



TUGAS AKHIR - KI141502

**Studi Kinerja Model Transmisi *TwoRayGround*  
dan Nakagami pada Protokol *Reactive di*  
Lingkungan *Mobile AdHoc Network (MANET)*  
menggunakan *Network simulator 2 (NS-2)***

AZI PRASTYO  
NRP 5110100225

Dosen Pembimbing  
Dr. Eng. RADITYO ANGGORO, S.Kom., M.Sc.  
Ir. F.X. ARUNANTO, M.Sc.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016





TUGAS AKHIR - KI141502

**Studi Kinerja Model Transmisi *TwoRayGround*  
dan Nakagami pada Protokol *Reactive* di  
Lingkungan Mobile Ad Hoc Network (MANET)  
menggunakan *Network simulator 2 (NS-2)***

AZI PRASTYO  
NRP 5110100225

Dosen Pembimbing  
Dr. Eng. RADITYO ANGGORO, S.Kom., M.Sc.  
F.X. ARUNANTO, Ir., M.Sc.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016





FINAL PROJECT - KI141502

**Performance Study of *TwoRayGround* and  
Nakagami Propagation Model For *Reactive*  
Protocol on *Mobile AdHoc Network (MANET)*  
using *Network simulator 2 (NS-2)***

AZI PRASTYO  
NRP 5110100225

Advisor  
Dr. Eng. RADITYO ANGGORO, S.Kom., M.Sc.  
F.X. ARUNANTO, Ir., M.Sc.

INFORMATICS DEPARTMENT  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



## LEMBAR PENGESAHAN

### STUDI KINERJA MODEL TRANSMISI TWORAYGROUND DAN NAKAGAMI PADA PROTOKOL *REACTIVE* DI LINGKUNGAN MOBILE ADHOC NETWORK (MANET) MENGGUNAKAN NETWORK SIMULATOR 2 (NS-2)

### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Arsitektur Jaringan Komputer  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**AZI PRASTYO**  
NRP. 5110100225

Disetujui oleh Pembimbing Tugas Akhir

1. Dr. Eng. Radityo Anggoro, S.Kom, M.Sc. ....  
NIP:198410162008121002 (pembimbing 1)
2. Ir. F.X. Arunanto, M.Sc. ....  
NIP:195701011983031004 (pembimbing 2)



**SURABAYA**  
**DESEMBER, 2016**





**STUDI KINERJA MODEL TRANSMISI TWORAYGROUND  
DAN NAKAGAMI PADA PROTOKOL REAKTIVE DI  
LINGKUNGAN MOBILE ADHOC NETWORK (MANET)  
MENGUNAKAN *NETWORK SIMULATOR 2 (NS-2)***

**Nama Mahasiswa** : Azi Prastyo  
**NRP** : 5110100225  
**Jurusan** : Teknik Informatika FTIf-ITS  
**Dosen Pembimbing I** : Dr. Eng. Radityo Anggoro, S.Kom.,  
M.sc.  
**Dosen Pembimbing II** : Ir. F.X. Arunanto, M.Sc.

**ABSTRAK**

Perangkat *mobile* seperti *notebook*, *handphone*, *tablet* mulai berkembang dengan adanya teknologi nirkabel (*wireless*) saat ini yang menjadi indikator kemajuan peradaban manusia memungkinkan perangkat komunikasi dapat berkomunikasi secara langsung dengan perangkat lainnya dalam posisi bergerak dan tanpa adanya jaringan infrastruktur yang tetap dan bersifat sementara, jaringan semacam ini disebut sebagai MANET (*Mobile Ad Hoc Network*). Pada MANET, setiap *node* bergerak secara bebas, sehingga jaringan dapat mengalami perubahan topologi dengan cepat. Karena *node* dalam MANET memiliki jarak transmisi yang terbatas, beberapa *node* tidak bisa berkomunikasi secara langsung dengan *node* lainnya.

Tugas Akhir ini meneliti skema MANET yang dihasilkan oleh *file node-movement* dan *traffic-pattern* yang telah ada pada distribusi *network simulator*. Penelitian ini menggunakan NS-2 sebagai *network simulator* dengan MANET jenis *protocol reactive* yaitu DSR (*Dynamic Source Routing*) dan AODV (*AdHoc On Demand Distance Vector Routing*) sebagai protokol *routing* yang digunakan serta menggunakan model transmisi TwoRayGround dan Nakagami sebagai pembandingan yang ada pada NS-2. Sistem pada Tugas Akhir ini diuji fungsionalitas dan performanya dengan melalui beberapa skenario yang telah ditentukan. Hasil dari

pengujian adalah peforma model transmisi TwoRayGround dan Nakagami pada protokol *routing* AODV dan DSR di lingkungan MANET dengan menggunakan NS-2. Performa protokol *routing* tersebut diukur berdasarkan *routing overhead*, *packet delivery ratio*, dan *delay* pengiriman paket dari satu *node* ke *node* lainnya (*End-to-end delay*). Pada pengujian menggunakan model *TwoRayGround*, AODV menunjukkan nilai lebih buruk pada *Packet delivery ratio* namun *delay* yang terjadi lebih cepat diatasi.

***Kata kunci: MANET, Nakagami, TwoRayGround, NS-2, DSR, AODV, Network simulator.***

**PERFORMANCE STUDY OF NAKAGAMI AND  
TWRAYGROUND PROPAGATION MODEL FOR *REACTIVE*  
*PROTOCOL* ON MOBILE ADHOC NETWORK USING  
*NETWORK SIMULATOR 2 (NS-2)***

**Name** : Azi Prastyo  
**NRP** : 5110100225  
**Major** : Informatics Department, FTIf-ITS  
**Advisor I** : Dr. Eng. Radityo Anggoro, S.Kom.,  
M.Sc.  
**Advisor II** : Ir. F.X. Arunanto, M.Sc.

**ABSTRACT**

*Mobile device such as notebook, handphone, tablet and others began evolving on wireless technology, which is now as the progress indicator of human civilization, enabling communication device communicate directly with another in moving motion and without any fixed or temporary infrastructure networks, this kind of networks are called as MANET (Mobile Ad Hoc Network). In MANET, each node move freely, so that the network sustain topology change fast. Cause node in MANET posses limited range, some nodes cannot communicate straightly another.*

*Therefore, this final project researched is the scheme of MANET which is produced by file node movement and traffic pattern that already exist on network simulator allocation. This research uses NS-2 as a network simulator with reactive MANET protocols type called DSR (Dynamic Source Routing) and AODV (AdHoc On Demand Distance Vector Routing), along using transmission model such as TwoRayGround and Nakagami as comparison on NS-2. Afterward, the system of this final project functionality and performance are examined pass through some determined scene. The results of the examination is performance*

*of transmission model TwoRayGround and Nakagami in reactive routing protocol (AODV and DSR) in MANET surrounding using NS-2. The performance of routing protocol is measured by routing overhead, packet delivery ratio, and delayed packet sent from a node to another. AODV shows bad results at Packet Delivery Ratio while using TwoRayGround propagation, but it shows best delay handling.*

***Keywords: MANET, Nakagami, TwoRayGround, NS-2, DSR, AODV, Network simulator.***

## KATA PENGANTAR

Bismillahirrahmanirrahim

Alhamdulillahirrahil'alamin, segala puji bagi Allah SWT, atas segala rahmat dan karunia-Nya yang tak terhingga sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **“STUDI KINERJA MODEL TRANSMISI TWRAYGROUND DAN NAKAGAMI PADA PROTOKOL REACTIVE DI LINGKUNGAN MOBILE ADHOC NETWORK (MANET) MENGGUNAKAN NETWORK SIMULATOR 2 (NS-2)”**

Terselesainya Tugas Akhir ini tidak terlepas dari bantuan banyak pihak. Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan sebesar-besarnya kepada pihak-pihak sebagai berikut:

1. Allah SWT, karena limpahan rahmat dan karunia-Nya lah penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Teknik Informatika ITS.
2. Orang tua penulis, Bunda Ewi Sudewi dan Ayah Sudibyo yang tiada henti memberikan semangat, doa serta dukungan penuh kepada penulis selama ini sehingga dapat menyelesaikan Tugas Akhir ini dan perkuliahan di Teknik Informatika ITS Surabaya.
3. Kakak penulis, Widityas Arta Palupi dan Pradirga Gustomo yang telah memberikan dukungan dan doa kepada penulis untuk terus semangat dalam penyelesaian Tugas Akhir ini.
4. Bapak Dr. Eng. Radityo Anggoro S.Kom., M.Sc. dan Bapak Ir. F.X. Arunanto, M.Sc. selaku dosen pembimbing dari penulis yang telah memberikan bimbingan, dukungan, masukan, nasihat dan banyak arahan kepada penulis dalam menyelesaikan Tugas Akhir ini.

5. Bapak Dr. Techn. Ir. Raden Venantius Hari Ginardi, M.Sc. selaku dosen wali dari penulis yang selalu memberi nasihat kepada penulis selama menjalani perkuliahan di Teknik Informatika ITS.
6. Bapak Dr. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Jurusan Teknik Informatika ITS.
7. Ayu yang selalu setia menemani, membantu, memberikan dukungan, doa serta semangat yang tiada henti kepada penulis dengan penuh sabar, suka, duka serta kepedulian yang sangat besar kepada penulis.
8. Sahabat super seperjuangan Dhiya'an Sabila Ramadhani, Guruh Adi Darma, Wiryo Febdila, Bima Bahteradi selalu menguatkan, memberikan dukungan, semangat, nasihat, hiburan dan canda tawa kepada penulis hingga saat ini.
9. Sukijo sahabat yang selalu mendengarkan setiap keluh kesah dan memberi angin segar. Terima kasih telah setia menemani dan mendampingi dikala penulis sedang mengalami kesulitan.
10. Teman-teman TC angkatan 2010, kakak angkatan, dan juga adik angkatan yang telah ramah dan berbaik hati membantu penulis selama berada di Teknik Informatika. Terima kasih atas rasa kekeluargaan yang telah kalian berikan kepada penulis
11. Pihak-pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis ke depannya. Penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum. Semoga Allah

SWT. memberkati dan membalas semua kebaikan yang telah dilakukan.

