

Studi Kinerja 802.11P pada Protokol *Ad Hoc On-Demand Distance Vector (AODV)* di Lingkungan *Vehicular Ad Hoc Network (VANET)* Menggunakan *Network Simulator 2 (NS-2)*

Ilmal Alifriansyah Rahardjo, Radityo Anggoro, dan F.X. Arunanto
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: onngo@if.its.ac.id

Abstrak—Dalam dunia jaringan nirkabel sudah tidak asing dengan salah satu konsep yang memungkinkan komunikasi antar kendaraan (*inter-vehicle*) dan komunikasi antara kendaraan dengan infrastruktur di sekitar jalan (*vehicle-to-roadside*). Konsep ini juga merupakan subset dari dari *Mobile Ad-Hoc Network (MANET)*, yaitu *Vehicular Ad-hoc Network (VANET)*. Pada VANET kendaraan yang ada bertindak sebagai *node* pada suatu jaringan. VANET terdiri dari banyak *node* yang juga berfungsi sebagai *router*. Berbeda dengan MANET, tingkat mobilitas pada VANET lebih tinggi. Tujuan utama dari adanya VANET ini adalah nantinya dapat digunakan untuk menciptakan dan menyediakan aplikasi-aplikasi transportasi untuk mendukung keamanan maupun kenyamanan bagi pengendara. Hal ini tentunya memerlukan implementasi protokol *routing* yang sesuai dengan karakteristik dari VANET dimana kendaraan-kendaraan yang notabene memiliki dinamika pergerakan *node* yang tinggi dalam suatu jaringan. Salah satu protokol *routing* reaktif yang telah teruji dalam VANET adalah *routing* protokol *Ad hoc On-Demand Distance Vector (AODV)*. Dan pada Studi ini dilakukan analisis performa dari *routing* protokol *Ad hoc On-Demand Distance Vector (AODV)* tersebut dengan parameter 802.11p dalam lingkungan VANET. Dari percobaan Studi ini dihasilkan suatu performa bahwa *routing* protokol AODV dengan parameter 802.11p pada skenario riil mengalami peningkatan nilai rata-rata pada *packet delivery ratio* dan *routing overhead* dan mengalami penurunan nilai rata-rata pada *delay* seiring dengan bertambahnya kepadatan kendaraan dibandingkan dalam skenario *grid*.

Kata Kunci—AODV, VANET, *Network Simulator*, NS-2, 802.11p.

I. PENDAHULUAN

DEWASA ini perkembangan dunia internet semakin berkembang dengan pesat, mulai digunakan untuk *browsing*, *chatting* dan lain sebagainya. Kegiatan tersebut merupakan termasuk suatu aplikasi yang berhubungan dengan proses pengiriman paket data dalam jaringan internet. Berdasarkan hasil survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) dengan Pusat

Kajian Komunikasi Universitas Indonesia (PUSKAKOM UI). Hasil survei menunjukkan bahwa jumlah pengguna internet di Indonesia untuk saat ini telah mengalami peningkatan sebesar 34,9% dari jumlah seluruh penduduk Indonesia. Jika pada tahun sebelumnya ada sekitar 71,9 juta pengguna internet di Indonesia, maka pada semester pertama 2015 ini jumlah tersebut mencapai sekitar 88,1 juta pengguna. Dengan semakin banyak pengguna internet ini semakin bertambah pula kebutuhan yang diperlukan untuk berkomunikasi antar pengguna dan pasti akan semakin kompleks. Dengan demikian diperlukan sebuah model baru untuk menyediakan layanan jaringan tanpa infrastruktur. Salah satu model baru tersebut adalah *Vehicular Ad-hoc Network (VANET)* [1]. VANET merupakan konsep subset dari *Mobile Ad-Hoc Network (MANET)* dimana kendaraan bertindak sebagai *node* pada jaringan. Mobilitas *node* yang tinggi merupakan karakteristik dasar VANET yang menyebabkan perubahan yang cepat pada topologi jaringan. Hal ini tentunya memerlukan implementasi protokol *routing* yang sesuai dengan karakteristiknya di dalam jaringan [2]. Implementasi pada lingkungan VANET dapat dilakukan dengan menggunakan simulasi sehingga penelitian ini dapat dilakukan untuk mempelajari sistem dengan baik. Simulasi dilakukan dengan menggunakan *Network Simulator 2 (NS-2)*. Implementasi ini akan dilakukan analisis performa 802.11p (WAVE) pada protokol *routing reactive* yaitu *Ad hoc On-Demand Distance Vector (AODV)*.

