

PENGEMBANGAN PROTOKOL MULTICAST AODV DENGAN MEMPERHITUNGGAN JARAK EUCLIDEAN BERDASARKAN POSISI, KECEPATAN DAN DELAY TRANSMISI PADA VANET

Nurdiansyah Rezkinanda¹⁾, dan Radityo Anggoro²⁾

^{1, 2)}Jurusan Teknik Informatika, Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember (ITS)

e-mail: nurdiansyah14@mhs.if.its.ac.id¹⁾, onggo@if.its.ac.id²⁾

ABSTRAK

Protokol routing dalam jaringan Vehicular Ad hoc Network (VANET) dikembangkan untuk memperbaiki kinerja transmisi paket data pada jaringan antar kendaraan. Berbagai pendekatan konektivitas antar jaringan kendaraan terus dilakukan karena VANET memiliki karakteristik yang berbeda dengan Mobile Ad hoc Network (MANET) yaitu dalam hal kecepatan node. Protokol Multicast Ad hoc On-Demand Distance Vector (MAODV) yang termasuk protokol routing reaktif dengan transmisi multicast, banyak dilakukan penelitian dan peningkatan dengan berbagai pendekatan. Pada penelitian ini dilakukan pengembangan protokol routing MAODV dengan memperhitungkan jarak Euclidean antar node berdasarkan posisi, kecepatan dan delay transmisi. Protokol routing Multicast Adaptif Structured-tree berbasis Reactive Euclidean Node Knowledge (MAS-BRENK) diusulkan untuk memperbaiki mekanisme multicast tree maintenance yaitu proses join dan prune. Perhitungan bobot *weighted product* digunakan untuk menghitung bobot antar node berdasarkan jarak euclidean, kecepatan dan delay transmisi. Protokol yang diusulkan tersebut diujikan ke dalam skenario jalan perkotaan. Hasil pengujian menunjukkan bahwa protokol MAS-BRENK mengalami peningkatan pengiriman paket MACT sebesar 1.3% dan penerimaan paket MACT dengan flag prune sebesar 1.2%. Selain itu penerimaan paket MACT dengan flag join menurun sebesar 1.2%. Hasil akhir yang didapatkan yaitu peningkatan PDR sebanyak 0.2% dan throughput sebanyak 2%, serta penurunan delay 14.3% dari protokol MAODV.

Kata Kunci: Euclidean Distance, MAODV, multicast, VANET, *weighted product*

ABSTRACT

Routing protocol on Vehicular Ad hoc Network (VANET) was developed to improve the performance of data packet transmission in vehicular network. Various approach of vehicular connectivity continued because VANET has different characteristics with Mobile Ad hoc Network (MANET) especially node velocity. Multicast Ad hoc On-Demand Distance Vector (MAODV) Protocol which includes a reactive routing protocol with multicast transmissions done much research and an improve in the variety of approaches. In this study, we improve MAODV routing protocol by calculating Euclidean distance between nodes based on position, velocity and transmission delay. We propose Multicast Adaptive Structured-tree Base on Euclidean Node Knowledge (MAS-BRENK) routing protocol to improve multicast tree maintenance mechanism that join and prune process. Weighted calculation of *weighted product* is used to calculate the weights between nodes based on Euclidean distance, velocity and transmission delay. The proposed protocol was tested in a scenario of urban roads. The test results showed that the MAS-BRENK protocol MACTs packet delivery increased by 1.3% and the MACTs packet receipt with prune flag by 1.2%. In additions MACT packet reception join flag decreased by 1.2%. The final result is an increase in PDR by 0.2% and throughput by 2%, and the delay decreased 14.3% from MAODV protocol.

Keywords: Euclidean Distance, MAODV, multicast, VANET, *weighted product*

I. PENDAHULUAN

Vehicular Ad hoc Network (VANET) merupakan perkembangan dari Mobile Ad hoc Network (MANET) dimana mobilitas *node* sangat tinggi pergerakannya dapat diprediksi dan topologi jaringan lebih dinamis dibandingkan dengan MANET. Mobilitas *node* yang sangat tinggi pada VANET berpengaruh besar terhadap perubahan topologi jaringan setiap waktu. Selain menyebabkan perubahan topologi, hal tersebut juga dapat menyebabkan rute terputus karena *node* keluar dari jangkauan sinyal transmisi [14]. Dengan tingginya mobilitas *node*, kegagalan *node*, dan karakteristik mobilitas *node* yang dinamis, *link* pada sebuah rute dapat menjadi tidak tersedia. Rute yang rusak memaksa *node* sumber mencari ulang rute untuk mentransmisikan data ke *node* tujuan. Hal ini dapat menyebabkan *delay* dan banyak paket yang hilang [2].

Protokol routing AODV merupakan salah satu protokol yang menggunakan transmisi *unicast* untuk melakukan pengiriman data dari *node* sumber menuju *node* tujuan [11]. AODV merupakan penggabungan dari mekanisme *route discovery* dan *route maintenance* pada protokol routing Dynamic Source Routing (DSR) serta penggunaan *hop-by-hop routing*, *sequence number* dan *periodic beacon* pada protokol routing Destination-Sequenced Distance Vector (DSDV). Kelebihan AODV yaitu menyimpan rute yang dibutuhkan sehingga kebutuhan memori sedikit, perbaikan rute yang rusak dapat ditangani dengan baik dan mempunyai skalabilitas yang tinggi pada jumlah *node*.

Prediksi jarak antar *node* diajukan berdasarkan posisi *node* dengan nilai *threshold* kurang dari sama dengan jangkauan transmisi *node*, serta *root node* dan *road static unit* (RSU) sebagai kontrol antar *node* [8], secara signifikan dapat meningkatkan *throughput* dan *packet delivery ratio* serta menurunkan *overhead* dan *end-to-end delay*. Namun protokol routing AODV yang masih menggunakan transmisi *unicast* masih perlu dioptimalkan.

Ketika *node* sumber melakukan pengiriman data menuju lebih dari satu *node* tujuan, metode transmisi pada protokol routing menjadi masalah penting. Karena pada transmisi *unicast*, *node* sumber mengirimkan paket data berulang kali bergantung pada jumlah *node* tujuan. Masalah protokol routing dengan transmisi *unicast* dapat ditangani dengan menggunakan transmisi *multicast* ketika *node* sumber mengirimkan paket data kepada lebih dari satu *node* tujuan. Salah satu pengembangan dari AODV dengan menerapkan penggunaan transmisi *multicast* adalah protokol routing Multicast Ad hoc On-Demand Distance Vector (MAODV). Protokol routing MAODV memungkinkan *node* sumber mendapatkan rute menuju beberapa *node* tujuan secara *multicast* dengan membentuk *multicast tree* yang terbagi dalam kelompok *multicast (multicast group)* tertentu.

