

# RANCANGAN MODIFIKASI ALGORITMA AODV PADA MANET UNTUK MENINGKATKAN PERFORMANSI SKEMA *ROUTING* DAN DETEKSI KONGESTI (STUDI LITERATUR)

Arif Fadllullah

Jurusan Teknik Elektro Universitas Borneo Tarakan, Tarakan, Kaltara, Indonesia

e-mail: arif.fadl@gmail.com

**Abstract**—AODV is one of the well-known reactive routing protocols on MANET that is built on routing requests. This protocol is widely used because of low routing overhead and high performance. However, sending data packets using conventional AODV often delay, when the routing path is interrupted as a result of the movement of nodes that change in the path. Therefore, this study proposes a modified AODV design that not only has the routing scheme feature of the main path chosen based on link throughput and minimum hop count, but also considers providing alternative paths when the link node is damaged or when the data packet queue exceeds the maximum capacity of a node until congestion status occurs at the node, especially to reduce packet delay delays that are prone to occur in conventional AODV.

**Keywords**—AODV, Modification AODV, Link Throughput, Congestion Detection

**Intisari**—AODV adalah salah satu protokol reaktif routing yang terkenal pada MANET yang dibangun berdasarkan permintaan routing. Protokol ini banyak dipakai karena overhead routing yang rendah dan performa tinggi. Hanya saja pengiriman paket data menggunakan AODV konvensional seringkali terjadi delay, manakala jalur routing terputus sebagai akibat adanya pergerakan node yang berubah-ubah dalam jalur. Oleh karena itu, penelitian ini mengusulkan rancangan AODV modifikasi yang tidak hanya memiliki fitur skema routing jalur utama yang dipilih berdasarkan link throughput dan minimum hop count, tetapi juga mempertimbangkan penyediaan jalur alternatif saat link node mengalami kerusakan maupun saat antrian paket data melebihi kapasitas maksimal suatu node hingga status kongesti terjadi pada node, khususnya untuk mengurangi delay pengiriman paket yang rentan terjadi pada AODV konvensional.

**Kata Kunci**—AODV, AODV Modifikasi, Link Throughput, Deteksi Kongesti

## I. PENDAHULUAN

Saat ini, internet telah menjadi salah satu bagian yang tidak terpisahkan untuk menunjang kebutuhan manusia dalam hal pengiriman data dan komunikasi jarak jauh. Agar dapat terhubung dengan jaringan internet, tidak hanya diperlukan perangkat yang memiliki *network interface connection* ataupun perangkat *wireless mobile*, melainkan juga harus memiliki perangkat penunjang, seperti *access point* dan *router* yang tersambung ke BTS (*base station*) infrastruktur internet melalui ISP (*internet service provider*) yang terdaftar. Tentu hal ini akan memakan biaya apalagi jika membutuhkan *bandwidth*

yang besar untuk mengakses informasi dalam internet baik berdasarkan *volume based* maupun *time based*.

Kondisi tersebut memunculkan ide bagaimana manusia bertukar data atau informasi tanpa tersambung ke jaringan internet. Salah satu alternatif adalah memanfaatkan layanan *wireless* yang terpasang pada perangkat tertentu atau di kenal dengan istilah MANET (*Mobile Ad Hoc Network*). Metode ini berdasarkan fakta bahwa sekarang ini mudah sekali ditemukan perangkat berbasis *mobile wireless*, seperti laptop, tablet, note, hp, dan lain sebagainya dalam jangkauan yang saling berdekatan. Dengan MANET, perangkat tersebut dapat berperan sebagai *host*, *router*, dan *forwarding packets* sehingga dapat melakukan pertukaran data tanpa memerlukan *base station* yang tetap sebagai *relay* transmisinya. Syaratnya, partisipan dalam jaringan ini harus mau menjadikan perangkatnya sebagai *router* untuk meneruskan paket data dan menjamin bahwa paket berhasil dikirimkan dari sumber ke tujuan. MANET ini sangat cocok untuk diterapkan pada daerah yang belum memiliki infrastruktur komunikasi internet atau sebagai alat komunikasi darurat jika infrastruktur rusak.

Ada beberapa hal yang perlu dipertimbangkan dalam mengimplementasikan MANET, termasuk bagaimana cara menghubungkan satu *mobile device* (selanjutnya disebut *node*) dengan node lainnya diluar jangkauan *wireless* sehingga seringkali harus melewati beberapa *node* penghubung atau *network multihop*. Belum lagi, node pada MANET yang sifatnya dinamik turut mempengaruhi perubahan secara periodik topologi jaringan dan jalur transmisi, termasuk jika jarak antara node-node menjadi sangat besar, maka power pengiriman lemah, transmisi data menjadi rendah, *delay* dan kegagalan membaca jalur pengiriman tinggi, sehingga *packet loss* sering terjadi.

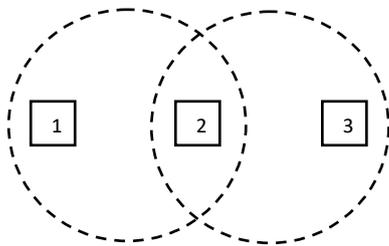
Perlu adanya standar *routing* protokol agar pengiriman data bisa dilakukan secara optimal. Salah satu alternatif protokol *routing* MANET yang banyak digunakan adalah AODV (*Ad Hoc Demand Distance Vector*). Berdasarkan beberapa penelitian, AODV unggul karena *overhead routing* yang rendah dan performa tinggi [1] [2] [3]. Akan tetapi, penentuan skema jalur utama hanya dengan menghitung minimal *hop count*, tanpa mempertimbangkan aspek lain seperti *link throughput*. Termasuk menyediakan jalur alternatif, ketika terdapat beberapa node penghubung yang mengalami kongesti atau kemacetan. Belum lagi pengiriman paket data dilakukan setelah pencarian jalur menjadi tidak efektif

manakala jumlah node yang terhubung dalam jaringan *ad hoc* berskala besar. Hal ini tidak hanya berimplikasi pada *delay* pengiriman yang meningkat, tetapi juga menghasilkan *routing* yang tidak valid, terlebih karena ada perubahan topologi saat pengiriman paket data dilakukan.

