

Modifikasi Pemilihan *Forwarding Node* pada *Dynamic Source Routing* (DSR) Berdasarkan Tingkat Kestabilan *Neighboring Node* di VANETs

Gleean Allan M, Radityo Anggoro, dan Muchammad Husni

Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: onngo@if.its.ac.id

Abstrak—*Vehicular Ad hoc Networks* (VANETs) merupakan pengembangan dari *Mobile Ad hoc Networks* (MANETs), dimana *node* memiliki karakteristik dengan mobilitas yang tinggi dan terbatas pada pola pergerakannya. Ada banyak *routing protocol* yang dapat diimplementasikan pada VANETs salah satunya adalah *Dynamic Source Routing* (DSR). DSR termasuk dalam klasifikasi *reactive routing protocol*. Sebuah *routing protocol* yang hanya akan membuat rute ketika ada paket yang ingin dikirim. Modifikasi akan dilakukan pada proses pengiriman paket *route request* (RREQ), dengan cara mengeliminasi jumlah *neighbor node* yang bertugas mengirim ulang (*rebroadcast*) paket RREQ. Hal ini dilakukan dengan cara melihat jumlah *node* tetangga dari tiap *node*, jika *node* tersebut memiliki jumlah *node* tetangga lebih dari *threshold*, maka *node* tersebut yang bisa melakukan proses *rebroadcast*. Jika paket RREQ sampai pada *node* tujuan, maka *node* tujuan akan mengirim paket *route reply* (RREP) ke *node* asal. Lalu rute untuk pengiriman paket akan terbentuk. Modifikasi yang dilakukan akan menghasilkan *routing overhead* dan *forwarded route request* yang lebih kecil daripada *routing protocol* DSR yang asli. Hasil pengujian yang dilakukan menunjukkan bahwa DSR yang telah dimodifikasi menghasilkan peningkatan rata-rata *packet delivery ratio* sebesar 5.41%, rata-rata penurunan *routing overhead* sebesar 50.72%, dan juga rata-rata penurunan *forwarded route request* sebesar 16.9%.

Kata Kunci—DSR, *Forwarding Node*, *Threshold*, VANETs.

I. PENDAHULUAN

INFORMASI merupakan suatu hal yang sangat mudah didapat dewasa ini. Teknologi internet saat ini juga digunakan sebagai suatu pemecahan suatu masalah. Contohnya di jalan raya antara lain yaitu penentuan rute untuk sampai ke tujuan lebih cepat, terjadi kecelakaan, dan terjadinya kemacetan dikarenakan waktu *traffic light* disamaratakan pada suatu persimpangan. Proses penentuan rute perjalanan berhubungan dengan rute pengiriman data informasi dalam jaringan internet. Proses penentuan rute ini disebut dengan *routing*. Untuk menanggulangi hal tersebut dapat memanfaatkan teknologi jaringan *Ad-Hoc* yang mana mendasari pembuatan *Vehicular Ad-Hoc Networks* (VANETs).

VANETs merupakan pengembangan dari *Mobile Ad-Hoc*

Networks (MANETs). Implementasi dari MANETs telah menciptakan beberapa *routing protocol*. Berdasarkan perlakuannya terhadap rute, dapat dibedakan menjadi dua, yaitu *routing protocol* bersifat proaktif dan *routing protocol* bersifat reaktif. Contoh untuk *routing protocol* proaktif adalah *Optimized Link State Routing Protocol* (OLSR). OLSR biasanya digunakan pada lingkungan MANETs, optimasi tersebut berdasarkan pada *link state* (LS) *protocol* tradisional, kuncinya berada pada *multi-point relay*. Hanya beberapa *node* yang terpilih sebagai *node* perantara untuk mengontrol paket, jadi untuk mengurangi *control packet* yang terlalu banyak. Untuk contoh *routing protocol* reaktif yaitu *Dynamic Source Routing* (DSR) dimana rute yang dibentuk oleh *routing protocol* tersebut berasal dari *node* sumber [1].