Pada protokol routing MAODV, terdapat proses *multicast tree maintenance* dimana memungkinkan *node* dikeluarkan atau *prune* dari *multicast group* dan melakukan perbaikan rute yang terputus dengan bergabung atau *join* terhadap *multicast group* tertentu untuk melakukan pencarian rute baru. Pada penelitian [5] ditunjukkan bahwa MAODV memiliki tingkat *scalability* dan *overhead* yang sedang dibandingkan dengan protokol Adaptive Demand-driven Multicast Routing (ADMR) dan Ad hoc Multicast Routing (AMR). Permasalahan utama pada protokol routing MAODV yaitu tidak disertakannya informasi jarak antar *node* baik dalam proses *multicast tree creation* maupun *multicast tree maintenance*. Penelitian [16] menunjukkan bahwa protokol MAODV mengalami penurunan *throughput* ketika kecepatan *node* meningkat dibandingkan dengan protokol ADMR dan On-Demand Multicast Routing Protocol (ODMRP).

Sehubungan dengan mobilitas *node* yang tinggi, *node* yang memiliki potensi mobilitas menjauh dari jangkauan *node* lain dapat menyebabkan kerusakan dan perpindahan rute. Perpindahan rute terjadi saat sebuah *node* mengirim pesan *error* ke *node* sumber. Meskipun rute baru dapat dibentuk ketika kerusakan terjadi, berulang kali membangun kembali rute baru dapat menimbulkan *delay* dan *overhead* yang besar [17]. Disisi lain, topologi jaringan VANET tidak terlepas dari informasi posisi, kecepatan dan *delay* transmisi yang selalu berubah setiap saat secara dinamis. Maka informasi tersebut hingga pergerakan setiap *node* dapat diprediksi atau tidak acak, hal ini dapat membantu mempermudah pengiriman paket data [9]. Diajukannya *Motion-MAODV* [7] dengan perhitungan *route stability* yang mempertimbangkan kecepatan dan jumlah *hop node* menuju *group leader*, membuktikan dapat meningkatkan *packet delivery ratio*, namun *end-to-end delay* juga mengalami peningkatan.

Berdasarkan penelitian dan permasalahan yang telah dijabarkan sebelumnya, diusulkan suatu mekanisme pembentukan dan pelepasan *multicast tree* yang berbeda pada protokol routing MAODV dengan memperhitungkan jarak *Euclidean* berdasarkan posisi, kecepatan dan *delay* transmisi. Dalam penelitian ini dinamakan dengan protokol routing Multicast Adaptif Structured-tree based on Reactive Euclidean Node Knowledge (MAS-BRENK). Dengan tingginya mobilitas kendaraan pada VANET, perhitungan Euclidean Node Knowledge (ENK) dapat dipergunakan untuk melakukan *pruning* atau pelepasan anggota *multicast group* sebelum rute terputus sehingga proses *prune* tidak lagi menunggu rute *timeout*. Selain itu, perhitungan tersebut menggunakan *node* sebagai pertimbangan dalam melakukan *join* kembali kepada *multicast group*, sehingga intensitas pembentukan rute secara berulang kali dapat diminimalisasi. Diharapkan mekanisme pemutusan rute dan pembentukan ulang rute baru yang diusulkan dapat mengurangi *end-to-end delay*, serta *packet delivery ratio* (PDR) dan *throughput* dapat ditingkatkan.

II. KAJIAN PUSTAKA

A. Vehicular Ad-hoc Network (VANET)

Vehicular Ad-hoc Network (VANET) merupakan pengembangan dari Mobile Ad-hoc Network (MANET) dimana pengembangannya difokuskan pada kendaraan atau *vehicle* yang dapat saling berkomunikasi maupun mengirimkan data. VANET adalah sebuah teknologi baru yang memadukan kemampuan komunikasi nirkabel kendaraan menjadi sebuah jaringan yang bebas infrastruktur serta memiliki karakteristik mobilitas yang sangat tinggi dan terbatas pada pola pergerakannya [6]. *Node* dalam jaringan dianggap sebagai *router* yang bebas bergerak dan bebas menentukan baik menjadi *client* maupun menjadi *router*. Protokol routing pada VANET memiliki dua model yaitu protokol routing reaktif yang membentuk tabel routing hanya saat dibutuhkan dan protokol routing proaktif yang melakukan pemeliharaan tabel routing secara berkala pada waktu tertentu.

Karena VANET merupakan komunikasi antar kendaraan diasumsikan memiliki perangkat *Global Positioning System* (GPS), maka informasi kecepatan, jalan, arah hingga pergerakan setiap *node* dapat diprediksi / tidak acak.

Hal ini dapat membantu mempermudah pengiriman paket data [9]. VANET akan membentuk jaringan *multi-hop* antar *node* untuk mengirimkan data menuju *node* lain ataupun *static intersection node* [10]. Pergerakan *node* bisa berubah setiap saat dan terbatas pada rute lalu lintas yang dapat ditentukan dari koordinat peta. Sehingga setiap *node* akan terus memperbarui informasi dalam tabelnya sesuai informasi dari *node* lain. Perubahan pergerakan menjadi salah satu permasalahan dalam pengiriman paket data sehingga dibutuhkan informasi jarak antar *node*, kecepatan dan *delay* transmisi.

B. Multicast Ad hoc On-Demand Distance Vector (MAODV)

Berdasarkan jumlah penerima paket, tipe transmisi data terbagi menjadi beberapa macam yaitu *unicast*, *broadcast*, *geocast* dan *multicast*. Beberapa aplikasi seperti *remote desktop* dan pengiriman *file* pada FTP mungkin membutuhkan transmisi *unicast* dengan satu alamat tujuan. Namun aplikasi seperti *video streaming* membutuhkan transmisi *multicast* untuk mengirimkan paket data ke beberapa tujuan dengan sekali pengiriman. Pada metode transmisi *multicast*, paket data yang dikirimkan oleh *node* sumber disertai dengan alamat *multicast* tujuan agar dapat diterima oleh lebih dari satu *node* tujuan. Dengan kata lain, dalam transmisi *multicast* paket data akan dikirimkan ke alamat dalam kelompoknya saja. *Node* yang menjadi tetangga dari *node* sumber akan menggandakan paket data dan menentukan tujuan alamat masing-masing paket data tersebut melalui rute yang telah ditentukan ke dalam kelompok *multicast* sampai diterima oleh *node* tujuan.