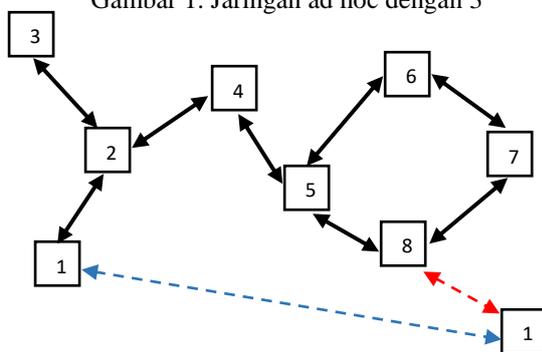
Oleh karena itu, dalam artikel ini akan diusulkan rancangan AODV dengan beberapa modifikasi dalam alur algoritmanya agar yang tidak hanya memiliki fitur skema *routing* jalur utama yang dipilih berdasarkan *link throughput* dan *minimum hop count*, tetapi juga mempertimbangkan penyediaan jalur alternatif saat link node mengalami kerusakan maupun saat antrian paket data melebihi kapasitas maksimal suatu node hingga status kongesti terjadi pada node, khususnya untuk mengurangi *delay* pengiriman paket yang rentan terjadi pada AODV konvensional.

## II. LANDASAN TEORI

### A. Protokol Routing



Gambar 1. Jaringan ad hoc dengan 3



Gambar 2. Jaringan ad hoc dengan 8 node

Pada Gambar 1, terdapat tiga buah node, yang mana node-1 berada di jangkauan transmisi node-2 tetapi di luar jangkauan node-3. Node-2 dapat menjangkau node-1 dan node-3. Serta, node-3 berada di jangkauan transmisi node-2 tetapi di luar jangkauan node-1. Sekarang jika node-1 dan node-3 ingin berkomunikasi satu sama lain, node tersebut harus memanfaatkan layanan node-2 untuk meneruskan komunikasinya.

Pada kenyataannya *ad hoc multinode* lebih rumit daripada contoh pada Gambar 1, karena nodenya tidak hanya 3, melainkan lebih seperti pada Gambar 2 yang terdapat 8 buah node. Awalnya jika node-1 ingin mengirimkan paket data ke node-7, maka jalur transmisinya sebagai berikut:  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ . Apabila di tengah-tengah proses pengiriman sedang berlangsung, tiba-tiba node-1 bergerak mendekati dan berada dalam jangkauan yang sama dengan node-8, maka jalur transmisi berubah menjadi:  $1 \rightarrow 8 \rightarrow 7$  dengan tetap menjaga agar tidak terjadi *packet loss* selama proses

*routing updating*. Agar proses *updating* tersebut berjalan dengan baik, perlu adanya protokol *routing*. Protokol *routing* dalam MANET bukan hal yang sederhana seperti *fix network*, karena suatu protokol harus mampu mengatasi masalah *multinode* yang bisa saja bergerak setiap saat, sehingga node yang sudah terdeteksi sebelumnya tiba-tiba menghilang atau sebaliknya ada node-node baru bermunculan sehingga pemetaan jalur harus bisa diupdate secara periodik dalam protokol *routing* MANET. Protokol *routing* pada MANET dapat dibedakan menjadi dua yaitu protokol *routing* reaktif dan *routing* proaktif. Salah satu alternatif protokol *routing* yang sering digunakan pada MANET adalah AODV.

### B. AODV

Protokol AODV (*Ad Hoc on Demand Distance Vector*) merupakan protokol reaktif yang dibangun berdasarkan permintaan *routing* dalam menentukan jalur hanya ketika node sumber ingin mengirimkan paket data ke node tujuan, dimana didalamnya terdapat penggabungan antara fitur *routing discovery* dan *routing maintenance* yang diambil dari DSR protokol, dan fitur *hop by hop routing* dan *destination sequence number* yang diambil dari DSDV protokol. AODV menggunakan RREQ (*route request*) dan RERR (*route error*) sebagai sinyal kontrol [4].

Proses *routing* pada AODV dibagi kedalam dua mekanisme, yakni *discovery route* dan *maintenance route* [5] [6].

#### • Discovery Route

Agar sebuah node sumber dapat mengirimkan data maupun melakukan komunikasi ke beberapa node tujuan, maka tidak hanya sekedar node sumber mengetahui IP (*internet protocol*) node tujuan, tetapi juga harus menginisialisasi jalur yang akan dilewati oleh paket data. Caranya melalui proses pencarian jalur, dimana node sumber melakukan *broadcast RREQ (route request)* ke semua node tetangga. Node yang mengetahui jalur ke penerima bisa dikatakan sebagai node penghubung atau *router*. Baik node penghubung maupun node tujuan akan menyimpan tabel *routing* baru yang dibawa oleh RREQ. Tabel *routing* baru ini berisi alamat node penghubung sebagai jalur menuju ke node penghubung lainnya atau node tujuan. Jika ada node yang mendapatkan pesan RREQ yang sama, maka pesan RREQ yang diterima terakhir itu akan dibuang. Selanjutnya node akan merespon pesan RREQ dengan meneruskan kembali ke node tetangga lainnya jika node tersebut bukan node tujuan. Akan tetapi jika node tujuan yang menerima pesan RREQ, maka selanjutnya node tersebut akan mengirimkan pesan balasan RREP (*route reply*) secara *unicasting*. Pesan RREP disalurkan sepanjang jalur yang dibentuk RREQ sebelumnya, melewati node-node yang memiliki jalur ke penerima. Hingga pesan RREP sampai ke node sumber, pesan ini berisi tentang IP pengirim, IP penerima, *hop count* (lompatan node yang dilewati), dan *time to live* (waktu pesan). Dengan RREP ini, secara sekuensial jalur *end to end* akan terbentuk sendiri antara node tujuan menuju node sumber. Jika jalur kadaluarsa atau tidak pernah digunakan, maka entri *routing* yang mengarahkan ke jalur tersebut otomatis akan terhapus dalam rentang waktu tertentu.

- *Maintenance Route*

Saat jalur telah terbentuk, mekanisme kerja *routing* AODV bergantung pada masing-masing entri *routing* tabel node itu sendiri, sehingga tidak diperlukan informasi *routing* dari node penghubung. Melalui pesan HELLO *broadcast* yang dikirim secara periodik antar node bertugas untuk menginformasikan dan memastikan bahwa topologi *link* node di depannya tidak mengalami perubahan ataupun kerusakan. Apabila itu terjadi, maka node yang terhubung pada *link* tersebut akan mengirimkan pesan RERR (*route error*) ke node penghubung yang aktif sebagai pesan untuk menjelaskan bahwa *link* telah rusak. Proses tersebut secara relay akan terus dikirimkan sampai pesan *error* diterima oleh semua node aktif, termasuk sampai ke node sumber. Setelah menerima pesan ini, node akan menghapus semua jalur yang berisi node gagal. Selanjutnya, *discover route* akan diaktifkan kembali dan node sumber harus mengirimkan ulang RREQ untuk menemukan jalur alternatif yang baru.