Dalam Studi ini penulis mengimplementasikan *routing* protokol AODV dengan parameter 802.11p dalam lingkungan VANET pada *network simulator NS-2*. Penulis melakukan studi kinerja performa pada *routing* protokol AODV dengan melakukan simulasi pada skenario yang memiliki jumlah *node* yang bervariasi. Kemudian dari hasil simulasi tersebut menghasilkan suatu nilai performa yang ditampilkan pada skenario *grid* dan skenario riil.

II. TINJAUAN PUSTAKA

A. Ad hoc On-Demand Distance Vector (AODV)

Ad hoc On-Demand Distance Vector (AODV) adalah protokol *routing* yang termasuk dalam klasifikasi reaktif protokol *routing*, yang hanya *me-request* sebuah rute saat dibutuhkan. AODV yang standar ini dikembangkan oleh C. E. Perkins, E.M. Belding-Royer dan S. Das pada RFC 3561. Ciri utama dari AODV adalah menjaga *timer-based state* pada setiap *node* sesuai dengan penggunaan tabel *routing*. Tabel *routing* akan kadaluarsa jika jarang digunakan. AODV memiliki *route discovery* dan *route maintenance*. *Route Discovery* berupa *Route Request* (RREQ) dan *Route Reply* (RREP). Sedangkan *Route Maintenance* berupa *Data*, *Route Update* dan *Route Error* (RRER).

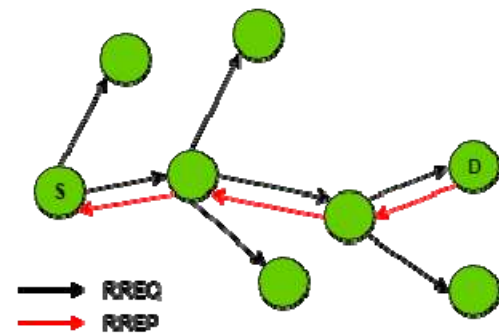
AODV memerlukan setiap *node* untuk menjaga tabel *routing* yang berisi *field* :

- *Destination IP Address* : berisi alamat IP dari *node* tujuan yang digunakan untuk menentukan rute.
- *Destination Sequence Number* : *destination sequence number* bekerjasama untuk menentukan rute.
- *Next Hop* : ‘Loncatan’ (*hop*) berikutnya, bisa berupa tujuan atau *node* tengah, *field* ini dirancang untuk meneruskan paket ke *node* tujuan.
- *Hop Count* : Jumlah *hop* dari alamat IP sumber sampai ke alamat IP tujuan.
- *Lifetime* : Waktu dalam milidetik yang digunakan untuk *node* menerima RREP.
- *Routing Flags* : Status sebuah rute; jika *up* (valid), jika *down* (tidak valid) atau sedang diperbaiki.

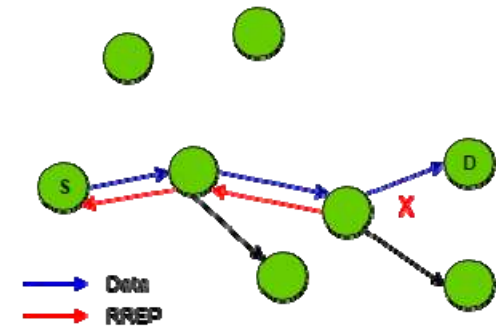
Penemuan jalur (*Path Discovery*) atau *Route Discovery* diinisiasi dengan menyebarkan *Route Reply* (RREP). Ketika RREP menjelajahi *node*, maka selanjutnya akan secara otomatis membuat suatu *path* baru. Jika sebuah *node* menerima RREP, maka *node* tersebut akan mengirimkan RREP lagi ke *node* atau *destination sequence number* seperti terlihat pada Gambar 1 dibawah ini.

Pada proses ini, *node* pertama kali akan mengecek *destination sequence number* pada tabel *routing* pada *node* yang menerima paket tersebut, apakah nilainya lebih besar daripada *Route Request* (RREQ), jika lebih besar, maka *node* akan mengirim RREP. Ketika RREP berjalan kembali ke *source* melalui jalur yang telah diatur, maka paket tersebut akan mengatur kembali jalur rute ke *source node* dan memperbarui waktu *timeout* [3].