Metode transmisi *multicast* diusulkan kedalam protokol routing AODV yang dinamakan *Multicast Ad hoc On-Demand Distance Vector* [12]. Pada protokol routing MAODV, ditambahkan tabel *routing* yaitu tabel *multicast group* yang berisi *multicast group IP address*, *multicast group leader IP address*, *multicast group sequence number*, *next hops*, *hop count to multicast group leader*, dan *hop count to multicast group member*. Selain itu juga ditambahkan tabel *group leader* berisi *multicast group IP address* dan *group leader IP address*. Dalam satu kelompok *multicast group*, mekanisme pengiriman data yang dipergunakan adalah protokol routing AODV. Antara satu sama lain *multicast group*, protokol routing MAODV yang berperan membentuk *tree* dan melakukan pemeliharaan rute *multicast*. Jika sebuah *node* ingin menjadi anggota dari *multicast group* tertentu, maka *node* tersebut harus melakukan mekanisme *route discovery* terlebih dahulu. Setiap *node* dalam *multicast group* diatur oleh satu *node* yaitu *group leader*. *Group leader* berperan penting memelihara *group sequence numbers* dari anggota *multicast group* dengan cara mengirimkan *group hello message* kepada anggotanya tiap periode tertentu.

Pada mekanisme pembentukan *multicast tree*, *node* sumber akan melakukan *broadcast* pesan *multicast RREQ* terlebih dahulu untuk mencari rute menuju *node* tujuan. Pesan *multicast RREQ* disertai *flag J* dikirimkan oleh *node* sumber jika ingin membentuk atau menjadi bagian kelompok *multicast group* dan begitu pula sebaliknya. Kemudian membuat daftar rute kosong pada tabel *multicast route* dan menjadikan dirinya sebagai *group member* tanpa *group leader address*. Hal tersebut dilakukan bilamana *node* sumber mendapatkan rute menuju *multicast group* lain yang sudah terbentuk, maka *node* sumber akan mengisi informasi mengenai tabel *multicast* dan tabel *group leader*. Pesan *multicast RREP* dengan *flag J* dipergunakan *node* tetangga untuk membalas pesan *node* sumber yang telah mengirimkan pesan *multicast route request* sebelumnya. Jika beberapa *node* tetangga mengirimkan pesan *multicast RREP* kepada *node* sumber, maka *node* sumber memilih rute dari *multicast RREP* dengan *sequence number multicast group* yang lebih besar. Pesan *multicast route activation* (MACT) dipergunakan untuk menggabungkan informasi rute dari *node* maupun *multicast group* ke dalam *multicast group* lainnya. Kemudian membentuk rute sesuai informasi yang diperoleh dan menyimpannya kedalam tabel *multicast route*.

Jika beberapa *node* tetangga yang menerima pesan *multicast RREQ* dengan mengirimkan pesan *multicast RREP* kepada *node* sumber, maka *node* sumber memilih rute dari *multicast RREP* dengan *sequence number multicast group* maupun *sequence number anggota group* dari *multicast group* yang lebih besar. Setiap *node* yang memiliki anggota *multicast group* maupun menjadi penghubung antar *multicast group* bertugas untuk memelihara rute. Dan setiap *node* tersebut disebut dengan *group leader*, dimana setiap waktu tertentu melakukan *broadcast* pesan *multicast group hello* (MGPRH). Anggota *multicast group* maupun *node* penghubung antar *multicast group* harus melakukan *update* terhadap tabel *group leader*. Untuk memelihara rute antar *node*, digunakan pesan *one-hop neighbor-hello*. Pada mekanisme pemeliharaan *multicast tree*, *node* pengirim pesan *multicast RREQ* disebut dengan *upstream* serta *node* pengirim pesan *multicast RREP* disebut dengan *downstream*. *Upstream* akan mengirimkan pesan *one-hop neighbor-hello* kepada *downstream* setiap periode tertentu. Jika *downstream* merupakan *group leader* dari *multicast group* dan tidak menerima pesan *broadcast* dari *upstream*, maka *downstream* menganggap bahwa rute telah terputus. Kemudian informasi rute menuju *upstream* pada tabel *multicast* dihapus dengan mengirimkan pesan *multicast activation* dengan *flag P* (MACT_P) kepada anggota *multicast tree*. Kemudian *downstream* akan mengirim pesan *multicast RREQ* kembali untuk mendapatkan rute yang baru sesuai dengan mekanisme pembentukan *multicast tree*. Setelah sebuah *node* terpilih menjadi *group leader* dari *upstream* tersebut, maka *group leader* akan mengirimkan *broadcast* pesan MGRPH dengan *flag U*

(*update flag*). Jika *upstream* tidak menerima pesan *multicast RREP* dari *downstream*, maka *upstream* akan melakukan hal yang sama yaitu menghapus informasi rute menuju *downstream* dengan mengirimkan *MACT_P* ke *multicast tree*.

C. Euclidean Distance

Setiap *node* yang bergerak maupun diam mempunyai informasi posisi *node* yang dapat diasumsikan sebagai titik koordinat (x,y) pada bidang kartesius. Perpindahan posisi *node* direpresentasikan dalam koordinat dua dimensi (x,y) yaitu $P_1 = (x_1,y_1)$ dan $P_2 = (x_2,y_2)$ pada Gambar 1.

Perhitungan jarak euclidean dari titik P_1 ke titik P_2 mendukung perhitungan lebih dari tiga (n) variabel antara kedua objek yang ditunjukkan pada Persamaan (1).

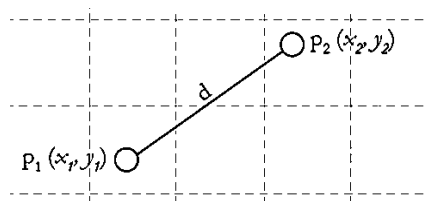
$$d_{op} = \sqrt{\sum_{i=1}^n |o_n - p_n|^2} \quad (1)$$

Dengan d_{op} adalah jarak antara titik o dan p dan n adalah banyaknya variabel yang dihitung, serta o_n merupakan variabel o ke n dan p_n adalah variabel p ke n . Vektor digunakan untuk menunjukkan arah dan posisi dalam suatu bidang atau ruang. Posisi titik materi pada suatu bidang dapat dinyatakan dalam bentuk vektor. Vektor satuan i dan j dan masing-masing menyatakan arah sumbu x dan y positif dalam bidang kartesian. Sebuah vektor A berada pada bidang xy , hasil komponen vektor A_x dan vektor satuan i adalah vektor $A_x i$ yang berada pada sumbu x dan besarnya $|A_x|$. Demikian dengan vektor $A_y j$ yang berada pada sumbu y dan besarnya $|A_y|$. Sehingga vektor satuan untuk vektor A adalah seperti ditunjukkan pada Gambar 2.