### III. METODE PENELITIAN

Metode penelitian ini didasarkan pada penerapan metode SLR sebagai suatu cara untuk melakukan ulasan penelitian dengan melakukan klasifikasi dan perbandingan terhadap beberapa penelitian yang telah ada baik berupa artikel, jurnal, atau *conference*. SLR menjelaskan proses pencarian dokumen secara relevan dari pertanyaan penelitian yang diajukan. Penelitian ini memiliki tiga tahapan: (i) Rencana *review* yang mendefinisikan protokol *review* untuk menentukan pertanyaan penelitian. (ii) Implementasi dibutuhkan kriteria inklusi dan eksklusi untuk mendapatkan kriteria kata kunci utama yang berhubungan dengan pertanyaan penelitian. (iii) Dokumentasi yang bertujuan untuk memberikan informasi yang diperoleh dari setiap ulasan berupa ringkasan.

#### A. *Research Question (RQ)*

*Research question* digunakan untuk mendefinisikan kriteria atau atribut pertanyaan yang menjadi acuan kerangka pertanyaan. Pertanyaan utama pada studi literatur penelitian ini adalah sebagai berikut:

- RQ1: Bagaimana perkembangan dan prinsip kerja algoritma AODV yang sudah ada?**  
**RQ2: Bagaimana skema pencarian jalur pada desain Modifikasi AODV usulan?**  
**RQ3: Bagaimana skema pemeliharaan jalur pada desain Modifikasi AODV usulan?**

#### B. *Proses Pencarian Pertanyaan*

Dari RQ diatas kemudian dicari kata kunci pertanyaan yang sesuai untuk studi literatur pada penelitian ini. Kata kunci yang dapat digunakan adalah **MANET, AODV, dan Modifikasi AODV**. Setelah mendefinisikan atribut/kata kunci untuk dijadikan kerangka acuan dalam menjawab RQ, langkah selanjutnya yaitu melakukan proses pencarian studi pustaka dengan menggunakan ekspresi operator logika AND, OR pada basis data jurnal seperti *ScienceDirect, IEEE, Google Scholar*.

#### C. *Kriteria Inklusi dan Eksklusi*

Untuk membatasi ruang lingkup pencarian maka dibutuhkan penyaringan kata kunci dengan menggunakan penyaringan berdasarkan kriteria inklusi sebagai berikut:

- 1) Fokus pada analisis modifikasi AODV.
- 2) Sumber dalam bahasa Inggris dan memberikan jawaban atas pertanyaan penelitian dan kata kunci penelitian.
- 3) Publikasi literatur dalam bentuk jurnal atau *conference* yang diseleksi oleh penulis.

Untuk mempercepat pencarian, maka diperlukan suatu cara agar dapat mempersempit ruang lingkup pencarian yang mengabaikan literatur hasil penyaringan eksklusi dengan kondisi sebagai berikut:

- 1) Literatur yang menggunakan bahasa selain bahasa Inggris.
- 2) Literatur yang tidak terkait dengan pertanyaan dan kata kunci penelitian.
- 3) Sumber yang hanya memberikan informasi dalam bentuk abstrak.
- 4) Literatur ganda dari sumber pencarian.

#### D. *Ekstraksi Data dan Sintesis*

Selanjutnya adalah ekstraksi jurnal, *conference* secara manual yang ada kaitannya dengan judul, abstrak, kata kunci dan referensi yang sesuai dengan pertanyaan penelitian dan kata kunci penelitian. Penelitian yang dilakukan ini akan memberikan desain usulan modifikasi AODV berdasarkan *review* studi pustaka yang diperoleh dari sumber literatur yang sudah ada.

#### E. *Dokumentasi dan Validasi Hasil*

Kumpulan jurnal dan *conference* hasil ekstraksi kemudian divalidasi dengan tujuan untuk mengetahui gambaran perkembangan algoritma modifikasi AODV yang kemudian dituangkan ke dalam desain usulan modifikasi AODV pada penelitian ini.

## IV. HASIL DAN PEMBAHASAN

Setelah ditentukan kriteria inklusi dan eksklusi dan pemilihan jurnal berdasarkan abstrak yang dibaca maka diperoleh beberapa jurnal atau *conference* yang relevan untuk selanjutnya dikaji lebih lanjut dan dijadikan pembahasan penelitian. Adapun pemaparan hasil *review* yang menjawab RQ studi literatur penelitian ini dijabarkan sebagai berikut:

#### **RQ1: Bagaimana perkembangan dan prinsip kerja algoritma AODV yang sudah ada?**

##### 1. Ide Awal Algoritma

Pada AODV konvensional, paket data dikirimkan setelah pencarian jalur utama selesai. Hal tersebut menjadi masalah ketika node yang saling terhubung berskala besar, dimana paket data diminta menunggu dalam waktu yang lama hingga jalur utama selesai terbentuk. Hal ini tidak hanya berimplikasi pada *delay* pengiriman yang tinggi, tetapi juga menghasilkan jalur yang tidak valid, terlebih karena ada perubahan topologi saat pengiriman paket data dilakukan. AODV konvensional dalam memilih jalur utama *routing* ke node tujuan hanya berdasarkan minimum *hop count*, sehingga

jalur alternatif lain yang *hop count*-nya lebih besar akan diabaikan. Padahal bisa jadi jalur yang *hop count*-nya besar, memiliki *link throughput* lebih baik bila dibandingkan dengan minimum *hop count*. Selain itu, jika jalur utama rusak maka jalur alternatif tersebut bisa digunakan, itupun harus melalui proses pencarian jalur kembali yang tentu akan meningkatkan *delay* pengiriman. Jalur sebuah node dikatakan terputus (*host unreachable*) jika daya node *down*, node rusak dan secara *mobile* node bergerak menjauhi jalur utama. Padahal gangguan jalur pada node juga bisa dikarenakan alur *density* antrian paket melebihi batas maksimal *buffer size* node yang berakibat pada kongesti, sehingga pengiriman mengalami RTO (*request time out*). Hal ini tidak menjadi pertimbangan dalam AODV konvensional, sehingga berakibat masih tingginya *loss rate* (kehilangan) paket data saat pengiriman berlangsung.