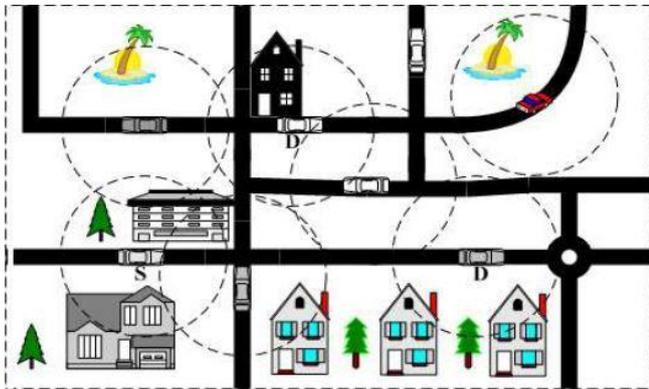
DSR akan dimodifikasi dengan mengimplementasikan metode dari OLSR yaitu dengan cara mengeliminasi jumlah *forwarding node* yang bertugas melakukan *rebroadcast* dengan melihat jumlah *neighboring node*. Uji coba akan dilakukan dan akan membandingkan performa protokol DSR dan DSR yang telah dimodifikasi dengan skenario *real* dengan beberapa variasi jumlah *node*.

II. ANALISIS DAN PERANCANGAN

A. *Vehicular Ad-Hoc Networks* (VANETs)

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, bertujuan untuk meningkatkan keselamatan mengemudi dan manajemen lalu lintas sementara menyediakan *driver* dan penumpang dengan akses Internet. Sebagai contoh, sebuah kendaraan dapat berkomunikasi dengan lampu lalu lintas cahaya melalui V2I komunikasi, dan lampu lalu lintas dapat menunjukkan ke kendaraan ketika keadaan lampu ke kuning atau merah. Ini dapat berfungsi sebagai tanda pemberitahuan kepada pengemudi, dan akan sangat membantu para pengendara ketika mereka sedang berkendara selama kondisi cuaca musim dingin atau di daerah asing. Hal ini dapat

mengurangi terjadinya kecelakaan. Melalui komunikasi V2V, pengendara bisa mendapatkan informasi yang lebih baik dan mengambil tindakan awal untuk menanggapi situasi yang abnormal. Untuk mencapai hal ini, suatu OBU (*On Board Unit*) secara teratur menyiarkan pesan yang terkait dengan informasi dari posisi pengendara, waktu saat ini, arah mengemudi, kecepatan, status rem, sudut kemudi, lampu sen, percepatan / perlambatan, kondisi lalu lintas [2]. Ilustrasi VANETs dapat dilihat pada Gambar 1.

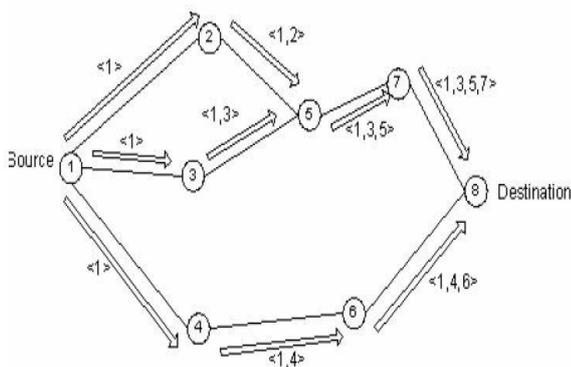


Gambar 1 Ilustrasi VANETs [2].