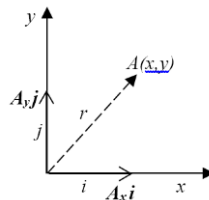
Kecepatan merupakan perpindahan posisi suatu benda terhadap satuan waktu. Ketika posisi benda berpindah dari titik awal A ke titik akhir B adalah Δr dengan selang waktu Δt maka kecepatan rata-rata (\bar{v}) dirumuskan sebagai perbandingan antara perpindahan (Δr) yang ditempuh benda terhadap selang waktu (Δt). Gerak sebuah benda pada bidang kartesian x,y dapat dirumuskan jika posisi benda berpindah terhadap selisih waktu tertentu.

D. Weighted Product

Metode *Weighted Product* merupakan metode penentuan prioritas dalam analisis lebih dari satu kriteria pada *multi atributte decision making* (MADM). Seperti pada penelitian [1] tentang penentuan algoritma penggunaan jaringan nirkabel (UMTS1, UMTS2, Wifi, WiMax) pada kasus pengiriman data tertentu. Contohnya pada aplikasi percakapan dan *streaming* yang sensitif terhadap *delay* dan *jitter* meskipun tidak menghabiskan trafik data yang besar. Dengan membandingkan beberapa metode MADM diantaranya *Preference Ranking Organization Method for Enrichment Evaluation* (PROMETHEE), *Simple Additive Weighting* (SAW) dan *Weighted Product Method* (WPM) untuk menentukan penggunaan jaringan nirkabel terbaik dalam pengiriman data tertentu seperti data percakapan dan streaming. Pada penelitian tersebut disimpulkan bahwa dengan menggunakan metode PROMETHEE menentukan penggunaan jaringan lebih optimal menggunakan jaringan wireless UMTS1 dibandingkan UMTS2, Wifi maupun WiMax ketika dibutuhkan untuk pengiriman data percakapan dan streaming yang membutuhkan atribut atau kriteria khusus seperti *delay* dan *jitter*. Ketika pengiriman data heterogen, baik percakapan, *streaming*, pengiriman *file*, aplikasi yang membutuhkan sinkronisasi, dan lain-lain berjalan bersama dalam melakukan pengiriman data, maka metode penentuan yang terbaik yaitu SAW dan WPM.



Gambar 1. Jarak d antara *node* P_1 dengan *node* P_2 [9]



Gambar 2. Vektor posisi

Pada penelitian lain [13] mengenai proses pengalihan penggunaan konektivitas perangkat mobile (proses *Handover*) juga membutuhkan penentuan dengan mempertimbangkan *delay*, *bandwidth*, *cost* dan *jitter*. Pada penelitian tersebut dipilih metode SAW dan WPM untuk menentukan penggunaan konektivitas perangkat yaitu Wifi dan WiMax. Dari hasil penelitian tersebut, metode WPM menunjukkan hasil penentuan yang lebih baik 35,75% dibandingkan metode SAW 12,64%.

Metode WPM mengevaluasi beberapa alternatif terhadap sekumpulan atribut atau kriteria, dimana setiap atribut saling tidak bergantung satu dengan yang lainnya. Menggunakan teknik perkalian untuk menghubungkan rating atribut, dimana rating tiap atribut harus dipangkatkan terlebih dahulu dengan bobot atribut seperti proses normalisasi. Preferensi untuk *alternative Si* yaitu sebagai berikut:

$$S_i = \prod_{j=1}^n X_{ic}^{w_j c} \tag{2}$$

Dengan alternatif *i* dan banyaknya kriteria *n* yang mempunyai nilai kriteria *X*, serta *w* sebagai bobot kriteria dan *c* sebagai atribut kriteria maka proses selanjutnya yaitu preferensi alternatif vektor *V* untuk langkah terakhir dilakukan tahap perankingan. Preferensi untuk *alternative Vi* dengan menghitung masing – masing preferensi *alternative Si* dibagi dengan jumlah keseluruhan preferensi *alternative Si*.

III. METODOLOGI PENELITIAN

Rancangan sistem protokol *routing Multicast Adaptif Structured-Tree Based on Reactive Euclidean Node Knowledge* (MAS-BRENK) dikembangkan dari protokol routing MAODV. Beberapa bagian yang dimodifikasi akan dijelaskan pada sub bab berikut.

A. Euclidean Node Knowledge (ENK)

Perhitungan jarak *euclidean* antar *node* dengan mempertimbangkan parameter posisi, kecepatan dan *delay* transmisi dalam penelitian ini disebut sebagai *euclidean node knowledge* (ENK). Perhitungan ENK dipergunakan untuk memutuskan proses *join multicast group* (jika belum mempunyai *multicast group*) atau *prune multicast group* (jika sudah menjadi anggota *multicast group*). Perhitungan tersebut juga dipergunakan untuk melakukan pemilihan *group leader* dalam *multicast group*. *Delay* transmisi menjadi pertimbangan penting, dikarenakan *delay* pengiriman paket data akan selalu berubah setiap waktu karena mobilitas kendaraan yang sangat dinamis. Maka perhitungan jarak *euclidean* diusulkan dengan menyertakan waktu pengiriman data ditambah dengan *delay* transmisi, dinotasikan sebagai *t_d*. Ketika sebuah *node* menerima *request* maupun *reply* yang berisi informasi posisi, kecepatan dari *node* pengirim. *Delay* transmisi menyebabkan informasi yang diterima menjadi kurang valid. Maka informasi *node* pengirim (*node i*) yang diterima oleh *node* penerima (*node j*) harus divalidasi.

Posisi *node* pengirim harus divalidasi oleh *node* penerima untuk menghitung jarak antar *node* tersebut. Jika kecepatan *node* pengirim adalah *v_{oi}* dengan menghitung perpindahan posisi *node* pengirim (*s_i*) dari posisi awal (x,y) menuju posisi akhir terhadap *delay* transmisi. Dengan perhitungan jarak, perpindahan posisi *node i* pada sumbu x dan y dapat dinotasikan sebagai berikut.

$$Pos_i(t_j) = \begin{bmatrix} x_i + (dx_i \cdot (t_j - t_i)) \\ y_i + (dy_i \cdot (t_j - t_i)) \end{bmatrix} \tag{3}$$

Setelah diketahui jarak *euclidean node j* terhadap *node i* sebagai salah satu parameter perhitungan ENK. Batas maksimal ENK diambil dari perhitungan terbesar perkalian bobot *node j* terhadap *node i* dan *node* lainnya yang telah mengirimkan informasi pada pesan *multicast activation*. Perhitungan batas maksimal ENK dihitung dengan rumus *weighted product* mempertimbangkan parameter jarak *euclidean*, kecepatan dan *delay* transmisi. Keempat

parameter tersebut menunjukkan banyaknya kriteria, dibagi menjadi empat bagian yaitu kriteria jarak (c_1), kriteria kecepatan (c_2), dan kriteria *delay* transmisi (c_3). Alternatif i merupakan node i yang mengirim pesan MACT berisi informasi node pengirim yang diterima oleh node j . Langkah perhitungan bobot ENK adalah sebagai berikut:

- 1) Alternatif i merupakan *node* yang mengirimkan pesan *multicast* MACT dan pesan tersebut diterima oleh *node* j . Contohnya yaitu jika ada 7 *node* yang mengirimkan pesan MACT, maka ketujuh *node* tersebut merupakan alternatif *node*.
- 2) Kriteria c yaitu kriteria jarak, kecepatan dan *delay* transmisi.
- 3) Bobot awal kriteria dibagi menjadi tiga tingkatan. Bobot terbesar dengan nilai 3, yaitu pada kriteria *delay* transmisi (w_c3). Jarak (w_c1) mempunyai nilai bobot 2. Kecepatan (w_c2) mempunyai nilai bobot 1.
- 4) Jumlah bobot keseluruhan Σw_c adalah 1. Dan bobot dari masing-masing kriteria (w_1, w_2, w_3) adalah bobot awal kriteria dibagi jumlah bobot keseluruhan.
- 5) Vektor S_i dihitung dari kriteria c berdasarkan masing-masing alternatif *node* dipangkatkan dengan bobot.
- 6) Perangkingan alternatif *node* dapat dilakukan dengan perhitungan vektor V_i dimana setiap vektor S_i dibagi jumlah keseluruhan dari vektor S_i .

Masing - masing alternatif *node* mempunyai nilai V_i yang dapat dipergunakan untuk menentukan batas maksimal dan minimal. Dari sini dapat ditentukan bahwa alternatif *node* dengan nilai V_i terbesar merupakan batas maksimal ENK. Alternatif *node* dengan nilai V_i terkecil dapat dijadikan sebagai pertimbangan pemilihan *group leader* pada *multicast group*.

B. Structured-tree

Pembentukan *structured-tree* pada protokol routing MAODV dilakukan dengan mengirimkan propagasi pesan *multicast* RREQ hingga mencapai *node* tujuan. Setelah pesan *multicast* RREQ diterima oleh *node* tujuan, maka *node* tujuan mengirimkan *multicast* RREP menuju *node* sumber. Sehingga *node* sumber dapat mengetahui rute mana saja yang dapat dilalui menuju *node* tujuan. Setelah rute terbentuk, *node* sumber dapat melakukan proses pembentukan *multicast group*. Pesan *multicast activation* (MACT) dikirimkan oleh *node* sumber untuk membentuk *multicast group*. Modifikasi protokol *routing* yang diusulkan, dilakukan dengan menambahkan perhitungan jarak *euclidean* menggunakan parameter posisi, kecepatan dan waktu pengiriman. Ketika sebuah *node* melakukan aktivasi jalur *multicast*, maka *node* tersebut mengirimkan pesan MACT dengan *flag J* (*join*) Pada pengiriman pesan MACT ditambahkan skrip parameter posisi, kecepatan dan *delay* transmisi. Paket *header* pada pesan MACT dimodifikasi dengan menambahkan parameter tersebut untuk dilakukan perhitungan bobot. Setelah rute *multicast* terbentuk, maka *node* akan melakukan aktifasi rute dengan mengirimkan pesan MACT. Hal tersebut bertujuan untuk melakukan validasi bahwa pengiriman data dilakukan dari rute yang sudah terbentuk.

Dari setiap pembentukan *multicast group* antar *node* dengan mobilitas kendaraan yang berubah-ubah setiap waktu, maka sangat dimungkinkan terjadi *link breakage* dan *membership revocation*. *Link breakage* merupakan kejadian kerusakan link pada *tree* karena kondisi tertentu. Terjadinya kerusakan link menyebabkan struktur *tree* akan terpecah. Terpecahnya struktur *tree* ditandai dengan pengiriman pesan MACT dengan *flag P*. Maka *node* tetangga harus melakukan *join* untuk memperbaiki *multicast tree* (*restructured-tree*), dimana *group leader* akan melakukan pencarian rute baru dengan mengirimkan pesan MACT ditandai dengan *flag J*. Penerima pesan MACT dengan *flag J* akan melakukan perhitungan ENK kembali dan memutuskan apakah *node* tersebut dapat bergabung dalam kelompok *multicast group*. *Membership revocation* merupakan pencabutan keanggotaan *node* oleh *group leader* dari anggota *multicast group* karena sebab tertentu. Dengan perhitungan ENK, maka keluarnya *node* dari keanggotaan *multicast group* tertentu dapat diprediksi sehingga akibat yang akan terjadi yaitu *link breakage* dapat segera ditangani. Keluarnya *node* dari keanggotaan *multicast group* (*pruning*) menandakan bahwa *node* tersebut diprediksi tidak mampu lagi melanjutkan pengiriman data karena sebab tertentu. Dengan perhitungan ENK, maka *group leader* akan mengirimkan MACT dengan *flag P* untuk mencabut keanggotaan *node* tersebut dari *multicast group*. Sehingga *node* yang menerima pesan tersebut akan menghapus rute *multicast*.

Anggota kelompok *multicast* (*multicast group member*) menjadi tanggung jawab *group leader* pada pemeliharaan rute. Meskipun bobot ENK masih berada pada nilai kurang dari batas maksimal, *group leader* bisa saja terputus dari *multicast tree* karena sebab tertentu. Jika hal tersebut terjadi maka *group member* harus melakukan pencarian rute kembali untuk membentuk *multicast tree* yang terputus. Dalam hal ini perhitungan ENK tidak lagi dipertimbangkan dalam melakukan penghapusan rute karena rute menuju *group leader* sudah dianggap terputus, sehingga *group member* akan melakukan proses pencarian rute kembali menuju *multicast tree*. Hal tersebut merupakan pertimbangan penting mengingat pergerakan pada VANET sangat dinamis.