Surabaya, Desember 2016

Azi Prastyo

*[Halaman ini sengaja dikosongkan]*



## DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
Abstrak .....	ix
Abstract .....	xi
KATA PENGANTAR.....	xiii
1 BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Permasalahan.....	2
1.3. Batasan Permasalahan .....	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Metodologi .....	3
1.7. Sistematika Penulisan .....	5
2 BAB II DASAR TEORI.....	7
2.1. <i>Mobile Ad Hoc Network</i> (MANET) .....	7
2.2. <i>AdHoc On Demand Distance Vector</i> (AODV).....	8
2.3. <i>Dynamic Source Routing</i> (DSR) .....	8
2.4. <i>TwoRayGround</i> .....	9
2.5. Nakagami .....	10
2.6. <i>Network simulator 2</i> (NS-2).....	11
2.7. <i>Node-Movement dan Traffic-connection Pattern</i> .....	12
2.7.1. File <i>Node-Movement</i> ( <i>Movement Generator</i> ).....	12
2.7.2. File <i>Traffic-connection Pattern</i> .....	14
2.8. <i>NS-2 Trace File</i> .....	15
2.9. Awk .....	16
3 BAB III ANALISIS DAN PERANCANGAN SISTEM.....	17
3.1. Deskripsi Umum.....	17
3.2. Perancangan Skenario .....	18
3.2.1. Skenario <i>Node-Movement</i> ( <i>Mobilty generation</i> )	19
3.2.2. <i>Traffic-connection Pattern Generation</i> .....	19
3.3. Perancangan Simulasi pada NS-2.....	20
3.4. Perancangan Metrik Analisis.....	21
3.4.1. <i>Packet delivery ratio</i> (PDR).....	21
3.4.2. <i>End-to-end delay</i> (E2D) .....	21

3.4.3.	Routing Overhead (RO).....	21
4	BAB IV IMPLEMENTASI .....	23
4.1.	Lingkungan Pembangunan Sistem .....	23
4.1.1.	Lingkungan Perangkat Lunak.....	23
4.1.2.	Lingkungan Perangkat Keras.....	23
4.2.	Implementasi Skenario .....	23
4.2.1.	Skenario File Node-Movement (Mobilty generation).....	23
4.2.2.	File Traffic-connection Pattern Generation .....	26
4.3.	Implementasi Simulasi pada NS-2 .....	28
4.4.	Implementasi Metrik Analisis .....	35
4.4.1.	Packet delivery ratio (PDR).....	35
4.4.2.	End-to-end delay (E2D).....	37
4.4.3.	Routing Overhead (RO).....	38
5	BAB V PENGUJIAN DAN EVALUASI .....	41
5.1.	Lingkungan Pengujian.....	41
5.2.	Kriteria Pengujian.....	41
5.3.	Analisis <i>Packet delivery ratio</i> .....	42
5.4.	Analisis <i>End-to-end delay</i> .....	46
5.5.	Analisis <i>Routing Overhead</i> .....	50
6	BAB VI KESIMPULAN DAN SARAN.....	55
6.1.	Kesimpulan.....	55
6.2.	Saran.....	57
7	LAMPIRAN .....	61

## DAFTAR GAMBAR

Gambar 2.1 Skema kerja protokol AODV .....	8
Gambar 2.2 Grafik <i>probability density functio</i> Nakagami .....	11
Gambar 2.3 Nam ( <i>Network Animation</i> ) adalah salah satu kegunaan dari NS-2.....	12
Gambar 2.4 <i>Command line</i> setdest.....	13
Gambar 2.5 Contoh <i>Command Line</i> ‘setdest’ .....	14
Gambar 2.6 <i>Command Line</i> skrip ‘cbrgen.tcl’ .....	14
Gambar 2.7 Contoh <i>Command Line</i> ‘cbrgen.tcl’ .....	15
Gambar 2.8 Contoh <i>Log</i> Pengiriman.....	15
Gambar 2.9 Contoh <i>Log</i> Penerimaan .....	16
Gambar 3.1 Rancangan simulasi pada NS-2 .....	18
Gambar 4.1 <i>Format Command Line</i> ‘setdest’ .....	24
Gambar 4.2 <i>Command line</i> skenario simulasi.....	24
Gambar 4.3 Posisi awal tiap <i>node</i> sumbu X, Y dan Z.....	25
Gambar 4.4 Perpindahan/Pergerakan <i>Node</i> .....	25
Gambar 4.5 GOD yang Berbentuk <i>Array</i> .....	26
Gambar 4.6 <i>Access Point</i> .....	26
Gambar 4.7 <i>Command line</i> untuk menjalankan cbrgen.tcl .....	27
Gambar 4.8 Implementasi pada skrip cbrgen.tcl .....	27
Gambar 4.9 Isi <i>file traffic-connection</i> .....	27
Gambar 4.10 Skrip TCL DSR dengan <i>Interface Type</i> ‘CMUPriQueue’ .....	29
Gambar 4.11 Konfigurasi <i>Trace File</i> , NAM dan Pergerakan <i>Node</i> pada NS-2 .....	30
Gambar 4.12 Konfigurasi pengiriman paket data NS-2 .....	32
Gambar 4.13 <i>Routing Protocol</i> AODV pada Model Propagasi <i>TwoRayGround</i> .....	32
Gambar 4.14 Eksekusi <i>Routing Protocol</i> AODV pada Model Propagasi Nakagami.....	33
Gambar 4.15 Menentukan posisi awal <i>node</i> pada simulasi.....	34
Gambar 4.16 <i>Pseudocode</i> PDR.....	36
Gambar 4.17 Perintah Menjalankan Beserta Hasil PDR.....	37

Gambar 4.18 <i>Pseudocode</i> E2D.....	37
Gambar 4.19 Perintah menjalankan beserta hasil E2D .....	38
Gambar 4.20 <i>Pseudocode</i> RO .....	39
Gambar 4.21 Perintah Menjalankan Beserta Hasil RO .....	39
Gambar 5.1 Grafik PDR <i>Routing Protocol</i> AODV .....	45
Gambar 5.2 Grafik PDR pada <i>Routing Protocol</i> DSR .....	45
Gambar 5.3 Grafik E2D pada <i>Routing Protocol</i> AODV .....	49
Gambar 5.4 Grafik E2D pada <i>Routing Protocol</i> AODV .....	50
Gambar 5.5 Grafik RO pada <i>Routing Protocol</i> DSR.....	53
Gambar 5.6 Grafik RO pada <i>Routing Protocol</i> AODV .....	54
Gambar 7.1 Posisi awal tiap <i>node</i> .....	63
Gambar 7.2 Konfigurasi GOD pada <i>file mobility generator</i> .....	65
Gambar 7.3 Pergerakan setiap <i>node</i> dari potongan Skenario .....	67
Gambar 7.4 Informasi pada GOD dari potongan Skenario .....	69
Gambar 7.5 Koneksi yang digunakan pada <i>cbr-2-0.25.txt</i> .....	70
Gambar 7.6 <i>File .tcl</i> untuk Protokol <i>Routing</i> DSR dengan Model <i>TwoRayGround</i> .....	75
Gambar 7.7 <i>File .tcl</i> untuk Protokol <i>Routing</i> AODV .....	79
Gambar 7.8 Implementasi <i>Packet delivery ratio</i> .....	81
Gambar 7.9 Implementasi <i>Packet delivery ratio</i> .....	82
Gambar 7.10 Implementasi <i>End-to-end delay</i> .....	85
Gambar 7.11 Perintah <i>update</i> pada Terminal Ubuntu .....	85
Gambar 7.12 Tampilan <i>command line</i> ‘ns’ dijalankan.....	86
Gambar 7.13 Tampilan <i>command line</i> ‘nam’ dijalankan .....	86

## DAFTAR TABLE

Table 2.1 Keterangan pada <i>Command Line</i> ‘setdest’ .....	13
Table 2.2 Keterangan <i>Command Line</i> ‘cbrgen.tcl’ .....	14
Table 2.3 Keterangan Contoh <i>Command Line</i> ‘cbrgen.tcl’ .....	15
Table 3.1 Parameter Skenario <i>Node-Movement</i> .....	19
Table 3.2 Parameter <i>Traffic-connection Pattern</i> .....	19
Table 3.3 Parameter pada NS-2.....	20
Table 4.1 Parameter Nam.....	35
Table 5.1 Spesifikasi Laptop yang Digunakan.....	41
Table 5.2 Kriteria Pengujian .....	41
Table 5.3 Hasil ujicoba selama 10 kali pada AODV dengan model Nakagami (PDR) .....	42
Table 5.4 Hasil ujicoba selama 10 kali pada AODV dengan model <i>TwoRayGround</i> (PDR).....	42
Table 5.5 Hasil ujicoba selama 10 kali pada DSR dengan model Nakagami (PDR) .....	43
Table 5.6 Hasil ujicoba selama 10 kali pada DSR dengan model <i>TwoRayGround</i> (PDR).....	43
Table 5.7 Rata-rata PDR Skenario <i>Node-Movement</i> AODV .....	44
Table 5.8 Rata-rata PDR Skenario <i>Node-Movement</i> DSR .....	44
Table 5.9 Hasil ujicoba selama 10 kali AODV dengan model Nakagami (E2D) .....	46
Table 5.10 Hasil ujicoba selama 10 kali AODV dengan model <i>TwoRayGround</i> (E2D) .....	47
Table 5.11 Hasil ujicoba selama 10 kali DSR dengan model Nakagami (E2D) .....	47
Table 5.12 Hasil ujicoba selama 10 kali AODV dengan model <i>TwoRayGround</i> (E2D) .....	48
Table 5.13 Rata-rata Rata-rata E2D Skenario <i>Node-Movement</i> AODV .....	48
Table 5.14 Rata-rata Rata-rata E2D Skenario <i>Node-Movement</i> DSR .....	48