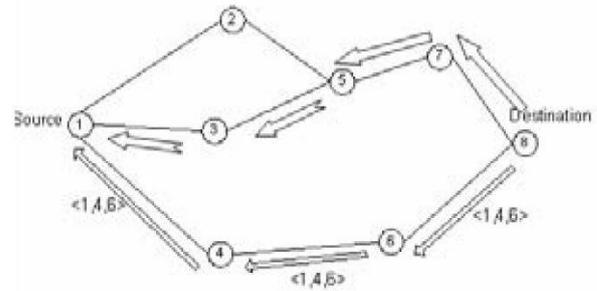
B. Dynamic Source Routing (DSR)

DSR adalah *routing protocol* yang sederhana dan efisien dirancang khusus untuk jaringan *ad hoc* nirkabel *multi-hop*. *Nodes* jaringan bekerja sama untuk meneruskan paket untuk yang lainnya untuk memungkinkan komunikasi melalui beberapa ‘hop’ antar *nodes* yang tidak secara langsung berada dalam jangkauan transmisi nirkabel satu sama lain [3].

Ketika *node* asal ingin mengirim paket ke *node* tujuan, jika tidak ditemukan rute menuju ke *node* tujuan, maka *node* asal menginisiasi mekanisme *route discovery* dengan cara *broadcast route request (RREQ)* ke tetangga. Jika tetangga mendapatkan *request* yang sama dari sumber yang berbeda atau alamatnya yang terdaftar pada RREQ, maka paket tersebut di *drop*. Sebaliknya jika itu adalah *node* tujuan, maka *node* tujuan tersebut mengirim *route reply (RREP)* kembali ke *node* asal. Ketika kondisi tersebut tidak terpenuhi, *node* yang menerima menambahkan alamat diri sendiri pada RREQ. RREP akan menelusuri rute yang sudah tersimpan sampai ke *node* asal [4]. Untuk ilustrasi pengiriman RREQ dan RREP pada DSR dapat dilihat masing-masing pada Gambar 2 dan Gambar 3.



Gambar 2. Mekanisme Pengiriman RREQ pada DSR [4].



Gambar 3. Mekanisme Pengiriman RREP pada DSR [4].

C. Network Simulator-2 (NS-2)

Network Simulator (Versi 2), yang banyak dikenal dengan NS-2, adalah sebuah alat simulasi berbasis aktivitas yang berguna dalam mempelajari sifat dinamis jaringan komunikasi. Simulasi fungsi jaringan kabel, nirkabel, dan protokol dapat dilakukan dengan menggunakan NS-2. NS-2 menggunakan dua bahasa utama yaitu Bahasa C++ dan *Object-oriented Tool Command Language (OTCL)*. Di NS-2 mendefinisikan mekanisme internal (*backend*) dari objek simulasi, dan OTCL mendefinisikan lingkungan simulasi eksternal (*frontend*) untuk perakitan dan konfigurasi objek. Setelah simulasi, NS-2 memberikan output simulasi baik dalam bentuk file NAM atau *trace file* [5].

D. Simulation of Urban Mobility (SUMO)

Simulation of Urban Mobility atau disingkat SUMO merupakan simulasi lalu lintas jalan raya yang *open source*, *microscopic*, dan *multi-modal*. SUMO mensimulasikan bagaimana permintaan lalu lintas yang terdiri dari beberapa kendaraan berjalan pada jalan raya yang telah ditentukan. Simulasi SUMO dapat menunjukkan beberapa topik manajemen lalu lintas dalam skala besar. SUMO murni *microscopic*, yang artinya setiap kendaraan dimodelkan secara eksplisit, memiliki rutenya sendiri, dan bergerak secara individu dalam jaringan [6].

