalam proses transisi antar rute. Sehingga sangat diperlukan rute yang stabil untuk mengatasi masalah akibat mobilitas *node* tersebut. Perubahan topologi dan mobilitas *node* dapat mengakibatkan rute memiliki kemungkinan terputus yang disebabkan *node* keluar dari jangkauan sinyal transmisi [4]. Namun di sisi lain perubahan topologi ini dapat dimanfaatkan untuk melakukan *redirection* agar mendapatkan rute baru yang handal dengan menghitung aspek energi pada *node*. Sehingga diharapkan mampu meningkatkan *throughput* dan menurunkan delay.

Setiap *wireless mobile device* pada MANET eksistensinya dibatasi oleh adanya ketersediaan energi. Bila energi sebuah *wireless mobile device* habis, maka *wireless mobile device* tersebut tidak eksis lagi pada

MANET [5]. Keterbatasan energi terjadi karena setiap *node* selalu bergantung kepada energi yang terbatas yang dimilikinya dalam rangka menyelesaikan tugas utamanya. Masalah yang dapat dilihat yakni skalabilitas dalam kebijakan *overhead* yang berimbas pada jumlah *node* yang beroperasi di jaringan, juga terkait erat dengan konsumsi energi karena jumlah paket kontrol yang lebih tinggi berarti lebih banyak konsumsi energi yang dihabiskan dalam transmisi, penerimaan (*reception*) dan mendengar (*overhearing*)[6]. Karena adanya keterbatasan energi terdapat kemungkinan terjadi putus komunikasi di tengah jalan pada *path* dikarenakan kehabisan energi. Pada penelitian ini diusulkan adanya proses perhitungan energi pada suatu *node* yang dapat digunakan sebagai rute baru yang berbasis energi dengan memanfaatkan perubahan topologi. Energi disini merupakan kekuatan aliran listrik yang digunakan dalam transfer paket data yang dinyatakan dengan satuan *joule*. Efisiensi energi pada *routing protocol* dapat dilakukan dengan menggunakan metode EA-SHORT (*Energy aware – Self Healing and Optimizing Routing Techniques*). Tujuan utama dari algoritma EA-SHORT adalah untuk menghemat energi pada MANET, dengan cara memperpanjang masa pakai *node* dan jaringan dengan merutekan paket melalui *node* yang memiliki sisa daya yang cukup dan menghindari *node* yang memiliki daya rendah. Sehingga dapat meningkatkan kinerja dan konservasi daya pada MANET.

Penelitian ini melakukan modifikasi pada kerangka protokol *routing* AOMDV untuk mengoptimalkan rute yang mempertimbangkan aspek energi dengan menggunakan algoritma EA-SHORT. Kinerja AOMDV dengan EA-SHORT akan diukur dari nilai parameter yang telah ditentukan yaitu *Packet Delivery Ratio* (PDR), *throughput* dan *end-to-end delay* (E2D) yang dilakukan pada sejumlah skenario berbeda, dimana pada penelitian ini akan dicari rute manakah yang paling handal di antara rute-rute yang disediakan oleh protokol *routing* yang digunakan.

## 2. TINJAUAN PUSTAKA

Penelitian terkait sebelumnya adalah mengenai analisis performansi protokol *routing* AOMDV dan AODV pada MANET [7]. Pada penelitian tersebut dilakukan perbandingan dan evaluasi protokol berdasarkan parameter uji yaitu PDR, *routing overhead*, *Normalized Routing Load* (NRL), *throughput* dan *average delay*. Hasil simulasi pada skenario perubahan jumlah *node* menunjukkan bahwa AODV lebih baik dari AOMDV dilihat dari parameter performansi PDR

sebesar 4.71%, *throughput* sebesar 9.64%, *routing overhead* sebesar 2936.12% dan NRL sebesar 2993.27%. Namun protokol *routing* AOMDV memiliki performansi lebih baik dari AODV, dilihat dari parameter performansi *average delay* sebesar 423.198% lebih baik. Selanjutnya hasil simulasi pada skenario perubahan kecepatan *node* juga menunjukkan bahwa AODV lebih baik dari AOMDV dilihat dari parameter performansi PDR sebesar 13.17%, *throughput* sebesar 16.51%, *routing overhead* sebesar 1480.17% dan NRL sebesar 1669.69%. Dan protokol *routing* AOMDV memiliki performansi lebih baik dari AODV, dilihat dari parameter performansi *average delay* sebesar 238.256% lebih baik. Sehingga dapat disimpulkan protokol *routing* AOMDV hanya lebih unggul dari segi parameter *average delay* daripada protokol *routing* AODV. Persamaan dari penelitian terkait dengan penelitian yang dilakukan yaitu menggunakan protokol *routing* AOMDV. Sedangkan perbedaan penelitian terkait dengan penelitian yang dilakukan yaitu pada penelitian yang dilakukan menggunakan metode EA-SHORT dan penelitian sebelumnya tidak menggunakan metode.

Penelitian terkait selanjutnya adalah melakukan analisis perbandingan protokol *routing*, AOMDV dan AODV yang telah dimodifikasi dengan penambahan algoritma perhitungan *Signal to Noise Ratio* (SNR) ke dalam mekanisme pencarian jalur [8]. Pada skenario perubahan besar paket, *throughput* mengalami peningkatan sebesar 20%, penurunan NRL sebesar 3%, dan penurunan *packet loss* sebesar 0,37%. Pada skenario perubahan jumlah *traffic*, *throughput* mengalami peningkatan sebesar 12%, penurunan NRL sebesar 0,5%, dan penurunan *packet loss* sebesar 0,21%. Penambahan algoritma *Signal to Noise Ratio* (SNR) terbukti dapat meningkatkan performansi protokol *routing* baik itu *throughput*, NRL dan *packet loss* yang lebih baik. Persamaan dari penelitian terkait dengan penelitian yang dilakukan yaitu menggunakan protokol *routing* AOMDV. Sedangkan perbedaan penelitian terkait dengan penelitian yang dilakukan yaitu pada penelitian yang dilakukan menggunakan metode EA-SHORT dan penelitian sebelumnya menggunakan metode SNR.

