

Implementasi Algoritma Pemilihan Node Tetangga Terbaik Pada Protokol Routing DSR di Jaringan MANET

(Implementation of the Best Neighboring Node Selection Algorithm on DSR Routing Protocol in MANET)

Andy Hidayat Jatmika, Susilo Pandu Waskito*, Ariyan Zubaidi

Program Studi Teknik Informatika, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: andy@unram.ac.id, susilopanduwaskito@gmail.com, zubaidi13@unram.ac.id

*Penulis Korespondensi

Abstract Mobile Ad Hoc Network, commonly abbreviated as MANET, is a wireless telecommunications technology consisting of a collection of nodes that exchange information and are dynamic. A common problem that usually occurs in MANET is the change in network topology that often changes continuously due to the movement of node mobility and causes the route to be easily interrupted. So that the process of finding a route back due to a route being interrupted can often occur. The routing protocol Dynamic Source Routing (DSR) in the route search process broadcasts RREQ to all neighboring nodes (intermediate nodes) without knowing whether the node is the best node or not. So that the possibility of the route being interrupted because it is not composed of the best nodes can occur. A modification of the DSR routing protocol is proposed by finding the best intermediate node based on parameters, namely bandwidth, RTT, packet loss ratio. Then the selected neighbor nodes (intermediate nodes) will arrange the delivery route from the source node to the destination node. The performance of the DSR and DSR routing protocols to be measured will then be measured using test parameters, namely throughput, packet delivery ratio (PDR) and average end to end delay. Based on the results of research trials, the application of the modified DSR method can improve the performance of the DSR protocol. Which after the application of the method, the average throughput increased by 25%, Packet Delivery Ratio (PDR) was 1.11% and the average end-to-end delay decreased by 17.64 %.

Key words: MANET, DSR, Neighbor Nodes, Routing Protocol, Reactive

I. PENDAHULUAN

Saat ini kebutuhan akan jaringan nirkabel sangat tinggi dikarenakan adanya aktivitas manusia yang cenderung bergerak dan membawa suatu perangkat komunikasi. Pergerakan manusia tersebut menyebabkan perangkat yang dibawa menjadi ikut bergerak bebas. Ada kalanya sesama perangkat bergerak tersebut membutuhkan suatu jaringan yang dapat dibangun setiap saat. Jaringan inilah yang disebut *Mobile Ad Hoc Network* (MANET). Ketika jaringan ini telah terbentuk,

hal yang perlu diperhatikan dan yang sangat penting adalah adanya rute pengiriman yang selalu berubah [1]. Dalam jaringan MANET diperlukan aturan *routing* yang dipakai untuk menemukan rute dari node sumber ke node tujuan. Salah satu aturan atau protokol tersebut adalah protokol *Dynamic Source Routing* (DSR). Protokol routing DSR akan membentuk suatu rute jika terjadi permintaan rute oleh node sumber [2].

Dalam proses menemukan rute, DSR mengirim paket *Route Request (RREQ)* ke semua *node* tetangganya tanpa melihat *node* tersebut yang terbaik atau bukan. Jika rute yang ditemukan bukan terdiri dari *node* terbaik, maka permasalahan yang terjadi adalah besar kemungkinan rute tersebut akan cepat terputus sehingga harus dilakukan proses pencarian ulang rute dari awal sehingga butuh waktu lama. Untuk mengatasi cara kerja DSR yang masih standar tersebut, maka diusulkan solusi dengan cara memodifikasi kerangka protokol DSR dengan memilih *node* tetangga terbaik dilihat dari 3 parameter atau perhitungan yang akan diukur ketika dilakukan pencarian rute yaitu parameter *bandwidth*, *Round Trip Time* (RTT), dan *packet loss ratio*. Penggunaan metode ini diharapkan agar *node-node* tetangga (*node intermediate*) yang berpartisipasi dalam pembentukan rute adalah *node-node* terpilih berdasarkan 3 parameter yang telah disebutkan tadi.

Metode pemilihan *node tetangga (intermediate node)* terbaik dimulai dari *node* sumber yang mengirimkan *hello packet* kepada *node* tetangga (*intermediate node*). Kemudian *node* tetangga akan membalas *hello packet* ke *node* pengirim. Selanjutnya *node* pengirim akan memilih *node* tetangga mana saja yang akan dimasukkan ke dalam *neighbor table* dengan cara melakukan penghitungan kriteria. Perhitungan kriteria yang dilakukan untuk menentukan *node* tetangga terbaik yaitu yang memiliki rasio *bandwidth* < 70%, RTT < 15% dan *packet loss ratio* < 15 % yang mana jika dijumlahkan semuanya akan menghasilkan nilai 100 %. Kemudian *node* pengirim akan memasukkan *node* tetangga terpilih ke dalam *neighbor table* miliknya. Setelah itu *node* pengirim

mengirim paket RREQ ke *node* tetangga terpilih. Prosedur perhitungan parameter ini akan terus dilakukan oleh *node tetangga* (*intermediate node*) jika *node* tujuan belum ditemukan [1].

Kinerja DSR yang dimodifikasi akan diuji berdasarkan parameter – parameter yang dijadikan pengujinya berupa *throughput*, *packet delivery ratio* (PDR) dan *average end to end delay* lalu akan dibandingkan juga kinerjanya dengan DSR standar untuk mengetahui apakah DSR modifikasi memiliki kinerja yang lebih baik.

Simulasi jaringan memakai perangkat lunak *Network Simulator 2* (NS2) 2.35 . NS2.35 akan diinstal pada OS Linux Ubuntu 14.04 LTS di dalam sebuah lingkungan mesin virtual yaitu VirtualBox. Bahasa pemrograman untuk simulasi adalah *Tools Command Language* (TCL). Bahasa pemrograman yang digunakan untuk menerapkan metode yang diusulkan menggunakan bahasa pemrograman C++. Untuk mengetahui nilai *throughput*, *packet delivery ratio* (PDR) dan *average end to end delay* digunakan Bahasa pemrograman AWK.