TABEL I
SKENARIO MOBILITAS

No	Nama Skenario	Jml. Pengirim	Jml. Penerima
1.	cbr-50-01-02		2
2.	cbr-50-01-05		5
3.	cbr-50-01-10	1	10
4.	cbr-50-01-30		30
5.	cbr-50-01-50		50
6.	cbr-50-02-02		2
7.	cbr-50-02-05		5
8.	cbr-50-02-10	2	10
9.	cbr-50-02-30		30
10.	cbr-50-02-50		50
11.	cbr-50-05-05		5
12.	cbr-50-05-10	5	10
13.	cbr-50-05-30		30
14.	cbr-50-05-50		50
15.	cbr-50-10-10		10
16.	cbr-50-10-30	10	30
17.	cbr-50-10-50		50
18.	cbr-50-30-30	30	30
19.	cbr-50-30-50		50

IV. MODEL MOBILITAS

Untuk mengetahui kinerja dari metode yang diusulkan maka perlu dilakukan uji coba. Dalam penelitian ini skenario mobilitas *node* menggunakan simulator SUMO untuk membangun sebuah peta grid [3]. Peta grid yang dirancang mempunyai jumlah garis *x* (horisontal) 10 dan jumlah garis *y* (vertikal) 10. Panjang persimpangan pada garis horisontal 200m dan garis vertikal 200m, sehingga ukuran peta grid keseluruhan adalah 2000m x 2000m. Model pergerakan kendaraan yang digunakan yaitu sesuai karakteristik perkotaan dengan kecepatan antara 0 - 10 m/s (36 km/jam) [4]. Kepadatan kendaraan juga disesuaikan dengan daerah perkotaan yaitu 50 *node* [4, 14].

V. SIMULASI

Parameter pengujian terdiri dari durasi simulasi yaitu 900 detik, jumlah *node* 50, trafik pengiriman data constant bit rate (CBR), dengan lingkungan topologi peta grid berukuran 2000m x 2000m. Simulasi pengujian dengan jumlah *node* pengirim dan *node* penerima terdapat sebanyak 19 simulasi yang bervariasi. Berikut variasi rincian simulasi berdasarkan jumlah *node* pengirim dan *node* penerima. Lebih lengkapnya ditunjukkan pada Tabel 1.

Untuk evaluasi kinerja dari metode yang diusulkan, data hasil penelitian diolah dengan dilakukan perbandingan terhadap beberapa parameter berikut:

- 1) *End-to-End Delay* ditentukan dari waktu yang dibutuhkan untuk mengirimkan paket data dari *node* sumber menuju *node* tujuan. Jika waktu *delay* yang dihasilkan dari pengujian semakin rendah, maka kinerja protokol semakin baik.
- 2) *Packet Delivery Ratio* (PDR) ditentukan dari perbandingan jumlah paket data yang diterima oleh *node* tujuan dengan jumlah paket data yang dikirim oleh *node* sumber.
- 3) *Throughput* ditentukan oleh kecepatan pengiriman data dari *node* sumber dalam ukuran tertentu yang dikirimkan menuju *node* tujuan dalam satuan waktu tertentu.

VI. HASIL DAN ANALISA

A. Aktivitas Pengiriman Pesan

Pada setiap pengiriman pesan MACT, *node* akan mengirimkan informasi posisi, kecepatan, dan waktu pengiriman pesan. Kemudian pengiriman setiap pesan dicatat dalam file teks yaitu *send_mact.txt*. Pada pesan MACT yang dikirimkan juga terdapat *flag* dimana masing – masing *flag* merupakan tanda dari isi pesan MACT baik *flag J* (*join*), *flag P* (*prune*) dan *GL* (*groupleader*). Aktivitas pengiriman pesan MACT ditunjukkan pada Gambar 3.

```

NODE: 38, Time: 63.500000, src: 38, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 76,
x: 1004.95, y: 882.48, dx: 0.00, dy: 1.00, speed: 9.32, sentTime: 63.500000
NODE: 18, Time: 67.000000, src: 18, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 42,
x: 721.45, y: 721.55, dx: -0.89, dy: -0.45, speed: 9.00, sentTime: 67.000000
NODE: 2, Time: 92.000000, src: 2, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 56,
x: 798.98, y: 848.83, dx: -0.06, dy: -1.00, speed: 9.63, sentTime: 92.000000
NODE: 18, Time: 92.011080, src: 18, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 50,
x: 605.53, y: 856.05, dx: -0.01, dy: 1.00, speed: 8.84, sentTime: 92.011080
NODE: 33, Time: 99.386986, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 300.19, y: 1199.72, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.386986
NODE: 33, Time: 99.392638, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 300.15, y: 1199.73, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.392638
NODE: 33, Time: 99.428413, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.89, y: 1199.75, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.428413
NODE: 33, Time: 99.462423, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.65, y: 1199.77, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.462423
NODE: 33, Time: 99.486330, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.48, y: 1199.79, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.486330
NODE: 33, Time: 99.491318, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.44, y: 1199.79, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.491318
NODE: 33, Time: 99.496067, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.41, y: 1199.80, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.496067
NODE: 33, Time: 99.506575, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.34, y: 1199.80, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.506575
NODE: 33, Time: 99.513544, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.29, y: 1199.81, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.513544
NODE: 33, Time: 99.521963, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.23, y: 1199.81, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.521963
NODE: 33, Time: 99.532483, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.15, y: 1199.82, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.532483
NODE: 33, Time: 99.542243, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.08, y: 1199.83, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.542243
NODE: 33, Time: 99.554053, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 299.00, y: 1199.83, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.554053
NODE: 33, Time: 99.562797, src: 33, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 40,
x: 298.93, y: 1199.84, dx: -1.00, dy: 0.09, speed: 7.18, sentTime: 99.562797
NODE: 42, Time: 100.500000, src: 42, flag: 2, hopCt: 0, grpDst: 234881024, seqno: 28,
x: 478.45, y: 404.95, dx: -1.00, dy: 0.00, speed: 9.11, sentTime: 100.500000
    
```

Gambar 3. Aktivitas pengiriman pesan MACT

Setiap pesan MACT yang diterima, akan dilakukan perhitungan bobot. Ketika bobot melebihi batas, maka node penerima akan mengirimkan pesan MACT dengan flag P kepada node pengirim. Aktivitas penerimaan pesan juga dicatat sebagai hasil dari metode yang diajukan untuk kemudian dibandingkan dengan metode sebelumnya. Pada simulasi pertama dengan satu *node* pengirim dan dua *node* penerima, analisa file *send_mact.txt* dari protokol MAODV mencatat pesan MACT sebanyak 259 pesan yang dikirimkan. Sedangkan protokol MAS-BRENK mengirimkan pesan MACT sebanyak 255 pesan atau lebih sedikit 1.54% dibandingkan protokol MAODV. Namun pada simulasi kedua dengan satu *node* pengirim dan dua *node* penerima, protokol MAODV mencatat pengiriman pesan MACT sebanyak 804 pesan. Sedangkan protokol MAS-BRENK mengirimkan pesan MACT sebanyak 824 pesan atau lebih banyak 2.49% dari protokol MAODV.