Table 5.15 Hasil ujicoba selama 10 kali AODV dengan model Nakagami (RO) .....	50
Table 5.16 Hasil ujicoba selama 10 kali AODV dengan model <i>TwoRayGround</i> (RO) .....	51
Table 5.17 Hasil ujicoba selama 10 kali DSR dengan model Nakagami (RO) .....	51
Table 5.18 Hasil ujicoba selama 10 kali DSR dengan model <i>TwoRayGround</i> (RO) .....	52
Table 5.19 Rata-rata RO Skenario <i>Node-Movement</i> AODV .....	52
Table 5.20 Rata-rata RO Skenario <i>Node-Movement</i> .....	53

# BAB I PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

## 1.1. Latar Belakang

Teknologi informasi mengalami perkembangan yang begitu pesat, yang didorong oleh keinginan manusia yang ingin serba praktis. Untuk mendukung mobilitas pengguna perangkat *mobile*, teknologi nirkabel (*wireless*) saat ini merupakan solusi yang tepat dalam penerapan teknologi komunikasi. Teknologi jaringan nirkabel (*wireless*) memberikan fasilitas pada perangkat *mobile* untuk dapat saling berkomunikasi dalam suatu area dimana membentuk sebuah jaringan yang bersifat sementara atau yang lebih dikenal dengan *Mobile Ad Hoc Network* (MANET).

Jaringan *Mobile Ad Hoc Network* (MANET) merupakan suatu jaringan yang terorganisir secara mandiri tanpa adanya dukungan infrastruktur. MANET melakukan pembentukan jaringan dan melakukan konfigurasi *node* sendiri sehingga mengurangi interfensi pengguna. MANET sangat cocok apabila diimplementasikan pada wilayah yang tidak memiliki infrastruktur tetap (kabel optik, menara pemancar, dll.). Dalam MANET setiap *mobile device* digambarkan sebagai *node* dan setiap *node* bergerak secara bebas, sehingga jaringan dapat mengalami perubahan topologi dengan cepat. *Node* dalam MANET memiliki jarak transmisi yang terbatas, maka beberapa *node* tidak bisa berkomunikasi secara langsung dengan *node* lainnya. Pada MANET, jalur *routing* mengandung beberapa *hop* dan setiap *node* berfungsi sebagai *router* untuk menentukan ke arah mana tujuan atau rute yang akan mereka pilih. Dalam menentukan setiap jalur *routing* pada MANET terdapat tiga jenis protokol *routing* yang diklasifikasikan menjadi tiga, diantaranya protokol *routing proactive*, *reactive* dan *hybrid* [1].

Protokol *routing reactive* membangun rute melalui pesan permintaan *Route Request Packets* (RREQ) dan *Route Reply* (RREP). Rute akan dibangun jika komunikasi yang diinisiasi oleh *node* yang tidak memiliki rute, maka protokol *routing* ini akan mencoba untuk membangun rute [2].

Dengan menggunakan NS-2 (*Network simulator 2*) implementasi pada lingkungan MANET dapat disimulasikan untuk mempelajari sistem. Implementasi yang dilakukan adalah analisa performa protokol *routing reactive* yaitu AODV (*AdHoc On Demand Distance Vector Routing*) dan DSR (*Dynamic Source Routing*) pada model transmisi *TwoRayGround* dan Nakagami.

Hasil dari Tugas Akhir ini adalah menunjukkan performa model transmisi *TwoRayGround* dan Nakagami pada protokol *routing reactive* (AODV dan DSR) dengan menggunakan *Network simulator 2*. Performa protokol *routing* tersebut diukur berdasarkan *routing overlay*, *packet delivery ratio*, dan *delay* pengiriman paket dari *node* ke *node* lainnya. Performa yang dihasilkan akan menjadi rujukan implementasi MANET pada kegiatan manusia.

## 1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana kinerja protokol *routing* AODV dan DSR pada model transmisi Nakagami?
2. Bagaimana kinerja protokol *routing* AODV dan DSR pada model transmisi *TwoRayGround*?
3. Bagaimana melakukan simulasi dan konfigurasi protokol *routing* AODV dan DSR?
4. Bagaimana melakukan analisis metrik *Packet delivery ratio* (PDR), *Routing Overhead* (RO) dan *End-to-end delay* (E2D) pada simulasi?

## 1.3. Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Protokol *routing* yang diujicobakan adalah AODV dan DSR.



2. Jumlah *node* yang digunakan sebanyak 50 *node*.
3. Kecepatan pergerakan *node* yang digunakan pada simulasi sebesar 5 m/s, 10 m/s dan 15 m/s.
4. Analisis kinerja didasarkan pada *Packet delivery ratio*, *End-to-end delay* dan *Routing Overhead*.

#### **1.4. Tujuan**

Tujuan dari pembuatan Tugas Akhir ini adalah untuk menunjukkan performa model transmisi *TwoRayGround* dan Nakagami menggunakan protokol *routing* AODV dan DSR di lingkungan MANET. Simulasi dipaparkan dengan menggunakan *Network simulator 2*.

#### **1.5. Manfaat**

Manfaat dari pembuatan Tugas Akhir ini adalah mengetahui performa protokol *routing* AODV dan DSR pada model transmisi *TwoRayGround* dan Nakagami di lingkungan MANET, sehingga penerapan *routing protocol* yang baik dapat digunakan. Contoh sensor kebakaran, gempa, dan tsunami.

#### **1.6. Metodologi**

Tugas Akhir ini menggunakan beberapa tahapan dalam proses pengerjaannya. Metodologi yang dilakukan dalam pengerjaan Tugas Akhir ini terdiri atas beberapa tahapan sebagai berikut.

##### **1. Penyusunan Proposal Tugas Akhir**

Pada tahap ini dilakukan penyusunan proposal Tugas Akhir yang berisi tentang deskripsi umum rancangan Tugas Akhir yang akan dibuat. Penyusunan ini terdiri dari menentukan judul Tugas Akhir, latar belakang, manfaat Tugas Akhir, tinjauan pustaka, ringkasan Tugas Akhir, metodologi, serta rencana jadwal kegiatan pengerjaan Tugas Akhir.

##### **2. Studi Literatur**

Pada tahap ini dilakukan studi literatur mengenai *tools* dan metode yang digunakan. Literatur yang dipelajari dan digunakan meliputi buku referensi, artikel, jurnal dan dokumentasi dari internet.

### **3. Implementasi Protokol *Routing***

Tahap ini dilakukan perancangan serta implementasi sistem berdasarkan studi literatur dan pembelajaran terhadap konsep teknologi dari perangkat lunak yang digunakan. Tahap ini merupakan tahap yang penting dimana sistem akan dibentuk sehingga dapat menunjukkan hasil yang sesuai. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

### **4. Pengujian dan Evaluasi**

Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terjadi kesalahan.

### **5. Penyusunan Buku Tugas Akhir**

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi yang telah dibuat. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain sebagai berikut.

1. Pendahuluan
  - a. Latar Belakang
  - b. Rumusan Permasalahan
  - c. Batasan Permasalahan
  - d. Tujuan
  - e. Manfaat
  - f. Metodologi
  - g. Sistematika penulisan
2. Tujuan Pustaka
3. Perancangan Sistem
4. Implementasi
5. Pengujian dan Evaluasi
6. Penutup

### 1.7. Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang dijelaskan sebagai berikut.

- Bab I. Pendahuluan  
Bab ini berisi latar belakang masalah, permasalahan, batasan masalah, tujuan Tugas Akhir, manfaat Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan buku Tugas Akhir.
- Bab II. Tinjauan Pustaka  
Bab ini membahas beberapa literatur penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.
- Bab III. Perancangan  
Bab ini berisi perancangan metode yang nantinya akan diimplementasikan dan dilakukan pengujian dari aplikasi yang akan dibangun.
- Bab IV. Implementasi  
Bab ini membahas implementasi dan rancangan sistem atau desain yang dilakukan pada tahap perancangan. Penjelasan berupa implementasi skenario mobilitas *node-node* pada jaringan *wireless* yang tidak mempunyai *router* tetap yang dibuat menggunakan *file node-movement* dan *traffic-pattern* yang ada pada *network simulator*, konfigurasi sistem dan skrip analisis yang digunakan untuk menguji performa protokol *routing*.
- Bab V. Pengujian dan Evaluasi  
Bab ini menjelaskan tahap pengujian sistem dan performa dalam skenario mobilitas *adhoc* yang dibuat oleh distribusi mobilitas dalam *network simulator*.
- Bab VI. Penutup  
Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan terhadap rumusan masalah yang ada serta saran pengembangan berikutnya.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **DASAR TEORI**

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan Tugas Akhir ini. Penjelasan dasar teori yang digunakan dalam Tugas Akhir ini bertujuan agar dapat memberi definisi tentang alat, protokol *routing*, serta data-data pendukung yang digunakan.