II. TINJAUAN PUSTAKA

Pada penelitian [1], protokol *Routing AODV* dimodifikasi dengan menggunakan algoritma *location aware*, yang dipilih adalah *node* tetangga terbaik berdasarkan *bandwidth*, *round trip time* dan *packet loss ratio*. *Node* tetangga yang terpilih akan membentuk jalur terbaik. Protokol *Routing* yang dimodifikasi menggunakan algoritma *location aware* memiliki beberapa kelebihan. Pertama, protokol *Routing AODV* dengan algoritma *location aware* bekerja secara efisien pada pencarian jalur dengan jumlah *node* yang banyak. Kedua, paket yang diteruskan dari *node* sumber ke *node* tujuan memiliki durasi yang relative pendek sehingga meningkatkan efisiensi waktu pengiriman. Ketiga, metode ini mengkonsumsi lebih sedikit energi, karena *node* untuk menyusun jalur dalam pengiriman paket hanya disusun oleh *node intermediate* (*node tetangga*) terbaik yang telah dipilih.

Pada penelitian [2], protokol DSR standar dimodifikasi dengan menerapkan algoritma *Path Aware-Short* untuk mengatasi proses pencarian rute yang memakan waktu lama. Setelah dilakukan pengujian antara protokol DSR modifikasi dengan DSR standar, hasil pencarian rute oleh DSR modifikasi lebih baik dibanding DSR standar karena tanpa harus melakukan pencarian rute lagi dan waktu yang dibutuhkan menjadi singkat.

Pada penelitian [3], dilakukan penelitian dengan membandingkan protokol *Ad Hoc on-Demand Distance Vector* (AODV) dengan protokol DSR saat terjadi keadaan *broadcast storm* yaitu keadaan jaringan yang di dalamnya terdapat *node* yang mengirim paket secara masif sehingga jaringan menjadi terbanjiri paket. Simulasi simulasi terdiri dari *node* yang tetap dan *node* yang bergerak. Banyak *node* untuk pengujian sebanyak 30, 40, dan 60 *node*. Parameter uji kinerja

jaringan menggunakan parameter *throughput*, *delay*, dan *packet loss*. Hasil pengujian menunjukkan protokol *routing AODV* memiliki kinerja yang lebih saat terjadinya *broadcast storm*.

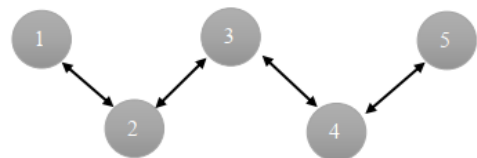
Pada penelitian [4], protokol AODV dan DSR dibandingkan menggunakan beberapa skenario pengujian menggunakan simulator OPNET Modeler 14.5. Uji kinerja kedua protokol diukur menggunakan parameter *latency*, *throughput*, *jitter* dan *packet loss*. Hasil pengujian menunjukkan bahwa protokol AODV mempunyai kinerja yang lebih bagus dibanding DSR dilihat dari parameter *latency*, *throughput*, dan *jitter*. Namun protokol *routing DSR* lebih bagus dibandingkan AODV dari sisi *packet loss*.

Pada penelitian [5], dilakukan uji perbandingan kinerja protokol DSR dan *Geographic Routing Protocol* (GRP). Kedua protokol tersebut dianalisis kinerjanya. Jumlah *node* yang dipakai pada skenario uji sebanyak 25 *node* dan 50 *node* dengan parameter pengujian kinerja yaitu *throughput*, *delay*, *load*, *media access delay*, *data dropped* dan *network load*. Hasil pengujian menunjukkan bahwa protokol GRP lebih bagus dibandingkan protokol DSR berdasarkan pengukuran nilai parameter *throughput*, *delay*, *media access delay* dan *data dropped*. Hanya saja untuk parameter uji *load*, DSR lebih bagus dibandingkan protokol GRP.

A. Mobile Ad-Hoc Network (MANET)

MANET merupakan suatu jenis jaringan yang dapat dibangun setiap saat, kapan saja, dan dimana saja ketika dibutuhkan tanpa menggunakan kabel sehingga perangkat yang saling terhubung mampu bergerak ke segala arah sambil dilakukan pertukaran data. Jaringan jenis ini biasanya digunakan ketika keadaan darurat seperti di medan perang, bencana alam, dan keadaan-keadaan darurat lainnya. Kelemahan dari jaringan ini adalah topologi yang setiap waktu dapat berubah akibat pergerakan tersebut [4]. Perangkat-perangkat yang tergabung dalam jaringan MANET atau yang disebut *node* memiliki level yang sama yaitu semua dapat sebagai *router* yang berfungsi sebagai alat yang menemukan rute dan meneruskan paket ke *router* lainnya.

Gambar 1 merupakan ilustrasi dari jaringan MANET sederhana.



Gambar 1. Mobile Ad-Hoc Network [3]

Pada Gambar 1 terdapat 5 buah titik yang disebut *node* dan terdapat angka di dalamnya yang menunjukkan ID dari *node-node* tersebut. Anak panah menunjukkan rute yang terbentuk dari jaringan MANET tersebut. *Node 2* berkomunikasi dengan *node 1* dan *3*, *node 3* berkomunikasi dengan *node 2* dan *4*, *node 4*

berkomunikasi dengan node 3 dan 5. Jika node 1 akan mengirim data ke node 5, maka akan melalui rute 1-2-3-4-5. Node 2, node 3, dan node 4 bertindak sebagai *node intermediate*. Node 1 bertindak sebagai node sumber, sedangkan node 5 bertindak sebagai node tujuan.

B. Protokol Routing

Untuk menemukan atau membangun sebuah rute pada jaringan MANET diperlukan suatu protokol routing. Protokol routing adalah sebuah aturan bagaimana paket akan diteruskan dan lewat mana paket tersebut akan melintas. Protokol ini akan bekerja pada semua node yang terdapat pada jaringan. Node-node akan meneruskan paket berdasarkan algoritma dari protokol routing yang digunakan, kemana paket akan diteruskan.

Node-node akan saling berkomunikasi menggunakan protokol routing. Paket yang diteruskan dari node sumber akan sampai ke node tujuan dari rute yang telah ditemukan [5].

