



PERBANDINGAN KINERJA PROTOKOL ROUTING HSR DAN DSDV PADA MOBILE AD-HOC NETWORK (MANET)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

KHALID HAMZAH HAZA AL HAMAMI
145150201111152



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020



PENGESAHAN

PERBANDINGAN KINERJA PROTOKOL ROUTING HSR DAN DSDV PADA MOBILE
AD-HOC NETWORK (MANET)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Khalid Hamzah Haza Al Hamami

NIM: 145150201111152

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Juli 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Ir. Primantara Hari Trisnawan, M.Sc
NIP: 19680912 199403 1 002

Dosen Pembimbing 2

Reza Andria Siregar, S.T, M.Kom
NIP: 19790621 200604 1 003

Mengetahui

Ketua Jurusan Teknik Informatika



Ichmad Basuki, S.T, M.MG, Ph.D
NIP: 19741118 200312 1 002



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 21 Juli 2020



Khalid Hamzah Haza Al Hamami

NIM: 145150201111152

KATA PENGANTAR

Puji syukur atas karunia Allah SWT yang Maha Pengasih lagi Maha Penyayang karena dengan rahmat dan karuniaNya, penulis dapat menyelesaikan skripsi yang berjudul ***"PERBANDINGAN KINERJA PROTOKOL ROUTING HSR DENGAN DSDV PADA MOBILE AD-HOC NETWORK (MANET)"*** sebagai salah satu syarat memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah memberi semangat, doa, bimbingan serta kritik dan nasehatnya. Dalam kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada :

1. Bapak Ir. Primantara Hari Trisnawan, M.Sc selaku dosen pembimbing I yang telah sabar dalam membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Reza Andria Siregar, S.T, M.Kom selaku dosen pembimbing II yang telah sabar dalam membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Agus Wahyu Widodo, S.T, M.Sc selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Segenap keluarga atas segala dukungan, nasehat, motivasi dan doa yang telah dipanjatkan untuk menyemangati penulis dalam menyelesaikan skripsi ini.
6. Teman – teman Fakultas Ilmu Komputer Universitas Brawijaya dan semua pihak yang belum disebutkan diatas, atas segala *support*, motivasi dan doa dalam menyelesaikan skripsi ini.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini, oleh karena itu, saran dan kritik yang membangun dari semua pihak sangat dibutuhkan demi kesempurnaan selanjutnya.

Malang, 21 Juli 2020

Penulis

khalidhamzah@student.ub.ac.id

ABSTRAK

Khalid Hamzah Haza A, Perbandingan Kinerja Protokol Routing HSR dan DSDV pada Mobile Ad-Hoc Network (MANET)

Pembimbing: Ir. Primantara Hari Trisnawan, M.Sc dan Reza Andria Siregar, S.T., M.Kom.

Mobile ad hoc Network (MANET) adalah jaringan node otonom portabel yang dilengkapi dengan antarmuka nirkabel, yang berkomunikasi melalui koneksi nirkabel tanpa administrasi terpusat. Jaringan ini dapat mengkonfigurasi sendiri (*self-configure*) dan memelihara topologi jaringan secara dinamis tanpa dukungan infrastruktur. Karena jangkauan transmisi mobile node terbatas, maka setiap node memiliki ketergantungan dengan node tetangganya untuk meneruskan paket. Analisis data kualitatif dilakukan pada kinerja protokol *Hierarchical State Routing (HSR)* dan *Destination-Sequenced Distance-Vector (DSDV)* dengan menafsirkan data dari hasil pengujian berupa *throughput*, *average jitter*, *average end to end delay*, *packet data delivery ratio*, *route overhead* dan *route convergence time*. Penelitian ini menggunakan Network Simulator 2.35 dengan skenario kepadatan node menggunakan variasi jumlah node, berjumlah 50 node hingga 120 node dengan kelipatan 10 node. Nilai terbaik parameter *throughput*, *average jitter*, *average end to end delay*, *packet data delivery ratio* dan *route overhead* berturut-turut adalah 189,312 Kbps, 27,18 ms, 2798,04 ms, 18,93 % dan 230,25. Hasil tersebut diperoleh pada protokol routing DSDV, sehingga dapat disimpulkan protokol routing DSDV memiliki kinerja yang lebih baik daripada HSR pada aspek kepadatan node. Nilai konvergensi yang lebih baik juga didapatkan oleh protokol DSDV dengan rata-rata waktu konvergensi 0,028617977 detik.

Kata kunci: MANET, routing, DSDV, HSR, Quality of Service, Network Simulator 2.35



ABSTRACT

Khalid Hamzah Haza A, *Performance Analysis of HSR and DSDV on Mobile Ad-Hoc Network (MANET)*

Supervisors: Ir. Primantara Hari Trisnawan, M.Sc dan Reza Andria Siregar, S.T., M.Kom.

Mobile ad hoc Network (MANET) is a portable autonomous node network equipped with a wireless interface, which communicates via a wireless connection without centralized administration. This network can self-configure and maintain network topology dynamically without infrastructure support. Because the mobile node's transmission range is limited, each node has a dependency on neighboring nodes to forward packets. Qualitative data analysis is performed on the performance of Hierarchical State Routing (HSR) and Destination-Sequenced Distance-Vector (DSDV) protocols by interpreting data from test results in the form of throughput, average jitter, average end to end delay, packet data delivery ratio, route overhead and route convergence time. This study uses Network Simulator 2.35 with node density scenarios using variations in the number of nodes, the amount of node is 50 nodes to 120 nodes with multiples of 10 nodes. The best value of parameters throughput, average jitter, average end to end delay, packet data delivery ratio and route overhead are 189.312 Kbps, 27.18 ms, 2798.04 ms, 18.93% and 230.25. These results were obtained in the DSDV routing protocol, so it can be concluded that the DSDV routing protocol has better performance than HSR on the node density aspect. A better convergence value is also obtained by the DSDV protocol with an average convergence time of 0.028617977 seconds.

Keywords: MANET, routing, DSDV, HSR, Quality of Service, Network Simulator 2.35



DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Rumusan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Batasan Masalah.....	3
1.7 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Kajian Pustaka.....	6
2.2 <i>Mobile Ad-Hoc Network (MANET)</i>	8
2.3 <i>Proactive Routing Protocol</i>	9
2.4 <i>Destination-Sequenced Distance-Vector (DSDV)</i>	9
2.3.1 Kelebihan DSDV.....	9
2.3.2 Kekurangan DSDV.....	10
2.3.3 Cara Kerja DSDV.....	10
2.5 <i>Hierarchical State Routing Protocol (HSR)</i>	10
3.3.3 Kelebihan HSR.....	11
3.3.3 Kekurangan HSR.....	11
3.3.3 Cara Kerja HSR.....	12
2.6 <i>Quality of Service (QoS)</i>	12
BAB 3 METODOLOGI PENELITIAN.....	15
3.1 Studi Literatur.....	16



3.2	Analisis Kebutuhan.....	16
3.2.1	Kebutuhan Fungsional	16
3.2.2	Kebutuhan Tambahan	16
3.2.3	Perangkat Keras	16
3.3	Perancangan Simulasi	17
3.3.3	Perancangan Parameter Lingkunan Simulasi	17
3.3.3	Perancangan Skenario Simulasi	17
3.3.3	Topologi Simulasi	18
3.4	Hasil dan Analisis.....	22
3.5	Kesimpulan dan Saran.....	22
BAB 4 IMPLEMENTASI		23
4.1	Implementasi	23
4.1.1	Instalasi <i>Network Simulator</i>	23
4.1.2	Membangun Simulasi.....	23
4.1.3	Konfigurasi lingkungan simulasi.....	24
4.2	Penarikan Data.....	25
BAB 5 HASIL DAN ANALISIS		27
5.1	Analisis Hasil.....	27
5.1.1	Pengujian <i>Average Throughput</i>	28
5.1.2	Pengujian <i>Average Jitter</i>	30
5.1.3	Pengujian <i>Average end to end delay</i>	32
5.1.4	Pengujian <i>Packet Data Delivery Ratio</i>	34
5.1.5	Pengujian Route Overhead.....	35
5.1.6	Pengujian <i>Route Convergence Time</i>	37
5.2	Perbandingan Hasil Analisis Pengujian Protokol DSDV dan HSR	39
BAB 6 PENUTUP		41
6.1	Kesimpulan.....	41
6.2	Saran	42
DAFTAR PUSTAKA		43
LAMPIRAN <i>SCRIPT AWK</i>		44
A.1	<i>Average Throughput</i>	44
A.2	<i>End to End Delay</i>	45
A.3	<i>Average Jitter</i>	46
A.4	<i>Packet Data Delivery Ratio</i>	47





DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian terkait.....	7
Tabel 3.1 Perancangan Parameter Lingkungan Simulasi.....	17
Tabel 5.1 Nilai average throughput.....	29
Tabel 5.2 Nilai average Jitter.....	30
Tabel 5.3 Nilai average end to end delay.....	32
Tabel 5.4 Nilai packet data delivery ratio.....	34
Tabel 5.5 Nilai route overhead.....	36
Tabel 5.6 Route DSDV.....	37
Tabel 5.7 Route HSR.....	38
Tabel 5.8 Perbandingan hasil analisis pengujian protokol.....	39



DAFTAR GAMBAR

Gambar 2.1 Contoh Tabel Routing DSDV.....	10
Gambar 2.2 Contoh Penalamatan HSR.....	12
Gambar 3.1 Flowchart Metode Penelitian.....	15
Gambar 3.2 Topologi Simulasi 50 Nodes.....	18
Gambar 3.3 Topologi Simulasi 60 Nodes.....	18
Gambar 3.4 Topologi Simulasi 70 Nodes.....	19
Gambar 3.5 Topologi Simulasi 80 Nodes.....	19
Gambar 3.6 Topologi Simulasi 90 Nodes.....	20
Gambar 3.7 Topologi Simulasi 100 Nodes.....	20
Gambar 3.8 Topologi Simulasi 110 Nodes.....	21
Gambar 3.9 Topologi Simulasi 120 Nodes.....	21
Gambar 3.10 Topologi Simulasi skenario konvergensi.....	22
Gambar 5.1 Grafik nilai average throughput.....	30
Gambar 5.2 Grafik nilai average jitter.....	31
Gambar 5.3 Grafik nilai average end to end delay.....	33
Gambar 5.4 Grafik nilai packet delivery ratio.....	35
Gambar 5.5 Grafik nilai route overhead.....	37



DAFTAR LAMPIRAN

LAMPIRAN SCRIPT AWK	44
A.1 Average Throughput	44
A.2 End to End Delay	45
A.3 Average Jitter	46
A.4 Packet Data Delivery Ratio	47
A.5 Route Overhead	48



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Mobile ad hoc Network (MANET) adalah jaringan node otonom portabel yang dilengkapi dengan antarmuka nirkabel, berkomunikasi melalui koneksi nirkabel tanpa administrasi terpusat. Jaringan MANET merupakan pendekatan di banyak tempat untuk membangun konektivitas untuk dipakai kapan saja dan dimana saja. Jaringan ini dapat mengkonfigurasi sendiri (*self-configure*) dan memelihara topologi jaringan secara dinamis tanpa dukungan infrastruktur (Sakshi, Goyal., 2017). Setiap perangkat di MANET bebas bergerak secara independen dan oleh karena itu node sering mengubah hubungannya dengan perangkat lainnya (Shruthi., 2017). Pada saat yang sama, node-node tersebut berfungsi sebagai router untuk mengirimkan data dari sumber ke tujuan mereka. Karena jangkauan transmisi mobile node terbatas, maka setiap node memiliki ketergantungan dengan node tetangganya untuk meneruskan paket. Ada banyak contoh pengaplikasian untuk MANET, seperti pada lapangan militer, operasi mencari dan menyelamatkan, atau setiap area geografis yang tidak memiliki base station untuk komunikasi. Jaringan MANET harus dapat memutuskan jalur terbaik antar node, meminimalkan waktu dan overhead bandwidth yang diperlukan untuk dapat mencakup setelah terjadi pergantian pada topologi. Untuk memenuhi persyaratan kinerja jaringan jenis ini, router harus bekerja secara efisien. Oleh karena itu, algoritma *routing* memainkan peran penting dalam desain arsitektur MANET (Bai dkk., 2017).

Ada tiga protokol *routing* utama yang digunakan dalam jaringan *ad-hoc mobile* (MANET) seperti *table driven* (Proactive), *On demand* (Reactive) dan *Hybrid protocol*. Tetapi menurut struktur jaringan, sebuah protokol *routing* dapat dibagi menjadi *flat routing*, *Hierarchical routing* dan *Geographic routing* (Singh dkk., 2017). Kinerja protokol *routing* bergantung pada lingkungan jaringan seperti kepadatan node, lalu lintas, mobilitas, waktu jeda, jenis lalu lintas, dll. Pada penelitian yang berjudul "MANET: Simulation Analysis and Performance Evaluation of *Routing Protocol* Using NS-3" yang dilakukan oleh Sakshi, Goyal., (2017) dengan dilakukannya perbandingan kinerja protokol *routing* dengan memvariasikan jumlah node didapatkan bahwa dalam kebanyakan hasil simulasi dari perbandingan kinerja protokol proaktif *Destination-Sequenced Distance-Vector* (DSDV), *Optimized Link State Routing* (OLSR) dan reaktif *Ad-hoc On demand Distance Vector* (AODV) pada MANET menunjukkan kinerja Protokol proaktif secara signifikan lebih baik daripada protokol *routing* reaktif. Protokol proaktif menunjukkan kinerja yang baik pada parameter *Quality of Service* (QoS) *Packet Delivery Ratio* (PDR) dan *end to end delay* sedangkan protokol reaktif menunjukkan kinerja yang baik dalam *routing load*. Didapatkan juga bahwa di bawah CBR (constant bit rate) *traffic*, protokol *routing* proaktif lebih unggul daripada protokol *routing on-demand* tetapi dengan *routing load* yang lebih tinggi. Lalu pada



pengukuran PDR dan *Throughput*, DSDV lebih unggul dari protocol lainnya. Sementara penelitian lain yang dilakukan oleh Kuber Singh, R. K. Singh (2017) yang membandingkan performansi protocol *Dynamic state routing* (DSR), *Fisheye State Routing* (FSR) dan *Hierarchical State Routing* (HSR) dalam sebuah scenario dengan 10 nodes didapatkan bahwa kinerja HSR Protocol lebih unggul dalam parameter QoS PDR dan diikuti dengan *average routing head* (ARH).