#### **2.1. *Mobile Ad Hoc Network* (MANET)**

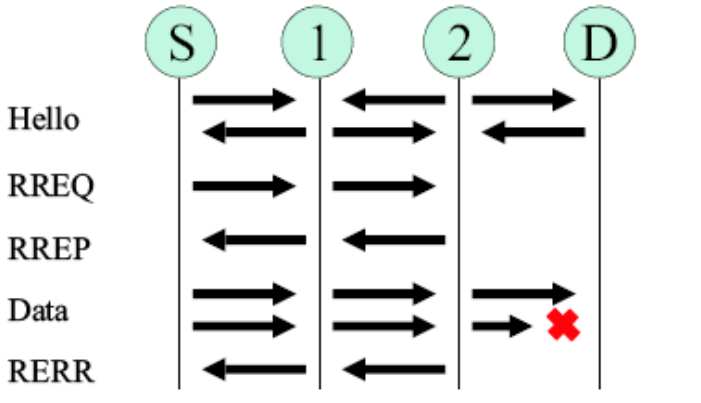
*Mobile Ad Hoc Network* (MANET) adalah kumpulan dari beberapa *wireless node* yang dapat di *set-up* secara dinamis dimana saja dan kapan saja tanpa menggunakan infrastruktur jaringan yang ada. MANET juga merupakan jaringan sementara yang dibentuk oleh beberapa *mobile node* tanpa adanya pusat administrasi dan infrastruktur kabel. Pada MANET, *mobile host* yang terhubung dengan *wireless* dapat bergerak bebas dan juga berperan sebagai *router*. Berdasarkan dokumen *Request for Comments* menjelaskan bahwa terdapat beberapa karakteristik *Mobile Ad Hoc Network* (MANET). Di sana dijelaskan bahwa MANET terdiri dari *mobile platform* (seperti *router* dan perangkat *wireless*) dalam hal ini disebut dengan "*node*" yang bebas berpindah-pindah kemana saja. *Node* tersebut bisa saja berada di pesawat, kapal, mobil dan dimana saja. Setiap *node* dilengkapi dengan *transmitter* dan *receiver wireless* menggunakan antena atau sejenisnya yang bersifat *omnidirectional (broadcast)*, *highly directional (point to point)*, memungkinkan untuk diarahkan, atau kombinasi dari beberapa hal tersebut. *Omnidirectional* maksudnya adalah gelombang radio dipancarkan ke segala arah oleh perangkat *transmitter wireless*. Sedangkan *highly directional* adalah gelombang yang dipancarkan ke satu arah tertentu [1].

Karakteristik dan kompleksitas dari MANET antara lain tidak memiliki infrastruktur, menggunakan *multi-hop routing*, jaringan topologi dinamis, dapat digunakan pada *device* yang heterogen, energi dan *bandwidth* yang terbatas, keterbatasan keamanan dan melakukan pembangunan serta pengaturan jaringan secara mandiri [9].

## 2.2. AdHoc On Demand Distance Vector (AODV)

*AdHoc On Demand Distance Vector* adalah *distance vector routing protocol* yang termasuk dalam klasifikasi *reactive routing protocol*, yang hanya dapat melakukan *request* sebuah rute saat dibutuhkan. AODV yang standar ini dikembangkan oleh C.E. Perkins, E.M. Belding-Royer dan S.Das.

AODV memiliki ciri utama yaitu menjaga *timer-based state* pada setiap *node* sesuai dengan penggunaan *Table routing*. *Table routing* akan kadaluarsa jika jarang digunakan. *Route discovery* pada AODV berupa *Route Request (RREQ)* dan *Route Reply (RREP)*, sedangkan *Route Maintenance* berupa *Data*, *Route Update* dan *Route Error (RRER)* [2].



Gambar 2.1 Skema kerja protokol AODV

## 2.3. Dynamic Source Routing (DSR)

*Dynamic Source Routing* adalah suatu mekanisme *routing* yang didesain untuk konfigurasi jaringan *mode ad hoc* dimana beberapa mekanisme *routing* konvensional tidak dapat berjalan optimal pada jaringan ini. Algoritma *routing* ini menggunakan mekanisme *source routing* dan menerapkan *link state routing*, dimana di setiap paket berisi *route* atau jalan dari *node* asal ke *node* tujuan yang diletakkan pada *header* di dalam paket. Keuntungan penggunaan DSR ini adalah *intermediate node* tidak perlu memelihara secara *up to date* informasi *routing* pada saat

melewatkan paket, karena setiap paket selalu berisi informasi *routing* di dalam *header*-nya. *Routing* jenis ini juga menghilangkan juga proses *periodic route advertisement* dan *neighbor detection* yang dijalankan oleh *routing ad hoc* lainnya. Dibandingkan dengan *on demand routing* lainnya DSR memiliki kinerja yang paling baik dalam hal *throughput*, *routing overhead* (pada paket) dan rata-rata panjang *path*, akan tetapi DSR memiliki *delay* waktu yang buruk bagi proses untuk pencarian *route* baru. Kerugian dari *routing* ini adalah mekanisme *route maintenance* tidak dapat memperbaiki *link* yang rusak atau *down*. Penggunaan *routing* ini akan sangat optimal pada jumlah *node* yang kecil atau kurang dari 200 *node*. Untuk jumlah yang lebih besar akan mengakibatkan *collision* antar paket dan menyebabkan bertambahnya *delay* waktu pada saat akan membangun koneksi baru [3].

#### 2.4. *TwoRayGround*

Dalam *mobile radio channel*, *single direct path* antara *base station* dan *mobile* terkadang hanya peralatan fisik biasa untuk propagasi dan rumus pada *free space* kurang akurat jika dalam penggunaannya berdiri sendiri. Model propagasi *TwoRayGround* merupakan model yang berguna karena berdasar pada optik geometri dan dapat digunakan untuk *direct path* dan refleksi dari *ground* antara *transmitter* dan *receiver*. Model ini dirasa sangat akurat untuk memperkirakan kekuatan sinyal dalam skala luas dengan jarak beberapa kilometer untuk sistem *mobile radio* dengan menggunakan menara yang tinggi. *Power* yang diterima dengan jarak  $d$  diberikan oleh persamaan (1):

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (1)$$

dimana  $h_t$  dan  $h_r$  adalah tinggi dari antenna *transmitter* dan *receiver*, nilai  $L$  diasumsikan sama dengan nilai  $L$  pada propagasi *free space*,  $L = 1$ . Untuk parameter yang lain, masih sama dengan parameter pada propagasi *free space*. Berdasarkan rumus matematika diatas,

*power loss* lebih cepat hilang dibandingkan dengan rumus matematika pada model propagasi *free space* ketika jaraknya bertambah. Model ini tidak memberikan hasil yang baik untuk jarak yang terlalu dekat dikarenakan osilasi yang disebabkan oleh konstruktif dan destruktif yang merupakan kombinasi dari model ini [4].

## 2.5. Nakagami

Model Nakagami-m-m bersifat lebih umum dan dapat diterapkan untuk berbagai kondisi *fading*, tergantung pada parameter-m yang digunakan. Kanal Nakagami-m-m memiliki *probability density function* (pdf) yang dinyatakan sebagai persamaan (2) [5].

$$p_R(R) = \frac{2m^m R^{2m-1}}{\Gamma(m)\Omega^m} \exp\left(-\frac{mR^2}{\Omega}\right) \quad R \geq 0 \quad (2)$$

dimana :

m = parameter *fading*,

m ≥ 0.5 sampai ∞ ( integer positif )

R = amplitudo *fading*

Γ = fungsi gamma

Ω = E [R<sup>2</sup>]

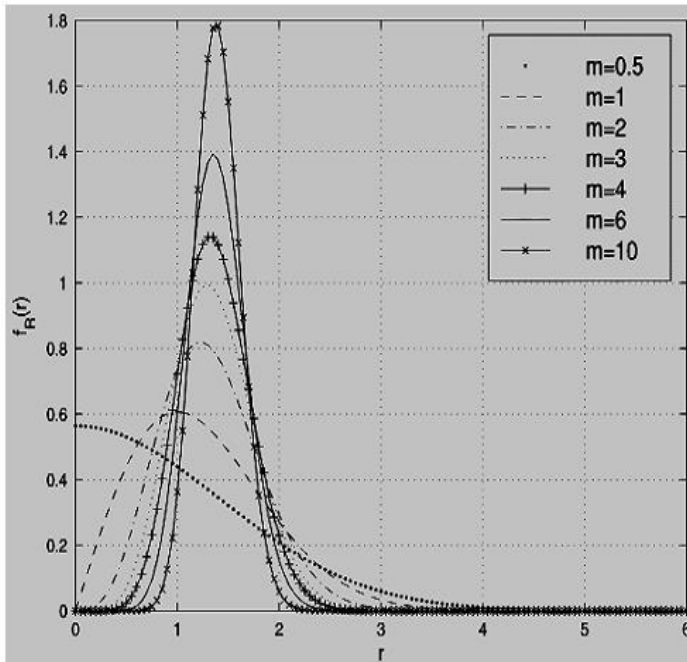
Pengaruh parameter-m pada kanal Nakagami-m-m :

1. Apabila  $m \leq 1$ , maka persamaan 3 menjadi *probability density function* dari kanal *fading* Rayleigh.
2. Apabila  $m > 1$ , merujuk ke kanal *fading* Ricean. Dimana kanal *fading* Ricean memiliki faktor K yang memiliki pengaruh disini. Sehingga nilai m adalah sebagai persamaan (3)[5].