Penelitian terkait selanjutnya adalah penelitian untuk mengimplementasikan metode perbaikan protokol *routing* [1]. Di mana penelitian *Optimalisasi routing* dengan menggunakan metrik yang berbeda seperti panjang lintasan dan pemakaian energi di sepanjang jalur didefinisikan. Kemudian diusulkan sebuah kerangka kerja SHORT untuk MANET. Saat menggunakan SHORT, semua *node* tetangga

memantau rute dan mencoba mengoptimalkannya bila ada jalur yang lebih baik. Pada penelitian ini, algoritma SHORT diterapkan untuk protokol *routing* AODV dan DSR. Algoritma SHORT meningkatkan optimalisasi *routing* dengan memonitor jalur *routing* secara terus menerus dan secara bertahap mengarahkan rute untuk lebih optimal. Hasil simulasi menunjukkan bahwa tingkat pengiriman yang lebih tinggi dan *network lifetime* yang lebih panjang. Tingkat pengiriman naik sekitar 8% pada AODV-SHORT dan 6% pada DSR-SHORT, serta *overhead* mengalami penurunan sekitar 10%. Persamaan dari penelitian terkait dengan penelitian yang dilakukan yaitu algoritma EA-SHORT. Sedangkan perbedaan penelitian terkait dengan penelitian yang dilakukan yaitu pada penelitian yang dilakukan menggunakan protokol *routing* AOMDV namun penelitian sebelumnya menggunakan protokol *routing* DSR dan AODV.

Penelitian terkait selanjutnya tentang peningkatan efisiensi yang dilakukan pada protokol *routing* AOMDV dengan menggunakan metode PA-SHORT di jaringan MANET [4]. Penelitian ini membandingkan kinerja protokol *routing* AOMDV dengan protokol *routing* Path Aware-AOMDV (PA-AOMDV). Protokol *routing* PA-AOMDV diperoleh melalui modifikasi kinerja protokol AOMDV dengan algoritma Path Aware SHORT. Algoritma Path Aware SHORT adalah metode untuk mengurangi jumlah hop. SHORT meningkatkan optimisasi rute dengan memantau rute dan mengoptimalkan rute ini yang memiliki jalur yang lebih baik. Kinerja kedua protokol akan dilihat berdasarkan parameter nilai *throughput*, *average E2D*, *packet delivery ratio*, dan *routing overhead*. Hasil menunjukkan bahwa *throughput* meningkat untuk 50 *node* adalah 61,84% dan untuk 100 *node* adalah 45,2%, *average E2D* menurun untuk 50 *node* adalah 0,066% dan untuk 100 *node* 0,12%, *packet delivery ratio* meningkat untuk 50 *node* adalah 60,87% dan untuk 100 *node* 82,02%, dan *routing overhead* menurun untuk 50 *node* adalah 67,07% dan 100 *node* 45,36%. Persamaan dari penelitian terkait dengan penelitian yang dilakukan yaitu menggunakan simulasi area 500m x 500m dan 1000m x 1000m serta menggunakan protokol *routing* AOMDV. Sedangkan perbedaannya yaitu pada penelitian yang dilakukan menggunakan algoritma EA-SHORT namun penelitian sebelumnya menggunakan algoritma PA-SHORT.

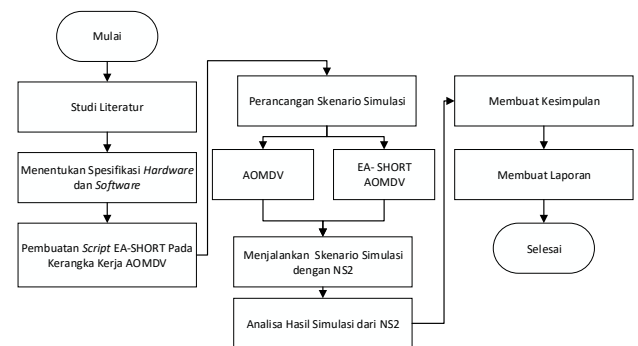
Penelitian terkait selanjutnya telah dilakukan analisis perbandingan kinerja protokol *routing* TORA dan TORA dengan menggunakan algoritma SHORT [9]. Hasil simulasi menunjukkan bahwa *throughput* yang lebih tinggi meningkat hingga 3-50%, *E2D* yang lebih

rendah hingga 50%, *delay* mengalami penurunan sebesar 50%, penurunan *packet loss* bervariasi dari 12-57%, perbaikan energi sekitar 43%, dan *lifetime* jaringan yang lebih tinggi. TORA yang dimodifikasi mengungguli protokol *routing* TORA dalam hal *throughput*, *packet loss*, *end-to-end delay*, dan energi. Persamaan dari penelitian terkait dengan penelitian yang dilakukan yaitu menggunakan algoritma EA-SHORT. Sedangkan perbedaannya yaitu pada penelitian yang dilakukan menggunakan protokol *routing* AOMDV namun penelitian sebelumnya menggunakan protokol *routing* TORA.

### 3. METODE PENELITIAN

#### 3.1. Diagram Alir Penelitian

Diagram alir penelitian penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

#### a. Studi literatur

Pada tahap studi literatur, dilakukan pengkajian mengenai berbagai jurnal dan penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan penulis. Pada tahap ini, dilakukan proses pembelajaran materi penelitian melalui pustaka-pustaka baik berupa buku maupun jurnal ilmiah.

#### b. Menentukan spesifikasi hardware dan software

Pada penelitian ini uji coba dan simulasi dilakukan dengan menggunakan Laptop HP 14-bs538TU dengan spesifikasi hardware (perangkat keras) yang dapat dilihat pada Tabel I.

TABEL I. SPESIFIKASI HARDWARE YANG DIGUNAKAN

Komponen	Spesifikasi
CPU	Intel(R) Celeron(R) CPU N3060 @1.60 GHz
Memori	4.00 GB (2 GB untuk virtual simulator)

Hard disk	500 GB (10 GB untuk <i>virtual simulator</i> )
-----------	--

Adapun *software* yang digunakan pada penelitian ini adalah sebagai berikut :

- *Oracle VM VirtualBox 6.0*, digunakan untuk menjalankan simulator. Di mana alokasi penggunaan memori komputer yang digunakan sebesar 2 GB dan penggunaan *harddisk* untuk menyimpan data sebesar 10GB.
- *Network Simulator 2 (NS2)* versi 2.35 untuk melakukan simulasi MANET.
- *Linux Ubuntu 16.04 LTS 64-bit* sebagai sistem operasi.
- *Microsoft Excel* untuk membuat grafik hasil simulasi.
- *Microsoft Visio* untuk membuat diagram
- *Microsoft Word* untuk dokumentasi laporan

#### c. Pembuatan *script* EA-SHORT AOMDV

Pembuatan *script* EA-SHORT AOMDV dilakukan dengan cara memodifikasi kerangka protokol *routing* AOMDV dengan menyisipkan Algoritma EA-SHORT pada protokol *routing* AOMDV. Penulisan kode dilakukan melakukan perhitungan berbasis energi pada *node* untuk menemukan rute hemat energi yang handal. Jika rute telah ditemukan maka jalur tersebut yang akan digunakan untuk melakukan pengiriman data.

#### d. Perancangan skenario simulasi

Pada bagian ini, peneliti melakukan perancangan skenario simulasi jaringan MANET untuk melakukan analisis performa algoritma EA-SHORT pada protokol *routing* AOMDV dengan kondisi, yaitu protokol AOMDV belum menggunakan EA-SHORT dan kondisi dimana AOMDV sudah dimodifikasi menggunakan EA-SHORT.