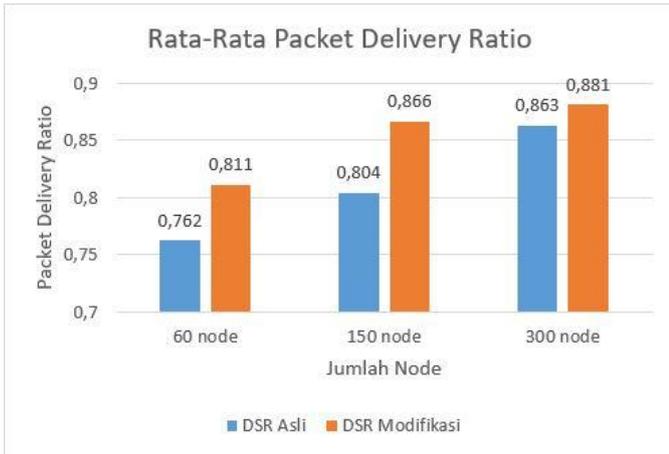
E. AWK

AWK merupakan sebuah bahasa pemrograman yang didesain untuk *text-processing* dan biasanya digunakan sebagai alat ekstraksi data dan pelaporan. AWK bersifat *data-driven* yang berisikan kumpulan perintah yang akan dijalankan pada data tekstural baik secara langsung pada *file* atau digunakan sebagai bagian dari *pipeline* [7].

III. UJI COBA

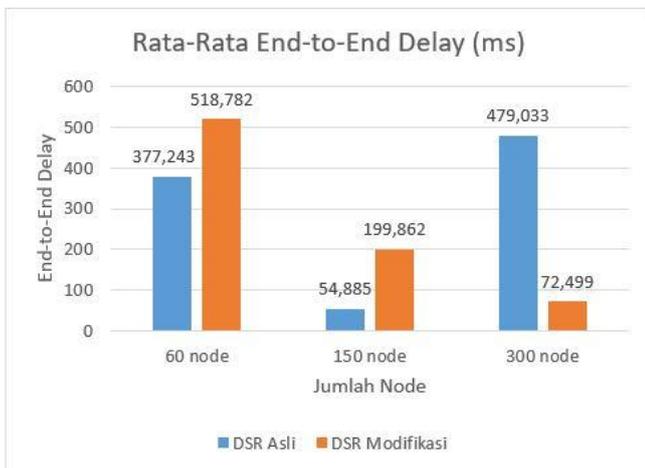
Uji coba menggunakan skenario *real* dengan kecepatan 20 m/s dengan menggunakan variasi lingkungan 60 *node*, 150 *node*, dan 300 *node*. Dari hasil uji coba yang dilakukan akan dibandingkan nilai *packet delivery ratio (PDR)*, *end-to-end delay*, *routing overhead (RO)*, dan *forwarded route request (RREQ F)* antara DSR asli dengan DSR yang telah dimodifikasi. Uji coba yang dilakukan pada DSR yang telah dimodifikasi menggunakan *threshold* yang berbeda beda untuk tiap lingkungan skenario. Pada lingkungan 60 *node* menggunakan *threshold* 14, sedangkan pada lingkungan 150 *node* menggunakan *threshold* 17, dan pada lingkungan 300 *node* menggunakan *threshold* 19. Untuk perbandingan hasil

pengujian dapat dilihat masing-masing pada Gambar 4, Gambar 5, Gambar 6, dan Gambar 7.



Gambar 4. Grafik Rata-Rata *Packet Delivery Ratio*.

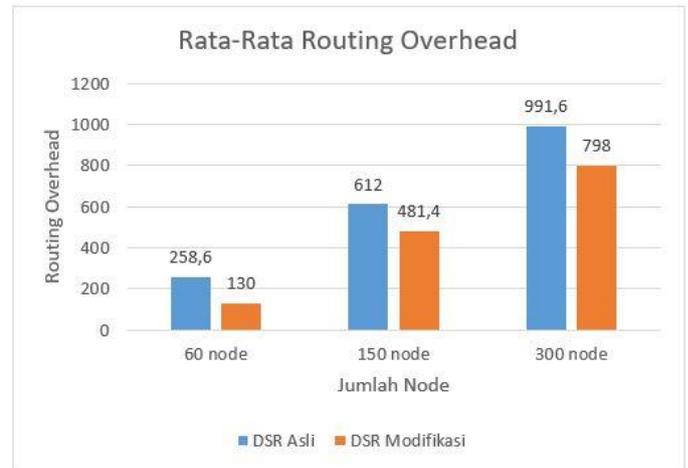
Berdasarkan grafik pada Gambar 4 terlihat bahwa PDR pada DSR yang telah dimodifikasi memiliki hasil yang lebih unggul daripada DSR asli. Pada lingkungan 60 node mengalami kenaikan sebesar 6.43% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 150 node mengalami kenaikan sebesar 7.71% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 300 node mengalami kenaikan sebesar 2.08% dari DSR asli menuju DSR yang telah dimodifikasi. Sehingga rata-rata PDR untuk ketiga lingkungan tersebut yaitu 5.41%. Semakin banyak jumlah node maka PDR yang dihasilkan semakin meningkat. Hal ini dikarenakan untuk lingkungan dengan jumlah node yang banyak menghasilkan rute yang banyak juga untuk pengiriman paket, sehingga jumlah PDR juga semakin meningkat.