$$m = \frac{(1+k)^2}{1+2k} \quad k \geq 0 \quad (3)$$



Pengaruh dari pada parameter- $m$  dapat dilihat pada grafik *probability density function* (pdf) kanal Nakagami- $m$ - $m$  seperti pada Gambar 2.2

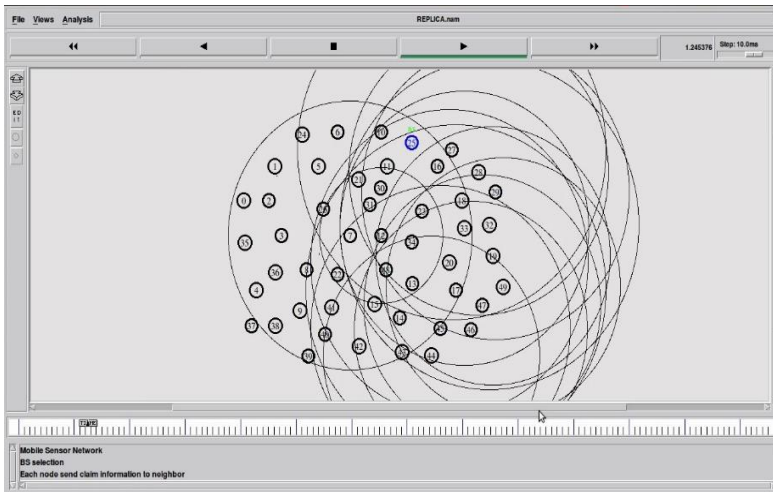


Gambar 2.2 Grafik *probability density function* Nakagami

## 2.6. Network simulator 2 (NS-2)

*Network simulator 2* (NS-2) adalah alat simulasi jaringan yang banyak digunakan dalam mempelajari struktur dinamik dari jaringan komunikasi. Simulasi dari jaringan nirkabel dan protokol seperti algoritma *routing*, TCP dan UDP dapat diselesaikan dengan baik menggunakan *simulator* ini. Beberapa keuntungan menggunakan NS-2 sebagai perangkat lunak adalah *network*

*simulator* dilengkapi dengan *tool* validasi sehingga pembuatan simulasi dengan menggunakan *network simulator* jauh lebih mudah daripada menggunakan *software developer* seperti Delphi atau C++, *network simulator* bersifat *open source* di bawah GPL(*Gnu Public License*), dapat digunakan pada sistem operasi Windows dan sistem operasi Linux. Pada Tugas Akhir ini *network simulator 2* akan dijalankan pada sistem operasi Linux [6].



**Gambar 2.3** Nam (*Network Animation*) adalah salah satu kegunaan dari NS-2

## 2.7. Node-Movement dan Traffic-connection Pattern

### 2.7.1. File Node-Movement (*Movement Generator*)

*Tools* yang disebut 'setdest' dikembangkan oleh CMU (Carnegie Mellon University) untuk menghasilkan pergerakan acak dari *node* dalam jaringan nirkabel. Penggunaan 'setdest' akan mempermudah dalam inisiasi *node-node* pada simulasi mulai dari posisi awal, lokasi berikutnya, serta kecepatan gerak dapat diatur. Lokasi 'setdest' berada pada direktori '~ns/indep-utils/cmu-scengen/setdest/'.

Sebelum dilakukannya simulasi, terlebih dahulu pengguna harus menjalankan perintah ‘setdest’. *Command Line* serta penjelasan parameter yang digunakan tertera pada Gambar 2.4 dan Table 2.1.

```
./setdest [-v version ] [-n num_of_nodes] [-p
pausetime] [-M maxspeed] [-t simtime] [-x maxx] [-y
maxy] > [outdir/movement-file]
```

**Gambar 2.4 Command line setdest**

**Table 2.1 Keterangan pada Command Line ‘setdest’**

Parameter	Keterangan
-v version	Versi ‘setdest’ simulator yang digunakan
-n num	Jumlah <i>node</i> dalam skenario
-p pausetime	Durasi ketika sebuah <i>node</i> tetap diam setelah tiba dilokasi pergerakan. Jika nilai diatur ke 0 maka <i>node</i> tidak akan berhenti ketika tiba dilokasi pergerakan dan akan terus bergerak
-M maxspeed	Kecepatan maksimum sebuah <i>node</i> . <i>Node</i> akan bergerak pada kecepatan acak dalam rentang [0, maxspeed]
-t simtime	Waktu simulasi
-x max x	Panjang maksimum area simulasi
-y max y	Lebar maksimum area simulasi

Hasil dari perintah ‘setdest’ berisi jumlah *node* dan mobilitas yang akan digunakan pada *file* Tcl selama simulasi berjalan. *File Output* dari perintah ‘setdest’ juga terdiri dari statistik perubahan *link* dan rute. Contoh *command line* ‘setdest’ seperti pada Gambar 2.5.

```
./setdest -v 1 -n 50 -p 2.0 -M 20.0 -t 200 -x 500
-y 500 > scen-20-test
```

**Gambar 2.5 Contoh Command Line ‘setdest’**

Perintah ‘setdest’ pada Gambar 3 berarti jumlah *node* yang ada pada skenario tersebut adalah 50 dan bergerak dengan kecepatan maksimum 20.0 m/s dengan jeda pergerakan selama 2 detik. Simulasi akan berhenti pada detik ke 200 dan luas area yang digunakan pada simulasi seluas 500 x 500 m<sup>2</sup>. *File output* disimpan dalam *file* “scen-20-test” [7].

### 2.7.2. File Traffic-connection Pattern

Simulasi pada *network simulator* dibuat acak sehingga dapat menggambarkan kondisi pada dunia nyata. Untuk mendapatkan *random traffic-connection* pada TCP dan CBR, skrip *traffic-skenario pattern generator* digunakan mengkonfigurasi mobilitas antar *node*. Skrip TCL ‘cbrgen’ membantu dalam menghasilkan *traffic load* berupa TCP atau CBR. Skrip TCL ini disimpan di dalam direktori `~ns/indep-utils/cmu-scen-gen`. *Command line* yang digunakan pada skrip ‘cbrgen.tcl’ serta penjelasannya ditunjukkan pada Gambar 2.6 dan Table 2.2 [7].

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed]
[-mc connections] [-rate rate] > traffic-
```

**Gambar 2.6 Command Line skrip ‘cbrgen.tcl’**

**Table 2.2 Keterangan Command Line ‘cbrgen.tcl’**

Parameter	Keterangan
-type cbr tcp	Jenis <i>traffic</i> yang digunakan dapat berupa TCP atau CBR
-nn <i>nodes</i>	Jumlah total <i>node</i>
-mc connection	Jumlah Koneksi
-seed seed	<i>Random seed</i>

-rate rate	Jumlah paket per detik. Pada <i>traffic</i> CBR ukuran paket konstan berukuran 512 <i>byte</i>
------------	------------------------------------------------------------------------------------------------

Pada Gambar 2.7 merupakan contoh penggunaan skrip ‘cbrgen.tcl’ dan Table 2.3 merupakan penjelasan parameter yang digunakan.

```
ns cbrgen.tcl -type cbr -nn 50 -seed 1.0 -mc 1 -rate
0.25 > cbr1.txt
```

**Gambar 2.7 Contoh Command Line ‘cbrgen.tcl’**

**Table 2.3 Keterangan Contoh Command Line ‘cbrgen.tcl’**

Parameter	Keterangan
-type cbr tcp	Jenis <i>traffic</i> yang digunakan CBR
-nn <i>nodes</i>	Jumlah total <i>node</i> 50
-mc connection	Jumlah Koneksi 1
-seed seed	<i>Random seed</i> (1.0)
-rate rate	4.0

## 2.8. NS-2 Trace File

NS-2 *Trace File* merupakan hasil keluaran dari eksekusi NS-2 yang berekstensi .tr. *Trace File* berisi informasi *log* dari semua pengiriman dan penerimaan paket yang terjadi selama simulasi berjalan. *Trace File* memiliki catatan berbagai jenis paket sesuai jenis *routing protocol* yang digunakan. Untuk mendapatkan performa masing-masing protokol setiap baris pada log harus dianalisis terlebih dahulu. Contoh pengiriman dan penerimaan data pada paket ditunjukkan pada Gambar 2.8 dan Gambar 2.9.

```
S 28.000000000 _1_AGT --- 618 cbr 64 [0 0 0 0] -
----- [1:0 0:0 35 0] [28] 0 0
```

**Gambar 2.8 Contoh Log Pengiriman**