#### e. Menjalankan skenario simulasi dengan NS2

Pada tahap ini perancangan skenario simulasi yang telah dibuat akan dijalankan pada simulator NS2 sehingga diperoleh hasil berdasarkan skenario simulasi yang diujikan. Terdapat beberapa parameter uji yang digunakan untuk melakukan analisa hasil pengujian dari skenario simulasi yang dilakukan yaitu *throughput*, *Packet Delivery Ratio (PDR)* dan *end to end delay (E2D)*.

#### f. Analisis hasil simulasi dari NS2

Pada tahap ini dilakukan proses analisis terhadap hasil yang diperoleh dari proses pengujian skenario

penelitian yang dilakukan, dengan membandingkan hasil simulasi protokol *routing* AOMDV dan protokol *routing* EA-SHORT AOMDV dengan percobaan pengiriman data dengan mobilitas *node* berdasarkan jumlah *node*, luas area, dan kecepatan *node*. Menggunakan *script* AWK akan dianalisis parameter uji yang telah ditentukan. Hasil Analisa simulasi ditampilkan dalam bentuk grafik.

#### g. Membuat kesimpulan

Pada tahap ini akan disimpulkan hasil modifikasi protokol *routing* AOMDV dan pengujian skenario, untuk mengetahui performa terbaik dari skenario yang telah diuji.

#### h. Membuat laporan

Pada tahap ini akan dilakukan pembuatan laporan dokumentasi terhadap penelitian yang telah dilakukan. Dokumentasi laporan ini diharapkan agar dapat membantu untuk melakukan penelitian berikutnya.

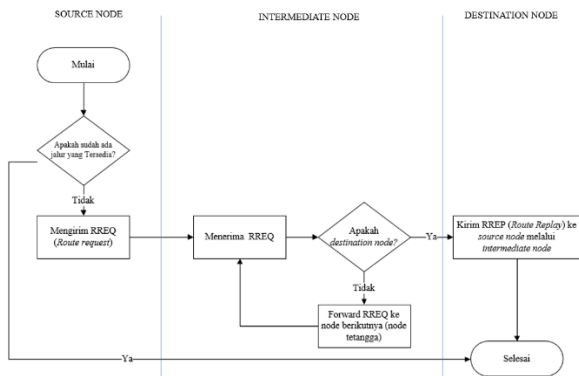
### 3.2. Algoritma Penemuan Rute Menggunakan Protokol *Routing* AOMDV

AOMDV merupakan konsep *distance vector* yakni pencarian rute berdasarkan jarak terdekat dan menggunakan pendekatan *hop-by-hop routing*[4]. AOMDV menemukan rute berdasarkan permintaan dengan menggunakan prosedur penemuan rute. Keuntungan adanya pada satu kali pencarian rute adalah ditemukannya beberapa rute, jika rute utama mengalami kerusakan maka akan digunakan rute lain yang masih ada pada *routing table* tanpa harus mencari rute lagi dari awal. Pencarian rute dari awal hanya dilakukan ketika semua rute yang ditemukan sudah tidak valid lagi.

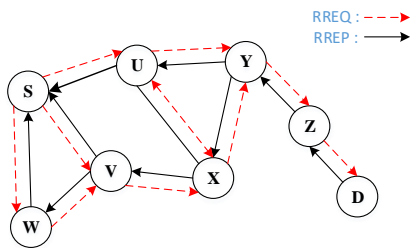
Misalkan, ketika *node* sumber *A* akan mengirimkan suatu paket data ke *node* tujuan *B*, pertama-tama *node* akan melihat tabel *routing* apakah sudah ada jalur yang tersedia. Jika jalur dari *A* ke *B* sudah tersedia dan siap digunakan, maka paket data akan dikirim menggunakan jalur yang sudah ada tersebut. Jika jalur dari *A* ke *B* belum tersedia, maka akan dimulai proses pencarian jalur dengan mengirimkan pesan RREQ secara *broadcast*. AOMDV menghitung *multiple path* dan melihat *route advertisement* untuk menentukan jalur alternatif dari sumber ke tujuan selama proses pencarian jalur. Untuk menghindari kemungkinan *loop*, digunakan *advertised hop count* yang ada dalam *routing table*. *Advertised hop count* dari sebuah *node* sumber *A* untuk tujuan *B* adalah kumpulan *hop count* maksimal untuk banyak jalur untuk tujuan *B*. *Advertised hop count* dimulai

setiap ada pembaruan *sequence number*. Dengan begitu, AOMDV hanya mengambil rute alternatif dengan jumlah *hop* yang lebih sedikit[10]. Pada AOMDV, propagasi RREQ dari sumber ke tujuan menetapkan beberapa *reverse paths* baik di *node* tengah maupun tujuan. Beberapa RREP melintasi lintasan mundur ini untuk membentuk beberapa lintasan maju ke tujuan di *node* sumber dan tengah. AOMDV juga menyediakan *node* tengah dengan jalur alternatif karena mereka bermanfaat untuk mengurangi frekuensi penemuan rute[11].

Alur proses *route discovery* pada protokol AOMDV dapat dilihat pada Gambar 2.



Gambar 2. Flowchart route discovery pada AOMDV

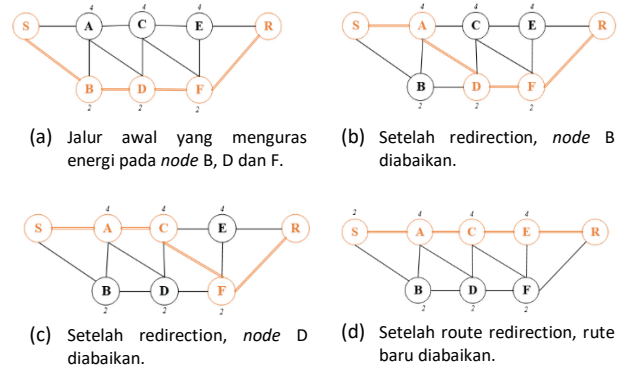


Gambar 3. Proses route discovery pada AOMDV

Proses *penemuan rute* pada AOMDV dapat dilihat pada Gambar 3, dimana *node S* merupakan *node sumber* dan *node D* merupakan *node tujuan*. Saat *node S* ingin mengirimkan paket ke *node D*. Pertama kali *node S* mengirimkan paket RREQ secara *broadcast* ke *node* tetangganya yaitu *node U*, *node V* dan *node W*. Karena *node U*, *node V* dan *node W* bukan *node* tujuannya, maka ketiga *node* tersebut akan meneruskan paket RREQ secara *broadcast* ke *node* tetangganya kecuali *source node*. Setelah itu *node U*, *node V* dan *node W* akan melakukan pengaturan rute cadangan. Proses tersebut akan dilakukan secara terus menerus sampai paket RREQ diterima oleh *node* tujuannya (*node D*). Kemudian *node D* akan memberikan paket RREP sebagai jawaban dari paket RREQ yang diterimanya.