DSDV adalah table-driven proactive *routing* protocol untuk MANET yang bekerja dengan prinsip algoritma Bellman-Ford untuk mencari rute yang bebas dari loop atau pengulangan (Sureshbai dkk., 2018). DSDV merupakan protokol *routing distance vector hop-by-hop*. Setiap entri ke tabel *routing* berisi informasi tentang node *hop* berikutnya dan jumlah *hop* untuk mencapai tujuan. *Broadcast* berkala dalam pembaruan *routing* berupaya agar tabel *routing* benar-benar diperbarui setiap saat (Agarkhed., 2017).

HSR adalah multi level cluster-based hierarchical *routing* protocol. Mobile nodes dikelompokkan dalam sebuah cluster dan sebuah clusterhead kan dipilih untuk setiap cluster (Liu, Kaiser., 2005). HSR adalah *routing* protokol berbasis *hierarchical link state*. Pada protokol ini mempertahankan topologi hirarkis, di mana *cluster head* di level terendah menjadi anggota level yang lebih tinggi. Clustering diatur dalam 2 level: level fisik dan level logis. Partisi logis memainkan peran penting dalam manajemen mobilitas. Cluster node fisik membroadcast informasi link mereka satu sama lain. *Cluster head* merangkum informasi cluster dan mengirimkannya ke *cluster head* tetangga melalui *gateway* (Shruthi., 2017). Pada HSR, hierarchical address di berikan kepada semua node. Hierarchical address merefleksikan network topology dan memberikan informasi yang cukup untuk paket dikirimkan pada network. Pada HSR logical address merefleksikan property kelompok mobile node dan hierarchical address merefleksikan lokasi fisik dari mobile node. Menyatukan skema addressing tersebut dapat menambah tingkat adaptasi dari sebuah *routing* protocol (Liu, Kaiser., 2005).

Berdasarkan pernyataan dan perbedaan cara kerja protocol *routing* diatas maka pada penelitian ini kinerja dari dua protocol *routing* proaktif yaitu Protokol DSDV (*Destination-Sequenced Distance-Vector*) dan Protokol HSR (*Hierarchical State Routing*) akan dianalisis dengan parameter *throughput*, *route overhead*, *Average End-to-End Delay*, *packet data deliver ratio*, *packet jitter* dan *route convergence time*. Skenario yang dirancang pada simulasi ini adalah memvariasikan jumlah node dan menggunakan jenis traffic komunikasi data Constant Bit Rate (CBR). Pemilihan skenario ini ditujukan untuk mencari protocol proaktif mana yang lebih baik kinerjanya dalam beberapa scenario yang akan disimulasikan menggunakan NS-2 (*Network Simulator*) pada jaringan MANET dan simulasi lebih spesifik untuk membuktikan pada scenario manakah *flat routing* dan *hierarchical routing* lebih optimal untuk diimplementasikan. Lalu hasil dari penelitian ini dapat digunakan untuk bahan referensi dalam pemilihan protokol *routing* untuk skenario simulasi yang akan diajukan pada penelitian ini.



1.2 Identifikasi Masalah

Berikut merupakan masalah yang telah diidentifikasi berdasarkan latar belakang diatas.

Protocol proaktif mana yang lebih baik kinerjanya dalam beberapa skenario yang akan disimulasikan menggunakan NS 2.35 (*Network Simulator*) pada jaringan MANET dan simulasi lebih spesifik untuk membuktikan pada skenario manakah *flat routing* dan *hierarchical routing* lebih optimal untuk diimplementasikan.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah diatas, rumusan masalah yang digunakan pada penelitian ini adalah sebagai berikut:

1. Bagaimana cara implementasi protocol DSDV dan HSR pada jaringan MANET berdasarkan skenario variasi jumlah node yang diajukan pada NS-2?
2. Bagaimana dampak dari penambahan jumlah node dalam mempengaruhi nilai dari parameter uji MANET menggunakan protokol DSDV dan HSR?

1.4 Tujuan

Berdasarkan rumusan masalah diatas, beberapa tujuan dari penelitian ini adalah sebagai berikut:

1. Dapat mengetahui cara kerja *routing* DSDV dan HSR.
2. Dapat mengetahui pengaruh variasi jumlah node terhadap parameter uji pada MANET menggunakan protokol DSDV dan HSR.

1.5 Manfaat

Manfaat dari penelitian tugas akhir ini adalah sebagai berikut:

1. Memberikan referensi hasil analisis kinerja protocol *routing* DSDV dan HSR untuk di implementasi ke dalam MANET.
2. Memberikan referensi hasil analisis kinerja protocol *routing* DSDV dan HSR untuk pengembangan protocol *routing* proaktif pada MANET.

1.6 Batasan Masalah

Batasan permasalahan yang digunakan untuk memfokuskan rumusan masalah dalam pelaksanaan penelitian ini adalah sebagai berikut:



1. Analisis hanya dilakukan berdasarkan parameter QOS (*Quality of Service*) *throughput*, *route overhead*, *Average End-to-End Delay*, *packet data delivery ratio*, *packet jitter* dan *route convergence time*.
2. Simulasi dilakukan menggunakan simulator NS 2.35
3. Jenis traffic komunikasi yang digunakan adalah CBR.
4. Kelajuan minimum dari node adalah 1 m/s dan kelajuan maksimum node adalah 2 m/s
5. jumlah variasi node yang akan digunakan adalah 50, 60, 70, 80, 90, 100, 120
6. Simulasi MANET menggunakan standar jaringan wireless 802.11
7. Luas lingkungan percobaan yang digunakan adalah 1000 x 1000 meter²
8. Durasi simulasi adalah 1000 detik
9. Besaran paket data yang dikirim selama simulasi adalah 512 byte
10. Jumlah node yang berkomunikasi dalam simulasi adalah 1 pasang pengirim dan penerima data.

1.7 Sistematika Pembahasan

Gambaran agar dalam penelitian skripsi ini dituliskan dalam Sistematika Pembahasan antara lain:

Bab 1 Pendahuluan

Pada bagian Bab ini membahas tentang gambaran umum penelitian yang berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

Bab 2 Landasan Kepustakaan

Bagian ini membahas tentang konsep *Mobile ad-hoc Network*, *Proactive routing protocol*, *Destination-Sequenced Distance-Vector Routing Protocol*, dan *Hierarchical State Routing Protocol*.

Bab 3 Metodologi Penelitian

Bagian ini berisi prosedur dalam mensimulasikan DSDV dan HSR pada NS-2 untuk dianalisis pada bab berikutnya. Pada bagian ini juga dibahas daftar kebutuhan perangkat lunak dan perangkat keras yang dibutuhkan untuk melakukan penelitian dan perancangan lingkungan simulasi dan perancangan skenario lingkungan simulasi pada sub bab perancangan.

Bab 4 Implementasi

Bab ini menjelaskan tentang detail langkah langkah implementasi yang dilakukan pada penelitian dalam membangun simulasi.



Bab 5 Hasil dan Analisis

Bagian ini menunjukkan hasil dari simulasi yang dilakukan serta analisis yang dilakukan berdasarkan data yang didapat dari simulasi berdasarkan skenario.

Bab 6 Penutup

Bagian ini membahas kesimpulan berdasarkan hasil penelitian “analisis perbandingan kinerja protokol *routing* dsdv dan hsr pada *mobile ad-hoc network* (manet)” dan saran dalam pengembangan simulasi selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada bab ini membahas kajian pustaka serta teori-teori pendukung terhadap penelitian yang dilakukan. Dan juga membahas penelitian sebelumnya yang digunakan sebagai dasar dalam membandingkan protokol *proactive* dengan perbandingan protokol yang digunakan pada penelitian ini.

Penelitian dengan judul “A COMPARISON ANALYSIS OF DSDV AND AODV ROUTING PROTOCOLS IN MOBILE AD-HOC NETWORKS” merupakan salah satu penelitian yang digunakan sebagai landasan pustaka. Penelitian tersebut menjelaskan tentang perbandingan kinerja antara protokol DSDV dan AODV menggunakan *Simulator* NS2 dengan metode *wired-cum-wireless*. Skenario yang digunakan adalah skenario campuran antara *wireless* dan *wired* domain. Percobaan melibatkan 20 *web service* dengan mendukung 10 metode pemanggilan web yang berbeda yang menyebabkan 500 request berbeda untuk mengetes kinerja protokol DSDV dan AODV. Kesimpulan yang didapatkan menyatakan bahwa dalam delay pengiriman paket protokol AODV lebih menonjol selanjutnya dalam pengiriman paket data DSDV lebih menonjol dibandingkan AODV. protokol AODV performanya akan terus menerus meningkat jika kepadatan node yang disimulasikan bisa lebih padat lagi dan kecepatan yang lebih cepat (Sureshkumar dkk., 2017).

Selanjutnya merupakan penelitian dengan judul “Simulation & Performance Comparison between DSR,FSR & HSR Routing Protocols in MANET Using NS2” yang membandingkan kinerja protokol HSR dengan protokol DSR dan FSR menggunakan NS2 dengan konfigurasi (1) Luas area 500 x 500 meter; (2) Paket data CBR; (3) Waktu simulasi 500 second; (4) Jumlah node 10. Lalu menggunakan Parameter uji *Average Overhead*, *packet delivery ratio*, *throughput* dan *Path optimality*. Kesimpulan yang didapatkan menyatakan bahwa protokol FSR memiliki kinerja yang lebih baik dalam parameter throughput daripada protokol DSR dan HSR dan rasio pengiriman paket protokol HSR adalah yang terbaik dalam hal *average routing head*. (Kuber Singh dkk., 2017).

Dari dua penelitian yang dilakukan sebelumnya, peneliti bermaksud untuk membandingkan protokol DSDV dan HSR. Kedua penelitian sebelumnya menggunakan *Simulator* yang sama dengan yang digunakan pada penelitian ini tetapi scenario dan parameter konfigurasi yang digunakan berbeda. Dan juga kombinasi perbandingan protokol routing yang digunakan berbeda. Pada tabel 2.1 dijelaskan tentang perbandingan penelitian penelitian terdahulu dengan penelitian yang akan dilakukan.



Tabel 2.1 Perbandingan penelitian terkait

No	Nama peneliti, tahun, judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana penelitian
1	A.Sureshkumar, V.Ellappan, K. Manivel, 2017, <i>A COMPARISON ANALYSIS OF DSDV AND AODV ROUTING PROTOCOLS IN MOBILE AD HOC NETWORKS.</i>	Menganalisis kinerja protokol DSDV menggunakan NS2.	Membandingkan kinerja protokol DSDV dengan protokol AODV pada simulator NS2 menggunakan <i>Wired-Cum-Wireless method.</i> Skenario yang digunakan adalah skenario campuran antara <i>wireless</i> dan <i>wired</i> domain. Percobaan melibatkan 20 <i>web service</i> dengan mendukung 10 metode pemanggilan <i>web</i> yang berbeda yang menyebabkan 500 request berbeda.	Membandingkan kinerja protokol DSDV dengan HSR menggunakan simulator NS-2.35 dan parameter uji <i>throughput, route overhead, Average End-to-End Delay, packet data delivery ratio, packet jitter dan route convergence time.</i> Penelitian ini menggunakan Skenario variasi jumlah <i>node</i> dengan dengan node berjumlah 50, 60, 70, 80, 90, 100, 120.
2	Kuber Singh, R. K. Singh, Sanjeev Singh Bisho, Monika Barowal, 2017, <i>Simulation & Performance</i>	Menganalisis kinerja protokol HSR menggunakan NS2.	Membandingkan kinerja protokol HSR dengan DSR dan FSR menggunakan NS2 dengan	Membandingkan kinerja protokol DSDV dengan HSR menggunakan simulator NS-2.35 dan parameter uji



Comparison between DSR,FSR & HSR Routing Protocols in MANET Using NS2.	konfigurasi (1) Luas area 500 x 500 meter; (2) Paket data CBR; (3) Waktu simulasi 500 second; (4) Jumlah node 10. Lalu menggunakan Parameter uji Average Overhead, packet delivery ratio, throughput dan Path optimality.	(1) throughput, route overhead, Average End-to-End Delay, packet data delivery ratio, packet jitter dan route convergence time. Penelitian ini menggunakan Skenario variasi jumlah node dengan dengan node berjumlah 50, 60, 70, 80, 90, 100, 120 dan mobility random way point.
--	---	--

2.2 Mobile Ad-Hoc Network (MANET)

Kumpulan node mobile membentuk jaringan *ad-hoc mobile multi-hop* mandiri tanpa menggunakan infrastruktur tetap. Transmisi data dilakukan secara tunggal atau *multi-hop*, karena rentang propagasi radio yang terbatas. Perubahan yang sering dan tak terduga dalam mobilitas node menurunkan stabilitas hubungan nirkabel. Beberapa node perantara bekerja sama dan membantu MANET dalam membangun komunikasi antar node yang jauh jaraknya melalui beberapa *hop*. Protokol *routing* harus didistribusikan dengan baik untuk melakukan *forwarding* data antara node sumber dan node destinasi (Natarajan, Mahadevan., 2017). Node *mobile* memiliki batasan *bandwidth*, topologi dinamis, operasi yang terbatas Energi, infrastruktur variabel dan kapasitas link (Lalar, Yadav., 2017).

Node dalam jaringan tidak hanya bertindak sebagai host tetapi juga sebagai router yang menemukan dan memelihara rute ke node lain di jaringan. Pembuatan jaringan yang cepat dan mudah membuat MANET layak untuk digunakan di militer, pemulihan daerah bencana dan di lingkungan lain dimana tidak ada infrastruktur atau telah hancur. Karena node *mobile* bergerak dalam berbagai arah sehingga menyebabkan link yang ada rusak dan menentukan rute baru, perutean di jaringan semacam ini merupakan salah satu tantangan dalam penerapan MANET. Mobilitas (bagaimana node bergerak) dari node mobile memainkan peran penting pada kinerja protokol *routing*. Rute antara dua node



yang saling berkomunikasi dapat terdiri dari beberapa *hop* melalui node lain dalam jaringan. (Gulati, Kumar., 2014).