Oleh karena itu, algoritma AODV pendekatan baru yang diusulkan akan mengkombinasikan beberapa algoritma, seperti AODV\_ODB, AODV\_M, AODV\_BRL, dan OLSR\_RED sehingga diharapkan mampu mengurangi *delay* yang terjadi saat paket data harus menunggu jalur selesai terbentuk melalui penyisipan paket data pada pesan RREQ. Termasuk mempertimbangkan pencarian jalur terpendek dengan tidak mengabaikan beberapa jalur alternatif melalui pemilihan jalur berdasarkan perhitungan nilai *throughput*, dan pengalihan jalur jika terdapat status kongesti pada node penghubung.

## 2. Pesan RREQ disisipkan Paket Data

Salah satu kondisi *delay* yang terjadi pada AODV konvensional adalah saat paket data menunggu dalam jangka waktu tertentu untuk dikirim selama proses pencarian jalur utama node sumber ke node tujuan. Berdasarkan penelitian yang telah dilakukan sebelumnya, *delay* tersebut dapat dikurangi melalui modifikasi pesan RREQ dengan pendekatan AODV\_ODB. Konsep AODV\_ODB [7] adalah saat menemukan *link error*, node akan melakukan *broadcast* sebuah paket *header*, yang tidak hanya berisi pesan RREQ, tetapi juga paket data. Artinya, ketika pesan RREQ menjangkau node tetangga bukan node tujuan, jalur baru terbentuk dan paket data juga diterima node tetangga, hingga paket di *broadcast* terus sampai ke node tujuan. Ketika paket *header* mencapai node tujuan, node tujuan akan mengirim RREP dan pada saat yang sama menerima paket data. Hal ini tidak hanya akan mempermudah pengaturan *routing* tetapi juga mengurangi *delay* pengiriman, karena paket data langsung dikirimkan saat pencarian jalur bersamaan dengan pengiriman pesan RREQ.

## 3. Inisialisasi Jalur berdasarkan *Link Throughput*

Spesifikasi *bandwidth*, daya, dan stabilitas sinyal yang berbeda pada masing-masing node atau peralatan *wireless mobile* erat kaitannya dengan *link throughput* yang bisa dilewati untuk pengiriman paket, sehingga nilai *throughput* dari *link bottleneck* (*link* dengan *throughput* terendah di antara semua *link* dalam rute tertentu) dapat dianggap sebagai nilai *throughput* untuk jalur tertentu yang menjadi acuan utama dalam menentukan skema *routing* AODV, selain dengan minimum *hop count*. Nilai

*Throughput* adalah *bandwidth* aktual jalur pengiriman paket dengan rumus dalam persamaan 1 berikut ini.

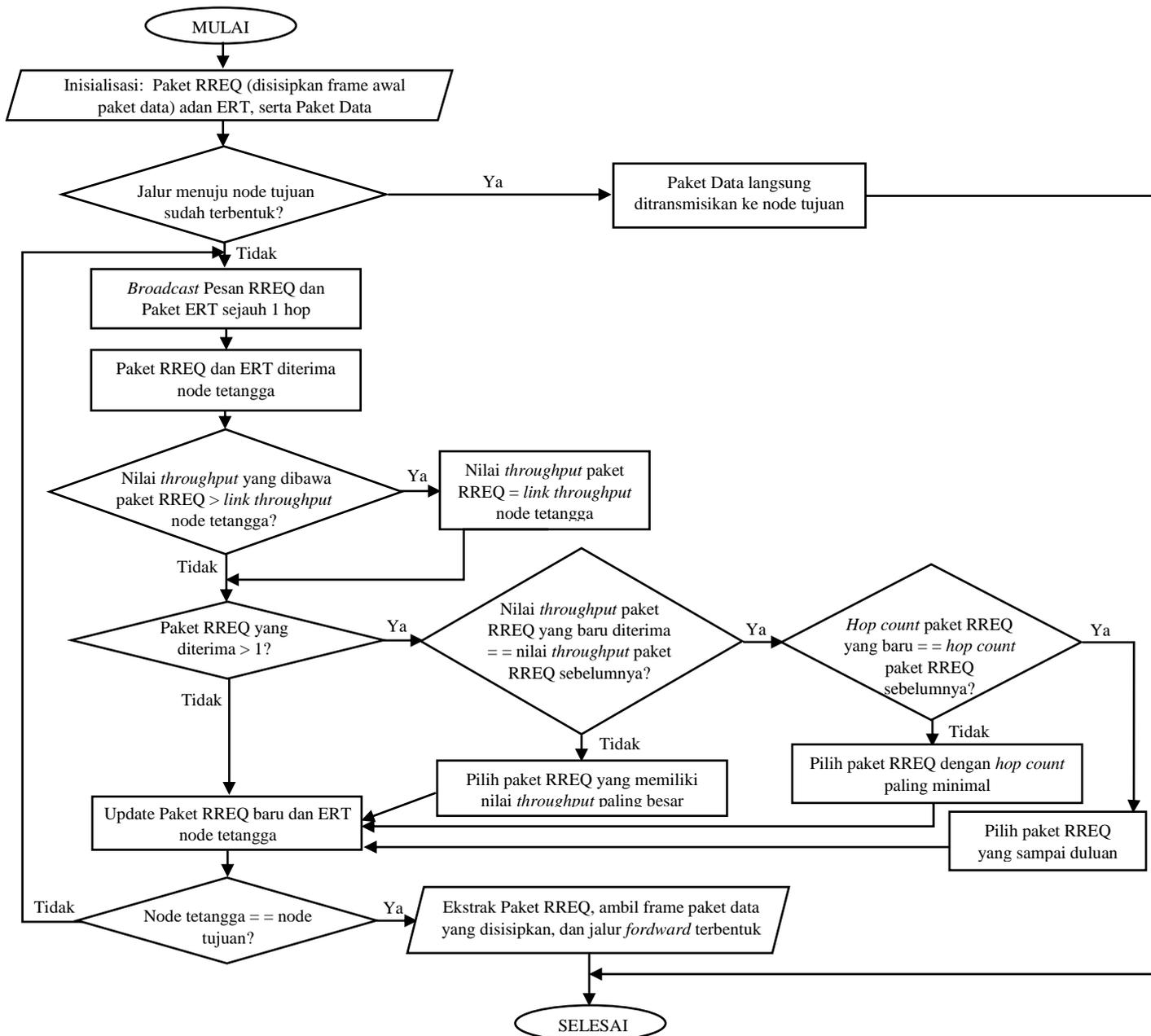
$$\text{Throughput} = \frac{\text{ukuran\_paket(bit)}}{\text{waktu\_paket\_diterima(detik)}} \quad [8] \quad (1)$$