Dan juga terlihat pada Gambar 2 bahwa jika sebuah *link* ke *hop* berikutnya tidak dapat dideteksi dengan metode penemuan rute, maka *link* tersebut akan diasumsikan putus dan *Route Error* (RRER) akan disebarkan ke *node* tetangganya. Dengan demikian sebuah *node* bisa menghentikan pengiriman data melalui rute ini atau meminta rute baru dengan menyebarkan RREQ kembali.



Gambar 1. Mekanisme Penemuan Rute (*Route Discover*).



Gambar 2. Mekanisme Pemrosesan Data (*Route Update*) dan *Route Error*

B. Network Simulator 2 (NS-2)

Network Simulator 2 (NS-2) adalah suatu interpreter yang *object-oriented*, dan *discrete event-driven* yang dikembangkan oleh University of California Berkeley dan USC ISI sebagai bagian dari proyek *Virtual Internet Testbed* (VINT) [4]. *Network Simulator* ini menjadi salah satu alat simulasi yang sangat berguna untuk menunjukkan simulasi jaringan melibatkan *Local Area Network* (LAN), *Wide Area Network* (WAN), dan dengan berkembangnya zaman berkembang pula fungsi dari alat simulasi ini, yaitu selama beberapa tahun belakangan ini alat simulasi ini melibatkan pada jaringan nirkabel (*wireless*) serta pada jaringan *ad hoc*. Adapun beberapa keuntungan menggunakan *Network Simulator* sebagai alat simulasi pembantu analisis dalam riset, antara lain adalah *Network Simulator* ini dilengkapi dengan *tools* validasi yang digunakan untuk menguji kebenaran pemodelan suatu sistem. Secara *default*, semua pemodelan akan dapat melewati proses validasi ini, seperti media, protokol dan komponen jaringan yang lengkap dengan perilaku trafiknya sudah disediakan pada *library Network Simulator*. Versi terbaru dari NS-2 adalah ns-2.35 yang dirilis pada tahun 2011.

C. 802.11p - Wireless Access Vehicular Environments (WAVE)

Untuk memanfaatkan potensi yang ada pada sistem komunikasi antar kendaraan, saat ini IEEE sedang mengembangkan suatu perubahan standar IEEE 802.11p atau yang biasa disebut dengan WAVE (*Wireless Access Vehicular Environments* [13]. WAVE merupakan penyempurnaan standar IEEE 802.11 yang diperlukan untuk mendukung

pengaplikasian ITS (*Intelligent Transportation Systems*). WAVE juga merupakan pengembangan sistem IEEE 802.11a dengan memperkenalkan *physical layer* dan *MAC layer* yang dapat meningkatkan sistem operasi dan aplikasi keselamatan dengan memberikan tingkat latency rendah. WAVE sendiri beroperasi pada band 5.9 GHz dengan menggunakan sistem multiplexing OFDM (*Orthogonal Frequency Division Multiplexing*) dan dapat mencapai kecepatan penransmisi data antara 6 – 27 Mbps. WAVE terdiri dari tujuh channel pada frekuensi 10 MHz yang dari satu *control channel* dan enam *service channel* pada *bandwidth* 5.9 GHz. *Service channel* digunakan untuk *public safety* dan *private service*, sedangkan *control channel* digunakan sebagai referensi *channel* untuk membangun *link* komunikasi antara RSU (*Road – Side Unit*) dan OBU (*On – Board Unit*). *Control channel* digunakan oleh OBU dan RSU untuk *broadcast application service*, *warning message* dan *safety status message*. Dari penjelasan di atas dapat diketahui bahwa aplikasi utama dari IEEE 802.11p adalah untuk sistem komunikasi antar kendaraan dengan sistem komunikasi yang digunakan adalah DSRC (*Dedicated Short Range Communication*). Pada Studi ini penulis menguji kinerja dan performa dari WAVE ini pada *routing protocol* AODV menggunakan NS-2.