```
r 28.011627928 _0_ AGT --- 618 cbr 64 [14a 0 18
500] ----- [1:0 0:0 30 0] [28] 6 0
```

**Gambar 2.9 Contoh Log Penerimaan**

Huruf “s” di kolom pertama menandakan pengiriman paket sedangkan huruf “r” menandakan penerimaan paket. Pada kolom kedua menandakan detik dilaksanakannya pengiriman atau penerimaan, pada kolom ketiga merupakan *node* tempat terjadinya proses pengiriman atau penerimaan paket, kolom ke empat menandakan pengiriman paket data, kolom ke lima tempat terjadinya *event* spesial seperti *collision*, kolom ke enam merupakan *id* unik paket, kolom ke tujuh berisi *type* paket yang dikirim yaitu *cbr*, kolom ke delapan adalah ukuran paket dalam *byte* sebesar 64.

## 2.9. Awk

Awk adalah bahasa pemrograman yang digunakan untuk melakukan manipulasi data dan membuat laporan. Nama awk diambil dari nama akhir pembuatnya yaitu Alfred Aho, Peter Weinberger, dan Brian Kerningham. Awk mempunyai beberapa versi yaitu awk versi pertama (awk), kemudian *nawk* atau *new version of awk* dan versi awk yang merupakan implementasi dari GNU Project yang disebut dengan *gawk* dengan standar POSIX 1003.2 untuk bahasa pemrograman dan utilitas, *BWK* awk, *xgawk*, *mawk* dan lain sebagainya. Pada beberapa sistem seperti linux, awk di-link-kan ke versi terbaru yang digunakan. Misalnya dalam distribusi Linux redhat, *gawk* merupakan link dari awk [8].

Awk dapat digunakan baik dalam *command line* maupun dimasukkan dalam sebuah skrip dengan melakukan *scan* baris per baris yang diperoleh dari standar *input*, *file* maupun *output* proses. Pada Tugas Akhir ini AWK digunakan mendapatkan skrip dalam penghitungan *Packet delivery ratio* (PDR), *End-to-end delay* (E2D), dan *Routing Overhead* (RO) dari hasil *trace* pada *Network simulator 2*.

## **BAB III**

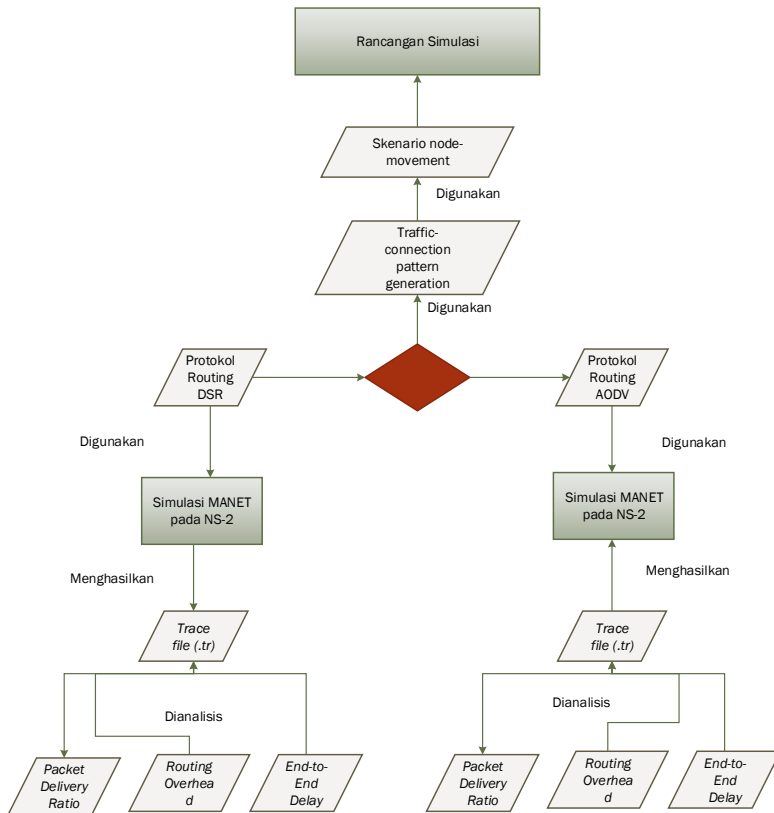
### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tahap perancangan dari sistem yang akan dibangun. Materi yang dibahas secara khusus deskripsi umum sistem, perancangan skenario, alur, serta gambaran implementasi sistem yang diterapkan *Network simulator 2* dan modul kelengkapannya.

#### **3.1. Deskripsi Umum**

Tugas Akhir ini bertujuan untuk melakukan analisis tentang performa dua protokol reaktif yaitu AODV dan DSR pada dua model transmisi yaitu Nakagami dan TwoRayGround. Implementasi yang dilakukan dengan menggunakan skenario MANET, yaitu dengan cara menjalankan Mobility Generator yang bersifat *Random way point*. Pada *Network simulator 2* (NS-2) *file node-movement* dan *file traffic-connection* di-generate sehingga menghasilkan data acak yang digunakan untuk simulasi. Rancangan simulasi dapat dilihat pada Gambar 3.1.

Masing-masing simulasi dilakukan dengan 2 *routing protocol* (AODV dan DSR) dan 2 model transmisi (Nakagami dan *TwoRayGround*) yang berbeda sehingga menjadi 4 hasil performa yang dapat dibandingkan. Pada tiap skenario, kecepatan maksimum pergerakan *node* satu ke *node* lainnya bervariasi yaitu 5 m/s, 10 m/s dan 15 m/s. Hasil proses uji coba dari tiap skenario akan menghasilkan *file trace* yang kemudian akan dilakukan analisis perhitungan *Packet Delivery Ratio* (PDR), *End-to-End delay* (E2D), dan *Routing Overhead* (RO). Dari hasil perhitungan tersebut kemudian dianalisis dan menghasilkan perbandingan performa dari kedua protokol *routing* pada masing-masing model transmisi.



**Gambar 3.1 Rancangan simulasi pada NS-2**

### 3.2. Perancangan Skenario

Perancangan skenario uji coba dengan menggunakan *mobility generation* yang bersifat *Random way point*. Kemudian menggunakan *file traffic-connection* untuk membuat koneksi pada masing-masing *node*. Pada Tugas Akhir ini pembuatan skenario dibagi menjadi 3 kecepatan maksimum yang berbeda yaitu 5 m/s, 10 m/s dan 15 m/s. Banyak koneksi yang digunakan adalah sebanyak 2 *node* untuk menentukan *node* pengirim dan *node*



penerima paket. Perancangan skenario pada mobility generator adalah sebagai berikut:

### 3.2.1. *Skenario Node-Movement (Mobilty generation)*

Skenario *mobilty generation* dibuat dengan menjalankan *tools* yang disebut ‘setdest’ yang akan menghasilkan *output* yang digunakan dalam *file* Tcl selama simulasi berjalan pada NS-2. Table 3.1 menunjukkan spesifikasi *node-movement* yang digunakan.

**Table 3.1 Parameter Skenario Node-Movement**

No.	Parameter	Spesifikasi
1	Jumlah <i>Node</i>	50
2	Waktu Simulasi	100 detik
3	Area	850 x 850
4	Kecepatan Maksimal	-5m/s -10m/s -15m/s
5	Sumber <i>Traffic</i>	CBR
6	Waktu Jeda	10 detik
7	Ukuran Paket	512 <i>byte</i>
8	<i>Rate</i> Paket	0.25 paket per detik
9	Jumlah Maksimal Koneksi	1
10	Model Mobilitas yang Digunakan	<i>Random way point</i>

### 3.2.2. *Traffic-connection Pattern Generation*

*Traffic-connection* dibuat dengan menjalankan *file* Tcl yang bernama *cbrgen.tcl*. *Output* dari program tersebut akan digunakan pada simulasi sebagai penghubung antara *node*. Table 3.2 menunjukkan spesifikasi yang digunakan pada *traffic-connection pattern*.

**Table 3.2 Parameter Traffic-connection Pattern**

No.	Parameter	Spesifikasi
1	-type cbr tcp	CBR

2	-nn nodes	2
3	-seed seed	1.0
4	-mc connection	1
5	-rate rate	0.25
6	Agent	UDP

### 3.3. Perancangan Simulasi pada NS-2

Perancangan kode pada NS-2 dengan konfigurasi MANET, dilakukan dengan cara menggabungkan skenario mobilitas dan *traffic-connection* serta skrip TCL dengan parameter-parameter yang telah ditentukan. Parameter simulasi ditunjukkan pada Table 3.3.

**Table 3.3 Parameter pada NS-2**

No.	Parameter	Spesifikasi
1	<i>Network simulator</i>	NS – 2.34
2	<i>Routing Protocol</i>	AODV dan DSR
3	Waktu Simulasi	100 detik
4	Waktu Pengiriman Paket Data	- <i>TwoRayGround</i> = 0 - 100 detik -Nakagami = 0 – 100 detik
5	Area	850m x 850 m
6	Banyak <i>node</i>	50
7	Radius transmisi	100 m
8	Tipe koneksi	UDP
9	Tipe data	<i>Constant Bit Rate (CBR)</i>
10	<i>Source / Destination</i>	Statik ( <i>Node 1 / Node 2</i> )
11	Kecepatan generasi paket	1 paket per detik
12	Ukuran paket data	512 bytes
13	Protokol MAC	IEEE 802.11
14	Mode Transmisi	- <i>TwoRayGround</i> - Nakagami
15	Tipe Antenna	OmniAntenna

16	Type <i>Interface Queue</i>	Droptail/PriQueue
17	Type Peta	MANET ( <i>random way point</i> )
18	Type kanal	<i>Wireless channel</i>
19	Type <i>trace</i>	<i>Old Wireless Format Trace</i>

### 3.4. Perancangan Metrik Analisis

Metrik yang akan dianalisis pada Tugas Akhir ini adalah *Packet delivery ratio* (PDR), *End-to-end delay* (E2D), dan *Routing Overhead* (RO). Berikut merupakan penjelasannya:

#### 3.4.1. *Packet delivery ratio* (PDR)

PDR dihitung dari perbandingan antara paket yang dikirim dengan paket yang diterima. PDR dihitung dengan menggunakan persamaan sebagai berikut:

$$PDR = \frac{\sum received}{\sum sent} * 100$$

Dimana *received* adalah banyaknya paket data yang diterima dan *sent* adalah banyaknya paket yang dikirimkan.

#### 3.4.2. *End-to-end delay* (E2D)

*End-to-end delay* dihitung dari rata-rata delay antar waktu paket diterima dan waktu paket dikirim. *End-to-end delay* dihitung dengan menggunakan persamaan sebagai berikut:

$$End\ to\ End\ Delay = \frac{\sum_{i \leq sent}^{i=0} t_{received[i]} - t_{sent[i]}}{sent}$$

Dimana  $t_{received[i]}$  adalah waktu penerimaan paket dengan urutan/id ke  $i$ , sedangkan  $t_{sent[i]}$  adalah waktu pengiriman paket dengan urutan/id ke  $i$ , dan  $sent$  adalah banyaknya paket data yang dikirimkan.

#### 3.4.3. *Routing Overhead* (RO)

*Routing Overhead* adalah jumlah paket kontrol *routing* yang ditransmisikan per data paket yang terkirim ke tujuan selama

simulasi terjadi. RO dihitung berdasarkan paket *routing* yang ditransmisikan. Baris yang mengandung RO pada *trace file* ditandai dengan paket yang bertipe *send* (s) / *forward* (f) dan terdapat masing-masing *header* paket dari protokol AODV dan DSR.

## **BAB IV IMPLEMENTASI**

Bab ini membahas tentang implementasi dari perancangan sistem. Implementasi yang dijelaskan antara lain lingkungan pembangunan perangkat lunak, implementasi skenario, implementasi pada NS-2, dan implementasi analisis metrik.

### **4.1. Lingkungan Pembangunan Sistem**

Pembangunan sistem dilakukan pada lingkungan pengembangan sebagai berikut:

#### **4.1.1. *Lingkungan Perangkat Lunak***

Perangkat lunak yang digunakan untuk pengembangan sistem dalam Tugas Akhir ini adalah sebagai berikut:

- Sistem Operasi Ubuntu 16 64 bit untuk lingkungan NS-2;
- *Network simulator 2* versi 2.35;
- Patch AODV dan DSR.

#### **4.1.2. *Lingkungan Perangkat Keras***

Spesifikasi perangkat keras yang digunakan untuk implementasi perangkat lunak pada Tugas Akhir ini ada sebagai berikut:

- *Processor* Intel(R) Core(TM) i7-4700HQ CPU @ 2.4GHz.
- Media penyimpanan 750GB;
- RAM sebesar 8 GB DDR4.

### **4.2. Implementasi Skenario**

Implementasi skenario mobilitas pada lingkungan MANET dilakukan dengan cara meng-*generate tools* yang disediakan pada NS-2. Fitur yang diberikan secara acak dengan jumlah *node* yang ditentukan, membuat simulasi menggambarkan lingkungan MANET. Implementasinya adalah sebagai berikut:

#### **4.2.1. *Skenario File Node-Movement (Mobilty generation)***

Implementasi skenario pada *mobilty generation* menggunakan *tool* yang dimiliki NS-2 yaitu 'setdest'. *File* skenario *node-movement (mobilty generation)* digunakan untuk setiap simulasi dengan jeda waktu tertentu. Dalam mempelajari mobilitas

dari masing-masing protokol pada setiap model transmisi, digunakan kecepatan yang berbeda-beda pada setiap simulasi. Program ‘setdest’ pada NS-2 bersifat *Random way point* sehingga hasil *output*-nya acak. *Command line* yang digunakan untuk meng-*generate* skenarionya seperti pada Gambar 4.1.