TABEL II
PERBANDINGAN JUMLAH PENGIRIMAN PESAN MACT

No.	Jml. Node Pengirim	Jml. Node Penerima	Σ MACT dikirim		Selisih Jml. Pesan	Σ MACTJ diterima		Selisih Jml. Pesan	Σ MACT_P diterima		Selisih Jml. Pesan
			A	B		A	B		A	B	
1.	1	2	259	255	-4	67	68	1	187	181	-6
		5	804	824	20	388	347	-41	368	426	58
		10	1011	1033	22	529	502	-27	409	469	60
		30	1035	1049	14	634	655	21	344	326	-18
		50	617	632	15	601	585	-16	-	33	33
2.	2	2	686	390	-296	91	63	-28	584	322	-262
		5	879	748	-131	384	321	-63	454	375	-79
		10	1127	1148	21	554	566	12	494	492	-2
		30	685	1080	395	470	679	209	174	337	163
		50	652	660	8	636	602	-34	-	36	36
3.	5	5	910	979	69	337	321	-16	524	611	87
		10	1456	1240	-216	577	554	-23	771	614	-157
		30	1105	1126	21	708	673	-35	317	389	72
		50	648	679	31	623	612	-11	2	40	38
		10	1428	1261	-167	555	506	-49	794	684	-110
4.	10	30	1139	1248	109	713	676	-37	358	488	130
		50	645	668	23	619	610	-9	-	30	30
		30	1173	1249	76	691	692	1	407	476	69
5.	30	50	650	727	77	607	623	16	1	67	66

*ket : A = MAODV
B = MAS-BRENK

TABEL III
PERBANDINGAN JUMLAH PENGIRIMAN PESAN MACT

No.	Jml. Pengirim	send_mact		recvMACTJ		recvMACT_GL		recvMACT_P	
		A	B	A	B	A	B	A	B
1.	1	745	759	444	431	15	15	262	287
2.	2	806	805	427	446	13	19	341	312
3.	5	1030	1,006	561	540	33	22	404	414
4.	10	1071	1,059	629	597	24	24	384	401
5.	30	912	988	649	658	13	17	204	272

*ket : A = MAODV
B = MAS-BRENK

TABEL IV
PERBANDINGAN PDR, THROUGHPUT DAN DELAY

No.	Jml. Pengirim	PDR (%)		Throughput (kbps)		Delay (s)	
		MAODV	MAS-BRENK	MAODV	MAS-BRENK	MAODV	MAS-BRENK
1.	1	29.736	29.936	275.9	283.2	0.369	0.340
2.	2	28.324	28.772	993.3	1,065.9	0.335	0.305
3.	5	22.700	23.053	6,095.4	6,327.3	0.219	0.256
4.	10	21.798	21.389	28,920.4	28,955.2	0.168	0.165
5.	30	17.886	17.733	247,807.0	238,416.0	0.205	0.061

Dari hasil rata-rata perbandingan jumlah paket data diatas, dengan jumlah *node* pengirim sebanyak 1 *node* protokol MAS-BRENK mengalami peningkatan jumlah pengiriman paket MACT rata - rata sebesar 1.8% dari 745 paket menjadi 759 paket dibandingkan dengan protokol MAODV dan 8.4% dengan jumlah *node* pengirim sebanyak 30 *node* dari 912 paket menjadi 988 paket. Namun pengiriman paket *send_mact* mengalami penurunan rata-rata sebesar 0.1% dengan jumlah *node* pengirim sebanyak 2 *node* dari 806 paket menjadi 805 paket dan mengalami penurunan 2.3% dengan jumlah *node* pengirim sebanyak 5 *node*. Dari keseluruhan rata-rata protokol MAS-BRENK mengalami peningkatan jumlah pengiriman paket *send_mact* sekitar 1.3% dibandingkan protokol MAODV. Pengiriman paket MACT berisi pengiriman paket dengan *flag* tertentu menuju *node* penerima. Jumlah paket MACT dengan *flag* J yang diterima pada protokol MAS-BRENK mengalami penurunan rata-rata keseluruhan yaitu sebesar 1.2% dibandingkan protokol MAODV. Dengan jumlah *node* penerima paket sebanyak 1 *node*, protokol MAS-BRENK mengalami penurunan jumlah paket rata-rata sebesar 2.8% dari 444 paket menjadi 431 paket dibandingkan dengan protokol MAODV dan 3.8% yaitu dari 561 paket menjadi 540 paket dengan jumlah *node* pengirim sebanyak 5 *node*. Serta mengalami penurunan dengan jumlah *node* pengirim sebanyak 10 *node* sebesar 5% dengan jumlah *node* pengirim sebanyak 10 *node*. Namun penerima paket MACT dengan *flag* J mengalami peningkatan rata-rata sebesar 4.5% dengan jumlah *node* pengirim sebanyak 2 *node*.

Protokol MAS-BRENK juga mengalami peningkatan penerimaan paket MACT dengan *flag* P yang cukup besar yaitu rata-rata 8.23%. Dari hasil rata-rata perbandingan tersebut, dengan jumlah *node* pengirim sebanyak 1 *node*, protokol MAS-BRENK mengalami peningkatan penerima paket rata-rata sebesar 9.7% dibandingkan dengan protokol MAODV dari 262 paket menjadi 287 paket dan 33% dengan jumlah *node* pengirim sebanyak 30 *node* yaitu dari 204 paket menjadi 272 paket. Namun mengalami penurunan rata-rata 8.4% dengan jumlah *node* pengirim sebanyak 2 *node* pengirim dari 341 paket menjadi 312 paket. Beberapa perbandingan hasil skenario diatas juga diambil rata-rata pembanding yaitu PDR, *throughput* dan *delay* terhadap jumlah pengirim. Didapatkan protokol MAS-BRENK mengalami peningkatan PDR dan *throughput* serta mengalami penurunan *end to end delay*. Perbandingan PDR, *Throughput*, dan *Delay* ditunjukkan pada Tabel 4.

Dari hasil rata-rata perbandingan diatas, dengan jumlah *node* pengirim sebanyak satu *node* protokol MAS-BRENK mengalami peningkatan PDR rata-rata sebesar 0.7% dari 29.736% menjadi 29.936% dibandingkan dengan protokol MAODV dan 1.5% dengan jumlah *node* pengirim sebanyak 2 dan 5 *node*. Namun PDR mengalami penurunan rata-rata sebesar 1.9% dengan jumlah *node* pengirim sebanyak 10 *node* dari 21.798% menjadi 21.389% dan mengalami penurunan 9% dengan jumlah *node* pengirim sebanyak 30 *node*. Dari keseluruhan rata-rata protokol MAS-BRENK mengalami peningkatan PDR sekitar 0.2% dibandingkan protokol MAODV.