D. Vehicular Ad hoc Network (VANET)

Vehicular Ad hoc Network (VANET) adalah sebuah jaringan terorganisir yang dibentuk dengan menghubungkan kendaraan dan RSU (*Roadside Unit*) disebut *Vehicular Ad Hoc Network (VANET)* dan RSU lebih lanjut terhubung ke jaringan *backbone* berkecepatan tinggi melalui koneksi jaringan. Kepentingan peningkatan baru-baru ini telah diajukan pada aplikasi melalui V2V (*Vehicle to Vehicle*) dan V2I (*Vehicle to Infrastructure*) komunikasi, dimana bertujuan untuk meningkatkan keselamatan mengemudi dan manajemen lalu lintas sementara menyediakan pengemudi dan penumpang dengan akses internet.

Dalam Studi ini penulis akan mengimplementasikan 802.11p pada *routing protocol* AODV dan menguji kinerja dalam lingkungan VANET.

III. DESAIN DAN PERANCANGAN

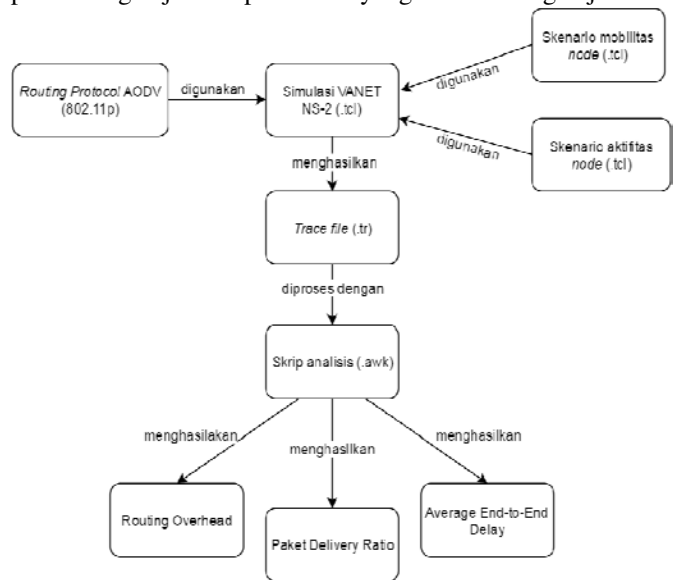
A. Deskripsi Umum

Pada Studi ini akan dilakukan implementasi dan analisis dari 802.11p pada *routing protocol* AODV pada NS-2. Diagram rancangan simulasi dapat dilihat pada Gambar 3. Dimana terdapat 2 (dua) jenis skenario yang digunakan sebagai perbandingan pengukuran lalu lintas kota Surabaya, yaitu skenario *grid* dan skenario riil. Pada skenario *grid* peta dibuat dengan bantuan aplikasi SUMO. Sedangkan dalam skenario riil peta yang diambil langsung keadaan saat pengambilan menggunakan OpenStreetMap dengan bantuan editor JOSM. Setelah *file* peta sudah berbentuk, dilakukan simulasi lalu lintas dengan SUMO. Hasil simulasi SUMO digunakan untuk mencari kinerja dari 802.11p pada simulasi

protokol AODV pada NS-2. Kemudian hasil simulasi dari NS-2 dianalisis dengan menggunakan skrip AWK untuk menghitung metrik analisis berupa *routing overhead*, *packet delivery* dan *average end-to-end delay*. Perhitungan metrik analisis bertujuan untuk mengukur performa dari protokol AODV untuk melihat performa dari 802.11p.

B. Perancangan Metrik Analisis

Berikut ini adalah perancangan beberapa metrik yang akan dijadikan analisis dalam Studi ini. Dimulai dari *Packet Delivery Ratio (PDR)*, *Average End-to-End Delay* dan *Routing Overhead (RO)*. Nilai dari PDR didapat dari perbandingan jumlah paket data yang dikirim dengan jumlah



Gambar 3. Diagram Alur Rancangan Simulasi.

paket data yang diterima. *Packet delivery ratio* dihitung menggunakan Persamaan 3.3, di mana *received* adalah jumlah paket data yang diterima dan *sent* adalah jumlah paket data yang dikirim.

$$PDR = \frac{Data_{received}}{Data_{sent}} \tag{3.1}$$

Average end-to-end delay merupakan waktu rata-rata dari setiap paket ketika sampai di tujuan. Semua paket, termasuk *delay* yang dikarenakan oleh paket *routing*, ini juga akan diperhitungkan dalam memperoleh nilai akhir. Paket yang akan dimasukkan ke dalam perhitungan hanya paket yang berhasil sampai tujuan. *Average end-to-end delay* dihitung menggunakan Persamaan 3.2, di mana *i* adalah nomor paket yang berhasil sampai di tujuan $t_{received} [i]$ adalah waktu ketika paket *i* dikirim, sedangkan $t_{sent} [i]$ adalah waktu ketika paket *i* diterima dan *pktCounter* adalah jumlah paket yang berhasil sampai di tujuan.