### 3.3. Algoritma EA-SHORT

Tujuan dari penyeimbangan muatan yang sadar energi adalah untuk mendistribusikan secara adil beban lalu lintas di antara semua *node* yang berpartisipasi di jaringan [1]. Misalnya, pertimbangkan bagian dari MANET ditunjukkan pada Gambar 3.4 (a), jalur S-B-D-F-R adalah jalur optimal untuk koneksi dari S ke R. Metrik untuk optimalitas dapat berupa jumlah *hop* untuk rute jalur terpendek. Dengan demikian, *node B*, *D* dan *F* akan terus digunakan dalam meneruskan lalu lintas, meninggalkan *node* lain bebas dari beban lalu lintas. Akibatnya, tingkat energi residu dari *node* menjadi sangat bervariasi. Jika perutean tidak sadar energi, itu akan tetap menggunakan jalur untuk koneksi S-R. *Node B*, *D* dan *F* pada akhirnya akan kehabisan persediaan baterai dan mati lebih awal. Namun, skema perutean yang sadar energi akan mencoba mengalihkan lalu lintas ke *node* lain. Pada penelitian ini diusulkan skema EA-SHORT berdasarkan *route redirection* (pengalihan rute). Dengan operasi *redirection* lokal yang berurutan, rute secara bertahap akan konvergen ke rute alternatif *node disjoint*.



Gambar 4. Contoh operator pengalihan rute lokal berturut-turut.

(Tautan yang ditunjukkan pada gambar adalah tautan nirkabel.)

Perhatikan contoh yang ditunjukkan pada Gambar 4 (a), *node B*, *D* dan *F*, ketiganya memiliki tingkat energi yang relatif rendah. Setelah adanya *route redirection*, *node B* diabaikan dan jalur S-B-D-F-R menjadi S-A-D-F-R, yang ditunjukkan pada Gambar 3.3 (b). Kemudian, setelah *redirection* pertama dan seterusnya seperti pada Gambar 3.3 (c) memberikan dampak, jalur S-A-C-E-R yang baru akan menggantikan yang asli, seperti yang ditunjukkan pada Gambar 3.3 (d).

Berikut ini langkah-langkah pencarian rute dengan algoritma EA-SHORT. Jika suatu *node* (i) mengetahui kedatangan paket P. Maka;

- a. *Node* akan melakukan pengecekan rute untuk paket P pada *tabel overhear*. *Tabel overhear*

- merupakan tabel yang digunakan oleh *node* untuk menyimpan rute suatu paket.
- b. Jika tidak ada rute yang cocok. Tambahkan *input* variabel  $e'$ , dimana rute ( $e'$ ) sama dengan rute ( $p$ ), *sequence number* ( $e'$ ) sama dengan *sequence number* ( $p$ ) lalu jadikan rute ( $e'$ ) sebagai list rute pertama pada *tabel overhear* dengan isi ( $e'$ ) berupa *hope count* ( $p$ ), level energi ( $p$ ), dan nomer id ( $p$ ). proses selesai.
  - c. Misalkan rute ( $p$ ) ditemukan pada *input*-an  $e$ , dengan kondisi *sequence number* ( $p$ ) kurang dari *sequence number* ( $e$ ), maka  $p$  akan diabaikan, dan proses selesai.
  - d. Jika *sequence number* ( $p$ ) lebih besar dari *sequence number* pada ( $e$ ), maka perbaharui  $e$  dengan ketentuan yakni *sequence number* ( $e$ ) sama dengan *sequence number* ( $p$ ), list rute pada *tabel overhear* ( $e$ ) direset sehingga memiliki satu *input*-an rute dengan isi *hope count* ( $p$ ), level energi ( $p$ ). nomer id ( $p$ ) dan proses selesai.
  - e. Jika *sequence number* ( $p$ ) sama dengan *sequence number* ( $e$ ), maka:
    1. Tambahkan *input*-an  $\langle hc(p), lvl(p), nid(p) \rangle$  ke *ov-list* ( $e$ ).
    2. Jika daftar *overhear* pada ( $e$ ) memiliki tiga *input*-an yakni  $A, B, C$  yang memenuhi kondisi berikut, maka *subpath* ditemukan.
      - *Hope count* pada ( $C$ ) sama dengan *hope count* ( $B$ )+1 dan sama dengan *hope count* ( $A$ )+2
      - Level energi pada (*node*  $i$ ) lebih besar atau sama dengan jumlah maksimal energi pada ( $A$ ) dan level energi pada ( $C$ );
      - level energi pada (*node*  $i$ ) – level energi ( $B$ )  $\geq 2$ , maka akan diaktifkan sub-rute baru, dan hapus *input*-an ( $e$ ) dari *tabel overhear*.
    3. Jika daftar *overhear* pada ( $e$ ) memiliki dua *input*-an, yakni  $A$  dan  $B$ , dengan kondisi *hope count* ( $B$ ) sama dengan *hope count* ( $A$ )+1 dan level energi pada *node* ( $i$ )  $\geq$  maksimal level energi ( $A$ ) dan dan level energi ( $B$ )+2, tambahkan indikator  $I$  pada daftar tunggu indikator (*Waiting Indicator List*), dengan kandidat ( $I$ ) sama dengan  $B$ . *sequence number* ( $I$ ) disamakan dengan ( $e$ ). rute ( $I$ ) sama dengan rute ( $e$ ), maka proses selesai.
    4. Jika daftar *overhear* ( $e$ ) memiliki dua *input*-an  $B$  dan  $C$ , dengan kondisi *hope count* ( $C$ ) sama dengan *hope count* ( $B$ )+1 dan level energi pada *node* ( $i$ )  $\leq$  dari maksimal level

energi pada ( $B$ )+2 dan level energi ( $C$ ), maka *node*  $I$  akan menyebarkan pesan SHORT untuk menginformasikan paket  $Q$  dengan ketentuan kandidat ( $Q$ ) sama dengan  $B$ , *sequence number* pada ( $Q$ ) sama dengan *sequence number* pada ( $e$ ), dan rute ( $Q$ ) sama dengan rute pada ( $e$ ).