Penelitian yang pernah dilakukan berkaitan penentuan jalur berdasarkan *throughput* menggunakan algoritma AODV\_M. Konsep AODV\_M [9] menyisipkan nilai *throughput* node sumber pada paket RREQ yang diteruskan ke node tetangga yang kemudian dibandingkan dengan *link throughput* node tetangga yang ada. Jika *link throughput* node tetangga lebih kecil dari nilai *throughput* yang dibawa paket RREQ, maka nilai *throughput* paket RREQ diubah dan digantikan dengan nilai *link throughput* node tetangga. Selanjutnya paket RREQ membawa nilai *throughput* yang baru diteruskan ke node tetangga lainnya dan jumlah *hop count* bertambah satu.

Jika sebuah paket RREQ dengan *destination sequence number* yang sama, tapi berasal dari jalur yang berbeda berhasil diterima oleh node tujuan. Maka untuk algoritma AODV konvensional, node tersebut hanya akan memilih paket dari jalur dengan *hop count* paling minimal atau paket yang terlebih dahulu diterima dan melakukan *drop* terhadap paket yang baru datang. Sedangkan dengan algoritma AODV\_M, nilai *throughput* paket RREQ yang baru datang dibandingkan dengan nilai *throughput* paket RREQ sebelumnya. Setelah dibandingkan, nilai *throughput* paket RREQ yang paling besar itulah yang kemudian akan diteruskan ke node selanjutnya. Akan tetapi, jika ternyata perbandingan nilai *throughput* menghasilkan nilai yang sama, maka baru jalur dengan minimum *hop count* yang dipilih. Begitu seterusnya, hingga paket RREQ sampai ke node tujuan. Kemudian node tujuan mengirimkan RREP dengan cara yang sama seperti pengiriman paket RREQ, dengan membandingkan nilai *throughput* RREP paket antar node yang telah dilewati paket RREQ sebelumnya, sehingga terbentuk jalur *end-to-end* antara node sumber ke node tujuan dengan memilih node penghubung sebagai jalur yang memiliki *throughput* paling besar.

## 4. Backup Jalur dengan LHF

Dalam AODV konvensional, node sumber mengirimkan data dan melakukan komunikasi ke node tujuan dengan memanfaatkan sebuah jalur tunggal yang terbentuk secara *end-to-end* melalui node-node penghubung berdasarkan minimum *hop count*. Jalur tunggal yang dibangun akan memiliki kendala ketika node penghubung rusak, sehingga inisialisasi jalur harus diulang lagi, yang pencariannya dimulai dari node sumber ke node tujuan. AODV\_BRL (*Backup Route with LHF*) merupakan satu dari sekian banyaknya algoritma *routing* guna mencari jalur alternatif pengiriman paket data. AODV\_BRL [10] mengkombinasikan teknik *backup* jalur dengan penambahan sebuah ERT (*Extended Routing Table*) yang di *broadcast* bersamaan dengan pesan RREQ. Ketika node tujuan mendapatkan pesan RREQ dari beberapa jalur, maka node tujuan mengirimkan paket RREP dan ERT secara *multicast* ke node penghubung hingga sampai ke node sumber, sehingga ada beberapa jalur alternatif yang dapat terbentuk dari node sumber ke



Gambar 3. Discovery Forwarding Routes

node tujuan, tidak hanya jalur utama yang hanya berdasarkan minimum *hop count*. Selain jalur alternatif yang dibangun oleh paket RREP, struktur *mesh* akan dibuat dengan mentransmisikan secara *broadcast Extended Hello Message* dari sebuah node tetangga ke node tetangga lainnya yang masih dalam jangkauan satu *hop*. *Hello message* akan dikirimkan setiap interval waktu 1,000 ms. Proses ini untuk mengetahui mana saja node tetangga yang masih aktif dan mengupdate ERT yang berasal dari node tetangga.

Jika dalam jangkauan satu *hop*, *hello message* mendeteksi adanya *link* yang rusak, maka proses pencarian jalur tidak diinisialisasi ulang dari node sumber, seperti pada AODV konvensional. Melainkan perbaikan *link* berdasarkan metode LHF yakni mensubstitusi alamat *routing* node yang rusak dengan alamat *routing* node penghubung baru yang lain yang punya jalur ke node tujuan pada ERT sebuah *downstream* node. Jika ternyata dari ERT *downstream* node tidak ditemukan jalur

alternatif, maka pengecekan dilakukan kembali dengan mengirimkan BRRQ (*backup route request*) dari *downstream* node ke node tetangganya. Jika terdapat node tetangga yang punya jalur ke node tujuan, maka node tersebut mengirimkan pesan balasan BRRP (*backup route reply*) dan node tersebut menjadi jalur alternatif yang baru. Hal ini tentu akan mengurangi *overhead* dan *delay* saat perbaikan *link*, karena pencarian jalur tidak diinisialisasi ulang dari node sumber.

#### 5. Status Kongesti sebagai Pertimbangan Link Error

*Link* sebuah node dalam MANET dapat dikatakan terputus, jika node rusak, mati, dan diluar jangkauan. Padahal penyebab terputusnya *link*, juga bisa karena meningkatnya jumlah *queue* (antrian) melebihi *buffer* node maksimal yang bisa mengakibatkan kongesti (kemacetan). Penelitian yang pernah dilakukan untuk mendeteksi adanya kongesti adalah penerapan algoritma RED (*Random Early Detection*) pada protokol AODV

[11] dan OLSR\_RED [12]. RED merupakan algoritma *active queue management* (AQM). RED menghitung rata-rata ukuran *queue* yang masuk ke node dengan memberikan nilai *Threshold* (ambang bata). Rata-rata ukuran *queue* dibandingkan untuk dua nilai *Threshold*, minimum *Threshold* dan maksimum *Threshold* [13]. Rumus nilai *Threshold* yang digunakan dapat dilihat pada persamaan 2 dan 3.