Gambar 5. Grafik Rata-Rata *End-to-End Delay*.

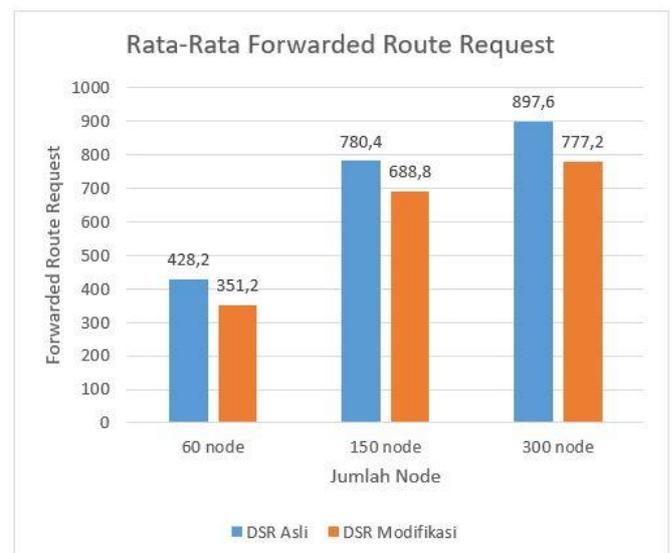
Berdasarkan grafik pada Gambar 5 terlihat untuk grafik *end-to-end delay* mengalami perubahan yang fluktuatif dari lingkungan dengan jumlah node yang sedikit menuju lingkungan dengan jumlah node yang banyak. Pada lingkungan 60 node mengalami kenaikan *end-to-end delay* sebesar 37.52% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 150 node mengalami kenaikan yang cukup jauh yaitu sebesar 264.15% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 300 node mengalami lonjakan penurunan yang sangat drastis yaitu

sebesar 560.74% dari DSR asli menuju DSR yang telah dimodifikasi. Hal ini dikarenakan waktu *delay* bergantung dari waktu rata-rata paket yang terkirim sampai node tujuan. Jika paket yang terkirim semakin banyak, maka semakin beragam pula jumlah *delay* tersebut.



Gambar 6. Grafik Rata-Rata *Routing Overhead*.

Berdasarkan grafik pada Gambar 6 terlihat bahwa RO pada DSR yang telah dimodifikasi memiliki hasil yang lebih unggul atau lebih sedikit dari DSR asli. Pada lingkungan 60 node mengalami penurunan sebesar 98.92% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 150 node mengalami penurunan sebesar 28.99% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 300 node mengalami penurunan sebesar 24.26% dari DSR asli menuju DSR yang telah dimodifikasi. Sehingga rata-rata penurunan RO yang terjadi sebesar 50.72%. Grafik RO pada Gambar 6 terlihat mengalami kenaikan dari lingkungan 60 node menuju lingkungan 300 node, hal ini dikarenakan semakin banyak node pada lingkungan tersebut maka semakin banyak pula jumlah RO yang terbentuk karena bisa saja satu node mengalami RO beberapa kali.



Gambar 7. Grafik Rata-Rata *Forwarded Route Request*.