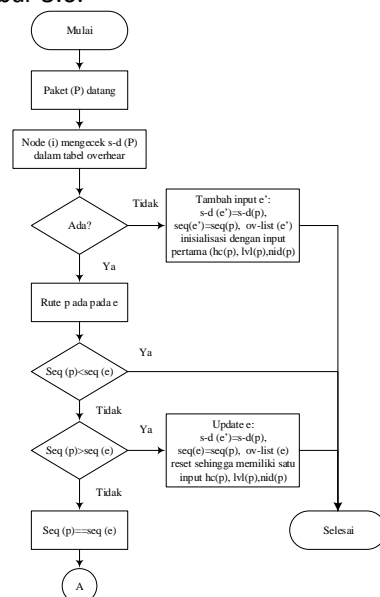
Ketika *node*  $i$  menerima paket informasi pesan SHORT  $Q$ , maka:

- Bandingkan bidang  $Q$  dengan *input*-an yang valid dalam *WaitingIndicator list*.
- Jika tidak ada yang cocok, maka abaikan paket  $Q$ ;

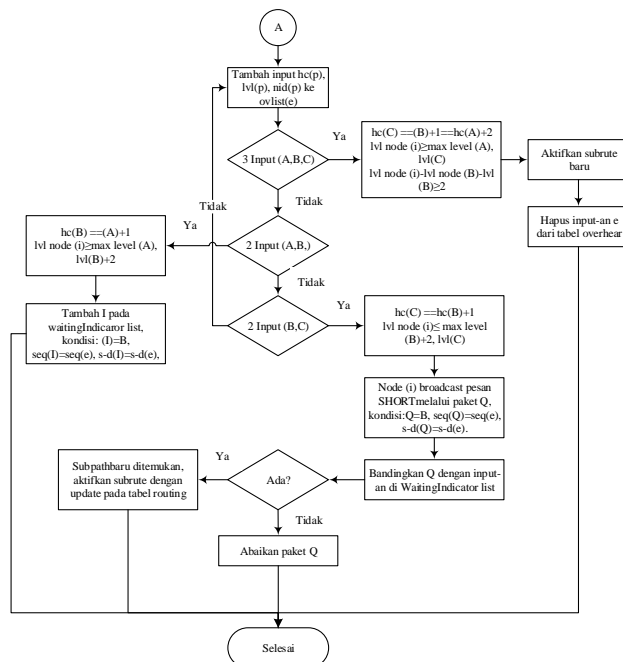
Jika kondisi tidak terpenuhi, maka subrute yang lebih baik ditemukan. Aktifkan subrute ini dengan memperbaharui *tabel routing*.

f. Proses selesai.

Untuk lebih jelasnya, pencarian rute menggunakan algoritma EA-SHORT ditunjukkan pada Gambar 3.5 dan Gambar 3.6.



Gambar 5. Flowchart algoritma EA-SHORT



Gambar 6. Flowchart algoritma EA-SHORT (Lanjutan Gambar 5)

### 3.4. Lingkungan Simulasi

Lingkungan simulasi pada penelitian ini akan diterapkan pada protokol *routing* AOMDV dan protokol *routing* AOMDV yang telah dimodifikasi dengan algoritma EA-SHORT. Pada penelitian ini simulasi dilakukan dengan menggunakan *Network Simulator 2* (NS-2). Pada NS-2 dibutuhkan pola *traffic*, skenario mobilitas *node* dan *script tcl*. Pola *traffic* mendefinisikan koneksi antara *source node* dengan *destination node* dan tipe koneksi yang digunakan (UDP). Skenario mobilitas *node* mendefinisikan posisi awal *node* dalam topologi jaringan, kecepatan mobilitas *node* dan tingkat mobilitas *node*. *Script tcl* merupakan kode program yang berfungsi untuk menciptakan objek simulator, mendefinisikan topologi jaringan, *load* pola *traffic* dan *load* skenario mobilitas *node*. Hasil simulasi dari performa protokol *routing* AOMDV dan protokol *routing* AOMDV yang telah dimodifikasi akan dibandingkan dengan menggunakan parameter pengujian yang telah ditentukan.

#### a. Parameter Pengujian

Adapun beberapa parameter uji yang digunakan untuk menganalisa hasil pengujian dari skenario simulasi yang dilakukan yaitu:

1. *Throughput* : Kecepatan (*rate*) jumlah paket data yang berhasil diterima di sisi penerima pada tiap detik[14]. Semakin tinggi nilai *throughput* maka semakin handal suatu jaringan, hal ini dikarenakan paket data yang

berhasil diterima semakin meningkat. *Throughput* dihitung menggunakan persamaan (1).

$$Throughput \text{ (Kbps)} = \frac{\text{ukuran data yang dikirim}}{\text{total waktu pengiriman data}} \quad (1)$$

2. *Average E2D* : rata-rata jeda yang terjadi pada saat pengiriman paket dari pengirim ke penerima paket. Semakin turun nilai *Average E2D* pada suatu jaringan maka semakin bagus jaringan, karena hal tersebut dapat menghemat penggunaan energi yang dipakai untuk proses *routing* selama waktu jeda. *Average E2D* dihitung menggunakan persamaan (2).

$$Average \text{ delay (s)} = \frac{\text{total delay}}{\text{total paket yang diterima}} \quad (2)$$

\*Di mana *delay* sama dengan waktu diterima dikurangi waktu dikirim.

3. *Packet Delivery Ratio* (PDR) : perbandingan paket yang dapat diterima. Semakin tinggi nilai PDR maka semakin baik, hal ini dikarenakan jumlah paket yang gagal diterima semakin berkurang. PDR dihitung menggunakan persamaan (3).

*Packet Delivery Ratio* (%)

$$= \frac{\text{paket data yang diterima}}{\text{paket data yang dikirim}} \times 100\% \quad (3)$$

#### b. Parameter skenario simulasi

Pada bagian ini merupakan skenario uji coba yang akan diimplementasikan menggunakan NS-2 versi 2.35. *Link layer* yang digunakan adalah berdasarkan pada standar IEEE 802.11. Fungsi koordinasi terdistribusi IEEE 802.11 untuk nirkabel LAN digunakan sebagai lapisan MAC. Jumlah *node* yang ada pada area sebanyak 50 dan 100 *node*[4]. Simulasi dilakukan pada area persegi (*network area*) dengan ukuran 500 × 500 m<sup>2</sup> dan 1000 × 1000 m<sup>2</sup>[4]. Tujuan simulasi dilakukan pada dua luas area yang berbeda adalah untuk mengetahui perbandingan kinerja kedua protokol dalam keadaan luas area yang kecil dan besar. Waktu simulasi yang dilakukan selama 600 *second*. Digunakan variasi kecepatan *node* untuk melihat perbandingan kinerja kedua protokol, serta menggunakan variasi *pause time* selama 45 s, 90 s, dan 180 s untuk mengetahui perbandingan kinerja kedua protokol. Jenis mobilitas yang digunakan pada simulasi adalah *Random Waypoint*. Setiap *node* secara independen memilih *destination* yang hendak dituju secara acak di dalam batasan jaringan. Parameter skenario simulasi yang lebih lengkap dapat dilihat pada TABEL II.