C.1 Protokol Routing Reaktif

Kelompok protokol routing reaktif merupakan jenis protokol dimana rute akan terbentuk jika ada sebuah *node* ingin mengirim data ke *node* tujuan. Berbanding terbalik dengan protokol routing jenis proaktif. Suatu node yang akan mengirim paket akan mengirim paket RREQ terlebih dahulu ke node-node tetangganya. Node tetangga yang menerima paket RREQ akan membacanya apakah paket tersebut ditujukan pada dirinya atau tidak, jika tidak maka akan diteruskan ke node lain yang berada disekitarnya, begitu seterusnya sampai paket RREQ tersebut tiba di node tujuan. Node tujuan akan membalas dengan paket *Route Reply* (RREP) melewati rute yang telah ditemukan. Salah satu contoh dari protokol jenis ini adalah DSR, AODV dan *Ad hoc On-Demand Multipath Distance Vector* (AOMDV) [6].

C.2 Protokol Routing Proaktif

Kebalikan dari jenis protokol routing reaktif. Protokol routing proaktif telah memiliki rute-rute dan akan selalu diperbaharui secara periodik. Semua *node* yang berada dalam jaringan memperbaharui informasi rute di dalam tabel routing miliknya masing-masing sehingga akan cepat diketahui jika ada perubahan topologi yang disebabkan pergerakan node dalam selang waktu tertentu. Beberapa protokol yang termasuk didalamnya adalah OLSR, BATMAN, DSDV, dan MDART [6].

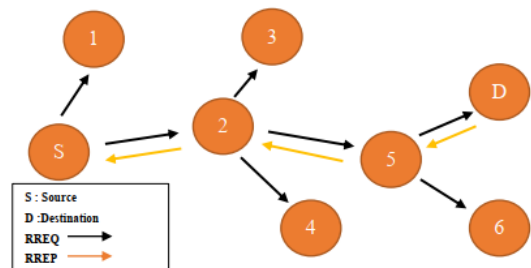
C. Dynamic Source Routing (DSR)

DSR termasuk protokol routing yang tergabung dalam kelompok jenis reaktif. Berdasarkan namanya, DSR memiliki mekanisme *source routing* dimana informasi rute akan diperbaharui oleh node sumber [5]. Protokol DSR memiliki 2 proses kerja yaitu *route discovery* dan *route maintenance*. *Route discovery* adalah proses penemuan rute dimana terdapat paket yang dikirimkan yaitu RREQ ke seluruh node yang saling terhubung satu sama lain. RREQ berisi informasi node sumber dan node tujuan. Node intermediate akan

meneruskan RREQ ini hingga sampai ke node tujuan. Selanjutnya node tujuan akan mengirimkan paket berupa RREP ke node sumber yang mengindikasikan bahwa node tujuan telah menerima RREQ dengan baik dan siap menerima data dari node sumber [3].

Route maintenance merupakan suatu proses dimana ada kejadian rute terputus saat pengiriman data berlangsung. Node yang mengalami putusnya jalur akan mengirimkan informasi berupa *route error* (RERR) ke node sumber yang mengindikasikan bahwa ada jalur yang terputus sehingga perlu dilakukan pencarian rute ulang [3].

Gambar 2 merupakan ilustrasi proses pencarian rute oleh protokol DSR.



Gambar 2. Pencarian Rute DSR [3]

Pada Gambar 2 terdapat 8 buah node yang tersebar di jaringan. Terdapat sebuah node sumber yang ditandai dengan huruf S yang akan mengirim paket data ke node tujuan yang ditandai dengan huruf D. Node S tidak secara langsung terhubung dengan node D sehingga perlu dilakukan pencarian rute menggunakan protokol DSR. Node S mengirim paket RREQ ke node tetangganya yaitu node 1 dan node 2. Karena bukan ditujukan untuk dirinya, node 2 meneruskan paket RREQ ke node-node tetangganya yaitu node 3, node 4, dan node 5. Node 5 meneruskan RREQ ke node tetangganya yaitu node D dan node 6. Node D membaca paket RREQ dan mengetahui paket tersebut untuk dirinya, maka node D akan mengirim RREP ke node sumber melalui node 5 dan node 2 hingga sampai ke node S. Ketika RREP sampai ke node S, maka node S akan mengirim paket data melalui rute S-2-5-D.

D. Prosedur memilih Node tetangga

Proses ini dilakukan dengan cara memeriksa berbagai parameter untuk memilih *node* tetangga terbaik yang dapat membentuk rute yang terdiri dari sekumpulan *node* tetangga (*intermediate node*) terbaik yang telah dipilih berdasarkan parameter yang telah ditentukan. Berbagai parameter yang digunakan dalam pencarian rute pada metode ini antara lain *node id*, *bandwidth*, RTT dan *packet loss ratio*. Parameter yaitu *bandwidth*, RTT dan *packet loss ratio* akan menjadi kunci untuk memilih *node* tetangga (*intermediate node*) terbaik dalam menyusun rute pengiriman dari *node* sumber ke *node* tujuan. *Node* tetangga terbaik yang dipilih adalah *node* yang memiliki rasio *bandwidth* > 70%, RTT < 15% dan *packet loss ratio*

< 15 %. Yang dimana apabila semuanya dijumlahkan maka akan menghasilkan perhitungan $bandwidth + RTT + packet\ loss\ ratio = 100\%$ [1].

E. Bandwidth

Bandwidth adalah ukuran kapasitas maksimum pengiriman paket dari satu titik ke titik lain pada suatu jaringan. Satuan yang digunakan yaitu *bit per second* (bps) atau *Byte per second* (Bps) dimana untuk setiap nilai 1 Byte setara dengan 8 bit. Hal ini menggambarkan seberapa banyak bit yang dikirim setiap detiknya [10].

F. Round Trip Time (RTT)

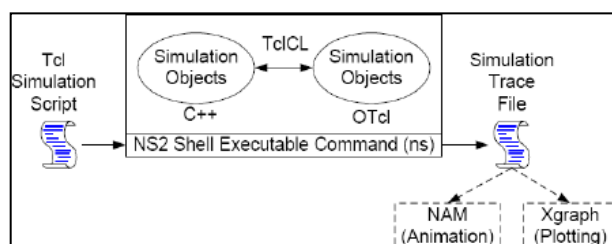
Waktu yang dibutuhkan oleh *node* sumber untuk mengirim sebuah permintaan kepada *node* tujuan hingga *node* tujuan memberikan tanggapan kembali kepada *node* sumber. Untuk dapat menghitung waktu *round trip time* dapat dilakukan uji coba pengiriman paket ICMP (ping), DNS atau bisa juga TCP. Nilai RTT dipengaruhi oleh besarnya paket permintaan yang dikirimkan oleh *node* sumber dan keadaan lalu lintas jaringan yang suka berubah. Jika keadaan lalu lintas jaringan cukup baik dan lancar, maka nilai RTT-nya kecil, namun apabila keadaan lalu lintas jaringan kurang baik atau lamban, maka RTT-nya besar [11].