2.3 Proactive Routing Protocol

Dalam protokol *routing* proaktif, setiap node memelihara dan memperbarui satu atau beberapa tabel *routing node* tetanganya. Informasi dalam tabel ini diperbarui secara konsisten sehingga informasi *routing* tetap konsisten dan diperbarui seiring perubahan status jaringan (Bai dkk., 2017). Protokol ini memelihara informasi *routing* dari setiap node *mobile* dan menyiarkannya ke jaringan. Oleh karena itu protokol disebut sebagai *tabel-driven*. Informasi tabel *routing* dapat diubah secara berkala ketika perubahan topologi jaringan terjadi dengan meninjau perubahan pada node tetangga. Meningkatnya jumlah node meningkatkan ukuran jaringan juga, dan hal ini overhead dapat dengan mudah terjadi (Shenbagapriya, Kumar., 2017).

2.4 Destination-Sequenced Distance-Vector (DSDV)

DSDV disebutkan sebagai algoritma *routing* yang bekerja dengan baik dalam bertukar pesan untuk rekonstruksi rute yang tidak valid di MANET yang memiliki mobilitas node yang tinggi. Protokol ini memiliki kinerja yang baik saat node yang digunakan lebih sedikit. DSDV merupakan protokol *routing distance vector hop-by-hop*. Setiap entri ke tabel *routing* berisi informasi tentang node *hop* berikutnya dan jumlah *hop* untuk mencapai tujuan. *Broadcast* berkala dalam pembaruan *routing* berupaya agar tabel *routing* benar-benar diperbarui setiap saat (Agarkhed., 2017). DSDV menggunakan *sequence number* untuk menghindari pengulangan dan mempertahankan kesegaran rute. *Sequence number* untuk rute ditetapkan oleh node tujuan dan ditingkatkan satu untuk setiap *advertisement* rute asli yang baru. pembaruan tabel *routing* dapat dikirim dengan dua cara: "*full dump*" atau "*incremental update*". Untuk *full dump* mengirimkan tabel semua informasi *routing* yang ada ke tetangga dan bisa berisi banyak paket sementara dalam *incremental update* hanya entri dari tabel *routing* yang memiliki perubahan drastis sejak update terakhir yang dikirim (Shruthi., 2017). Menurut penelitian yang berjudul "A COMPARISON ANALYSIS OF DSDV AND AODV ROUTING PROTOCOLS IN MOBILE AD HOC NETWORKS" yang dilakukan oleh Sureshkumar, A., Ellappan, V. dan Manivel, K. (2017) didapatkan beberapa kelebihan dan kelemahan DSDV antara lain:

2.3.1 Kelebihan DSDV

- Lebih cocok untuk penerapan *ad hoc network* dengan skala kecil
- Menyelesaikan masalah *routing loop*
- Mengurangi kemungkinan *count to infinity*
- Hanya mempertahankan jalur terbaik dari pada *multiple path* untuk sebuah *destination*

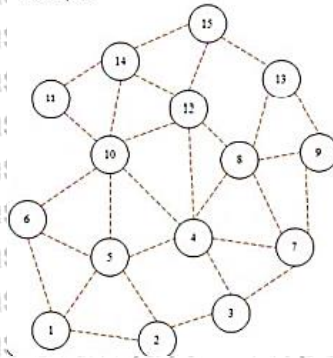


2.3.2 Kekurangan DSDV

- Penggunaan *battery power* yang besar karena intensitas update route yang tinggi
- Penggunaan *bandwidth* bahkan saat *network idle*
- Tidak cocok untuk *dynamic network*

2.3.3 Cara Kerja DSDV

Example:



Routing table for Node 1

Dest	NextNode	Dist	seqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	6	3	176
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

Gambar 2.1 Contoh Tabel *Routing* DSDV

Gambar 2.1 menunjukkan tabel *routing* untuk node 1 menggunakan protokol DSDV. Entri pertama dalam tabel adalah untuk Node 2. Bila node 2 adalah node tujuan, kolom *Dest* diisi dengan angka 2, dan kolom *Next Node* juga diisi dengan angka 2, karena node 2 terhubung langsung ke node 1. Kolom *Dist* memberikan informasi tentang jarak antar node. Kolom ini diisi dengan nomor 1 sejak mereka terhubung langsung seperti yang disebutkan sebelumnya. Dan *sequence number* yang sesuai dengan pesan yang akan dikirim selanjutnya adalah 22 seperti yang ditugaskan oleh node 2.

2.5 Hierarchical State Routing Protocol (HSR)

Hierarchical State Routing (HSR) Protocol adalah *routing* protokol berbasis *hierarchical link state*. Pada protokol ini mempertahankan topologi hirarkis, di mana *cluster head* di level terendah menjadi anggota level yang lebih tinggi. Tujuan clustering adalah pemanfaatan yang efisien dari radio *channel resource* dan mengurangi *overhead routing* jaringan. Clustering diatur dalam 2 level: level fisik dan level logis. cluster fisik memiliki hubungan *wireless one-hop* di antara mereka dan didasarkan pada hubungan geografis antar node. Cluster logis didasarkan pada hubungan logis tertentu dimana partisi bergantung pada afinitas logis atau fungsional antar node. Partisi logis memainkan peran penting dalam manajemen mobilitas. Cluster node fisik membroadcast informasi link mereka satu sama lain. *Cluster head* merangkum informasi cluster dan mengirimkannya ke *cluster head* tetangga melalui *gateway* (Shruthi., 2017).



Sebuah node di setiap level flooding informasi yang diperolehnya ke level yang lebih rendah setelah algoritma dijalankan pada level itu. Jadi level bawah memiliki informasi topologi hirarkis. Setiap node memiliki alamat hirarkis. Salah satu cara untuk menetapkan alamat hirarkis adalah dengan menggabungkan angka cluster yang ada di jalan dari root ke node. Sebuah gateway dapat dicapai dari root melalui banyak jalur sehingga gateway dapat memiliki beberapa alamat hirarkis. Alamat hierarkis sudah cukup untuk memastikan pengiriman dari manapun di jaringan ke host. Setiap node memiliki identifier unik yang disebut Node ID. Node ID tidak lain adalah alamat perangkat keras fisik (yaitu *MAC address*). Dalam HSR, *HID (Hierarchical ID)* dari sebuah node digunakan sebagai *sequence MAC address* untuk node dari hirarki atas ke node itu sendiri. Setiap sub-jaringan dilengkapi dengan *Location Management Server (LMS)*. Semua node yang termasuk dalam sub-jaringan tertentu mendaftarkan alamat logis mereka dengan LMS masing-masing. LMS *broadcast* alamat hierarkis mereka ke tingkat atas dan informasinya dikirim ke LMS lain juga. *Transport layer* mengirimkan paket data ke *network layer* beserta alamat logis yang berkaitan dengan tujuan. *Network layer* menetapkan alamat hierarkis LMS tujuan dari semua LMS lain yang ada dan meneruskan paket ke dalamnya. Begitu source dan destination mengetahui alamat hierarki masing-masing, mereka dapat dengan mudah melewati LMS saat mengirim atau menerima paket data (Shruthi., 2017).

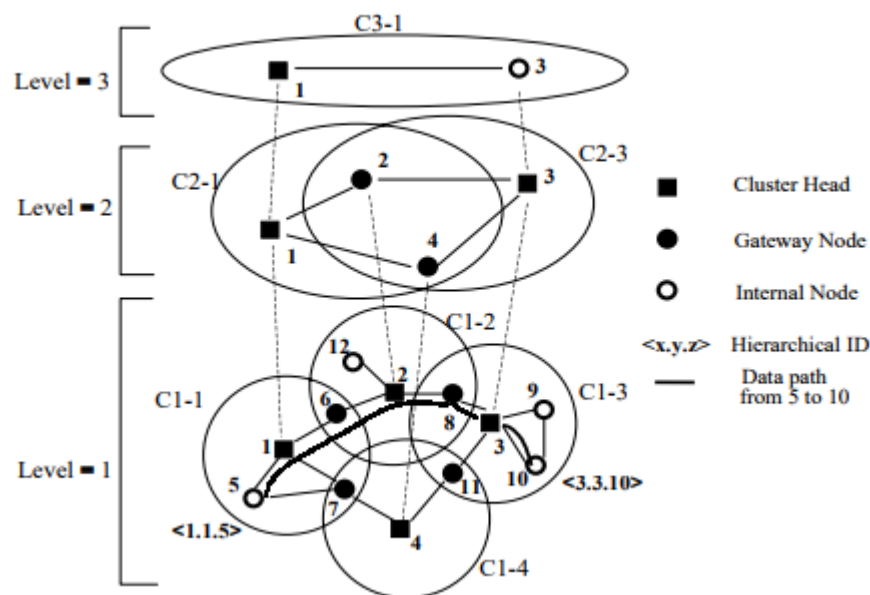
3.3.3 Kelebihan HSR

Mengurangi ukuran *routing table* karena menggunakan informasi hirarkis

3.3.3 Kekurangan HSR

Proses pertukaran informasi yang melibatkan semua tingkat hirarki serta proses pemilihan pemimpin di setiap cluster membuatnya cukup bermasalah untuk jaringan *ad hoc*.

3.3.3 Cara Kerja HSR



Gambar 2.2 Contoh Pengalamatan HSR

Gambar 2.2 merupakan ilustrasi dari contoh Pengalamatan *hierarchical routing*. Alamat hirarkis dari node 12, HID (12), adalah <3:2:12>. Dalam contoh ini, node 3 adalah anggota cluster hirarkis teratas (level 3). Ini juga merupakan cluster head C2-3. Node 2 adalah anggota C2-3 dan merupakan cluster head C1-2. Node 12 adalah anggota C1-2 dan dapat dicapai langsung dari node 2. Alamat HID dari node 12 adalah rangkaian alamat MAC dari node di sepanjang jalur untuk mencapai simpul 12 dari hierarki atas. Dengan kata lain, node 3 adalah representasi hierarki atas dari simpul 12, simpul 2 adalah representasi hierarki tingkat berikutnya dari simpul 12.

pengiriman paket dari node 5 ke node 10. HID (5) = <1:1:5> dan HID (10) = <3:3:10>. Paket diteruskan ke atas menuju hirarki atas oleh node 5 (ke node 1). Node 1 mengirimkan paket ke node 3, yang merupakan node hierarki teratas untuk tujuan 10. Node 1 memiliki "virtual link", yaitu tunnel, ke node 3, yaitu jalur (1,6,2,8, 3). Dengan demikian, paket ke node 3 dikirimkan di sepanjang jalur ini. Akhirnya, node 3 mengirimkan paket ke node 10 di sepanjang jalur hierarki ke bawah, yang dalam hal ini mengurangi satu hop tunggal. Keuntungan dari skema alamat hirarkis ini adalah bahwa setiap node dapat secara dinamis dan lokal memperbarui HID sendiri saat menerima update *routing* dari node-node yang lebih tinggi dalam hirarki dan tidak ada kontrol pusat yang diperlukan.

2.6 Quality of Service (QoS)

Quality of Service (QoS) mengacu pada seperangkat persyaratan layanan yang harus dipenuhi oleh jaringan saat mengirimkan aliran paket dari node sumber menuju node destinasi. Dalam kata lain QoS merupakan level layanan yang dapat diukur dan dikirimkan kepada pengguna jaringan, yang dapat dikarakterisasi oleh



packet loss, available bandwidth, end to end delay, dll (Kasthuri, Sivakumar, Chandrasekar., 2017).

Penggunaan rekomendasi analisis QoS ini dapat dipakai dalam memperbaiki layanan-layanan jaringan. Pada penelitian ini menggunakan padameter QoS *throughput, route overhead, average end to end delay, packet data delivery ratio, packet jitter* dan *route convergence time*.

Pengukuran kinerja protokol *routing* dilakukan menggunakan parameter sebagai berikut:

- a. **Throughput** merupakan jumlah pengiriman paket yang sebenarnya melalui jaringan menggunakan media transmisi per satuan waktu. *Throughput* pada sistem komunikasi dapat dipengaruhi oleh banyak faktor, mulai dari teknologi fisik yang digunakan maupun dari kekuatan pemrosesan yang ada pada komponen sistem. *Average throughput* dapat diperoleh dari menghitung jumlah ukuran paket data yang diterima oleh *node* tujuan selama durasi waktu pengiriman data pada proses pengamatan dibagi dengan lama waktu pengiriman paket data tersebut.

$$\text{Average Throughput} = \frac{\text{Total jumlah data diterima (bit)}}{\text{Lama pengiriman data (second)}}$$

Dimana jumlah ukuran data diterima merupakan jumlah ukuran paket data yang berhasil sampai dan diterima oleh *node* tujuan dengan satuan *bit* (1 Byte = 8 bit). Sedangkan durasi pengiriman data adalah lama waktu yang ditempuh untuk mengirimkan paket data dari *node* sumber menuju *node* tujuan dengan satuan detik atau *second*.

- b. **Route Overhead** adalah Jumlah total *routing (route request, route replay, route error)* paket yang dikirimkan selama simulasi tetapi tidak termasuk MAC, dan paket ARP.
- c. **Average End-to-End Delay** adalah rata-rata waktu yang dibutuhkan oleh paket untuk mencapai tujuan melalui jaringan. Satuan untuk menghitung *average end to end delay* adalah *millisecond (ms)*. *Delay* dapat terjadi karena adanya antrian paket yang akan dikirim, atau menghindari kemacetan lalu lintas jaringan dengan mengambil jalur/rute lain. *Average end to end delay* dapat diperoleh dengan menghitung total selisih antara waktu paket dikirim dengan waktu paket diterima dari paket yang sampai pada *node* tujuan dibagi dengan banyaknya jumlah paket yang diterima atau paket yang sampai pada *node* tujuan.

$$\text{Average end to end delay} = \frac{\text{Total delay (ms)}}{\text{Total paket diterima}}$$

Dimana total *delay* merupakan jumlah dari *delay* yang terjadi selama pengiriman paket. Sedangkan jumlah paket diterima adalah banyaknya paket yang telah sampai pada *node* tujuan.