TABEL II PARAMETER SKENARIO

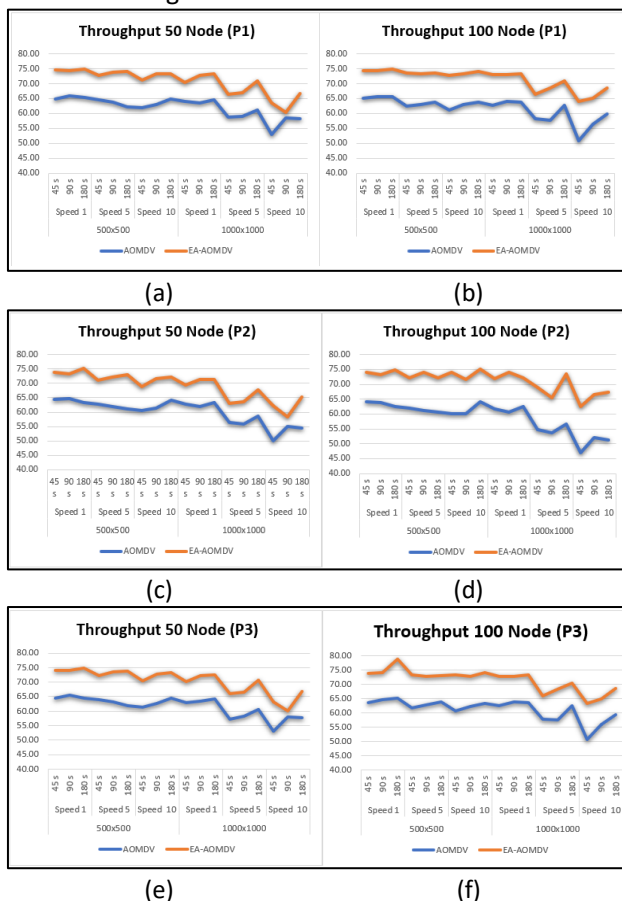
Parameter Skenario	
MAC layer	IEEE 802.11

Jumlah <i>node</i>	50 <i>node</i> , 100 <i>node</i>
Network area	500x500 m <sup>2</sup> , 1000x1000 m <sup>2</sup>
Waktu simulasi	600 s
Kecepatan <i>node</i>	1 m/s, 5 m/s, 10 m/s
Pause time	45 s, 90 s, 180 s
Pergerakan <i>node</i>	Random Waypoint
Propagation	TwoRayGround
Antena	Omni antenna
Pola traffic	CBR
Channel	Wireless

#### 4. HASIL DAN PEMBAHASAN

##### 4.1 Analisis *throughput*

Skenario uji coba pertama adalah mencari rata-rata jumlah paket data yang berhasil diterima di sisi penerima setiap detik (*throughput*). *Throughput* adalah rata-rata data yang dikirim dalam suatu jaringan. *Throughput* merujuk pada besar data yang dibawa oleh semua traffic jaringan dan selalu dikaitkan dengan *bandwidth*.



Gambar 7. Grafik perbandingan nilai *throughput* pada 3 percobaan. (a) Node 50 percobaan 1; (b) Node 100 percobaan 1; (c) Node 50 percobaan 2; (d) Node 100 percobaan 2; (e) Node 50 percobaan 3; (f) Node 100 percobaan 3

Berdasarkan ketiga percobaan yang dilakukan, rata-rata hasil uji coba *throughput* pada luas area 500x500 m<sup>2</sup> untuk 50 *node* pada EA-AOMDV adalah 73,04 Kbps dan 63,44 Kbps pada AOMDV. Sedangkan rata-rata hasil uji coba untuk 100 *node* pada EA-AOMDV adalah 73,76 Kbps dan 62,94 Kbps pada AOMDV. Rata-rata hasil uji coba pada luas area 1000x1000 m<sup>2</sup> untuk 50 *node* pada EA-AOMDV adalah 67,13 Kbps dan 59,11 Kbps pada AOMDV. Sedangkan rata-rata hasil uji coba untuk 100 *node* pada EA-AOMDV adalah 69,12 Kbps dan 58,15 Kbps pada AOMDV.

Pengiriman paket data oleh *throughput* dapat dipengaruhi oleh paket RREQ yang dikirimkan seperti jumlah *node* yang menentukan kepadatan lalu lintas untuk sampai ke *node* tujuan atau kecepatan transfernya. Pada grafik Gambar 4.18 nilai *throughput* dengan *pause time* 45s relatif selalu turun dan naik pada saat berada di kondisi *pause time* 180s, perubahan signifikan dapat dilihat di *network area* 100x1000 m<sup>2</sup>. Hal ini disebabkan oleh adanya kemungkinan jarak antara *node* sumber dengan *node* tujuan yang sangat jauh di luas area yang besar. Ketika *node* sumber dan *node* tujuan saling bergerak berjauhan maka akan mengakibatkan semakin banyak *route* yang terputus dan mengakibatkan proses pergantian *route* semakin sering dilakukan. Pada percobaan ini semakin tinggi *pause time*, maka semakin lama waktu yang dibutuhkan tiap *node* untuk tetap stabil sebelum pergerakan *node* berikutnya.

Selain itu pada grafik terlihat nilai *throughput* pada *node* 100 relatif lebih tinggi dibandingkan *node* 50 terutama nilai EA-AOMDV. Adanya penambahan jumlah *node* menyebabkan nilai *throughput* pada protokol AOMDV mengalami peningkatan, hal ini disebabkan oleh banyak *node* yang terdapat pada lingkungan simulasi akan menentukan kepadatan lalu lintas jaringan sehingga semakin kecil kemungkinan terjadinya *link* terputus pada jalur komunikasi dan menyebabkan *lifetime* jaringan bertahan lebih lama.