```
./setdest [-v version] [-n num of nodes] [-p pausetime] [-M
maxspeed] [-t simtime] [-x maxx] [-y maxy] > [outdir/movement-file]
```

**Gambar 4.1 Format Command Line ‘setdest’**

Skenario yang digunakan pada Tugas Akhir ini yaitu dengan jumlah *node* 50, maksimal jumlah koneksi antara masing-masing *node* 2, waktu jeda 10 detik, kecepatan yang digunakan masing-masing 5 m/s, 10 m/s, dan 15 m/s, waktu simulasi yaitu 100 detik, sedangkan panjang dan lebar area simulasi sebesar 850m. *File output* yang dihasilkan setelah menjalankan perintah ‘setdest’ secara otomatis tersimpan pada direktori “~ns /indep-utils/CMU-scen-gen/setdest/”. Pada Gambar 4.2 memperlihatkan implementasi 50 *node* dan kecepatan maksimal yang berbeda. Masing-masing kecepatan maksimal pergerakan *node* akan di-*generate* sebanyak 10 kali dan menghasilkan skenario yang dapat menggambarkan situasi *node-node* pada MANET.

```
./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x
850 -y 850 > sc1-5.txt
./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x
850 -y 850 > sc1-10.txt
./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x
850 -y 850 > sc1-15.txt
```

**Gambar 4.2 Command line skenario simulasi**

*Output* dari penggunaan ‘setdest’ untuk menciptakan skenario acak, menunjukkan posisi awal setiap *node* dalam area simulasi. X, Y dan Z adalah perwakilan dari sumbu yang digunakan Gambar 4.3.

```
#
# nodes: 50, pause: 10.00, max speed: 5.00, max x: 500.00, max y: 500.00
#
$node_(0) set X_ 308.013863877133
$node_(0) set Y_ 394.501613233317
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 423.371522934473
$node_(1) set Y_ 267.830788322912
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 9.746416594660
$node_(2) set Y_ 6.000948196323
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 329.728474394307
$node_(3) set Y_ 236.646824335339
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 488.179561808088
$node_(4) set Y_ 457.946864830551
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 14.829582205171
$node_(5) set Y_ 329.996487973088
$node_(5) set Z_ 0.000000000000
$node_(6) set X_ 197.785578816990
$node_(6) set Y_ 478.714062616351
$node_(6) set Z_ 0.000000000000
$node_(7) set X_ 263.033070999868
$node_(7) set Y_ 71.610892865589
$node_(7) set Z_ 0.000000000000
```

**Gambar 4.3** Posisi awal tiap *node* sumbu X, Y dan Z

```
$ns_ at 10.000000000000 "$node_(0) setdest 160.044527214314 319.724148954876 2.484608400101"
$ns_ at 10.000000000000 "$node_(1) setdest 362.903473895025 487.819560213589 0.598236247765"
$ns_ at 10.000000000000 "$node_(2) setdest 2.681357566668 201.613282886232 1.086565131789"
$ns_ at 10.000000000000 "$node_(3) setdest 67.657441034156 146.244352548110 1.905630703869"
$ns_ at 10.000000000000 "$node_(4) setdest 451.280082079176 271.391160541531 3.967153226297"
$ns_ at 10.000000000000 "$node_(5) setdest 43.736529908422 52.194505081618 4.549453363588"
$ns_ at 10.000000000000 "$node_(6) setdest 478.019610961918 199.065545086198 1.218548188409"
$ns_ at 10.000000000000 "$node_(7) setdest 153.487380693056 334.519740329149 4.326094072272"
$ns_ at 10.000000000000 "$node_(8) setdest 275.251198269485 378.928105537092 1.033067505313"
$ns_ at 10.000000000000 "$node_(9) setdest 185.435416193345 439.603256047116 3.960396093279"
$ns_ at 10.000000000000 "$node_(10) setdest 409.231566502751 185.402832940172 2.336097001310"
$ns_ at 10.000000000000 "$node_(11) setdest 308.420078398515 27.543875791395 2.004427280491"
$ns_ at 10.000000000000 "$node_(12) setdest 16.581432416322 482.214239961599 0.456832273635"
$ns_ at 10.000000000000 "$node_(13) setdest 499.706078422923 197.253142839450 0.770650022965"
$ns_ at 10.000000000000 "$node_(14) setdest 30.712996816333 144.406232525710 3.067329496379"
$ns_ at 10.000000000000 "$node_(15) setdest 385.922597318865 404.523733337628 4.881099772236"
```

**Gambar 4.4** Perpindahan/Pergerakan *Node*

Pergerakan *node* diperlihatkan pada Gambar 4.4 dan dijalankan selama simulasi. Posisi awal dan pergerakan *node* berikutnya telah diatur pada *file* skenario. Pada potongan dari *sc1.txt* mendefinisikan bahwa *node* (0) pada detik ke 10 mulai bergerak ke arah tujuan (160.04, 319.72) dengan kecepatan 2.5m/s.

GOD atau *General Operations Director* berfungsi sebagai media penyimpan informasi global tentang jumlah dan pergerakan *node*. Pada Gambar 4.5 fungsi GOD digunakan sebagai penyampai *array* terpendek dari *hop* yang diperlukan dari satu *node* ke *node* yang lain.

```
$god_set-dist 0 1 1
$god_set-dist 0 2 3
$god_set-dist 0 3 1
$god_set-dist 0 4 1
$god_set-dist 0 5 2
$god_set-dist 0 6 1
$god_set-dist 0 7 2
$god_set-dist 0 8 1
$god_set-dist 0 9 2
$god_set-dist 0 10 1
$god_set-dist 0 11 2
$god_set-dist 0 12 1
$god_set-dist 0 13 1
$god_set-dist 0 14 2
```

**Gambar 4.5** GOD yang Berbentuk *Array*

Kemudian informasi yang dimuat pada objek GOD yaitu jalan terpendek dari sebuah *node* ke *node* yang lain. Pada Gambar 4.6 menunjukkan informasi pada objek GOD yaitu perubahan *node* 6 ke *node* 40 menjadi 2 *hop* pada detik ke 10.670817767092.