- d. **Packet data delivery ratio** adalah rasio perbandingan total paket yang diterima oleh node tujuan dengan total paket yang dikirim oleh node sumber.

$$\text{Packet delivery ratio (PDR)} = \frac{\text{Total paket yang diterima}}{\text{Total paket yang dikirim}} \times 100\%$$

Dimana total paket yang diterima didapatkan dari jumlah total paket keseluruhan yang berhasil diterima oleh node tujuan selama pengiriman paket. Sedangkan total paket yang dikirim didapatkan dari jumlah total paket keseluruhan yang berhasil dikirimkan oleh node sumber menuju node tujuan selama pengiriman paket.

- e. **Packet Jitter** adalah variasi perbedaan waktu delay paket yang dikirim karena adanya *congestion* atau traffic dalam jaringan. *Average jitter* dapat diakibatkan oleh variasi-variasi panjang antrian dalam waktu pengolahan data dan juga penghimpunan ulang beberapa paket setelah proses pengiriman paket selesai serta peningkatan trafik yang terjadi secara tiba-tiba sehingga menimbulkan *congestion* dan penyempitan *bandwidth*. *Average jitter* dapat diperoleh dengan menghitung total variasi *delay* selama paket dikirim dibagi dengan banyaknya jumlah paket yang diterima atau paket yang sampai pada node tujuan.

$$\text{Average jitter} = \frac{\text{Total variasi delay (ms)}}{\text{Jumlah paket diterima} - 1}$$

Dimana total variasi *delay* didapatkan dari jumlah selisih antar *delay* yang terjadi selama pengiriman paket. Sedangkan jumlah paket diterima adalah banyaknya paket yang telah sampai pada node tujuan.

- f. **Route Convergence Time** adalah ukuran seberapa cepat sekelompok router mencapai keadaan konvergensi (keadaan dimana satu set router dapat memiliki informasi topologi yang sama mengenai internetwork tempat router router tersebut beroperasi). Konvergensi bertujuan untuk melengkapi tabel *routing* dari masing-masing *router* yang berperan dalam proses pengiriman paket. Konvergensi terjadi mulai dari awal pembentukan jalur atau ketika perubahan topologi jaringan terjadi.



BAB 3 METODOLOGI PENELITIAN

Pada bab ini dijelaskan hal-hal mengenai prosedur yang akan dilakukan penelitian ini bertipe analitis dimana akan dibahas tentang perbandingan kinerja protocol *routing* proaktif DSDV dan HSR pada jaringan MANET. Pada gambar 3.1 menunjukan urutan prosedur yang perlu dilakukan dalam melakukan penelitian ini.



Gambar 3.1 Flowchart Metode Penelitian



3.1 Studi Literatur

Menelaah lebih dalam berkaitan dengan kinerja *routing* protocol proaktif dari berbagai sumber literatur, hal yang ditelaah diantaranya adalah:

- a. MANET (Mobile Ad-hoc Network)
- b. Simulasi DSDV
- c. Simulasi HSR

Buku, jurnal, dan sumber-sumber penelitian lainnya merupakan beberapa contoh literatur yang dimanfaatkan pada penelitian ini.

3.2 Analisis Kebutuhan

Kebutuhan baik dari segi fungsional maupun kebutuhan tambahan yang diperlukan ketika penelitian dilakukan.

3.2.1 Kebutuhan Fungsional

Simulasi yang dirancang dianggap berhasil jika dapat menjalankan fungsi fungsi sebagai berikut:

- a. Network Simulator 2.35
- b. Network Animator (NAM)
- c. Portocol *Destination-Sequenced Distance-Vector* (DSDV)
- d. Protocol *Hierarchical State Routing* (HSR)

3.2.2 Kebutuhan Tambahan

Dibawah ini adalah beberapa fungsi-fungsi tambahan yang dapat ditambahkan untuk memperlancar jalannya simulasi:

- a. Sistem operasi Linux
- b. gawk
- c. Xgraph

3.2.3 Perangkat Keras

Sebuah laptop digunakan untuk melakukan penelitian ini dengan uraian sebagai berikut:

- a. Processor : Intel(R) Core(TM) i7 CPU L640 @ 2.13GHz (4 CPUs), ~2.1GHz
- b. Memory : 4096MB RAM
- c. VGA : Intel(R) HD Graphics 1696 MB
- d. HDD : 250 GB Serial ATA, 500 GB Serial ATA



3.3 Perancangan Simulasi

Perancangan simulasi dimulai dari perancangan parameter lingkungan simulasi dan perancangan skenario simulasi. Perancangan parameter lingkungan simulasi dibuat dengan menentukan parameter-parameter lingkungan simulasi berupa luas area simulasi, jumlah node, waktu simulasi, jenis traffic yang digunakan, jenis mobility model yang digunakan dan protocol yang akan disimulasikan. perancangan skenario simulasi dilakukan dengan menentukan kasus kasus atau rangkaian kejadian yang akan disimulasikan beserta parameter kinerja protocol *routing* yang akan dibandingkan.

3.3.3 Perancangan Parameter Lingkungan Simulasi

Batasan-batasan yang akan digunakan dalam pembuatan lingkungan simulasi *routing* DSDV dan HSR dalam jaringan MANET akan dirancang pada bagian ini. Pada tabel 3.1 dibawah ini dapat dilihat skema perancangan lingkungan simulasi.

Tabel 3.1 Perancangan Parameter Lingkungan Simulasi

Jumlah mobile node	50, 60, 70, 80, 90, 100, 110, 120
Tipe Antena	<i>Omni-directional</i> (RF)
Tipe MAC	802.11
Tipe Propagasi	<i>Two Ray Ground</i>
Ukuran Paket	512 byte
Jenis Traffic	UDP (CBR) 1500 byte
Waktu Simulasi	1000 detik
Luas Area Simulasi	1000 x 1000 meter ²
Kecepatan Pergerakan Node	Acak, Min 1 m/s, Maks 2 m/s
Tipe Mobilitas	<i>Random Waypoint</i>
Transmission Rate	1 Mbps

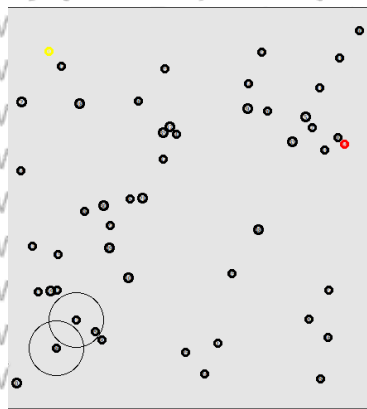
3.3.3 Perancangan Skenario Simulasi

Skenario simulasi MANET pada penelitian ini dilakukan dengan dua jenis protocol proaktif yang berbeda, yaitu DSDV (Protokol Proaktif Flat) dan HSR (Protokol Proaktif Hierarchical). Satu koneksi UDP dengan packet rate 1500 byte dan model traffic CBR digunakan pada perancangan ini. Variasi jumlah node yang berbeda akan disimulasikan pada setiap skenario pengujian, yakni 50, 75, 100, 125, 150, 175 dan 200 dengan masing-masing berdurasi 1000 detik. 512 byte merupakan besar paket yang dikirimkan. Node akan langsung bergerak secara



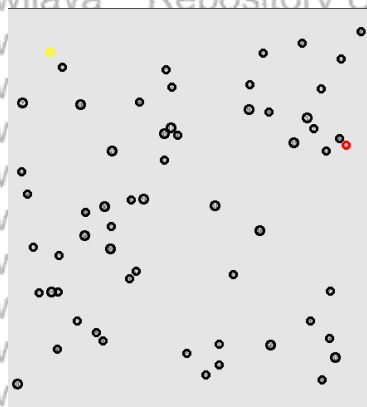
random setelah simulasi dijalankan. Pengiriman paket data oleh node sumber menuju node destinasi dilakukan pada detik 25 sampai detik ke 1000.

3.3.3 Topologi Simulasi



Gambar 3.2 Topologi Simulasi 50 Nodes

Pada gambar 3.2 sebanyak 50 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.

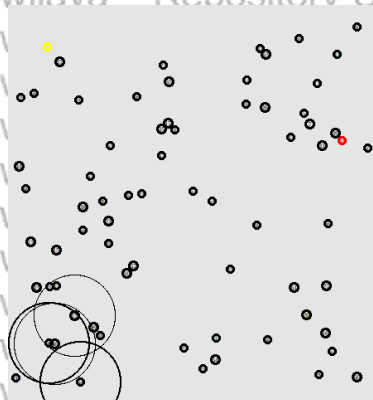


Gambar 3.3 Topologi Simulasi 60 Nodes

Pada gambar 3.3 sebanyak 60 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1

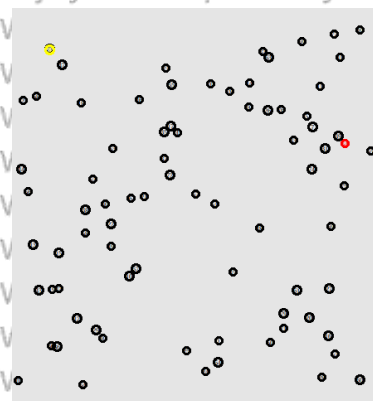


m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



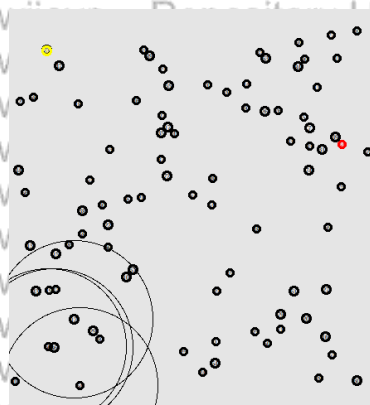
Gambar 3.4 Topologi Simulasi 70 Nodes

Pada gambar 3.4 sebanyak 70 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



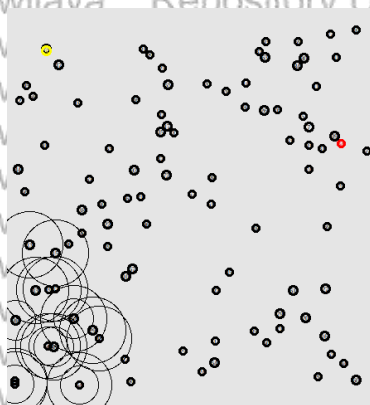
Gambar 3.5 Topologi Simulasi 80 Nodes

Pada gambar 3.5 sebanyak 80 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



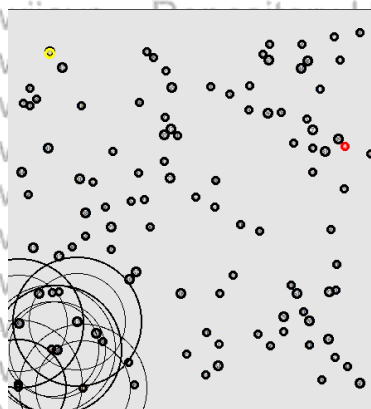
Gambar 3.6 Topologi Simulasi 90 Nodes

Pada gambar 3.6 sebanyak 90 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



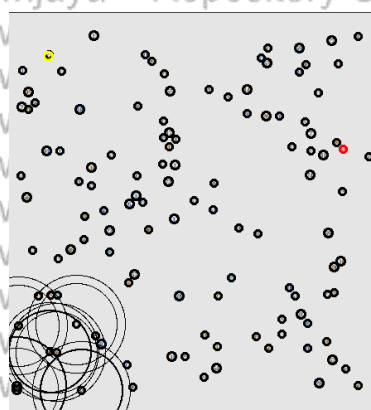
Gambar 3.7 Topologi Simulasi 100 Nodes

Pada gambar 3.7 sebanyak 100 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



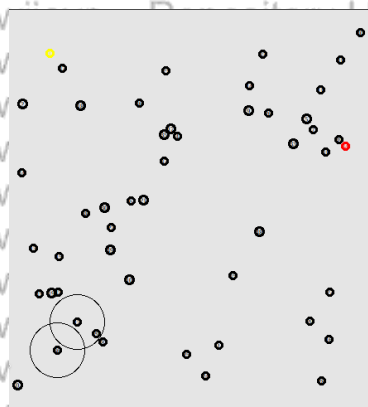
Gambar 3.8 Topologi Simulasi 110 Nodes

Pada gambar 3.8 sebanyak 110 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



Gambar 3.9 Topologi Simulasi 120 Nodes

Pada gambar 3.9 sebanyak 120 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi.



Gambar 3.10 Topologi Simulasi skenario konvergensi

Pada gambar 3.10 sebanyak 50 *node* ditempatkan secara acak menggunakan *pseudo random* pada suatu area dengan luas 1000x1000 meter². Model mobilitas *random way point* merupakan model mobilitas yang digunakan pada simulasi ini, dimana pergerakan *node* akan acak dengan tujuan koordinat tertentu. Layaknya pergerakan manusia pada umumnya, kecepatan pergerakan *node* dipilih antara 1 m/s sampai dengan 2 m/s secara acak. Jenis koneksi UDP digunakan antara *node* sumber dan *node* destinasi. Yang berbeda pada simulasi ini adalah lamanya simulasi dijalankan, pada simulasi ini waktu simulasi diset hanya sampai 100 detik saja.

3.4 Hasil dan Analisis

Rancangan skenario lalu diimplementasikan dalam bentuk simulasi dan data yang dihasilkan dari simulasi tersebut akan diolah dan dianalisis untuk mengetahui kinerja protokol *routing* berdasarkan parameter *Quality of Service*.

3.5 Kesimpulan dan Saran

Pada tahap terakhir ini kesimpulan diambil berdasarkan hasil perencanaan, implementasi, pengujian dan analisis dalam menjawab rumusan masalah serta saran dalam melakukan percobaan selanjutnya.



BAB 4 IMPLEMENTASI

Bab ini menjelaskan tentang detail langkah langkah implementasi yang dilakukan pada penelitian dalam membangun simulasi.

4.1 Implementasi

Network Simulator 2.35 digunakan sebagai simulator untuk menerapkan sistem simulasi agar proses pengujian protokol dapat dilakukan. Untuk gambaran lebih detail tentang cara instalasi NS-2.35 dan tahapan membangun simulasi akan dijelaskan pada bagian Implementasi ini.