$$Delay = \frac{\sum_{i=0}^n t_{received} [i] - t_{sent} [i]}{pktCounter} \quad (3.2)$$

Routing overhead merupakan jumlah paket routing control yang ditransmisikan selama simulasi terjadi. Paket kontrol yang dihitung adalah jumlah Route Request (RREQ), Route Reply (RREP) dan Route Error (RRER). Rumus dari routing overhead dapat dilihat pada Persamaan 3.1.

$$RO = RREQ_{sent} + RREP_{sent} + RRER_{sent} \quad (3.3)$$

IV. IMPLEMENTASI

A. Implementasi Skenario Grid dan Skenario Riil

Pada skenario grid ini dijelaskan bagaimana skenario yang digunakan untuk membuat suatu rute yang berbentuk seperti jala-jala atau grid dengan bantuan tool netgenerate yang telah disediakan oleh SUMO. Skenario grid ini dibuat dengan panjang jalan 200 meter dan luas peta pada skenario grid ini adalah 1200 meter x 1200 m. Jumlah ujung persimpangan antara jalan vertikal dan horizontal sebanyak 6 ujung x 6 ujung. Kecepatan kendaraan yang diperbolehkan untuk melintasi jalan pada skenario ini diatur dengan kecepatan maksimal 15 m/s. Pada skenario riil parameter sama dengan skenario grid, hanya sedikit berbeda dalam pemakaian peta. Peta yang digunakan skenario riil adalah peta asli dan berkecstensi file (.osm), dan berbeda luas, pada peta skenario riil luas peta adalah 1200 meter x 800 meter.

B. Implementasi Network Skenario 2

Pada implementasi ini dilakukan penggabungan antara hasil skenario grid dan skenario riil berupa file (mobility.tcl) dengan skrip TCL dari kode NS-2 yang diberikan parameter-parameter untuk selanjutnya dilakukan percobaan simulasi VANET pada NS-2. Berikut parameter simulasi perancangan sistem VANET yang dapat dilihat pada Tabel 1.

Tabel 1. Keterangan Parameter dan Spesifikasi pada Simulasi NS-2.

No	Parameter	Spesifikasi
1	Network simulator	NS-2.35
	Routing protocol	AODV
3	Area simulasi	1200 m x 1200 m (grid) 1200 m x 800 m (riil)
4	Waktu	360 detik
5	Radius transmisi	50 m
6	Agen	Constant Bit Rate
7	Source/Destination	Stationary
8	Jumlah kendaraan	50, 100, 150, 200

9	Kecepatan statis	15 m/s
10	Packet Rate	2 kB/s
11	Packet Interval	1 paket/detik
12	Ukuran Paket	512 Bytes
13	Protokol MAC	IEEE 802.11p
14	Model Propagasi	Two-ray Ground

V. UJI COBA DAN ANALISIS HASIL

Uji coba ini dilakukan dengan menggunakan sebuah komputer dan dalam lingkungan yang dibatasi. Berikut ini adalah spesifikasi perangkat dan lingkungan yang digunakan dalam simulasi ini.

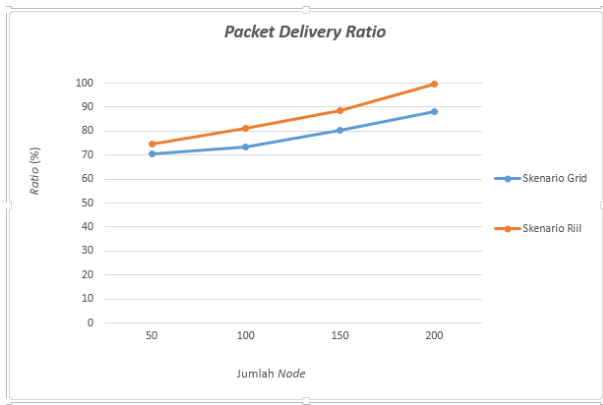
- CPU Intel (R) Core TM i3-2130 CPU @3.40GH x 4.
- Sistem Operasi Linux Ubuntu 12.04 LTS 32-bit.
- Memori 3.8 GB.
- Harddisk 500 GB.
- SUMO versi 0.26.0
- JOSM versi 11223
- Network Simulator ns2.35

Trace file hasil menjalankan program skenario grid dan skenario riil ini kemudian dianalisis nilai PDR, Delay dan RO-nya dengan script masing-masing awk_PDR, awk_DLY, awk_RO ini merupakan proses awk. Adapun nilai dari PDR setelah simulasi dan dengan perhitungannya, didapatkan nilai seperti pada Tabel 2 dan Gambar 4.