$$Min_{Thr} = 0,25 \times Buffer_{size} \tag{2}$$

$$Max_{Thr} = 3 \times Min_{Thr} \tag{3}$$

$$AvQ = ((1 - w_q) \times Avg) + (Cur_{Que} \times w_q) \tag{4}$$

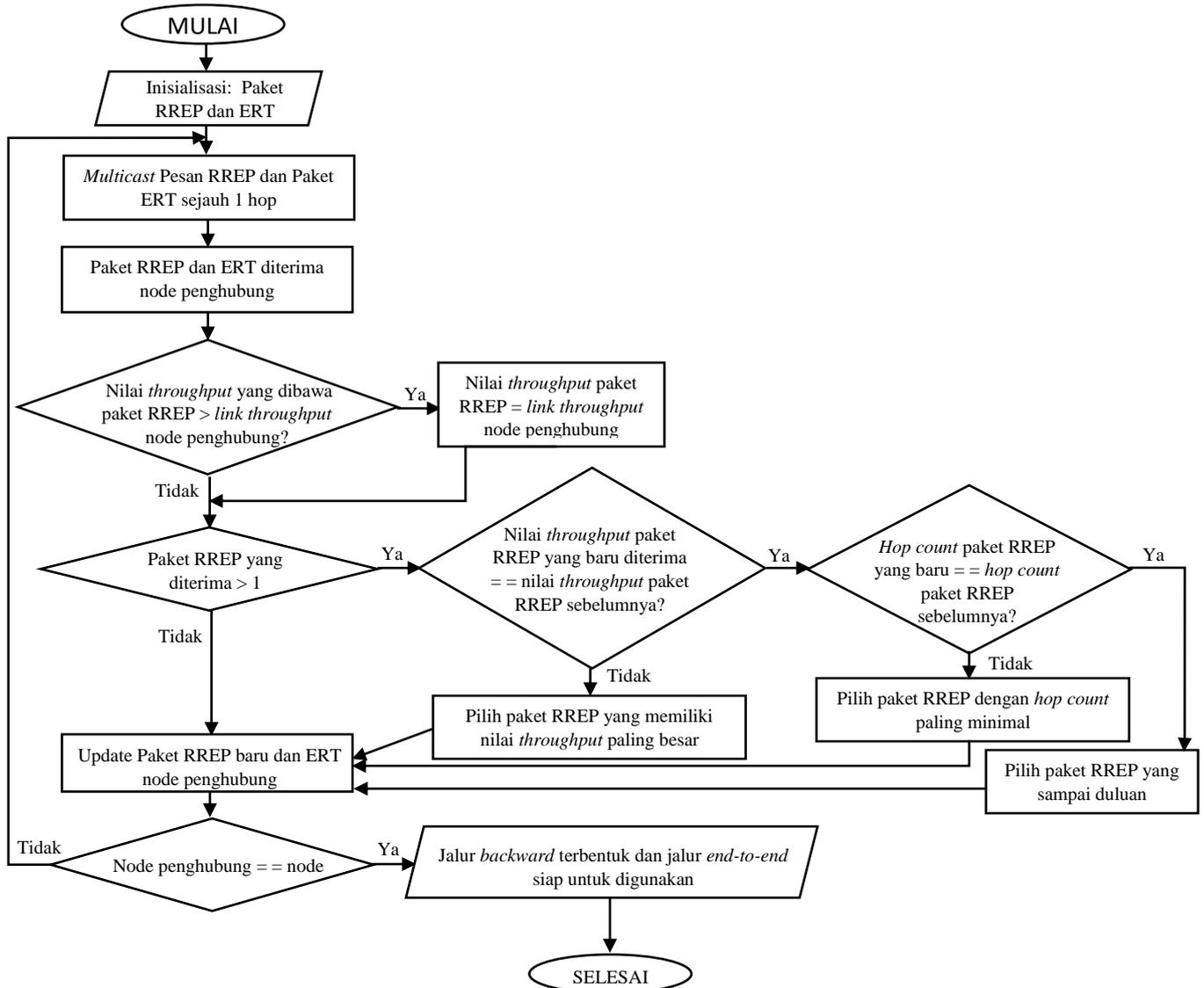
$$Status_{Que} = Cur_{Que} - AvQ \tag{5}$$

$Min_{Thr}$  adalah nilai batas minimum *Threshold*.  $Max_{Thr}$  adalah nilai batas maksimum *Threshold*.  $AvQ$  adalah nilai rata-rata dari ukuran paket antrian (persamaan 4).  $w_q$  adalah bobot (*weight*) antrian dengan nilai konstan 0,002.  $Cur_{Que}$  adalah ukuran paket antrian saat ini, sedangkan  $Status_{Que}$  merupakan kondisi dari nilai aktual antrian (persamaan 5). Ukuran antrian semakin berkurang

manakala paket data yang datang dalam jaringan akan memenuhi antrian. Hasil perbandingan  $Status_{Que}$  akan diklasifikasikan ke dalam nilai CC. Jika  $Status_{Que}$  kurang dari  $Min_{Thr}$ , maka node diklasifikasikan ke area aman dari kongesti (Area I) dengan nilai CC=000. Apabila  $Status_{Que}$  lebih dari  $Min_{Thr}$  dan  $Cur_{Que}$  kurang dari  $Max_{Thr}$ , maka node diklasifikasikan ke dalam area mendekati kongesti (Area II) dengan nilai CC=001, sedangkan jika  $Cur_{Que}$  lebih dari  $Max_{Thr}$  maka node diklasifikasikan ke dalam area dalam keadaan kongesti (Area III) dengan nilai CC=010.