4.1.1 Instalasi *Network Simulator*

Sebelum menjalankan simulasi yang sudah direncanakan sebelumnya diperlukan untuk melakukan Instalasi NS-2.35. Dalam melakukan instalasi NS-2.35 diperlukan perangkat lunak tambahan yang mendukung kinerja dari aplikasi NS-2.35. Langkah-langkah instalasi NS-2.35 lebih lanjut dijelaskan pada lampiran A1. Setelah dilakukan instalasi, NS-2.35 dapat digunakan untuk menjalankan simulasi. Protokol *Hierarchical State Routing* (HSR) merupakan pengembangan dari protokol *routing Ad hoc On-Demand Distance Vector* (AODV). Protokol ini tidak disediakan langsung pada NS-2.35 yang telah dipasang sebelumnya. Tidak diperlukan adanya *patch* HSR untuk dipasangkan pada NS-2.35 karena protokol *routing* tersebut dapat dijalankan dengan menambah fungsi fungsi pada *routing* protokol AODV. Pada protokol *routing Destination sequenced distanced vector* (DSDV) disediakan langsung secara default pada NS-2.35.

4.1.2 Membangun Simulasi

Setelah NS-2.35 dan protokol *routing* yang digunakan dapat dijalankan pembangunan simulasi baru dapat dilakukan. Skenario yang sudah dirancang sebelumnya akan di ubah dalam bentuk script dan simpan dalam sebuah file simulasi dengan ekstensi “.tcl”. file tersebut juga dapat digunakan untuk mengelola simulasi baik berupa skenario maupun parameter-parameter lingkungan simulasi.

Berbagai kebutuhan simulasi seperti jenis *routing*, lamanya simulasi, luas area simulasi dikonfigurasi pada file tcl. Pada Simulasi ini, protokol *routing* HSR dan DSDV akan digunakan dengan skenario penambahan jumlah *node* dan untuk mengetahui waktu konvergensi tiap protokol akan digunakan skenario yang berbeda. *Constant Bit Rate* (CBR) akan digunakan sebagai jenis traffic paket data yang terhuung dengan protokol *User Datagram Protokol* (UDP). Setelah seluruh konfigurasi file tcl sesuai dengan yang direncanakan, jalankan file tersebut menggunakan perintah ns beserta nama file “.tcl”.



Banyaknya jumlah skenario menentukan banyaknya jumlah *File tcl* yang perlu dibuat. Pada simulasi ini, banyak *file* yang digunakan adalah 9 *file* yang berisi 8 *file* dengan jumlah *node* berbeda dan 1 *file* untuk melihat nilai konvergensi. *Output* dari menjalankan *file tcl* adalah *file* berektensi “.tr” dan “.nam. *File .tr (trace file)* yang dihasilkan berisi hasil *log* yang merekam segala aktifitas selama simulasi dijalankan, *file* ini digunakan untuk menganalisis simulasi dan melakukan pengujian data. Lalu yang berfungsi untuk menampilkan animasi jalannya simulasi merupakan *file* nam. jalankan *file* tersebut menggunakan *syntax* nam beserta nama *file* “.nam”

4.1.3 Konfigurasi lingkungan simulasi

Lamanya simulasi diatur dengan waktu sebesar 1000 detik, dengan batas area simulasi seluas 1000x1000m. Pergerakan *node* diatur menggunakan model random way point, dengan kelajuan *node* sebesar 1 m/s hingga 2,0 m/s. *Pseudo random seed* digunakan pada default nilai *random*, agar proses jalannya simulasi akan tetap sama sampai 5 kali perulangan. Variasi jumlah *node* yang digunakan sesuai dengan skenario pengujian. Variasi *node* tersebut adalah 50, 60, 70, 80, 90, 100, 110 dan 120 *node*. Mengacu pada *file tcl*, Berikut merupakan contoh *source code* konfigurasi lingkungan simulasi.

```
set val(minSpeed) 1.0 ;# movement
minimum speed [m/s]
set val(maxSpeed) 2.0 ;# movement
maximum speed [m/s]
set val(minPause) 0.0 ;# movement
minimum pause time [s]
set val(maxPause) 0.0 ;# movement
maximum pause time [s]
set val(stop) 1000 ;# time of
simulation end
set val(mobility) Static ;# mobility model
set val(nn) 50 ;# number of
mobilenodes
set val(seed) 1 ;# general
pseudo-random sequence generator
set val(x) 1000 ;# X dimension
of topography
set val(y) 1000 ;# Y dimension
of topography
```

Pada bagian peletakan dan mobilitas *node* digunakan pseudo random agar peletakan dan mobilitas *node* selalu sama setiap menjalankan simulasi. Selanjutnya dalam penentuan *node* yang akan mengirim dan menerima pesan,



pada simulasi ini diatur agar pada node 34 akan mengirimkan pesan dengan koneksi UDP dan node 3 akan menerimanya.

```
set udp0 [new Agent/UDP]
$ns attach-agent $node(34) $udp0
set null0 [new Agent/LossMonitor]
$ns attach-agent $node(3) $null0
$ns connect $udp0 $null0
$udp0 set packetSize_ 1500
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 512
$cbr0 set rate_ 1.0Mb
$cbr0 set random_
```

Dan untuk pengiriman pesan diatur akan dikirim pada detik ke 25

```
$ns at 25.0 "$cbr0 start"
```

4.2 Penarikan Data

Penarikan data dapat dilakukan setelah simulasi berhasil dijalankan. *File* *.tr* (*trace file*) yang dihasilkan berisi hasil *log* yang merekam segala aktifitas selama simulasi dijalankan, *file* ini digunakan sebagai sumber data dalam analisis pengujian. *File* dengan ekstensi “.awk” digunakan Untuk mengambil data dari *trace file*. *File* ini dapat memfilter baris dan kolom pada *trace file* dan melakukan proses komputasi berdasar *code* yang diimplementasikan pada *file* ini. Berikut adalah beberapa program *awk* yang digunakan pada simulasi ini, penghitung *average throughput*, *average end to end delay*, *route overhead*, *packet data delivery ratio* dan *average jitter*.

Kemampuan Program *awk* dalam memanipulasi *string*, digunakan agar *output* yang didapatkan dapat digunakan untuk melakukan perhitungan sesuai dengan yang diinginkan. Berikut adalah contoh *syntax* untuk menjalankan program *awk*.

```
$ awk -f jitter.awk name_file.tr >> output.xg
```

file “*output.xg*” lalu akan dikeluarkan untuk menyimpan hasil komputasi *file* *awk*. Berikut adalah contoh *syntax* untuk memuat *file* tersebut menjadi sebuah grafik menggunakan *xgraph*.



```
$ xgraph output.xg
```

Penggambaran nilai dan hasil dari percobaan akan muncul dalam bentuk grafik yang sudah dipanggil dengan parameter yang dalam simulasi ini berupa jumlah node.

BAB 5 HASIL DAN ANALISIS

5.1 Analisis Hasil

Pembahasan mengenai hasil pengujian dan kinerja protokol HSR dan DSDV akan dijelaskan pada bagian ini. Dalam pembahasan hasil akan dilakukan analisis yang sesuai dengan skenario pengujian untuk membandingkan hasil kinerja protokol yang dapat digunakan sebagai rujukan kesimpulan.

File tcl yang telah dibuat akan dijalankan sesuai skenario penambahan jumlah *node* sebanyak 50, 60, 70, 80, 90, 100, 110 dan 120. Secara bergantian, Simulasi dilakukan antara protokol HSR dan DSDV dimulai dari *node* 50 hingga mencapai simulasi *node* 120. Untuk mengetahui kinerja protokol *routing* pada saat Kepadatan *node* tertentu dapat dilihat pada Skenario penambahan jumlah *node* ini. Dengan menggunakan pseudo random hasil simulasi pun mencapai nilai yang sama walaupun simulasi dilakukan sebanyak 5 kali. Dengan menggunakan parameter *Quality of Service* (QoS) hasil pengujian akan ditampilkan. Berikut adalah parameter-parameter QoS yang akan ditampilkan pada penelitian ini *average throughput*, *average end to end delay*, *route overhead*, *packet data delivery ratio* dan *average jitter* code AWK yang digunakan terlampir pada lampiran laporan. Berikut ini merupakan contoh dari penggalan *raw data* dari *trace file* simulasi DSDV pada skenario 50 *nodes*.

```
s -t 25.000000000 -Hs 34 -Hd -2 -Ni 34 -Nx 883.05 -Ny 713.70 -Nz 0.00 -Ne -  
1.000000 -NI AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 34.0 -Id 3.0 -It cbr -Il 512 -  
If 0 -li 239 -lv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0  
r -t 25.000000000 -Hs 34 -Hd -2 -Ni 34 -Nx 883.05 -Ny 713.70 -Nz 0.00 -Ne -  
1.000000 -NI RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 34.0 -Id 3.0 -It cbr -Il 512 -  
If 0 -li 239 -lv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
```

Cara kerja *awk* adalah dengan mengambil data perkolom yang diinisialisasikan pada *code program* dan mengambil informasi dari data tersebut untuk dihitung dan analisis *Quality of Service* dari suatu simulasi. Seperti pada saat menghitung *throughput* yang di cek adalah jika kolom ke 2 adalah “-t” lalu jika kolom ke 19 adalah “AGT” dan kolom ke-1 adalah “s” atau “+” dan kolom ke 37 lebih dari atau sama dengan 512 maka bisa didapatkan nilai *start time* berdasarkan nilai dari kolom ke 3. Lalu dicek lagi jika kolom ke 19 adalah “AGT” dan kolom ke 1 adalah “r” dan kolom ke 37 lebih dari atau sama dengan 512 maka bisa didapatkan nilai *stop time* berdasarkan nilai dari kolom ke 3 lalu menghitung total ukuran paket yang diterima berdasarkan nilai dari kolom ke 37. Dan terakhir dihitung total dari ukuran paket yang telah didapatkan dibagi selisih dari stop time dengan start time dan dikali 0.008 untuk mengubah menjadi satuan *kbps*.



Lalu untuk menghitung *End to End Delay* maka akan dilihat paket ID nya pada kolom ke 41 dan dicatat waktunya berdasarkan kolom ke 3 lalu bisa didapatkan waktu dikirimnya paket. juga dilihat pada kolom ke 1 jika nilainya adalah “r” dan kolom ke 35 adalah “cbr” dan kolom ke 19 adalah “AGT” dapat diketahui waktu diterimanya paket pada tujuan. Lalu dihitung semua selisih waktunya berdasarkan paket IDnya, lalu dirata rata dan dikali 1000.

Kemudian untuk menghitung *Average Jitter* yang di lihat adalah jika kolom ke 2 adalah “-t” lalu jika kolom ke 19 adalah “AGT” dan send time paket ID pada kolom ke 41 kolom ke 1 adalah “s” atau “+” dan kolom ke-37 lebih dari atau sama dengan 512 maka bisa didapatkan waktu dikirimkannya paket data berdasarkan nilai dari kolom ke 3. Lalu jika kolom ke 2 adalah “-t” lalu jika kolom ke 19 adalah “AGT” dan send time paket ID pada kolom ke 41 kolom ke 1 adalah “r” dan kolom ke-37 lebih dari atau sama dengan 512 maka bisa didapatkan waktu diterimanya paket data berdasarkan nilai dari kolom ke 3. Data data tersebut lalu disimpan dan diolah untuk mencari nilai *jitter* dan dirata rata.

Selanjutnya untuk menghitung *Packet Data Delivery Ratio* dilihat pada kolom ke 1 jika nilainya adalah “s” dan pada kolom ke 19 adalah “AGT” maka nilai total dari data yang berhasil dikirim akan bertambah, lalu jika pada kolom ke 1 jika nilainya adalah “r” dan pada kolom ke 19 adalah “AGT” maka nilai total dari data yang berhasil diterima akan bertambah. Hasil terakhir dari total data yang diterima akan dibagi dengan total data yang dikirim. Dan didapatkan *Packet Data Delivery Ratio*.

Lalu untuk menghitung *Route Overhead* dilihat pada kolom ke 1 jika nilainya adalah “r” dan pada kolom ke 35 adalah “cbr” dan pada kolom ke 19 adalah “AGT” maka nilai total dari paket berhasil dikirim akan bertambah, lalu jika pada kolom ke 1 jika nilainya adalah “s” atau “f” dan pada kolom ke 19 adalah “RTR” dan pada kolom ke 35 adalah “message” atau “DSDV” maka nilai total dari *routing packet* yang berhasil dikirim akan bertambah. Hasil terakhir dari total paket data yang diterima akan dibagi dengan total *routing packet* yang dikirim. Dan didapatkan *Route Overhead* dari simulasi tersebut.

5.1.1 Pengujian Average Throughput

Pada pengujian *average throughput* akan didapatkan Kecepatan transfer paket data antara *source Node* dan *destination node* dapat digambarkan pada pengujian *average throughput* ini. Nilai *average throughput* dapat diperoleh dengan menghitung jumlah total paket data yang telah sukses sampai ke node tujuan selama sela waktu tertentu lalu dibagi dengan durasi simulasi. Dapat dilihat pada tabel 5.1 merupakan nilai *throughput* yang telah sesuai dengan skenario yang telah dijalankan.