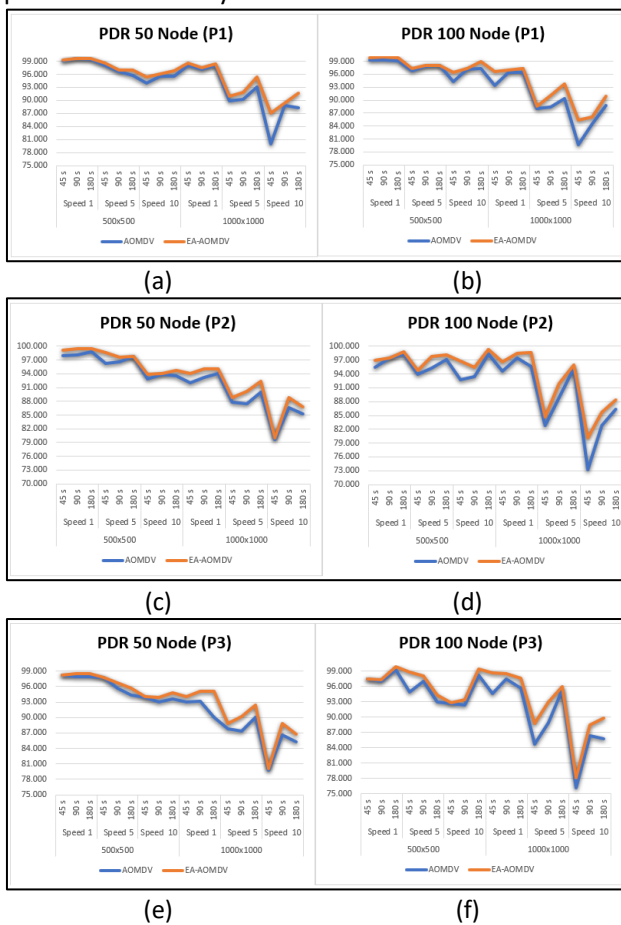
Berdasarkan grafik hasil uji coba yang telah dilakukan, pada ketiga percobaan dapat diketahui bahwa nilai *throughput* yang di peroleh pada EA-AOMDV selalu lebih tinggi dibandingkan AOMDV standar. Hal ini disebabkan oleh algoritma EA-SHORT mampu meminimalisir terjadinya *link failure* dengan membandingkan *node-node* tetangga dari suatu *node* asal dan memilih *node* dengan energi paling tinggi untuk menghindari *node* kehabisan energi di tengah sebelum sampai pada *node* tujuan. Jadi berdasarkan grafik pada Gambar 7 nilai *throughput* pada protokol



routing AOMDV dengan diterapkan algoritma EA-SHORT memberikan peningkatan nilai sebesar 10, 2245 Kbps, Ini membuktikan protokol routing EA-AOMDV lebih baik dari segi throughput jika dibandingkan dengan AOMDV.

**4.2 Analisis Packet Delivery Ratio (PDR)**

Skenario uji coba kedua adalah mencari nilai perbandingan antara paket data yang terkirim dengan jumlah paket data yang dikirimkan oleh node sumber. Packet Delivery Ratio (PDR) menunjukkan keberhasilan protokol dalam mengirim data, tinggi nilai PDR salah satunya disebabkan oleh berhasilnya sebuah protokol dalam melakukan pencarian dan pemeliharaan rutenya.



Gambar 8. Grafik perbandingan nilai packet delivery ratio pada 3 percobaan. (a) Node 50 percobaan 1; (b) Node 100 percobaan 1; (c) Node 50 percobaan 2; (d) Node 100 percobaan 2; (e) Node 50 percobaan 3; (f) Node 100 percobaan 3

Berdasarkan ketiga percobaan yang dilakukan, rata-rata hasil uji coba packet delivery ratio pada luas area 500x500 m<sup>2</sup> untuk 50 node pada EA-AOMDV

adalah 97,258 % dan 96,368 % pada AOMDV. Sedangkan rata-rata hasil uji coba untuk 100 node pada EA-AOMDV adalah 97,513% dan 96,631 % pada AOMDV. Rata-rata hasil uji coba pada luas area 1000x1000 m<sup>2</sup> untuk 50 node pada EA-AOMDV adalah 91,870 % dan 89,797 % pada AOMDV. Sedangkan rata-rata hasil uji coba untuk 100 node pada EA-AOMDV adalah 91,512 % dan 89,417 % pada AOMDV.

Pada protokol routing AOMDV dan EA-AOMDV pause time mempengaruhi pergerakan node, semakin sering node bergerak, maka zona yang terbentuk akan semakin sering berubah hal tersebut menyebabkan proses penemuan rute semakin sering dilakukan. Seperti terlihat pada grafik Gambar 8 nilai packet delivery ratio pada pause time 45s selalu mengalami penurunan. Selain itu semakin tinggi kecepatan node, maka packet delivery ratio yang dihasilkan semakin rendah. Dapat dilihat pada grafik saat kecepatan node menunjukkan angka tertinggi yaitu 10 m/s nilai packet delivery ratio semakin menurun terlebih pada pause time 45s. Hal ini terjadi karena kecepatan gerak node yang meningkat sehingga sering terjadi perubahan posisi node, yang menyebabkan rute yang akan dilalui untuk mengirimkan paket juga akan semakin cepat berubah karena rute mengalami putus link, sehingga terjadi hilangnya paket data yang lewat (packet drop). Perubahan signifikan dapat dilihat di network area 100x1000 m<sup>2</sup>. Hal ini disebabkan oleh adanya kemungkinan jarak antara node sumber dengan node tujuan sangat jauh di luas area yang besar, sehingga membutuhkan banyak node untuk mengirim paket data.

Berdasarkan grafik hasil uji coba yang telah dilakukan, pada ketiga percobaan dapat diketahui bahwa nilai packet delivery ratio yang di peroleh pada EA-AOMDV relative lebih tinggi dibandingkan AOMDV standar. Jadi berdasarkan grafik pada Gambar 8 nilai throughput pada protokol routing AOMDV dengan diterapkan algoritma EA-SHORT memberikan peningkatan nilai sebesar 1, 48464%. Ini membuktikan protokol routing EA-AOMDV lebih baik dari segi packet delivery ratio jika dibandingkan dengan AOMDV.

**4.3 Analisis average end-to-end delay**

Skenario uji coba ketiga adalah mencari nilai rata-rata selang waktu mulai dari paket dikirimkan oleh node sumber sampai paket data tersebut berhasil diterima oleh node tujuan.



Gambar 9. Grafik perbandingan nilai *end-to-end delay* pada 3 percobaan. (a) Node 50 percobaan 1; (b) Node 100 percobaan 1; (c) Node 50 percobaan 2; (d) Node 100 percobaan 2; (e) Node 50 percobaan 3; (f) Node 100 percobaan 3

Berdasarkan ketiga percobaan yang dilakukan, rata-rata hasil uji coba *average E2D* pada luas area 500 x 500 m<sup>2</sup> untuk 50 *node* pada EA- AOMDV adalah 0,01048 s dan 0,01291 s pada AOMDV. Sedangkan rata-rata hasil uji coba untuk 100 *node* pada EA-AOMDV adalah 0,01196 s dan 0,01536 s pada AOMDV. Rata-rata hasil uji coba pada luas area 1000x1000 m<sup>2</sup> untuk 50 *node* pada EA- AOMDV adalah 0,02117 s dan 0,02526 s pada AOMDV. Sedangkan rata-rata hasil uji coba untuk 100 *node* pada EA-AOMDV adalah 0,02418 s dan 0,03072 s pada AOMDV. Nilai *E2D* yang dihasilkan protokol *routing* EA-AOMDV lebih rendah dibandingkan dengan protokol *routing* AOMDV. Ini membuktikan protokol *routing* EA- AOMDV lebih baik dari segi *average E2D* jika dibandingkan dengan AOMDV.