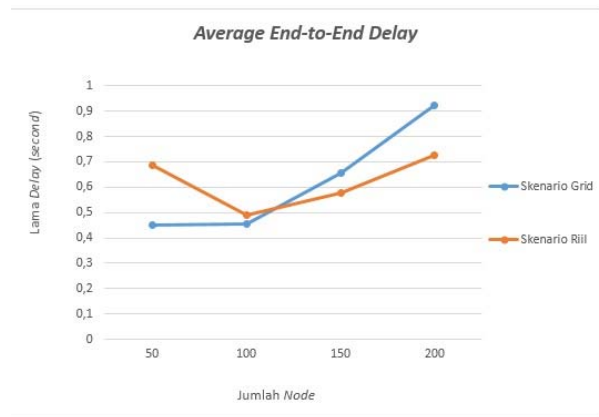
Berdasarkan data tersebut, dapat dilihat bahwa pada skenario riil nilai rata-rata PDR lebih tinggi dibandingkan dengan skenario grid. Pada jumlah node 50, nilai PDR pada skenario riil memiliki 4% lebih baik dibandingkan pada skenario grid. Pada jumlah node 100 dan 150, pada skenario riil nilai PDR naik ke angka 7,7% hingga 8,5% lebih baik dibandingkan dengan skenario grid. Dan pada jumlah node tertinggi, yaitu 200, pada skenario riil nilai PDR naik lagi ke angka 11,4% lebih baik dibandingkan dengan skenario grid. Banyaknya jumlah node yang pada skenario grid dan skenario riil sama-sama berpengaruh terhadap nilai PDR. Terbukti dari jumlah kepadatan rendah, 50 node, hingga kepadatan paling tinggi, 200 node, nilai PDR dari kedua skenario sama-sama meningkat.

Tabel 2. Hasil Perhitungan Rata-rata PDR pada Skenario Grid dan Skenario Riil.

Jumlah Node	Skenario Grid (%)	Skenario Riil (%)	Perbedaan (%)
50	70,7	74,7	4
100	73,5	81,2	7,7
150	80,2	88,7	8,5
200	88,3	99,7	11,4



Gambar 4. Grafik PDR pada Skenario *Grid* dan Skenario Riil.



Gambar 5. Grafik PDR pada Skenario *Grid* dan Skenario Riil.

Namun pada skenario riil nilai PDR sedikit lebih meningkat dibandingkan dengan skenario *grid*. Hal ini disebabkan perbedaan jumlah paket RREQ dari AODV yang tersebar dalam jaringan. Semakin banyak paket RREQ yang tersebar, maka semakin banyak pula kemungkinan rute yang akan terjadi. Karena pada skenario riil rute yang ada lebih bervariasi dalam pencarian rute karena berbentuk asimetris dibandingkan dengan skenario *grid* yang lebih simetris dan pencarian rute yang lebih statis.

Selanjutnya adalah nilai dari Delay setelah dilakukan simulasi dapat dilihat pada Tabel 3 dan Gambar 5.

Berdasarkan data tersebut, dapat dilihat bahwa pada skenario riil nilai rata-rata Delay pada kepadatan awal dimana jumlah *node* masih 50, nilai Delay jauh lebih tinggi hingga memperoleh selisih 0,237 detik dibandingkan dengan skenario *grid*. Namun pada kondisi dengan jumlah *node* 100, nilai Delay turun drastis hingga selisih dengan skenario *grid* berkurang menjadi hanya 0,036 detik saja. Namun seiring bertambahnya kepadatan kendaraan dalam skenario riil, yaitu pada jumlah *node* 150 hingga 200, nilai Delay ikut naik menjadi 0,076 hingga 0,197 detik, namun kali ini lebih rendah dibandingkan dengan skenario *grid*. Pada kepadatan rendah, AODV yang ada pada skenario riil memerlukan waktu yang lebih lama untuk melakukan proses *route discovery*. Ini karena rute dari skenario riil yang lebih bervariasi dari skenario *grid* yang selalu sama di setiap 200 m, berbeda dengan skenario riil yang setiap jalan mempunyai panjang yang berbeda-beda.