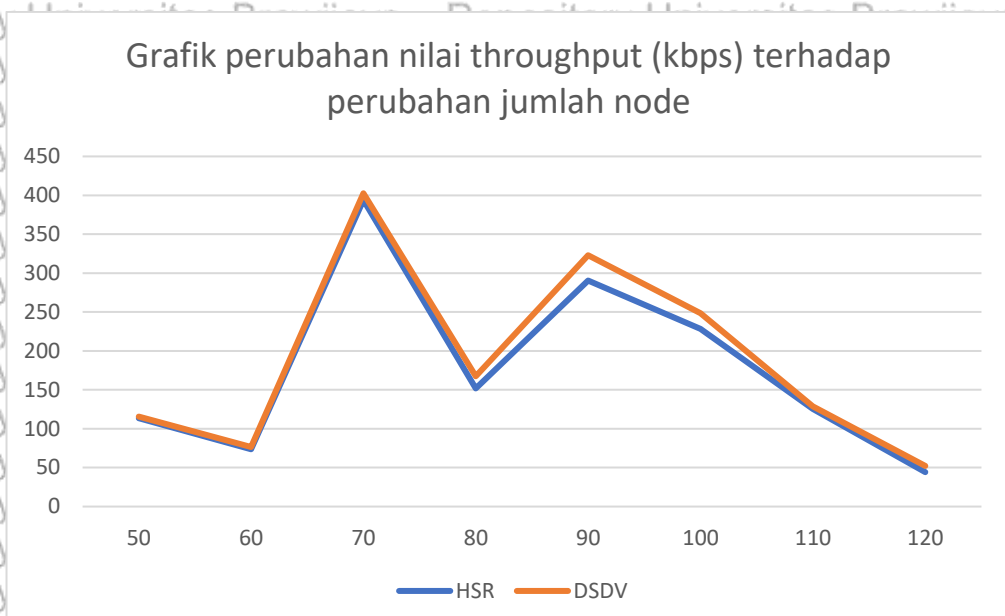


Tabel 5.1 Nilai average throughput

Jumlah Node	Average Throughput (Kbps)	
	HSR	DSDV
50	113,48	115,57
60	73,75	76,54
70	393,93	402,63
80	151,62	167,64
90	290,43	322,91
100	228,49	248,44
110	125,57	128,57
120	44,23	52,20
Rata-rata	177,687	189,312

Pada tabel 5.1 dapat dilihat nilai dari average throughput terhadap penambahan jumlah *node*. HSR mendapatkan nilai *average throughput* yang lebih kecil dibandingkan nilai *average throughput* yang diperoleh DSDV. Selanjutnya nilai *average throughput* pada pengujian dengan 50 *node*, didapatkan *average throughput* HSR sebesar 113,48 Kbps sedangkan nilai *average throughput* DSDV sebesar 115,57 Kbps. Selanjutnya nilai *average throughput* pada pengujian dengan 60 *node*, didapatkan *average throughput* HSR sebesar 73,75 Kbps sedangkan nilai *average throughput* DSDV sebesar 76,54 Kbps. Selanjutnya nilai *average throughput* pada pengujian dengan 70 *node*, didapatkan *average throughput* HSR sebesar 393,93 Kbps sedangkan nilai *average throughput* DSDV sebesar 402,63 Kbps. Selanjutnya nilai *average throughput* pada pengujian dengan 80 *node*, didapatkan *average throughput* HSR sebesar 151,62 Kbps sedangkan nilai *average throughput* DSDV sebesar 167,64 Kbps. Selanjutnya nilai *average throughput* pada pengujian dengan 90 *node*, didapatkan *average throughput* HSR sebesar 290,43 Kbps sedangkan nilai *average throughput* DSDV sebesar 322,91 Kbps.

Lalu nilai *average throughput* pada pengujian dengan 100 *node*, didapatkan *average throughput* HSR sebesar 228,49 Kbps sedangkan nilai *average throughput* DSDV sebesar 248,44 Kbps. Nilai *average throughput* pada pengujian dengan 110 *node*, didapatkan *average throughput* HSR sebesar 125,57 Kbps sedangkan nilai *average throughput* DSDV sebesar 128,57 Kbps. Nilai *average throughput* pada pengujian dengan 120 *node*, didapatkan *average throughput* HSR sebesar 44,23 Kbps sedangkan nilai *average throughput* DSDV sebesar 52,20 Kbps.



Gambar 5.1 Grafik nilai *average throughput*

Grafik perubahan nilai *throughput* yang dapat dilihat pada gambar 5.1 merupakan hasil perbandingan perubahan nilai *average throughput* antara protokol HSR dan DSDV terhadap penambahan jumlah *node*. Dari grafik tersebut menunjukkan bahwa nilai *average throughput* protokol DSDV dan HSR yang tidak stabil dan cenderung mengalami penurunan.

5.1.2 Pengujian *Average Jitter*

Pada pengujian *average jitter* akan didapatkan ragam lamanya waktu sebuah *node* untuk mengirim paket data kepada *node* tujuan dapat digambarkan pada pengujian *average jitter* ini. Nilai *average jitter* dapat diperoleh dengan menghitung variasi durasi sebuah *node* untuk mengirim paket data kepada *node* tujuan. Dapat dilihat pada tabel 5.2 merupakan nilai hasil pengujian yang sesuai dengan skenario yang telah dijalankan berupa *jitter*.

Tabel 5.2 Nilai *average jitter*

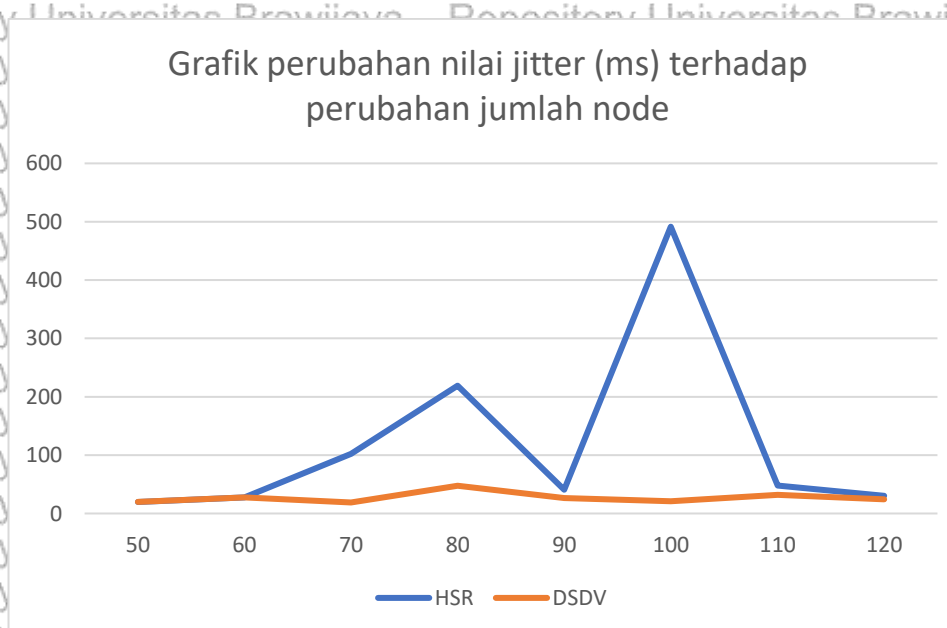
Jumlah Node	Average Jitter (ms)	
	HSR	DSDV
50	19,74	19,74
60	27,84	27,84
70	102,04	18,88
80	218,93	47,54
90	40,84	26,67
100	491,52	20,81



110	47,61	31,84
120	30,16	24,12
Rata-rata	122,335	27,18

Pada tabel 5.2 dapat dilihat nilai dari *average jitter* terhadap variasi jumlah *node*. DSDV mendapatkan nilai *average jitter* yang lebih kecil dibandingkan nilai *average jitter* yang diperoleh HSR. Nilai *average jitter* pada pengujian dengan 50 *node*, didapatkan *average jitter* HSR sebesar 19,74 ms sedangkan nilai *average jitter* DSDV sebesar 19,74 ms. Lalu nilai *average jitter* pada pengujian dengan 60 *node*, didapatkan *average jitter* HSR sebesar 27,84 ms sedangkan nilai *average jitter* DSDV sebesar 27,84 ms. Selanjutnya nilai *average jitter* pada pengujian dengan 70 *node*, didapatkan *average jitter* HSR sebesar 102,04 ms sedangkan nilai *average jitter* DSDV sebesar 18,88 ms. Nilai *average jitter* pada pengujian dengan 80 *node*, didapatkan *average jitter* HSR sebesar 218,93 ms sedangkan nilai *average jitter* DSDV sebesar 47,54 ms. Selanjutnya nilai *average jitter* pada pengujian dengan 90 *node*, didapatkan *average jitter* HSR sebesar 40,84 ms sedangkan nilai *average jitter* DSDV sebesar 26,67 ms.

Lalu nilai *average jitter* pada pengujian dengan 100 *node*, didapatkan *average jitter* HSR sebesar 491,52 ms sedangkan nilai *average jitter* DSDV sebesar 20,81 ms. Nilai *average jitter* pada pengujian dengan 110 *node*, didapatkan *average jitter* HSR sebesar 47,61 ms sedangkan nilai *average jitter* DSDV sebesar 31,84 ms. Nilai *average jitter* pada pengujian dengan 120 *node*, didapatkan *average jitter* HSR sebesar 30,16 ms sedangkan nilai *average jitter* DSDV sebesar 24,12 ms.



Gambar 5.2 Grafik nilai *average jitter*



Grafik perubahan nilai *average jitter* yang dapat dilihat pada gambar 5.2 merupakan hasil perbandingan perubahan nilai *average jitter* antara protokol HSR dan DSDV terhadap penambahan jumlah *node*. Dari grafik tersebut menunjukkan bahwa nilai *average jitter* protokol DSDV cenderung lebih stabil, sedangkan pada protokol HSR tidak stabil sehingga dapat terlihat perbedaan kinerja pada parameter *average jitter* yang cukup signifikan.

5.1.3 Pengujian *Average end to end delay*

Pada pengujian *average end to end delay* akan didapatkan durasi yang dibutuhkan sebuah *node* sumber untuk mengirim paket data kepada *node* tujuan dapat digambarkan pada pengujian *average end to end delay* ini. Nilai *average end to end* dapat diperoleh dengan menghitung selisih lama waktu sebuah *node* untuk mengirim paket dan waktu data sampai kepada *node* tujuan. Dapat dilihat pada tabel 5.3 merupakan nilai hasil pengujian yang sesuai dengan skenario yang telah dijalankan berupa *end to end delay*.

Tabel 5.3 Nilai *average end to end delay*

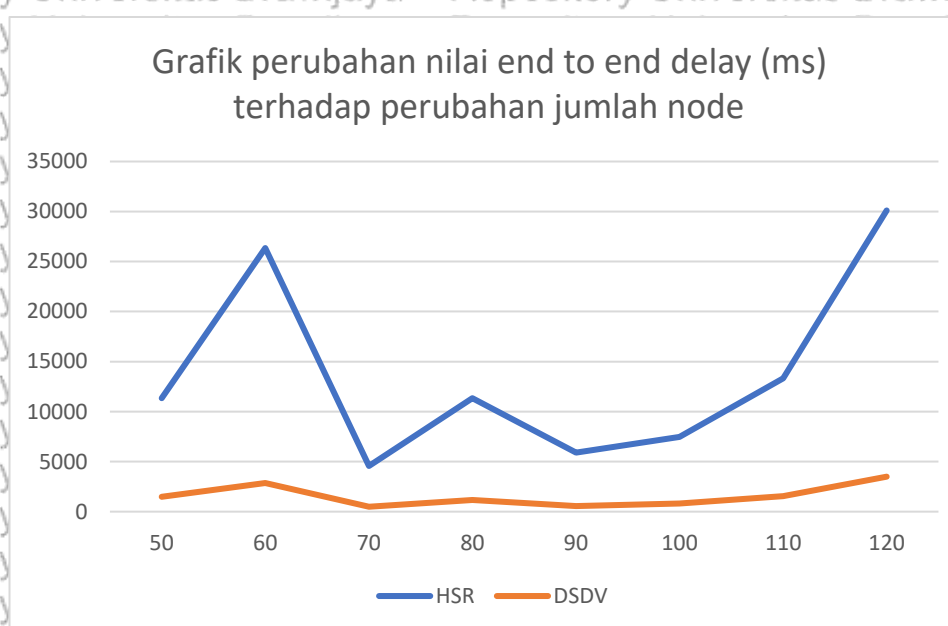
Jumlah <i>Node</i>	<i>Average end to end delay</i> (ms)	
	HSR	DSDV
50	11347,26	1471,66
60	26356,76	2852,29
70	4563,83	488,71
80	11326,94	1164,48
90	5905,65	560,02
100	7482,39	798,96
110	13313,22	1552,00
120	30107,81	3496,23
Rata-rata	13800,48	2798,04

Pada tabel 5.3 dapat dilihat nilai dari *average end to end delay* terhadap variasi jumlah *node*. DSDV mendapatkan nilai *average end to end delay* yang lebih kecil dibandingkan nilai *average end to end delay* yang diperoleh HSR. Nilai *average end to end delay* pada pengujian dengan 50 *node*, didapatkan *average end to end delay* HSR sebesar 11347,26 ms sedangkan nilai *average end to end delay* DSDV sebesar 1471,66 ms. Lalu nilai *average end to end delay* pada pengujian dengan 60 *node*, didapatkan *average end to end delay* HSR sebesar 26356,76 ms sedangkan nilai *average end to end delay* DSDV sebesar 2852,29 ms. Selanjutnya nilai *average end to end delay* pada pengujian dengan 70 *node*, didapatkan



average end to end delay HSR sebesar 4563,83 ms sedangkan nilai *average end to end delay* DSDV sebesar 488,71 ms. Nilai *average end to end delay* pada pengujian dengan 80 *node*, didapatkan *average end to end delay* HSR sebesar 11326,94 ms sedangkan nilai *average end to end delay* DSDV sebesar 1164,48 ms. Selanjutnya nilai *average end to end delay* pada pengujian dengan 90 *node*, didapatkan *average end to end delay* HSR sebesar 5905,65 ms sedangkan nilai *average end to end delay* DSDV sebesar 560,02 ms.

Lalu nilai *average end to end delay* pada pengujian dengan 100 *node*, didapatkan *average end to end delay* HSR sebesar 7482,39 ms sedangkan nilai *average end to end delay* DSDV sebesar 798,96 ms. Nilai *average end to end delay* pada pengujian dengan 110 *node*, didapatkan *average end to end delay* HSR sebesar 13313,22 ms sedangkan nilai *average end to end delay* DSDV sebesar 1552,00 ms. Nilai *average end to end delay* pada pengujian dengan 120 *node*, didapatkan *average end to end delay* HSR sebesar 30107,81 ms sedangkan nilai *average end to end delay* DSDV sebesar 3496,23 ms.



Gambar 5.3 Grafik nilai *average end to end delay*

Grafik perubahan nilai *end to end delay* yang dapat dilihat pada gambar 5.3 merupakan hasil perbandingan perubahan nilai *average end to end delay* antara protokol HSR dan DSDV terhadap penambahan jumlah *node*. Dari grafik tersebut menunjukkan bahwa nilai *average end to end delay* protokol DSDV cenderung lebih stabil dan mengalami penurunan, sedangkan pada protokol HSR tidak stabil bahkan cenderung mengalami peningkatan.



5.1.4 Pengujian Packet Data Delivery Ratio

Pada pengujian *packet data delivery ratio* akan didapatkan rasio perbandingan jumlah paket data yang diterima *destination node* dengan jumlah data yang dikirimkan *source node* dapat digambarkan pada pengujian *packet data delivery ratio* ini. Nilai *packet data delivery ratio* dapat diperoleh dengan membandingkan jumlah paket data yang diterima *destination node* dengan jumlah data yang dikirimkan *source node*. Dapat dilihat pada tabel 5.4 merupakan nilai hasil pengujian yang sesuai dengan skenario yang telah dijalankan berupa *packet data delivery ratio*.