Dari sisi *throughput* protokol MAS-BRENK juga mengalami peningkatan rata-rata keseluruhan yaitu sebesar 2% dibandingkan protokol MAODV. Dengan jumlah *node* pengirim sebanyak 1 *node* protokol MAS-BRENK mengalami peningkatan *throughput* rata-rata sebesar 2.7% dari 275.9kbps menjadi 283.2kbps dibandingkan dengan protokol MAODV dan 7.3% yaitu dari 993.3kbps menjadi 1065.9kbps dengan jumlah *node* pengirim sebanyak 2 *node*. Serta mengalami peningkatan dengan jumlah *node* pengirim sebanyak 5 *node* sebesar 3.8% dan 0.12% dengan jumlah *node* pengirim sebanyak 10 *node*. Namun *throughput* mengalami penurunan rata-rata sebesar 3.8% dengan jumlah *node* pengirim sebanyak 30 *node*.

Protokol MAS-BRENK juga mengalami penurunan *delay* yang cukup besar yaitu rata-rata 14.3%. Dari hasil rata-rata perbandingan tersebut, dengan jumlah *node* pengirim sebanyak satu *node* protokol MAS-BRENK mengalami penurunan *delay* rata-rata sebesar 7.9% dibandingkan dengan protokol MAODV dari 0.369s menjadi 0.364s, mengalami penurunan 8.9% dengan jumlah *node* pengirim sebanyak 2 *node* pengirim dari 0.335s ke 0.305s dan 1.7% dengan jumlah *node* pengirim sebanyak 10 *node*. Namun *delay* mengalami peningkatan rata-rata sebesar 17.2% dengan jumlah *node* pengirim sebanyak 5 *node* dari 0.219s menjadi 0.256s. Penurunan *delay* cukup besar yaitu 70.3% terjadi ketika jumlah *node* pengirim sebanyak 30 *node* yaitu dari 0.205s menjadi 0.061s.

VII. KESIMPULAN

Protokol MAS-BRENK mempertimbangkan beberapa parameter berdasarkan bobot posisi, kecepatan, dan delay transmisi. Tidak semua *node* dapat melakukan *join* menjadi anggota *multicast*. Dan tidak semua *node* menunggu rute *time-out* untuk melakukan *prune* dari anggota *multicast*. Jumlah *node* pada protokol MAS-BRENK yang melakukan *join* lebih sedikit daripada MAODV. Sehingga didapatkan sedikit peningkatan pada PDR dan throughput. Disisi lain jumlah *node* yang melakukan *pruning* pada protokol MAS-BRENK juga lebih banyak daripada protokol MAODV. Sehingga *end-to-end delay* yang dihasilkan lebih kecil. Meskipun secara detail dari skenario jumlah pengirim, dari beberapa skenario terjadi penurunan PDR dan *throughput* serta peningkatan *delay*. Rata-rata PDR dan *throughput* protokol MAS-BRENK mengalami peningkatan dan rata-rata *delay* juga mengalami penurunan dibandingkan protokol MAODV.

DAFTAR PUSTAKA

- [1] Anupama, K.S.S., Gowri, S.S., PrabhakaraRao, B., Rajesh. P., "Application of MADM Algorithms of Network Selection", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), 2015, Vol.3 Issue 6.
- [2] Balakrishna, R., Rao, U. R., & Geethanjali, N., "Performance issues on AODV and AOMDV for MANETS". International Journal of Computer Science and Information Technologies (IJCSIT), 2010, (hal. 38-43).
- [3] Dimiyati, M., Anggoro, R. & Wibisono, W., "Pemilihan *Node Rebroadcast* untuk Meningkatkan Kinerja Protokol *Multicast* AODV (MAODV) pada VANETs". 2016, Volume 14, Nomor 2. JUTI.
- [4] Dorle, S., Vidhale, B. & Chakole, M., "Evaluation of Multipath, Unipath and Hybrid Routing Protocols for Vehicular Ad Hoc Networks". In Fourth International Conference on Emerging Trends in Engineering & Technology., 2011. IEEE Press.
- [5] Ghazemi, M., & Bag-Mohammady, M., "Classification of Multicast Routing Protocols for Mobile Ad Hoc Network", International Journal of IEEE, ICTC12, 2012, pp. 789-794.
- [6] Harri, J., Filali, F. & Bonnet, C., "Mobility Models for Vehicular Ad Hoc Network: A Survey and Taxonomy". Research Report. Sophia Antipolis: Eurecom, 2006
- [7] Jemaa, I. B., Shagdar, O., Martinez, F. J., Garrido, P., Nashashibi, F., "Extended Mobility Management and Routing Protocols for Internet-to-VANET Multicasting", IEEE 12th Consumer Communication and Networking Conferences, 2015, pp. 904-909.
- [8] Joshi, A., Kaur, R., "A Novel Multicast Routing Protocol for VANET", International Journal of IEEE, IACC15, 2015, pp. 41-45.
- [9] Menouar, H., Filali, F. and Lenardi, M., "A survey and qualitative analysis of MAC protocols for vehicular ad hoc networks". Wireless Communications, IEEE, 13 (5), 2006, pp. 30--35.
- [10] Nakamura, M., Kitani, T., Sun, W., Shibata, N., Yasumoto, K. and Ito, M., "A method for improving data delivery efficiency in delay tolerant vanet with scheduled routes of cars". 2010, pp. 1--5.
- [11] Perkins, C.E., Royer, E.M., "Ad hoc On-Demand distance Vector routing", Proceedings of 2nd IEEE workshop on mobile computing systems and applications, 1999, pp 90-100.
- [12] Royer, E.M., Perkins, C.E., 2000, "Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing", IETF, Internet Draft: draft-ietf-manet-maodv-00.txt.
- [13] Savitha, K., Chandrasekar, C., "Vertical Handover Decision Schemes using SAW and WPM for Network Selection in Heterogeneous Wireless Networks", Global Journal of Computer Science and Technology, 2011, Vol.11 Issue 9.
- [14] Su, W., Lee, S.-J., & Gerla, M., (2000). "Mobility Prediction and Routing in AdHoc Wireless Networks". International Journal of Network Management.
- [15] Vidhale, B. & Dorle, S., 2011. "Performance Analysis of Routing Protocols in Realistic Environment for Vehicular Ad Hoc Network". In 21st International Conference on System Engineering. IEEE Press.
- [16] Shurdi, O., Miho, R., Kamo, B., Kholici, V., Rakipi. A., 2011, "Performance Analysis of Multicast Routing Protocols MAODV, ODMRP, and ADMR for MANETS", International Conference of IEEE NBIS'11.
- [17] Xia, H., Yu, J., Xia, S., Jia, Z., Sha, E. H. M., "Applying Link Stability Estimation Mechanism to Multicast Routing in MANETS". Journal of System Architecture 60, 2014, pp 467-480.