Adapun tren *delay* yang semakin meningkat pada skenario *grid* ketika jumlah *node* semakin banyak ini dikarenakan selain AODV terlalu lama dalam melakukan proses *route discovery*,

Tabel 3.

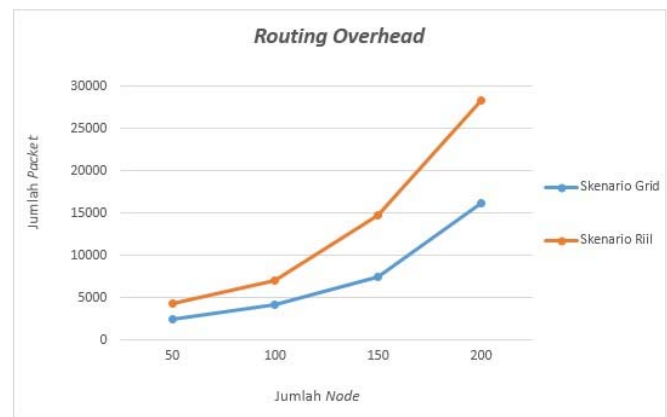
Hasil Perhitungan Rata-rata Delay pada Skenario *Grid* dan Skenario Riil.

Jumlah Node	Skenario <i>Grid</i> (detik)	Skenario Riil (detik)	Perbedaan (detik)
50	0,450	0,687	0,237
100	0,455	0,491	0,036
150	0,654	0,578	0,076
200	0,921	0,724	0,197

Tabel 4.

Hasil Perhitungan Rata-rata RO pada Skenario *Grid* dan Skenario Riil.

Jumlah Node	Skenario <i>Grid</i> (paket)	Skenario Riil (paket)	Perbedaan (paket)
50	2493	4314	1821
100	4100	7062	2962
150	7416	14688	7272
200	16179	28368	12189



Gambar 6. Grafik Routing Overhead pada Skenario *Grid* dan Skenario Riil

juga karena terjadi kemacetan jalan di setiap perempatan yang tedapat lalu lintas. Karena AODV melakukan *broadcast*, *buffer* paket dari setiap *node* di sekitar area perempatan menjadi sangat panjang. Berbeda dengan skenario riil yang lebih sedikit memiliki perempatan dan karena jarak antar lampu lalu lintas yang satu dengan yang lain yang bervariasi dan termasuk lebih jauh dibandingkan dengan pada skenario *grid*.

Dan yang terkahir adalah nilai dari RO setelah dilakukan simulasi dapat dilihat pada Tabel 4 dan Gambar 6.

Berdasarkan data tersebut, dapat dilihat bahwa skenario riil memiliki nilai rata-rata RO yang lebih tinggi dibandingkan dengan Skenario *Grid*. Terlihat pada kepadatan awal, dengan jumlah *node* 50, skenario riil memiliki nilai RO lebih banyak 1821 paket dibandingkan dengan Skenario *Grid*. Kemudian pada jumlah *node* 100 hingga *node* 200, masing-masing dari skenario sama-sama meningkat nilai dari RO. Karena memang pada skenario *grid* dan skenario riil ini berbeda dalam rute perjalanan pada saat AODV melakukan *broadcast*, sehingga

secara otomatis banyak jumlah paket *routing* yang tersebar dalam jaringan.

Hasil analisis yang dapat ditarik melihat nilai masing-masing metrik diatas menyebabkan beberapa perbedaan. Adapun faktor yang menyebabkan terjadinya perbedaan ini adalah lebih kepada bentuk jalan, luas daerah yang menyebabkan rute yang dilewati *node* cukup berbeda.