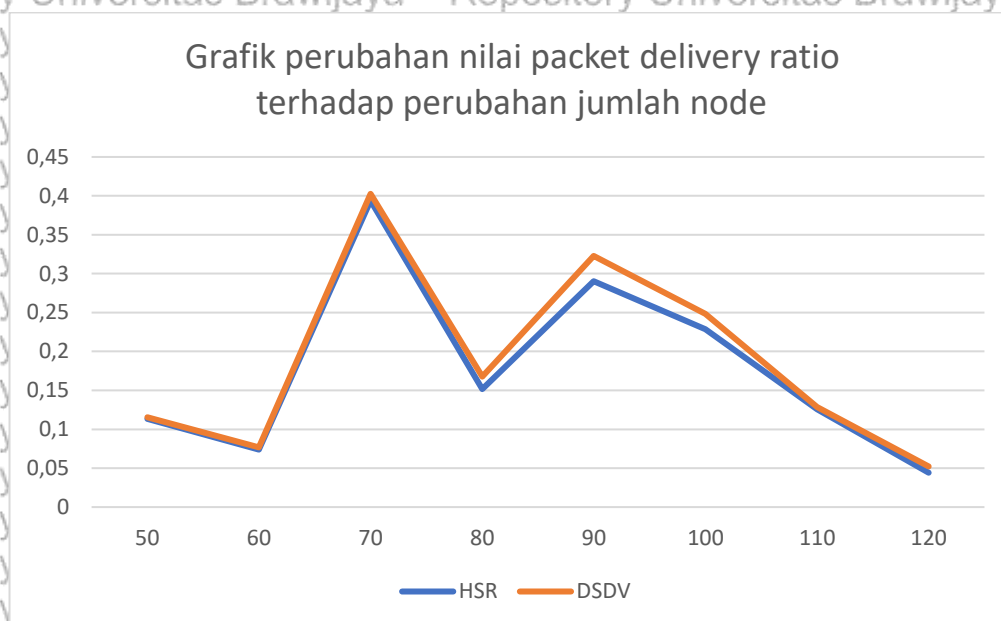
Tabel 5.4 Nilai packet data delivery ratio

Jumlah Node	Packet data delivery ratio	
	HSR	DSDV
50	0,1135	0,1156
60	0,0738	0,07657
70	0,3939	0,4026
80	0,1516	0,1676
90	0,2904	0,3229
100	0,2285	0,2484
110	0,1256	0,1286
120	0,0442	0,0522
Rata-rata	0,1777	0,1893

Pada tabel 5.4 dapat dilihat nilai dari *packet data delivery ratio* terhadap variasi jumlah *node*. DSDV mendapatkan nilai *packet data delivery ratio* yang lebih kecil dibandingkan nilai *packet data delivery ratio* yang diperoleh HSR. Nilai *packet data delivery ratio* pada pengujian dengan 50 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,1135 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,1156. Lalu nilai *packet data delivery ratio* pada pengujian dengan 60 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,0738 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,07657. Selanjutnya nilai *packet data delivery ratio* pada pengujian dengan 70 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,3939 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,4026. Nilai *packet data delivery ratio* pada pengujian dengan 80 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,1516 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,1676. Selanjutnya nilai *packet data delivery ratio* pada pengujian dengan 90 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,2904 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,3229.



Lalu nilai *packet data delivery ratio* pada pengujian dengan 100 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,2285 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,2484. Nilai *packet data delivery ratio* pada pengujian dengan 110 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,1256 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,1286. Nilai *packet data delivery ratio* pada pengujian dengan 120 *node*, didapatkan *packet data delivery ratio* HSR sebesar 0,0442 sedangkan nilai *packet data delivery ratio* DSDV sebesar 0,0522.



Gambar 5.4 Grafik nilai *packet delivery ratio*

Grafik perubahan nilai *packet delivery ratio* yang dapat dilihat pada gambar 5.4 merupakan hasil perbandingan perubahan nilai *packet delivery ratio* antara protokol HSR dan DSDV terhadap penambahan jumlah *node*. Dari grafik tersebut menunjukkan bahwa nilai *packet delivery ratio* protokol DSDV dan HSR yang tidak stabil dan cenderung mengalami penurunan.

5.1.5 Pengujian Route Overhead

Pada pengujian *route overhead* akan didapatkan jumlah total *routing* (*route request*, *route replay*, *route error*) paket yang dikirimkan selama simulasi tetapi tidak termasuk MAC, dan paket ARP dapat digambarkan pada pengujian *packet data delivery ratio* ini. Nilai *route overhead* dapat diperoleh dengan menjumlahkan total *routing* (*route request*, *route replay*, *route error*) paket yang dikirimkan selama simulasi tetapi tidak termasuk MAC, dan paket ARP. Dapat dilihat pada tabel 5.5 merupakan nilai hasil pengujian yang sesuai dengan skenario yang telah dijalankan berupa *route overhead*.

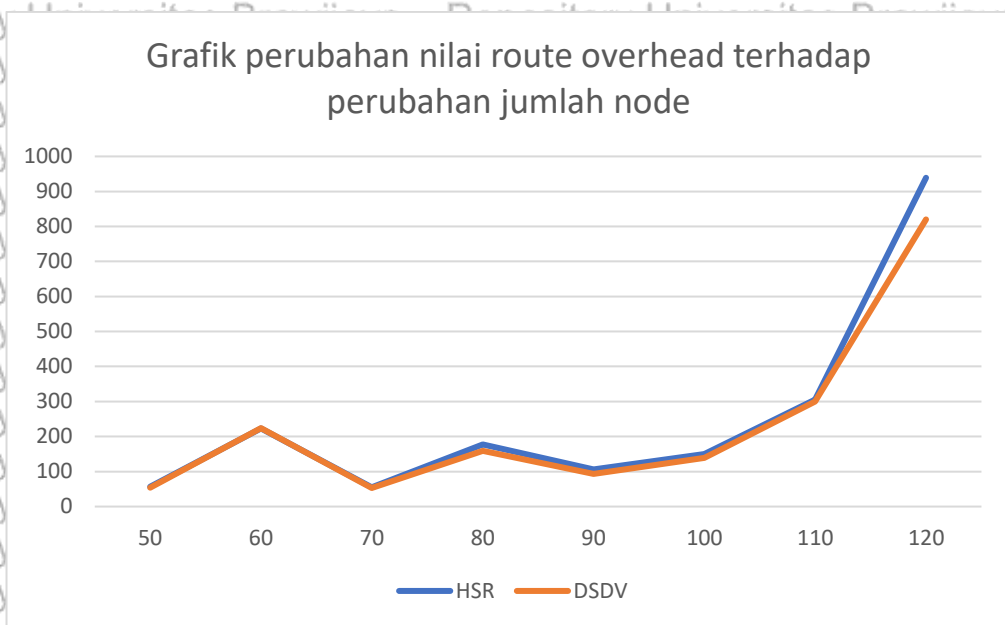


Tabel 5.5 Nilai route overhead

Jumlah Node	Route overhead	
	HSR	DSDV
50	57	54
60	223	224
70	55	53
80	177	159
90	106	93
100	150	139
110	305	300
120	939	820
Rata-rata	251,5	230,25

Pada tabel 5.5 dapat dilihat nilai dari *route overhead* terhadap variasi jumlah *node*. DSDV mendapatkan nilai *route overhead* yang lebih kecil dibandingkan nilai *route overhead* yang diperoleh HSR. Nilai *route overhead* pada pengujian dengan 50 *node*, didapatkan *route overhead* HSR sebesar 57 sedangkan nilai *route overhead* DSDV sebesar 54. Lalu nilai *route overhead* pada pengujian dengan 60 *node*, didapatkan *route overhead* HSR sebesar 223 sedangkan nilai *route overhead delay* DSDV sebesar 224. Selanjutnya nilai *route overhead* pada pengujian dengan 70 *node*, didapatkan *route overhead* HSR sebesar 55 sedangkan nilai *route overhead* DSDV sebesar 53. Nilai *route overhead* pada pengujian dengan 80 *node*, didapatkan *route overhead* HSR sebesar 177 sedangkan nilai *route overhead* DSDV sebesar 159. Selanjutnya nilai *route overhead* pada pengujian dengan 90 *node*, didapatkan *route overhead* HSR sebesar 106 sedangkan nilai *route overhead* DSDV sebesar 93.

Lalu nilai *route overhead* pada pengujian dengan 100 *node*, didapatkan *route overhead* HSR sebesar 150 sedangkan nilai *route overhead* DSDV sebesar 139. Nilai *route overhead* pada pengujian dengan 110 *node*, didapatkan *route overhead* HSR sebesar 305 sedangkan nilai *route overhead* DSDV sebesar 300. Nilai *route overhead* pada pengujian dengan 120 *node*, didapatkan *route overhead* HSR sebesar 939 sedangkan nilai *route overhead* DSDV sebesar 820.



Gambar 5.5 Grafik nilai route overhead

Grafik perubahan nilai *route overhead* yang dapat dilihat pada gambar 5.5 merupakan hasil perbandingan perubahan nilai *route overhead* antara protokol HSR dan DSDV terhadap penambahan jumlah *node*. Dari grafik tersebut menunjukkan bahwa nilai *route overhead* protokol HSR dan DSDV cenderung mengalami kenaikan sejalan dengan banyaknya *node*.

5.1.6 Pengujian *Route Convergence Time*

Untuk mengukur waktu konvergensi protokol DSDV dan HSR, dijalankan lah sebuah skenario tambahan. *File tcl* yang telah dibuat berdasarkan perancangan skenario tambahan, dijalankan sesuai skenario untuk melakukan pengujian. Selisih waktu antara waktu saat paket terakhir diterima node tujuan lalu terputus dari jalur yang sudah dibuat dan berhasil menemukan jalur baru dan paket berhasil diterima lagi oleh node tujuan merupakan waktu Konvergensi yang didapatkan. Simulasi dilakukan sebanyak 5 kali dan menghasilkan nilai yang sama, hal ini terjadi karena penempatan *node* menggunakan *pseudo random*, sehingga nilai yang dihasilkan pada tiap pengujian sama.

Tabel 5.6 Route DSDV

No.	Jalur routing	Waktu diterimanya paket pada node tujuan
1	34 → 14 → 17 → 46 → 23 → 3	83,772773345
2	34 → 14 → 37 → 23 → 3	83,801391322
	Selisih	0,028617977



Tabel 5.7 Route HSR

No.	Jalur <i>routing</i>	Waktu diterimanya paket pada node tujuan	Selisih
1	34 → 9 → 17 → 14 → 4 → 46 → 47 → 23 → 3	31,80747079	0,042107
2	34 → 9 → 17 → 14 → 4 → 46 → 47 → 3	31,84957777	
3	34 → 9 → 17 → 14 → 4 → 46 → 47 → 3	32,8049554	0,353983
4	34 → 9 → 17 → 14 → 4 → 46 → 47 → 23 → 3	33,15893831	
5	34 → 9 → 17 → 14 → 4 → 46 → 47 → 23 → 3	33,15893831	0,03472
6	34 → 9 → 17 → 14 → 4 → 46 → 47 → 3	33,19365846	
7	34 → 9 → 17 → 14 → 4 → 46 → 47 → 3	61,93971995	0,806198
8	34 → 9 → 17 → 14 → 35 → 6 → 44 → 18 → 3	62,74591808	
9	34 → 9 → 17 → 14 → 35 → 6 → 44 → 18 → 3	63,75452587	0,173682
10	34 → 9 → 17 → 14 → 37 → 23 → 3	63,92820791	
Rata Rata			0,282138

Jalur *routing* dari protokol DSDV dan HSR dapat kita lihat pada tabel 5.6 dan Tabel 5.7 mulai dari awal terbentuknya jalur antara *source Node* dan *destination node* sampai dengan perubahan jalur *routing*. Dari Tabel 5.6 diketahui bahwa jalur *routing* pertama yang terbentuk pada protokol DSDV adalah ketika *node* sumber atau *node*(34) berhasil melakukan pengiriman *packet* menuju *node*(3) menggunakan jalur *routing* 34 → 14 → 17 → 46 → 23 → 3 dalam waktu 25 detik.

Pada Tabel 5.7, diketahui bahwa jalur *routing* yang pertama kali terbentuk pada protokol HSR adalah ketika *node* sumber atau *node*(34) berhasil melakukan pengiriman *packet* menuju *node*(3) menggunakan jalur *routing* 34 → 9 → 17 → 14 → 4 → 46 → 47 → 23 → 3 dengan waktu 25.1057579 detik. Rata-rata waktu konvergensi protokol DSDV adalah 0,028617977 detik. Sedangkan rata-rata waktu konvergensi protokol HSR adalah 0,282138 detik.

DSDV mengalami 1 kali perubahan jalur, hal ini dapat dilihat pada tabel 5.6 dimana jalur antara *source Node* dan *destination node* yang tercipta hanya 2 jalur.



34 → 14 → 17 → 46 → 23 → 3, 34 → 14 → 37 → 23 → 3 merupakan jalur yang digunakan protokol DSDV dalam mengirimkan data. Dan dapat diketahui sela waktu yang dibutuhkan protokol DSDV dalam membentuk jalur baru 34 → 14 → 37 → 23 → 3 adalah 0,028617977 detik. Maka rata rata waktu konvergensi protokol DSDV adalah 0,028617977 detik Karena pada skenario *convergence time* DSDV hanya terjadi 1 kali perubahan jalur.

HSR mengalami 5 kali perubahan jalur, hal ini dapat dilihat pada tabel 5.7 dimana jalur antara *source node* dan *destination node* yang tercipta. 34 → 9 → 17 → 14 → 4 → 46 → 47 → 23 → 3, 34 → 9 → 17 → 14 → 4 → 46 → 47 → 3, 34 → 9 → 17 → 14 → 35 → 6 → 44 → 18 → 3, 34 → 9 → 17 → 14 → 37 → 23 → 3. merupakan jalur yang digunakan protokol HSR dalam mengirimkan data. Maka rata rata waktu konvergensi protokol HSR adalah 0,282138 detik hasil tersebut merukan hasil rata rata sela waktu dari tiap pembentukan jalur dan perubahan jalur Karena pada skenario *convergence time* HSR terjadi 5 kali perubahan jalur.