G. Packet Loss Ratio

Total paket yang hilang saat proses pengiriman paket dari *node* sumber ke *node* tujuan. Pengiriman paket dilakukan dalam suatu *interval* waktu tertentu saat transmisi. Keadaan ini terjadi ketika satu atau lebih paket yang dikirim tidak berhasil sampai ke tujuannya. *Packet loss* juga biasanya disebabkan oleh kesalahan dalam transmisi data saat melalui jaringan nirkabel, kemacetan jaringan, atau tabrakan lalu lintas data dalam sebuah jaringan. Satuan yang digunakan adalah persentase dari paket yang hilang dengan paket yang dikirim. [12].

H. Network Simulator (NS-2)

Jenis simulator populer yang banyak digunakan oleh banyak peneliti. NS-2 diciptakan dengan 2 bahasa yaitu kumpulan *library* dan bahasa pemrograman C++ yang digunakan untuk *event scheduler*, protokol, komponen jaringan, dan bahasa pemrograman Tcl/OTcl untuk skrip simulasinya yang dapat juga dilihat batang tubuh dari NS-2 pada Gambar 3 [13].



Gambar 3. Batang tubuh NS-2 [13]

File *trace* ber-ekstensi *.tr* dapat di-filter memakai skrip AWK dan dapat juga dikonversikan dalam bentuk grafik dengan *tool* XGraph [12].

I. Bahasa Pemrograman TCL

TCL merupakan singkatan dari *tool command language*. TCL berisi *library package* yang terdapat pada *tools* sebagai penterjemah perintah dasar. Tcl menyediakan fitur seperti sebuah parser untuk bahasa perintah tekstual sederhana, kumpulan perintah *built-in utility*, dan sebuah *interface* bahasa C yang sebagai alat untuk menambah perintah *built-in* dengan perintah yang *tool-specific*. Ada empat faktor pada desain bahasa TCL yaitu :

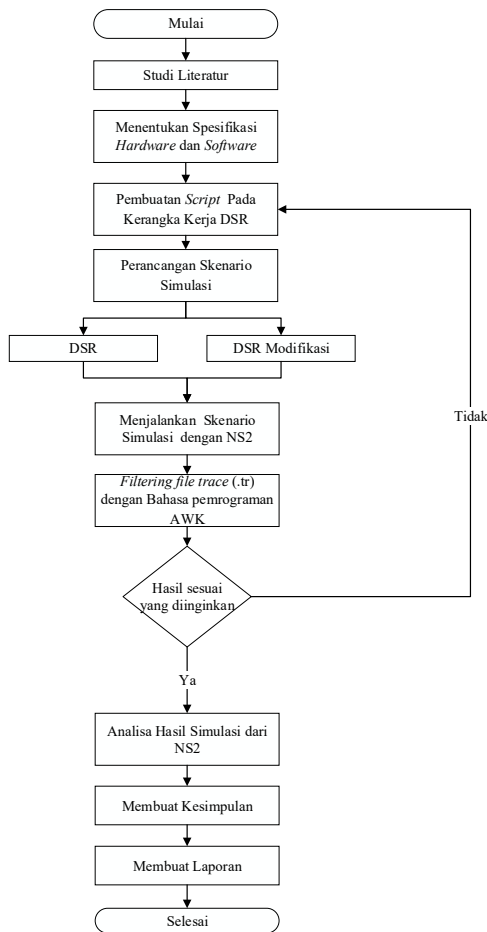
- 1) Bahasa yang digunakan untuk perintah. Hampir semua program Tcl pendek, yang memiliki satu *line* saja. Kebanyakan program akan ditulis dan dieksekusi sesekali atau beberapa kali, kemudian dibuang. Ini menandakan bahasa Tcl harus memiliki *syntax* sederhana sehingga lebih mudah untuk menulis perintah, dimana biasanya bahasa pemrograman memiliki *syntax* yang kompleks.
- 2) Bahasa yang digunakan untuk programming. Bahasa Tcl memiliki *constructor programming* seperti variabel, prosedur, kondisi, dan *loop*, sehingga *users* dapat memperluas set perintah *built-in* dengan menulis prosedur Tcl.
- 3) Bahasa yang mengizinkan *interpreter* yang sederhana dan efisien. *Library* Tcl terdapat pada program-program berukuran kecil, dengan kata lain mesin tanpa fasilitas *shared-library*, *interpreter* haruslah tidak memakan banyak *memory*. Mekanisme penerjemahan perintah Tcl haruslah cepat digunakan untuk *event* yang bekerja ratusan kali tiap detik, seperti pergerakan mouse.
- 4) Bahasa yang mengizinkan *interface* sederhana untuk aplikasi C. Bahasa haruslah mudah digunakan untuk aplikasi C untuk diterjemahkan dan mudah untuk diperluas perintah *built-in* dengan perintah yang spesifik dengan aplikasi.

J. AWK Script

Skrip ini digunakan untuk mencari pola dalam teks file pada suatu baris tertentu. Apabila suatu baris cocok dengan sebuah pola, dilakukan sebuah aksi pada baris itu. AWK tetap memproses baris input sampai akhir input files terpenuhi. Skrip awk berbeda jika dibandingkan dengan bahasa pemrograman lain, karena skrip AWK berupa *data-driven*, artinya dapat mendeskripsikan data yang ingin dikerjakan kemudian mengolahnya. Bahasa pemrograman lain sifatnya prosedural, lebih susah untuk menentukan data yang akan diproses pada program. AWK diciptakan agar lebih mudah menulis maupun membaca program data.

III. METODE PENELITIAN

A. Tahapan Penelitian



Gambar 4. Diagram Alir Penelitian

Gambar 4 menjelaskan urutan Langkah yang dilakukan, dimulai suatu studi literatur dengan tujuan untuk menambah dasar penelitian serta wawasan penulis dalam melakukan penelitian ini. Studi literatur bersumber dari berbagai jurnal – jurnal baik yang tarakreditasi maupun yang tidak terakreditasi (jurnal nasional dan jurnal internasional) serta buku-buku yang berisi materi terkait penelitian ini.