Dalam proses pengiriman paket data, terdapat banyak faktor yang mengakibatkan *delay*. Salah

satunya terjadi ketika jumlah komunikasi bertambah akibat kepadatan *node*, kepadatan *node* dapat menyebabkan semakin banyak *node* perantara yang terpakai untuk mengirimkan layanan dari sumber ke tujuan sehingga waktu pengiriman semakin lama. Hal ini dapat dilihat dari nilai rata-rata keseluruhan *average E2D* pada jumlah 100 *node* selalu lebih tinggi dibanding nilai *average E2D* pada jumlah 50 *node*. Hasil nilai *average end-to-end delay* yang dihasilkan pada percobaan 1, percobaan 2, dan percobaan 3 terdapat beberapa perubahan yang signifikan pada luas area 1000x1000 m<sup>2</sup> untuk protokol *routing* AOMDV. Hal ini disebabkan oleh adanya kemungkinan jarak antara *node* sumber dengan *node* tujuan sangat jauh di luas area yang besar, sehingga membutuhkan banyak *node* untuk mengirim paket data.

Adanya penerapan algoritma EA-SHORT pada protokol *routing* AOMDV terbukti dapat menurunkan nilai *delay*. Hal ini dikarenakan algoritma EA-SHORT meningkatkan optimasi *routing* dengan melakukan perhitungan energi sehingga memperkecil kemungkinan rute terputus sebelum *node* sumber sampai pada *node* tujuan. Pada Gambar 9 hasil menunjukkan nilai *E2D* pada protokol *routing* AOMDV tanpa menggunakan algoritma dibandingkan dengan nilai *E2D* pada protokol *routing* dengan menggunakan algoritma EA-SHORT terbukti dapat menurunkan hasil *delay* sebesar 0.04115 s, sehingga dapat dikatakan uji coba pada penelitian ini berhasil.

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

- Berdasarkan hasil uji coba penelitian, penerapan algoritma EA-SHORT mampu meningkatkan performa protokol AOMDV. Di mana setelah penerapan algoritma EA-SHORT, rata-rata *throughput* mengalami peningkatan sebesar 13,92 %, PDR sebesar 1,57 % dan *average E2D* menurun sebesar 19,53 %.
- Pada protokol *routing* AOMDV dan EA-AOMDV *pause time* dan kecepatan *node* mempengaruhi pergerakan *node*. Semakin tinggi *pause time* dan rendah kecepatan *node*, maka semakin lama waktu yang dibutuhkan tiap *node* untuk tetap stabil sebelum pergerakan *node* berikutnya. Semakin rendah *pause time* dan tinggi kecepatan *node*, semakin sering *node* bergerak, maka rute yang terbentuk akan semakin sering berubah hal tersebut menyebabkan proses penemuan *route* semakin sering dilakukan sehingga menyebabkan

penurunan pada nilai *throughput* dan *packet delivery ratio* dan kenaikan pada nilai *end-to-end delay*.

- c. Perubahan signifikan nilai parameter uji selalu terjadi pada *network area* 1000x1000 m<sup>2</sup>. Hal ini disebabkan oleh adanya kemungkinan jarak antara *node* sumber dengan *node* tujuan yang sangat jauh di luas area yang besar. Ketika *node* sumber dan *node* tujuan saling bergerak berjauhan akan mengakibatkan semakin banyak *route* yang terputus dan mengakibatkan proses pergantian *route* semakin sering dilakukan.

### 5.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka peneliti dapat memberikan saran-saran sebagai berikut :

- Melakukan penelitian selanjutnya dengan menggunakan protokol *routing* dan algoritma *energy aware* yang berbeda pada jaringan MANET.
- Melakukan penelitian selanjutnya dengan menggabungkan algoritma EA-SHORT dengan algoritma lainnya untuk mencari hasil yang lebih optimal.
- Melakukan penelitian dengan menggunakan luas area simulasi yang bervariasi dan jumlah kepadatan *node* dan parameter uji lainnya yang bervariasi pula, untuk mendapatkan hasil yang valid terkait dengan pengaruh luas area, pause time, kecepatan *node* dan jumlah koneksi terhadap simulasi penelitian di jaringan MANET.

### DAFTAR PUSTAKA

- [1] C. Gui and P. Mohapatra, "A framework for self-healing and optimizing routing techniques for mobile ad hoc networks," *Wirel. Networks*, vol. 14, no. 1, pp. 29–46, 2008.
- [2] A. Z. I Made Windra Yudistiana, Andy Hidayat Jatmika, "Analisis Optimasi Kinerja Protokol Routing AODV Dan AOMDV Dengan Menggunakan Metode RFAP Untuk Mencegah RREQ Flooding Attacks Pada Jaringan MANET," *Jtika*, vol. 1, no. 1, pp. 9–18, 2019.
- [3] I. Aulia, F. A. Yulianto, and A. Herutomo, "Analisis Performansi Routing Protocol AOMDV dan DYMO pada Vehicular Ad Hoc Network," pp. 1–6, 2013.
- [4] Y. P. Wulandari, A. H. Jatmika, and F. Bimantoro, "Meningkatkan Efisiensi Rute Pada Protokol Routing AOMDV Menggunakan Metode PA-SHORT," *Jtika*, vol. 1, no. 1, pp. 77–85, 2019.
- [5] A. H. Jatmika, S. Djanali, and M. Husni, "Optimasi Routing pada Jaringan MANET," pp. 1–7, 2011.
- [6] W. A. Rachman, P. H. Trisnawan, and M. A. Fauzi, "Analisis Konsumsi Energi Protokol Routing Fisheye State Routing ( FSR ) Pada Mobile Ad Hoc Network ( MANET )," vol. 3, no. 5, pp. 4455–4466, 2019.
- [7] A. Fauzio, N. M. Adriansyah, and L. V. Yovita, "Analisis Performansi Ad-Hoc On-Demand Multipath Distance Vector Routing (AOMDV) dan Ad-Hoc On-Demand Distance Vector Routing (AODV) Pada MANET," pp. 1–8, 2007.
- [8] L. F. Z. Fauzi, "Analisis Performansi Multipath Routing Pada Jaringan Mobile Adhoc," Telkom University, 2016.
- [9] R. Asokan, A. M. Natarajan, and C. Venkatesh, "Quality-of-Service Routing Using Path and Power Aware Techniques in Mobile Ad Hoc Networks," *J. Comput. Syst. Networks, Commun.*, vol. 2008, no. 160574, pp. 1–7, 2008.
- [10] A. B. Prasetyo, "Analisis Performansi Multipath Routing Pada Jaringan Mobile Adhoc," Universitas Telkom, 2006.
- [11] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wirel. Commun. Mob. Comput.*, vol. 6, no. 7, pp. 969–988, 2006.