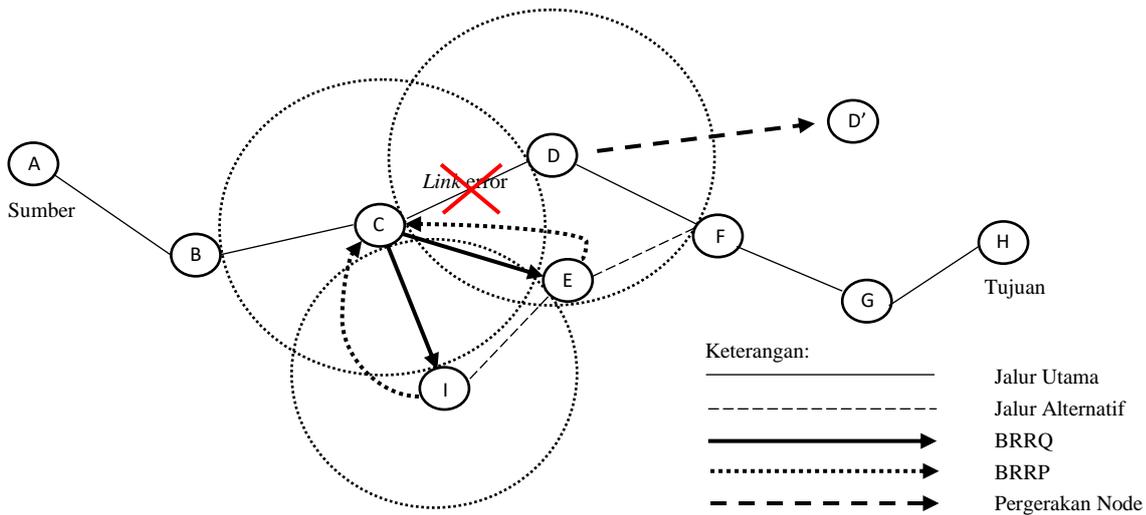
Jika sebuah node penghubung memiliki nilai CC=010, artinya *link error* dan node tersebut segera mentransmisikan CSP (*Congestion Status Packet*) sejauh satu *hop* ke node tetangganya. Pesan CSP yang dikirim berisi nilai status kongesti (CC) dan parameter *routing* (*Source S, Destination D, Hop Count hop, Sequence Number Seq, Neighbors Information N\_list*). Node tetangga yang menerima CSP akan mencari jalur alternatif dan menghapus *link* yang *error* dari ERT-nya.

**RQ2: Bagaimana skema pencarian jalur pada desain Modifikasi AODV usulan?**

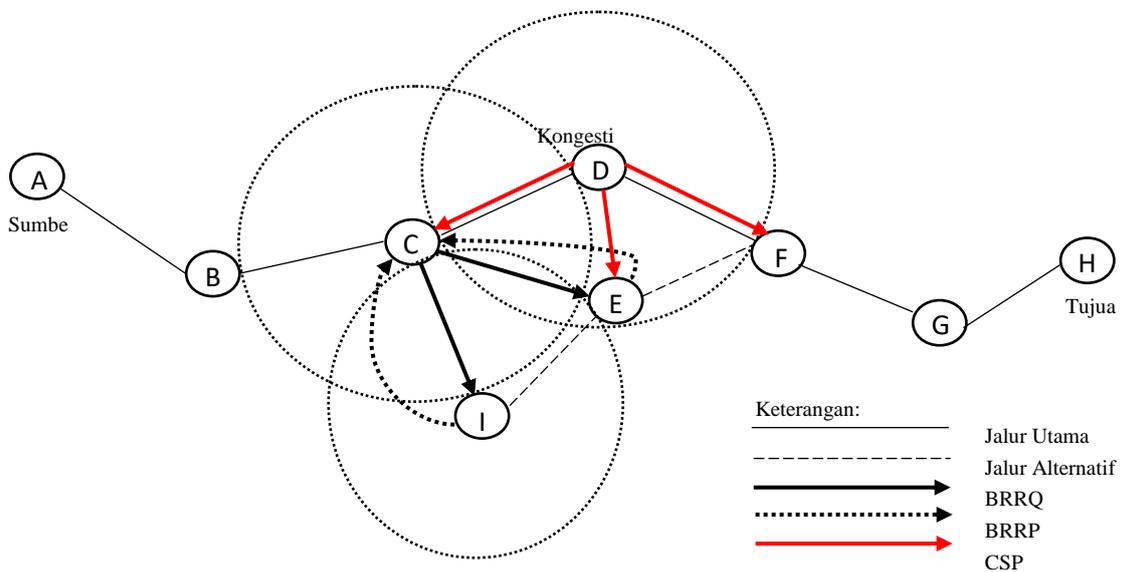


Gambar 4. Discovery Backwarding Routes





Gambar 5. Proses transmisi pesan BRREQ untuk mencari jalur alternatif ketika link error



Gambar 6. Skema pemeliharaan jalur saat node sedang mengalami kongesti

Saat pengiriman paket data atau saat *hello message* ditransmisikan, ternyata node C mendeteksi adanya error *link* di depannya, maka node C memilih jalur alternatif lain berdasarkan ERT miliknya. Namun, jika ERT node C tidak menyediakan jalur alternatif, maka node C mentransmisikan secara *broadcast* pesan BRRQ sejauh 1 *hop* ke node tetangganya, yaitu node E dan I. Pesan ini kemudian direspon oleh node E dan I yang memiliki jalur ke node tujuan. Responsi berupa pengiriman pesan balasan BRRP dan ERT node tetangga. Jika node C mendapatkan pesan BRRP lebih besar dari 1, maka node C memilih jalur alternatif dengan *throughput* paling besar, Jika *throughput* sama, maka node C memilih jalur alternatif yang jumlah *hop*-nya paling minimal. Jika *throughput* dan *hop* sama, maka node C memilih jalur alternatif berdasarkan pesan BRRP mana yang sampai duluan.

Misal node E memiliki jalur *throughput* yang tinggi dan *hop count* paling minimal, maka node E yang menjadi node penghubung alternatif menggantikan node D, sehingga data ERT node E, yakni  $F \rightarrow G \rightarrow H$  disubstitusikan ke tabel data ERT node C, tentu dengan penambahan node E sebagai *next hop*<sup>1</sup> untuk node C,

sehingga jalur alternatif terbentuk menjadi  $A \rightarrow B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow H$ . Berikut adalah ERT node C setelah diupdate:

Tabel II  
Extended Routing Table untuk Node C setelah di update

Entri	Tujuan	Next hop <sup>1</sup> (C)	Next hop <sup>2</sup> (C)	Next hop <sup>3</sup> (C)	...	Next hop (C)	Throughput
...	...	...	...	...	...	...	...
i-1	A	B	...	...	...	...	5,5 Mbps
I	H	E	F	G	...	...	5,5 Mbps
...	...	...	...	...	...	...	...