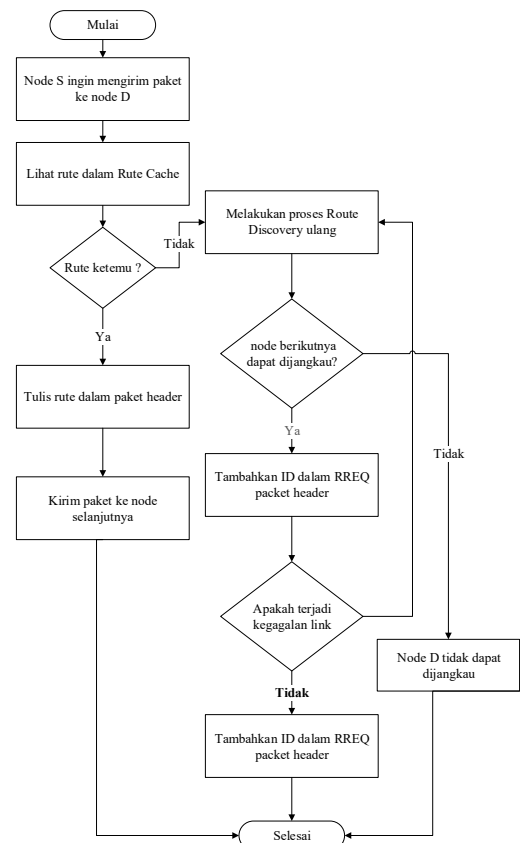
Perangkat keras yaitu spesifikasi komputer yang digunakan pada penelitian ini untuk melakukan proses simulasi jaringan yang diusulkan. Sedangkan perangkat lunaknya berupa aplikasi seperti network simulator (NS-2 versi 2.35), sistem operasi linux ubuntu versi 14.04 LTS, mesin virtual VirtualBox, dan Microsoft Excel.

Langkah selanjutnya dilakukan pembuatan skrip dari metode yang diusulkan ke dalam kerangka protokol routing DSR berdasarkan perhitungan *bandwidth*, *round trip time (RTT)*, dan *packet loss ratio*. Skrip dibangun menggunakan Bahasa pemrograman C++ pada modul protokol routing yang terdapat pada simulator NS-2.

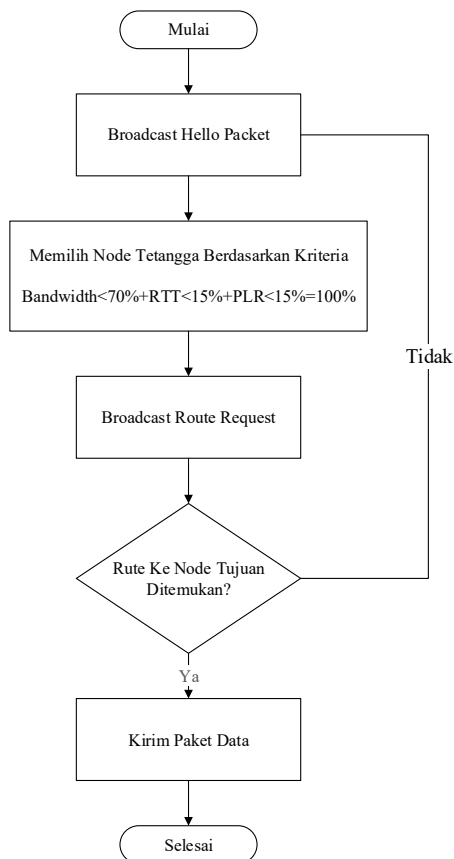
Langkah selanjutnya membuat skrip skenario simulasi menggunakan bahasa pemrograman TCL dimana file ini akan ber-ekstensi .tcl dengan menggunakan protokol DSR standar dan DSR. Hasil simulasi akan menghasilkan

luaran berupa *file trace* ber-ekstensi .tr. File trace ini akan difilter menggunakan bahasa pemrograman AWK untuk mendapatkan parameter uji yang diinginkan penulis seperti *throughput*, *packet delivery ratio (PDR)* dan *average end to end delay*. Nilai-nilai parameter uji tersebut akan dimasukkan dan diolah menggunakan aplikasi microsoft excel untuk divisualisasikan ke dalam bentuk grafik sehingga akan mudah dibaca dan dianalisis hasilnya. Dari hasil analisis akan diperoleh kesimpulan pada penelitian ini yang selanjutnya akan dituang ke dalam bentuk laporan.

Gambar 5 merupakan diagram alir dari cara kerja protokol routing DSR. Diawali dengan sebuah node S yang akan mengirim paket ke node D. Node S akan mengecek di table routingsnya apakah rute sudah ada atau belum. Jika rute sudah ada maka paket akan langsung dikirim ke node tujuan. Jika rute belum ada pada tabel routing, maka node S akan mencari rute dengan mengirim paket RREQ ke node tetangganya. Node tetangga yang juga berfungsi sebagai *node intermediate* akan meneruskan paket RREQ tersebut hingga sampai ke node D. Node D akan membalas dengan mengirimkan paket RREP ke node S yang mengindikasikan bahwa node D siap menerima paket melalui rute yang telah ditemukan. Proses transmisi paket akan dilakukan melalui rute yang telah dibentuk. Jika terjadi kegagalan rute atau putusnya rute, maka akan dikirimkan paket RERR ke node S dengan tujuan agar node S mengetahui bahwa rute tersebut sudah tidak dapat dilalui dan harus dilakukan pencarian rute kembali.



Gambar 5. Diagram Alir Dynamic Source Routing (DSR)

Gambar 6. Diagram Alir Algoritma Pemilihan *Node* Tetangga

Dari Gambar 6 dapat diketahui bahwa pada diagram ini dimulai dengan melakukan *Broadcast Hello Packet*, kemudian dilanjutkan dengan memilih *node* tetangga. Kualitas *node* tetangga yang dipilih berdasarkan kriteria yang merupakan perhitungan nilai dari *Bandwidth* tersedia yang kurang dari 70 % ditambah *Round Trip Time* kurang dari 15% ditambah *Packet Loss Ratio* kurang dari 15 % sehingga menghasilkan persentase 100 %. Kemudian dilanjutkan dengan *broadcast Route Request*. Jika rute menuju *node* tujuan ditemukan maka akan langsung mengirimkan paket data, namun jika belum ditemukan maka akan melakukan *Broadcast Hello Packet* kembali.