Berdasarkan data tersebut terbukti bahwa kemampuan routing protocol DSDV dalam mencari jalur baru setelah terputus dari jalur lamanya lebih cepat dibanding HSR. Baik dalam waktu pembentukan awal jalur maupun perubahan jalur.

5.2 Perbandingan Hasil Analisis Pengujian Protokol DSDV dan HSR

Perbandingan antara protokol DSDV dan HSR akan dianalisis pada bagian ini untuk mengetahui protokol mana yang memiliki kinerja yang lebih baik. Pada tabel 5.8 diitunjukkan hasil perbandingan antar hasil pengujian protokol.

Tabel 5.8 Perbandingan hasil analisis pengujian protokol

Skenario		DSDV	HSR	Keterangan
Variasi jumlah node	<i>Average Troughput</i>	189,312 Kbps	177,687 Kbps	DSDV lebih baik
	<i>Average Jitter</i>	27,18 ms	122,335 ms	DSDV lebih baik
	<i>Average end to end delay</i>	2798,04 ms	13800,48 ms	DSDV lebih baik
	<i>Packet data delivery ratio</i>	18,93 %	17,77 %	DSDV lebih baik
	<i>Route Overhead</i>	230,25	251,5	DSDV lebih baik
Konvergensi	Waktu Konvergensi	0,028617977 detik	0,282138 detik	DSDV lebih baik



Hasil analisis perbandingan kinerja antara protokol DSDV dan HSR dalam skenario penambahan *node* dapat dilihat pada tabel 5.8. Penerapan tiap scenario pengaruh terhadap parameter uji yang digunakan dikarenakan penambahan jumlah *node* dapat dilihat pada tabel 5.8. Pada pengujian skenario penambahan jumlah *node*, kinerja protokol DSDV lebih unggul daripada protokol HSR dari segi parameter uji *average throughput*, *average end to end delay*, *average jitter*, *Packet data delivery ratio* dan *Route Overhead*. Dan dapat dilihat juga perbedaan yang cukup signifikan pada parameter *jitter* dan *end to end delay*. Waktu konvergensi routing DSDV lebih baik dari pada HSR. Hal ini dapat dilihat pada tabel 5.8 karena nilai waktu konvergensi DSDV lebih kecil dibandingkan dengan nilai waktu HSR. Perhitungan didapatkan dari rata-rata sela waktu antara pembentukan jalur dan perubahan jalur.

Peningkatan nilai parameter *average throughput*, *packet data delivery ratio* menunjukkan kinerja yang bagus, sedangkan penurunan parameter *average throughput*, *packet data delivery ratio* menunjukkan sebaliknya. Sementara peningkatan parameter *average end to end delay*, *average jitter*, dan *Route Overhead* menunjukkan kinerja yang buruk, sedangkan penurunan parameter *average end to end delay*, *average jitter*, dan *Route Overhead* menunjukkan sebaliknya.



BAB 6 PENUTUP

6.1 Kesimpulan

Hasil yang didapatkan selama penelitian yang telah dilakukan akan ditunjukan pada bagian ini. Kesimpulan yang didapatkan dari pengujian, dan analisis hasil pengujian yang telah dilakukan adalah sebagai berikut:

1. Pada pengujian diketahui bahwa penerapan scenario variasi jumlah node berpengaruh terhadap kinerja protokol *routing* yang digunakan yaitu *average throughput*, *average jitter*, *average end to end delay*, *Packet data delivery ratio* dan *Route Overhead*.

Dari pengujian kinerja protokol *routing* terhadap skenario variasi jumlah *node*, diketahui perbandingan kinerja antara protokol DSDV dan HSR menunjukkan bahwa protokol DSDV lebih unggul daripada protokol HSR. Dari segi parameter uji *average throughput*, *average jitter*, *average end to end delay*, *packet data delivery ratio* dan *route overhead* diketahui nilai parameter uji protokol DSDV berturut-turut adalah 189,312 Kbps, 27,18 ms, 2798,04 ms, 18,93 % dan 230,25 sedangkan nilai parameter uji protokol HSR berturut-turut adalah 177,687 Kbps, 122,335 ms, 13800,48 ms, 17,77 % dan 251,5.

Lalu pada pengujian *route convergence time* diketahui nilai rata rata waktu konvergensi protokol DSDV adalah 0,028617977 detik sedangkan nilai rata rata waktu konvergensi protocol HSR adalah 0,282138 detik.

Secara keseluruhan, Protokol DSDV memiliki knerja lebih baik dari pda HSR hal ini dapat dibuktikan berdasarkan hasil pengujian yang telah dilakukan.

2. Pengaruh skenario terhadap parameter uji kinerja protocol DSDV dan HSR bervariasi.

Pertambahan jumlah node menyebabkan penumpukan *traffic* pada suatu titik tertentu tertentu, yang mengakibatkan kinerja protokol DSDV pada parameter *average jitter* mengalami peningkatan, sementara parameter *average throughput*, *average end to end delay*, *Packet data delivery ratio* dan *Route Overhead* mengalami penurunan. Sedangkan pada protokol HSR mengakibatkan peningkatan pada parameter *average jitter* sementara pada parameter *average throughput*, *average end to end delay*, *Packet data delivery ratio* dan *Route Overhead* mengalami penurunan.

Peningkatan parameter *average throughput*, *packet data delivery ratio* menunjukkan kinerja yang bagus, sedangkan penurunan parameter *average throughput*, *packet data delivery ratio* menunjukkan sebaliknya. Sementara peningkatan parameter *average end to end delay*, *average jitter*, dan *Route Overhead* menunjukkan kinerja yang buruk, sedangkan



penurunan parameter *average end to end delay*, *average jitter*, dan *Route Overhead* menunjukkan sebaliknya.

6.2 Saran

Dibawah ini merupakan saran-saran yang bisa disampaikan dalam pengembangan penelitian selanjutnya yang serupa antara lain:

1. Diperlukannya skenario yang berbeda dalam pembangunan simulasi untuk menganalisis kinerja pada protokol DSDV dan HSR.
2. Diperlukannya tambahan skenario berbagai macam serangan, dalam analisis kinerja pada protokol DSDV dan HSR.

DAFTAR PUSTAKA

- Agarkhed, J. 2017. A Survey on DSDV *Routing* Protocol in Ad hoc Network. IJETTCs: International Journal of Emerging Trends & Technology in Computer Science, (Volume 6, Issue 5, pp. 114-117)
- Bai, Y., Mai, Y. dan Wang, N. 2017. Performance comparison and evaluation of the proactive and reactive *routing* protocols for MANETs. IEEE: In *Wireless Telecommunications Symposium (WTS), 2017* (pp. 1-5).
- Gulati, M.K. dan Kumar, K. 2014. Performance comparison of mobile Ad Hoc network *routing* protocols. IJCSN: *International Journal of Computer Networks & Communications*, 6(2), p.127.
- Larlar, S., Yadav, A.K. 2017. *Comparative Study of Routing Protocols in MANET*. India: Oriental Scientific Publishing Co.
- Natarajan, K., Mahadevan, G. 2017. *Evaluation of seven MANET Routing Protocols using Scalability Scenario*. IJCSN: International Journal of Computer Science and Network (Volume: 6, Issue: 2, April 2017).
- Sakshi., Goyal, N. 2017. *MANET: Simulation Analysis and Performance Evaluation of Routing Protocol Using NS-3*. IJCSN: International Journal of Advanced Research in Computer Science and Software Engineering (Volume: 7, Issue: 4, April: 2017).
- Sharma, A. dan Kumar, R. 2016. December. Performance comparison and detailed study of AODV, DSDV, DSR, TORA and OLSR *routing* protocols in ad hoc networks. IEEE: In *Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference on* (pp. 732-736).
- Shenbagapriya, R., Kumar, N. 2014. A survey on proactive *routing* protocols in MANETs. IEEE: In *Science Engineering and Management Research (ICSEMR), 2014 International Conference on* (pp. 1-7).
- Shruthi, S. 2017. Proactive *routing* protocols for a MANET—A review. IEEE: In *I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on* (pp. 821-827).
- Singh, K., et al. 2017. Simulation & Performance Comparison between DSR, FSR & HSR *Routing* Protocols in MANET Using NS2. ACSIS: *The Second International Conference on Research in Intelligent and Computing in Engineering, 2017* (volume: 10, pp. 83-88).
- Sureshkumar, A., Ellappan, V. dan Manivel, K. 2017. A comparison analysis of DSDV and AODV *routing* protocols in mobile AD HOC networks. IEEE: In *Emerging Devices and Smart Systems (ICEDSS), 2017 Conference on* (pp. 234-237).

LAMPIRAN SCRIPT AWK

A.1 Average Throughput

```
BEGIN {
    recvdSize = 0
    startTime = 1e6
    stopTime = 0
}
{
    # Trace line format: new
    if ($2 == "-t") {
        event = $1
        time = $3
        node_id = $5
        flow_id = $39
        pkt_id = $41
        pkt_size = $37
        flow_t = $45
        level = $19
    }
    # Store start time
    if (level == "AGT" && (event == "+" || event == "s") &&
        pkt_size >= 512) {
        if (time < startTime) {
            startTime = time
        }
    }
    # Update total received packets' size and store packets
    arrival time
    if (level == "AGT" && event == "r" && pkt_size >= 512) {
        if (time > stopTime) {
            stopTime = time
        }
    }
    # Rip off the header
    hdr_size = pkt_size % 512
    pkt_size -= hdr_size
    # Store received packet's size
}
```



```

recvSize += pkt_size
}
}
END {
printf("%.2f\t%.2f\n", nodes, (recvSize/(stopTime-
startTime)) * (8/1000))
}

```

A.2 End to End Delay

```

BEGIN {
sends=0;
recvs=0;
routing_packets=0.0;
droppedBytes=0;
droppedPackets=0;
highest_packet_id =0;
sum=0;
recvnum=0;
}
{
# Trace line format: new
time = $3;
packet_id = $41;

# CALCULATE PACKET DELIVERY FRACTION
if (( $1 == "s" ) && ( $35 == "cbr" ) && ( $19=="AGT" )) {
sends++; }

if (( $1 == "r" ) && ( $35 == "cbr" ) && ( $19=="AGT" ))
{ recvs++; }

# CALCULATE DELAY
if ( start_time[packet_id] == 0 ) start_time[packet_id] =
time;
if (( $1 == "r" ) && ( $35 == "cbr" ) && ( $19=="AGT" )) {
end_time[packet_id] = time; }
else { end_time[packet_id] = -1; }

# DROPPED PACKETS
if (( $1 == "d" ) && ( $35 == "cbr" ) && ( $3 > 0 ))
{
droppedBytes=droppedBytes+$37;
droppedPackets=droppedPackets+1;
}

#find the number of packets in the simulation
if (packet_id > highest_packet_id)
highest_packet_id = packet_id;
}
END {
for ( i in end_time)
{

```



```

start = start_time[i];
end = end_time[i];
#printf("EndTime\t\t %.2f\t\t%.2f\t\t%.2f\n",i,start,end);
packet_duration = end - start;
if (packet_duration > 0)
{
    sum += packet_duration;
    recvnum++;
}
delay=sum/recvnum;
printf("%.2f\t%.2f\n",nodes,delay*1000)
}

```

A.3 Average Jitter

```

BEGIN {
    num_recv=0
}
{
    # Trace line format: new
    if ($2 == "-t") {
        event = $1
        time = $3
        node_id = $5
        flow_id = $39
        pkt_id = $41
        pkt_size = $37
        flow_t = $45
        level = $19
    }
    # Store packets send time
    if ((level == "AGT" && sendTime[pkt_id] == 0 && (event ==
    "+" || event == "s") && pkt_size >= 512) {
        sendTime[pkt_id] = time
    }
    # Store packets arrival time
    if ((level == "AGT" && event == "r" && pkt_size >= 512) {
        recvTime[pkt_id] = time
        num_recv++
    }
}
END {

```



```
# Compute average jitter
jitter1 = jitter2 = tmp_recv = 0
prev_time = delay = prev_delay = processed = 0
prev_delay = -1
for (i=0; processed<num_recv; i++) {
    if(recvTime[i] != 0) {
        tmp_recv++
        if(prev_time != 0) {
            delay = recvTime[i] - prev_time
            e2eDelay = recvTime[i] - sendTime[i]
            if(delay < 0) delay = 0
            if(prev_delay != -1) {
                jitter1 += abs(e2eDelay -
prev_e2eDelay)
                jitter2 += abs(delay-prev_delay)
            }
            prev_delay = delay
            prev_e2eDelay = e2eDelay
        }
        prev_time = recvTime[i]
    }
    processed++
}
printf("Mean Jitter =
%.2f\n", jitter1*1000/tmp_recv);
printf("%.2f\t%.2f\n", nodes, jitter2*1000/tmp_recv)
printf("Current Jitter =
%.2f\n", jitter2*1000/tmp_recv);
}

function abs(value) {
    if (value < 0) value = 0-value
    return value
}
```

A.4 Packet Data Delivery Ratio

```
BEGIN {
    sendLine = 0;
```



```

recvLine = 0;
forwardLine = 0;
}
$0 ~/^s.* AGT/ {
    sendLine ++;
}
$0 ~/^r.* AGT/ {
    recvLine ++;
}
$0 ~/^f.* RTR/ {
    forwardLine ++;
}
$0 ~/^D.* cbr/ {
    dropLine ++;
}
END {
    printf("%.2f\t%.4f\n", nodes, (recvLine/sendLine))
}

```

A.5 Route Overhead

```

BEGIN{
    recvd = 0; # to calculate total number of data packets received
    rt_pkts = 0; # to calculate total number of routing packets received
}
# Check if it is a data packet
if (( $1 == "r" ) && ( $35 == "cbr" || $35 == "tcp" ) && ( $19 == "AGT" )) recvd++;
# Check if it is a routing packet
if (( $1 == "s" || $1 == "f" ) && $19 == "RTR" && ($35 == "AODV" || $35 == "message" || $35 == "DSR" || $35 == "DSDV" )) rt_pkts++;
}
END{
    printf("%.2f\t%.2f\n", nodes, rt_pkts/recvd)
}

```