Selanjutnya, berikut skema pemeliharaan jalur menggunakan jalur alternatif saat node sedang mengalami kongesti dapat dilihat pada Gambar 6. Misal pada node D terjadi kondisi penumpukan antrian paket data, dimana *StatusQue* antrian paket lebih besar dari *MaxThr* *buffer* node D, maka node D dikategorikan sebagai node kongesti, dengan nilai  $CC=010$ . Node D segera mengirimkan CSP (*Congestion Status Packet*) ke node

tetangganya sejauh 1 hop, yakni node C, E, dan F. Kemudian node C, E, dan F mengupdate ERT mereka dengan menghapus *routing* yang melewati hop node D. Jika ternyata setelah penghapusan *routing*, node C tidak menemukan jalur ke node tujuan, maka skema pencarian jalur alternatif menggunakan cara yang sama, seperti ketika pencarian jalur dilakukan saat kerusakan fisik atau mobilitas node terjadi.

## V. KESIMPULAN DAN SARAN

Untuk pencarian jalur pada desain modifikasi AODV yang diusulkan penelitian ini, selain mempertimbangkan minimum *hop count*, juga dipertimbangkan jalur *throughput* yang bisa dilewati untuk pemrosesan seleksi jalur. Selain itu, pencarian jalur dilakukan dengan mengirimkan pesan RREQ yang disisipi frame paket data, serta mengirimkan paket ERT (*Extended Routing Table*) untuk mengupdate tabel ERT untuk setiap node yang telah dilewati. Untuk pemeliharaan jalur, ada dua kondisi yang dipertimbangkan. Kondisi pertama saat node mengidentifikasi node tetangganya yang rusak saat pengiriman paket data berlangsung maupun melalui pengiriman pesan *hello message*. Jika ditemukan rusak sejauh 1 hop, maka *routing* yang mengarah ke node tersebut akan dihapus. Kondisi kedua saat node sedang mengalami *density* (kepadatan) jalur diakibatkan antrian pengiriman paket yang mendekati kapasitas maksimal *buffer* node, node tersebut kemudian mengirimkan CSP (*Congestion Status Packet*) ke node tetangganya sejauh 1 hop. Node yang menerima CSP kemudian mengalihkan jalur ke *routing* alternatif lain dan menghapus *routing* yang mengarah ke node yang sedang kongesti. Kedua kondisi tersebut mengakibatkan jalur utama terputus, sehingga diperlukan pencarian jalur alternatif. Untuk pencarian jalur *routing* alternatif itu sendiri tidak dilakukan dengan menginisialisasi ulang dari node sumber ke node tujuan, melainkan cukup dengan pertukaran pesan BRRQ dan BRRP di sekitaran node penghubung. Jika alternatif *routing* ditemukan dari data *routing* node tetangga tertentu, maka node penghubung cukup mensubstitusi data *routing* yang lama dengan data *routing* yang berasal dari node tetangga tersebut pada tabel ERT-nya.

Saran untuk penelitian selanjutnya adalah hasil desain rancangan usulan penelitian ini dapat diujicobakan ke dalam simulasi skala lab untuk kemudian dilakukan penilaian performansi dengan membandingkan nilai PDR, *throughput*, *overall routing traffic*, dan *overall end-to-end delay* terhadap data kontrolnya, yaitu AODV\_BRL dan AODV konvensional. Algoritma diterapkan dalam skenario yang sama pada *software* simulasi OPNET Modeler 16.0. Diharapkan dengan perbandingan ini, algoritma AODV modifikasi baru menghasilkan performa yang lebih baik daripada AODV konvensional.

## REFERENSI

- [1] R. F. Sari, A. Syarif and B. Budiarjo, "Analisis Kinerja Protokol *Routing* AODV Pada Jaringan Ad Hoc Hybrid: Perbandingan Hasil Simulasi Dengan NS-2 dan Implementasi Pada Testbed dengan PDA," *Makara, Teknologi*, vol. 12, no. 1, pp. 7-18, 2008.
- [2] S. Mittal and P. Kaur, "Performance Comparison of AODV, DSR, and ZRP *Routing* Protocols in MANET'S," in *International Conference on Advances in Computing, Control, and Telecommunication Technologies*, 2009.
- [3] A. B. Malany, V. S. Dhulipala and R. M. Chandrasekaran, "Throughput and Delay Comparison of MANET *Routing* Protocols," *Int. J. Open Problems Compt. Math*, vol. 2, no. 3, pp. 461-468, 2009.
- [4] C. E. Perkins and E. M. Royer, "Ad Hoc Demand Distance Vector *Routing*," in *Proc. IEEE Symp. The 2th IEEE Workshop on Mobile Systems and Application*, 1999.
- [5] E. M. Reyer, "A Review of Current *Routing* Protocols for Ad Hoc Mobile Wireless Networks," in *IEEE Personal Communication*.
- [6] H. Yang and Z.-y. Li, "Simulation and Analysis of a Modified AODV *Routing* Protocols," in *International Conference on Computer Science and Network Technology*, 2011.
- [7] B. LI, Y. Liu and G. Chu, "Optimized AODV *Routing* Protocol for Vehicular Ad Hoc Networks," in *IEEE*, 2010.
- [8] E. S. Dewo, Bandwidth dan Throughput, IlmuKomputer.com, 2003.
- [9] L. U. Khan, S. A. Mahmud, M. H. Zafar, G. M. Khan and H. S. Al-Raweshidy, "M-AODV: Modified Ad Hoc On-demand Distance Vector *Routing* Scheme," in *9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP)*, 2014.
- [10] L. Yujun and H. Lincheng, "The Research on an AODV-BRL to Increase Reliability and Reduce *Routing* Overhead in Manet," in *International Conference on Computer Application and System Modeling (ICCASM)*, 2010.
- [11] S. T. Kumaran and V. Sankaranarayanan, "Early Congestion Detection and Adaptive *Routing* in MANET," *Egyptian Informatics Journal*, vol. 12, no. 3, pp. 165-175, 2011.
- [12] I. N. B. Hartawan and W. Wibisono, "Mekanisme Pemilihan MPR dengan Congestion Detection dalam OLSR Pada MANET," *Jurnal Ilmiah Ilmu Komputer Universitas Udayana*, vol. 6, no. 2, pp. 11-17, 2013.
- [13] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions Networking*, vol. 1, no. 4, pp. 397-413, 1993.
- [14] Y. Wang, Y. Zhou\*, Y. Yu, Z. Wang and S. Du, "AD-AODV: A Improved *Routing* Protocol Based on Network Mobility and Route Hops," 2012.
- [15] C. T. Cuong, V. T. Tu and N. T. Hai, "MAR-AODV: Innovative *Routing* Algorithm in MANET based on Mobile Agent," in *27th International Conference on Advanced Information Networking and Applications Workshops*, 2013.