B. Lingkungan Simulasi Penelitian

Tahap ini merupakan lingkungan simulasi yang digunakan dan diterapkan pada protokol *routing* DSR standar dan protokol *Routing* DSR modifikasi. Simulasi akan dilakukan pada NS-2 dimana membutuhkan pola trafik, mobilitas *node* dan skrip simulasi. Pola trafik merupakan koneksi antara *node* sumber menuju *node* tujuan dan tipe koneksi yang digunakan yaitu *User Datagram Protocol* (UDP). Selain itu digunakan juga beberapa parameter dalam memilih *node* tetangga untuk menyusun rute pengiriman dari *node* sumber ke *node* tujuan.

B.1 Parameter Pemilihan *Node* Tetangga

Adapun beberapa parameter pemilihan *node* tetangga yang digunakan untuk memilih *node* tetangga yaitu:

- 1) *Bandwidth*: ukuran kapasitas maksimum pengiriman paket dari satu titik ke titik lain pada suatu jaringan. Satuan yang digunakan yaitu bit per second (bps) atau Byte per second (Bps) dimana untuk setiap nilai 1 Byte setara dengan 8 bit. Hal ini menggambarkan seberapa banyak bit yang dikirim setiap detik.
- 2) *Round Trip Time (RTT)*: Waktu yang dibutuhkan oleh *node* sumber untuk mengirim sebuah permintaan kepada *node* tujuan hingga *node* tujuan memberikan tanggapan kembali kepada *node* sumber.
- 3) *Packet Loss Ratio (PLR)*: Total paket yang hilang saat proses pengiriman paket dari *node* sumber ke *node* tujuan. Pengiriman paket dilakukan dalam suatu interval waktu tertentu saat transmisi. Keadaan ini terjadi ketika satu atau lebih paket yang dikirim tidak berhasil sampai ke tujuannya. Packet loss juga biasanya disebabkan oleh kesalahan dalam transmisi data saat melalui jaringan nirkabel, kemacetan jaringan, atau tabrakan lalu lintas data dalam sebuah jaringan.

B.2 Parameter skenario simulasi

Pada bagian ini akan dijelaskan mengenai parameter yang digunakan pada skenario simulasi. Skenario uji coba memakai NS-2 versi 2.35. Standar komunikasi data IEEE yang digunakan adalah 802.11. Jumlah *node* yang digunakan sebanyak 25, 50 dan 100. Simulasi jaringan diterapkan pada *network area* $500 \times 500 \text{ m}^2$ dan $1000 \times 1000 \text{ m}^2$ [17]. Simulasi jaringan dijalankan selama 600 detik. Kecepatan yang digunakan sebesar 1 m/s, 5 m/s, dan 10 m/s. Jenis mobilitas yang digunakan adalah *Random Waypoint* agar setiap *node* secara mandiri dan bebas akan memilih tujuan yang akan dicapai secara acak namun tetap dalam batasan jaringan yang telah ditentukan sebelumnya.

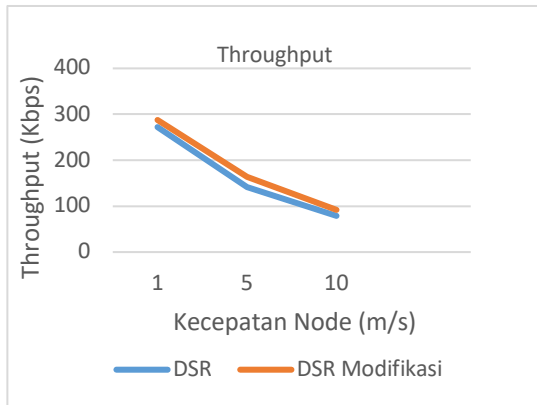
IV. HASIL DAN PEMBAHASAN

A. Hasil Simulasi dan Analisis Hasil Simulasi

Ujicoba dilakukan dengan menambahkan kriteria parameter yaitu 25 *node*, 50 *node*, 100 *node*, luas area $500 \times 500 \text{ m}^2$ dan $1000 \times 1000 \text{ m}^2$, kecepatan sebesar 1 m/s, 5 m/s dan 10 m/s.

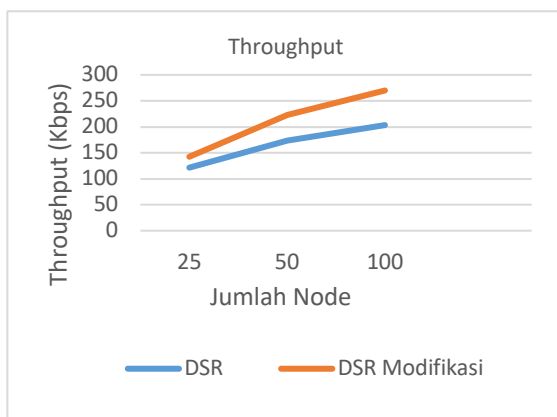
A.1. Pergerakan *Node* Terhadap *Throughput*

Ujicoba ke-1 untuk mendapatkan nilai *throughput*. Skenario uji coba dilakukan sebanyak tiga kali dengan kecepatan *node* di 1 m/s, 5 m/s, dan 10 m/s.



Gambar 7. Hasil uji coba *throughput*

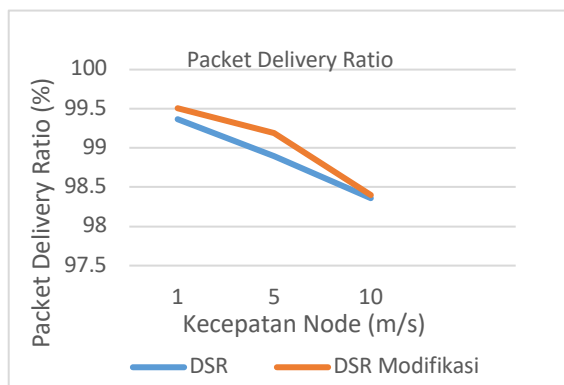
Setelah dilakukan uji coba sebanyak tiga kali kemudian didapatkan nilai rata-rata hasil uji coba *throughput* yang disajikan menggunakan grafik pada Gambar 8.