VI. KESIMPULAN DAN SARAN

Dari hasil uji coba yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil uji coba perbedaan nilai dari skenario *grid* dengan skenario riil didapatkan hasil bahwa pada skenario riil protokol AODV memiliki nilai lebih baik daripada dalam skenario *grid*. Seperti dilihat pada nilai rasio perbedaan berikut:

- Nilai PDR dari 4% pada kepadatan minimum kemudian naik ke 11,4% pada kepadatan maksimum.
- Nilai Delay dari 65,5% pada kepadatan minimum kemudian naik hingga 127% pada kepadatan maksimum.
- Nilai RO dari 11,4% pada kepadatan minimum dan naik ke angka 57% pada kepadatan maksimum.

Nilai ini semua tergantung seiring bertambahnya jumlah kepadatan kendaraan pada skenario.

2. Faktor-faktor yang dapat mempengaruhi nilai PDR, *Delay* dan RO yang dihasilkan dari simulasi dengan skenario *grid* dan skenario riil adalah:

- Bentuk rute dari masing-masing skenario yang berbeda
- Pergerakan *node* yang dibuat secara acak.
- Parameter pada masing-masing skenario.

Saran yang dapat diberikan dari hasil uji coba dan evaluasi dari Studi ini untuk pengembangan simulasi kedepan, antara lain:

1. Studi lebih lanjut mengenai pengaruh tipe *routing* protokol lain (proaktif atau *hybird*) dengan kepadatan kendaraan yang lebih variatif pada skenario *grid* dan skenario riil.
2. Dilakukan penambahan model transmisi selain *TwoRay-Ground*.
3. Dilakukan pengurangan dan penambahan pada kecepatan *node* maksimal.
4. Diperlukan penyempurnaan sistem lampu lalu lintas agar lebih realistis. Lampu lalu lintas yang diimplementasikan pada Studi ini ditangani sepenuhnya oleh *simulator*, bukan berdasarkan desain penulis.

DAFTAR PUSTAKA

- [1] A. M. C dan T. C, "A CLUSTER BASED ENHANCEMENT TO AODV FOR INTER-VEHICULAR COMMUNICATION IN VANET," *Int. J. Grid Comput. Appl. IJGCA*, vol. 3, hal. 41–50, Sep 2012.
- [2] "A Noble Routing Protocol for Vehicular ad hoc Networks (VANETs) with Less Routing Overheads," *Int. J. Future Gener. Commun. Netw.*, vol. 7, hal. 23–34, 2014.
- [3] B. Ramakrishnan, "Performance Analysis of AODV Routing Protocol In Vehicular Ad-hoc Network Service Discovery Architecture," *ARPN J. Syst. Softw.*, vol. 2, hal. 65–72, Feb 2012.
- [4] G. Gautam dan B. Sen, "Design and Simulation of Wireless Sensor Network in NS2," *Int. J. Comput. Appl. 0975 – 8887*, vol. 113, hal. 14–16, Mar 2015.
- [5] "DLR - Institute of Transportation Systems - SUMO – Simulation of Urban Mobility." [Daring]. Tersedia pada: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/. [Diakses: 18-Jan-2017].
- [6] "NETCONVERT - Sumo." [Daring]. Tersedia pada: <http://sumo.dlr.de/wiki/NETCONVERT>. [Diakses: 01-Jan-2017].
- [7] P. TS, "SUMO, Open Street Maps and NS2 - A Real Traffic Simulation."
- [8] K. Thenmozhi dan D. E. Ramaraj, "Routing Overhead in MANET - Minimization with Multipath Local Route Discovery Routing Protocol," *Int. J. Eng. Trends Technol. IJETT*, vol. 13, hal. 144–147, Jul 2014.
- [9] P. Rohal, R. Dahiya, dan P. Dahiya, "Study and Analysis of Throughput, Delay and Packet Delivery Ratio in MANET for Topology Based Routing Protocols (AODV, DSR and DSDV)," *Int. J. Adv. Res. Eng. Technol.*, vol. 1, no. II, hal. 54–58, Mar 2013.
- [10] "NETGENERATE - Sumo." [Daring]. Tersedia pada: <http://sumo.dlr.de/wiki/NETGENERATE>. [Diakses: 01-Jan-2017].
- [11] "OpenStreetMap," *OpenStreetMap*. [Daring]. Tersedia pada: <https://www.openstreetmap.org/>. [Diakses: 30-Des-2016].
- [12] Jagadevi dan D. S. Terdal, "Performance Evaluation of IEEE 802.11p MAC Protocol for VANETs," *Int. J. Sci. Res. IJSR*, vol. 4, no. 9, hal. 711–715, Sep 2015.