Berdasarkan grafik pada Gambar 7 terlihat bahwa RREQ F pada DSR yang telah dimodifikasi memiliki hasil yang lebih unggul atau lebih sedikit dari DSR asli. Pada lingkungan 60

node mengalami penurunan sebesar 21.92% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 150 *node* mengalami penurunan sebesar 13.3% dari DSR asli menuju DSR yang telah dimodifikasi. Pada lingkungan 300 *node* mengalami penurunan sebesar 15.49% dari DSR asli menuju DSR yang telah dimodifikasi. Sehingga rata-rata penurunan RREQ F yang terjadi sebesar 16.9%. Grafik RREQ F pada Gambar 7 terlihat mengalami kenaikan dari lingkungan 60 *node* menuju lingkungan 300 *node*, hal ini dikarenakan semakin banyak *node* pada lingkungan tersebut maka semakin banyak pula jumlah RREQ F karena akan semakin banyak *node* yang *forward* paket *route request*.

IV. KESIMPULAN DAN SARAN

Setelah dilakukan pengujian dan evaluasi, didapatkan kesimpulan sebagai berikut:

- 1) Dengan menambahkan perhitungan jumlah tetangga pada *routing protocol* DSR dan membatasi *node* mana saja yang bisa *rebroadcast* paket RREQ dengan menggunakan *threshold* dari jumlah tetangga, maka dapat mengurangi jumlah *forwarding node* yang melakukan *rebroadcast*.
- 2) Dampak pembatasan *forwarding node* terhadap performa *routing protocol* DSR secara keseluruhan untuk skenario *real* menghasilkan peningkatan rata-rata *packet delivery ratio* sebesar 5.41%, rata-rata penurunan *routing overhead* sebesar 50.72%, dan juga rata-rata penurunan *forwarded route request* sebesar 16.9%.

Adapun saran-saran yang diberikan untuk pengembangan sistem ini kedepannya adalah sebagai berikut:

- 1) Untuk mendapatkan hasil uji coba yang lebih baik, dapat dilakukan uji coba yang lebih banyak, misal lebih

dari 10 kali percobaan untuk tiap skenario.

- 2) Kedepannya dapat melakukan perhitungan jumlah *node* tetangga yang lebih dinamis pada *routing protocol* DSR dan untuk lingkungan yang dinamis, misalnya pada *node* yang diluar jangkauan transmisi, dapat menghapus *node* tersebut dari list simpanan *node* tetangga sehingga perhitungan jumlah *node* tetangga lebih akurat.
- 3) Ide untuk implementasi pengurangan jumlah *forwarding node* sudah bagus, kedepannya bisa mengimplementasikan dengan menambahkan aspek lain yang dijadikan untuk pembatasan *forwarding node* seperti energi, kecepatan, arah, dan lain-lain.

DAFTAR PUSTAKA

- [1] S. N. Ferdous and M. S. Hossain, "Randomized Energy-Based AODV Protocol For Wireless Ad-Hoc Network," in *IEEE*, 2016.
- [2] L. Khan, N. Ayub, and A. Saeed, "Anycast Based Routing in Vehicular Adhoc Networks (VANETS) using Vanetmobisim," in *COMSATS IIT Abbottabad*, 2009.
- [3] K. Majumder and S. K. Sarkar, "Performance Analysis of AODV and DSR Routing Protocols in Hybrid Network Scenario," in *IEEE*, 2009.
- [4] M. Tamilarasi, U. M. H. S. S. V. R, C. Somathilaka, N. R. Babu, S. Chandramathi, and T. G. Palanivelu, "Scalability Improved DSR Protocol for MANETs," in *IEEE*, 2007.
- [5] N. M. Mittal and S. Choudhary, "Comparative Study of Simulators for Vehicular Ad-hoc Networks (VANETS)," in *IJETAE*, 2014.
- [6] SUMO, "SUMO," 2017. [Online]. Available: http://www.sumo.dlr.de/userdoc/Sumo_at_a_Glance.html.
- [7] A. V. Aho, B. W. Kernighan, and P. J. Weinberger, *The AWK Programming Language*. Boston: Addison-Wesley, 1988.