Gambar 8 Rata-rata *throughput*

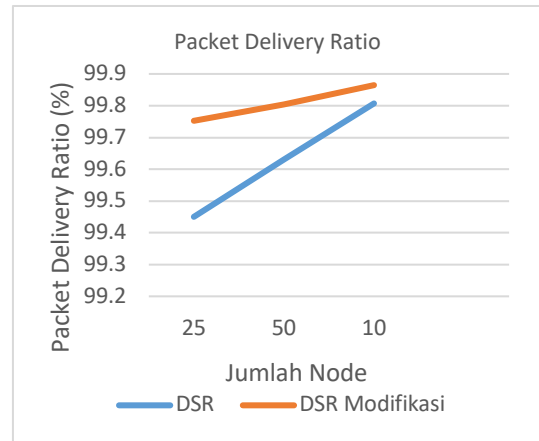
A.2. Pergerakan Node Terhadap Packet Delivery Ratio (PDR)

Ujicoba ke-2 untuk mendapatkan nilai *packet delivery ratio* (PDR). Skenario uji coba dilakukan sebanyak tiga kali dengan kecepatan *node* di 1 m/s, 5 m/s, dan 10 m/s yang dapat dilihat pada Gambar 9.



Gambar 9. Hasil uji coba *packet delivery ratio*

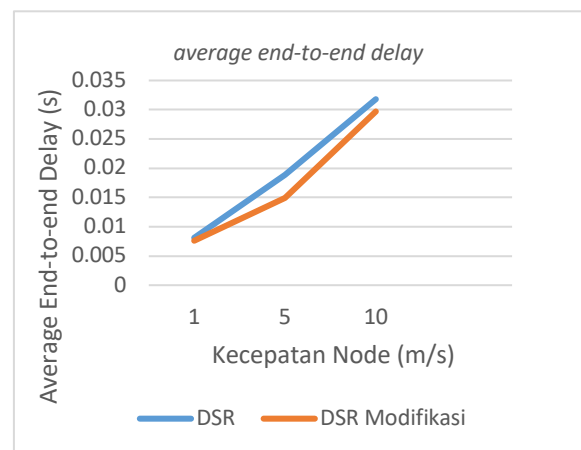
Setelah uji coba tiga kali kemudian didapatkan nilai rata-rata hasil uji coba *packet delivery ratio* yang disajikan menggunakan grafik pada Gambar 10.



Gambar 10. Rata-rata *packet delivery ratio*

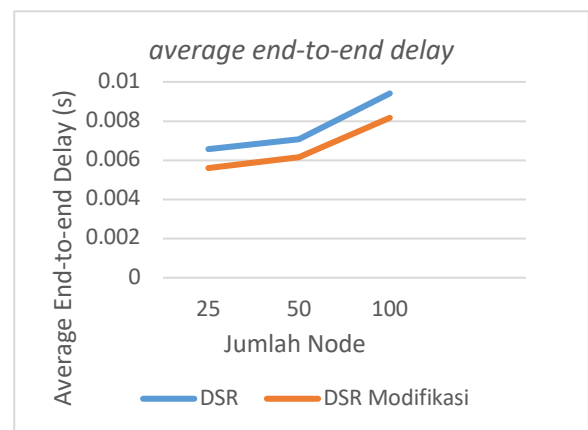
A.3. Pergerakan Node Terhadap Average End-to-End Delay

Ujicoba ke-3 untuk mendapatkan nilai *average end to end delay*. Skenario uji coba dilakukan sebanyak tiga kali dengan menggunakan variasi kecepatan *node* di 1 m/s, 5 m/s, dan 10 m/s yang dapat dilihat pada Gambar 11.



Gambar 11. Hasil uji coba *average end-to-end delay*

Setelah uji coba tiga kali kemudian didapatkan nilai rata-rata hasil uji coba *average end to end delay* yang disajikan menggunakan grafik pada Gambar 12.



Gambar 12. Rata-rata *average end-to-end delay*

A.4. Analisis Kinerja Protokol Routing DSR dan DSR Modifikasi

Setelah dilakukan ketiga percobaan pada protokol routing DSR dan DSR modifikasi, dapat diketahui bahwa terdapat beberapa faktor yang mempengaruhi nilai *throughput*, *packet delivery ratio* dan *average end-to-end delay* antara lain yaitu jika kecepatan *node* bertambah dan kepadatan bertambah maka nilai *throughput* dan *packet delivery ratio* mengalami penurunan namun nilai *average end-to-end delay* mengalami peningkatan. Hal tersebut disebabkan karena jika kecepatan bertambah maka posisi *node* akan sering mengalami perubahan dan menyebabkan rute pengiriman yang telah terbentuk sering mengalami putus hubungan sehingga menyebabkan paket data yang dikirim akan mudah hilang. Jika jumlah *node* bertambah maka nilai *throughput*, *packet delivery ratio* dan *average end-to-end delay* menunjukkan kenaikan. Hal tersebut disebabkan karena jika jumlah *node* bertambah maka jumlah *node* yang dilewati dalam pengiriman mengalami penambahan. Terjadi juga perbedaan nilai yang cukup signifikan untuk *throughput*, *packet delivery ratio* dan *average end-to-end delay* pada luas area 1000 x 1000 m² dibandingkan luas area 500 x 500 m², hal tersebut disebabkan karena jarak yang semakin luas menyebabkan jarak antar setiap *node* akan semakin renggang.

Kemudian berdasarkan percobaan yang telah dilakukan dapat diketahui juga bahwa protokol routing DSR modifikasi lebih baik dibandingkan protokol routing DSR. Peningkatan kinerja dapat diketahui setelah dilakukannya perbandingan perhitungan nilai berdasarkan *throughput*, *packet delivery ratio* dan *average end-to-end delay*. Untuk *throughput* peningkatan yang terjadi adalah sebesar 25 %, untuk *Packet Delivery Ratio* (PDR) terjadi peningkatan sebesar 1,11 % dan untuk *average end-to-end delay* terjadi penurunan sebesar 17,64 %.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil pengujian, dapat diambil beberapa kesimpulan yaitu :

- 1) Penerapan metode yang diusulkan mampu meningkatkan performa protokol DSR dibuktikan dengan hasil uji coba dimana nilai *throughput* meningkat sebesar 25 %, *Packet Delivery Ratio* (PDR) meningkat sebesar 1,11 % dan *average end-to-end delay* turun sebesar 17,64 %.
- 2) Semakin cepat *node* bergerak maka menyebabkan turunnya nilai *throughput* dan *packet delivery ratio* dan angkanya naik pada *average end-to-end delay*.
- 3) Jika jumlah *node* bertambah maka nilai *throughput*, *packet delivery ratio* dan *average end-to-end delay* akan mengalami peningkatan.

- 4) Perbedaan nilai yang signifikan terjadi pada *network area* 1000x1000 m² dibandingkan nilai pada *network area* 500x500 m² akibat jarak renggang karena luas bertambah.

B. Saran

Saran yang dapat digunakan untuk keperluan penelitian selanjutnya adalah :

- 1) Peningkatan nilai PDR kurang signifikan sehingga perlu adanya metode lain yang sejenis untuk diujicoba.
- 2) Melakukan penelitian selanjutnya dengan menggabungkan pemilihan *node* tetangga terbaik berdasarkan *Bandwidth*, *Round Trip Time*, dan *Packet Loss Ratio* pada protokol routing yang berbeda dengan metode lain untuk mencari nilai yang lebih optimal.
- 3) Melakukan penelitian selanjutnya dengan menerapkan jumlah *node* yang lebih bervariasi, luas area yang lebih bervariasi, dan parameter uji lainnya untuk mendapatkan hasil yang lebih valid terkait pengaruh kepadatan *node* dan pengaruh luas area.

DAFTAR PUSTAKA

- [1] A. Raich and A. Vidhate, "Best path finding using location aware AODV for MANET," *Int. J. Adv. Comput. Res.*, vol. 3, no. 11, pp. 336–340, 2013, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.406.349&rep=rep1&type=pdf>.
- [2] N. Fahriani, S. Djanali, and A. M. Shiddiqi, "Efisiensi Rute Pada Protokol Dynamic Source Routing Menggunakan Path Aware-Short," *Eksplora Inform.*, vol. 2, no. 1, pp. 37–48, 2012, [Online]. Available: <https://eksplora.stikom-bali.ac.id/index.php/eksplora/article/view/196/142>.
- [3] F. F. Laksono and I. Nurcahyani, "Simulasi Dan Analisis Perbandingan Kinerja Routing Protocol Ad Hoc on Demand Distance Vector (Aodv) Dan Dynamic Source Routing (Dsr) Saat Melakukan Data Broadcast Storm Pada Jaringan Manet.
- [4] B. S. Kusuma, "Analisis Perbandingan Performansi Protokol Aodv Dan Zrp Pada Mobile Adhoc Network," *Kinetik*, vol. 2, no. 3, pp. 15–17, 2017, doi: 10.22219/kinetik.v2i3.91.
- [5] F. Amilia, Marzuki, and Agustina, "Analisis Perbandingan Kinerja Protokol Dynamic Source Routing (DSR) Dan Geographic Routing Protocol (GRP) Pada Mobile Ad Hoc Network (MANET)," *J. Sains, Teknol. dan Ind.*, vol. 12, no. 1, pp. 9–15, 2014..
- [6] B. N. Hairani, A. Hidayat Jatmika, and F. Bimantoro, "Penerapan Algoritma EA-SHORT pada Protokol Routing AOMDV untuk Menemukan Rute yang Handal Berbasis Energi di Jaringan MANET," *J. Teknol. Informasi, Komputer, dan Apl. (JTika)*, vol. 3, no. 1, pp. 13–23, 2021, doi: 10.29303/jtika.v3i1.110..
- [7] H. Abdulwahid, B. Dai, B. Huang, and Z. Chen, "Scheduled-links multicast routing protocol in MANETs," *J. Netw. Comput. Appl.*, vol. 63, pp. 56–67, 2016, doi: 10.1016/j.jnca.2015.12.013.

- [8] I. Chlamtac, M. Conti, and J. J. N. Liu, "Mobile ad hoc networking: Imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003, doi: 10.1016/S1570-8705(03)00013-1.
- [9] M. A. Bahari, P. H. Trisnawan, and R. A. Siregar, "Analisis Kinerja Protokol AODV (Ad Hoc On-Demand Distance Vector) Dan AOMDV (Ad Hoc On-Demand Multipath Distance Vector) Terhadap Serangan Aktif Pada Jaringan MANET (Mobile Ad Hoc Network)," vol. 3, no. 4, pp. 3235–3244, 2018.
- [10] M. Mustaziri, "Sistem Pakar Fuzzy Untuk Optimasi Penggunaan Bandwidth Jaringan Komputer," *J. Sist. Inf. Bisnis*, vol. 2, no. 1, pp. 27–35, 2014, doi: 10.21456/vol2iss1pp027-035..
- [11] F. Syahrulah, A. Bhawiyuga, and M. Data, "Implementasi Sistem Pendeteksi Rogue Access Point Dengan Metode Perhitungan Nilai Round Trip Time," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 7367–7373, 2018.
- [12] M. S. Obaidat, P. Nicopolitidis, and F. Zarai, *Modeling and simulation of computer networks and systems: Methodologies and applications*. 2015.
- [13] P. Tcl and S. Pada, "Agency for Advanced Research Projects," pp. 1–11, 1995.
- [14] J. K. Ousterhout, "Tcl: An Embeddable Command Language," *USENIX Conf. Proc.*, no. May 1995, 1989..
- [15] G. N. U. Awk, *GNU Awk*. Free Software Foundation 51 Franklin Street, Fifth Floor Boston, MA 02110-1301 USA
- [16] I. Hartawan and W. Wibisono, "Mekanisme Pemilihan Mpr Dengan Congestion Detection Dalam Olsr Pada Manet," *J. Ilmu Komput.*, vol. 6, no. 2, pp. 11–17, 2013.
- [17] Y. P. Wulandari, A. H. Jatmika, and F. Bimantoro, "Meningkatkan Efisiensi Rute Pada Protokol Routing AOMDV Menggunakan Metode PA-SHORT di Jaringan MANET," *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 1, no. 1, pp. 77–85, 2019, doi: 10.29303/jtika.v1i1.11.