```
Sns_at 10.101091746617 "$god_set-dist 12 24 1"
Sns_at 10.200107532099 "$god_set-dist 20 32 1"
Sns_at 10.276408236989 "$god_set-dist 7 47 1"
Sns_at 10.384471335595 "$god_set-dist 32 41 1"
Sns_at 10.417140904013 "$god_set-dist 18 23 1"
Sns_at 10.445412204693 "$god_set-dist 6 9 2"
Sns_at 10.445412204693 "$god_set-dist 9 13 2"
Sns_at 10.445412204693 "$god_set-dist 9 23 1"
Sns_at 10.445412204693 "$god_set-dist 9 25 2"
Sns_at 10.445412204693 "$god_set-dist 9 28 2"
Sns_at 10.579076655818 "$god_set-dist 35 47 1"
Sns_at 10.670817767092 "$god_set-dist 6 40 2"
Sns_at 10.670817767092 "$god_set-dist 40 45 1"
Sns_at 10.688769534171 "$god_set-dist 15 17 1"
Sns_at 10.692512607350 "$god_set-dist 6 24 1"
Sns_at 10.766427740070 "$god_set-dist 8 16 1"
```

**Gambar 4.6** *Access Point*

#### 4.2.2. *File Traffic-connection Pattern Generation*

Implementasi *traffic-connection generation* untuk TCP dan CBR dilakukan dengan menggunakan skrip *traffic-connection generator*. Skrip *traffic-connection generator* berada di direktori `~ns/indep-utils/cmu-scen-gen` dan berbentuk *file* dengan nama



cbrgen.tcl. *Command line* yang digunakan untuk menghasilkan gerakan pada *node* pada Gambar 4.7.

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed]
[-mc connections] [-rate rate] > traffic-
```

**Gambar 4.7 Command line untuk menjalankan cbrgen.tcl**

Implementasi koneksi TCP/CBR secara acak dengan menjalankan skrip cbrgen.tcl. Parameter yang digunakan antara 2 *node* maksimal berjumlah 1 dengan nilai *seed* 1.0 dan jumlah paket per detik sebanyak 0.25 dan disimpan dengan nama *file* cbr1.txt. Gambar 4.8 memperlihatkan *command* yang dijalankan pada cbrgen.tcl

```
ns cbrgen.tcl -type cbr -nn 2 -seed 1.0 -mc 1 -rate
0.25 > cbr1.txt
```

**Gambar 4.9 Implementasi pada skrip cbrgen.tcl**

```
#
# nodes: 2, max conn: 1, send rate: 0.25, seed:
1.0
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
```

**Gambar 4.8 Isi file traffic-connection**

Keluaran yang dihasilkan setelah menjalankan *file* *cbrgen.tcl*, Gambar 4.9 merupakan potongan dari *file* *cbr1.txt*. Koneksi CBR yang dihasilkan menggunakan *UDP Agent*, koneksi UDP merupakan koneksi antara *node* 1 dan 2. Pengiriman paket data yang dilakukan setiap satu detik dengan ukuran paket sebesar 512 *byte* dan maksimal pengiriman paket 10.000.

### 4.3. Implementasi Simulasi pada NS-2

Simulasi yang akan dijalankan adalah dua protokol *reactive* yaitu AODV dan DSR masing-masing dijalankan dengan dua model transmisi Nakagami dan *TwoRayGround*. Skenario *node-movement* dan *traffic-connection generation* yang akan digunakan dalam simulasi disimpan dalam bentuk *.txt*. Untuk dapat berjalan dengan menggunakan NS-2 parameter yang dibutuhkan dibuat dengan bahasa Tcl.

Gambar 4.10 merupakan *set* awal atau konfigurasi yang dibutuhkan ketika menjalankan simulasi MANET pada NS-2. Baris pertama merupakan tipe *channel* yang digunakan yaitu *Wireless Channel*, baris kedua menunjukkan tipe transmisi yang digunakan pada Tugas Akhir ini yaitu Nakagami dan *TwoRayGround*. Perlu diperhatikan untuk baris ke lima pada DSR berbeda dengan AODV. Pada AODV nilai *interface type* “Queue/DropTail/PriQueue” sedangkan untuk DSR nilai *interface type*-nya adalah “CMUPriQueue”. *Link layer*, *antenna*, dan jumlah maksimal paket pada *interface queue* diatur pada baris selanjutnya. Baris ke 9 hingga baris ke 17 berturut-turut *routing protocol* panjang dan lebar x (*axis*) dan y (*axis*) yang digunakan yaitu 500 meter, *seed* 0.0, nama *trace file*, jumlah *node* yang ada, skenario yang digunakan ketika simulasi ‘*sc3.txt*’, *traffic-connection* yang digunakan ketika simulasi yaitu dengan nama *file* ‘*cbr-50*’, dan simulasi akan berhenti pada detik ke 100.

```

set val(ifq)           Queue/DropTail/PriQueue
set val(ll)           LL
set val(ant)          Antenna/OmniAntenna
set val(ifqlen)       100
set val(rp)           DSR
set val(x)            500
set val(y)            500
set val(seed)         0.0
set val(tr)           D-Nak.tr
set val(nn)           50
set val(cp)           "sc3.txt"
set val(sc)           "cbr-50"
set val(stop)         100.0
set ns_                [new Simulator]

}

if { $val(rp) == "DSR" } {
set val(ifq)           CMUPriQueue
} else {
set val(ifq)           Queue/DropTail/PriQueue
}

set tracefd [open D-Nak3.tr w]
$ns_ trace-all $tracefd
set namtrace [open D-Nak3.nam w]
    $ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
Phy/WirelessPhy set RXThresh_ 1.42681e-08 ;#100m

#Sns_use-newtrace

```

**Gambar 4.10** Skrip TCL DSR dengan *Interface Type* ‘CMUPriQueue’

Transmission range memiliki banyak pilihan dengan jarak yang bervariasi, akan tetapi pengaturan *transmission range* yang digunakan pada simulasi ini adalah *RXThresh\_* (*Receiver Sensitivity Threshold*) dengan nilai  $1.42681e-08$  yang memiliki jarak jangkauan sejauh 100 meter. Nilai ‘Phy/WirelessPhy’ pada skrip tcl diganti dengan ‘RXThresh\_  $1.42681e-08$ ’.

```

set tracefd [open A-Trg.tr w]
$ns_ trace-all $tracefd
set namtrace [open A-Trg.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x)
$val(y)
Phy/WirelessPhy set RXThresh_ 1.42681e-
08 ;#100m
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
    $ns_ node-config -adhocRouting $val(rp) \
                    -llType $val(ll) \
                    -macType $val(mac) \
                    -ifqType $val(ifq) \
                    -ifqLen $val(ifqlen) \
                    -antType $val(ant) \
                    -propType $val(prop) \
                    -phyType $val(netif) \
                    -channel $chan_1_ \
                    -topoInstance $topo \
                    -agentTrace ON \
                    -routerTrace ON \
                    -macTrace
OFF
    for {set i 0} {$i < $val(nn) } {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion
0        ;
    }

```

**Gambar 4.11 Konfigurasi *Trace File*, *NAM* dan *Pergerakan Node* pada *NS-2***

Skrip yang ditunjukkan pada Gambar 4.11 merupakan pengaturan simulasi. *Set tracefd* dan *set namtrace* merupakan

pengaturan *trace file* dan nam *file* yaitu animasi yang menggambarkan situasi ketika simulasi. *Trace file* yang digunakan pada Tugas Akhir ini adalah *old trace* format. *Set topo* merupakan pengaturan objek topografi sesuai dengan luas area simulasi yang telah ditentukan. *Create-god* dan *node-config -channelType* merupakan konfigurasi pada *node-node* yang akan dibuat. *Create-god* mengimplementasikan parameter pada *set val(nn)* sedangkan pada *node-config -channelType* merupakan implementasi dari konfigurasi yang telah ditentukan, seperti *link layer*, tipe *mac*, dan tipe transmisi.

Skrip yang ditunjukkan pada Gambar 4.12 merupakan bagian terakhir skrip yang digunakan untuk menginisialisasi penempatan awal *node-node* yang dibuat pada skenario *node-movement (mobilty generation)*, pergerakan *node* selama simulasi berjalan dan melakukan konfigurasi pengiriman yang akan dilakukan dan hasilnya dicatat pada *trace file*.

```
puts "Loading connection pattern..."
source $val(cp)

puts "Loading skenario file..."
source $val(sc)

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ initial_node_pos $node_($i) 50
}

for {set i 0} {$i < $val(nn) } {incr i} {

    $ns_ at $val(stop).000000001 "$node_($i)
reset";
```

```

}
$ns_ at $val(stop).000000001 "puts \"NS
EXITING...\"; $ns_ halt"

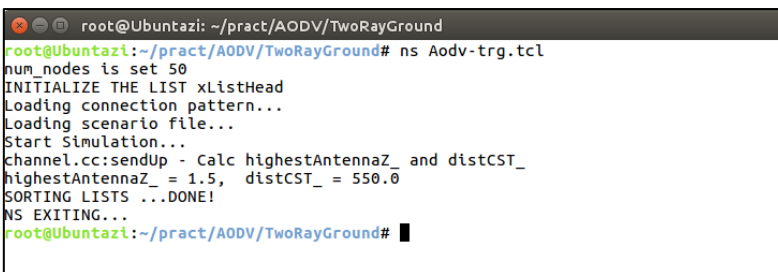
proc stop {} {
    global ns_ tracefd
    namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
}

puts "Start Simulation..."
$ns_ run

```

**Gambar 4.12 Konfigurasi pengiriman paket data NS-2**

Pada Gambar 4.13 dan Gambar 4.14 merupakan contoh eksekusi *file* .tcl menggunakan transmisi Nakagami dan *TwoRayGround* pada *routing protocol* AODV. Jumlah *node* yang digunakan 50 yang bersifat acak sesuai dengan skenario *node-movement* yang telah di-generate.



```

root@Ubuntazi: ~/pract/AODV/TwoRayGround
root@Ubuntazi:~/pract/AODV/TwoRayGround# ns Aodv-trg.tcl
num_nodes is set 50
INITIALIZE THE LIST xListHead
Loading connection pattern...
Loading scenario file...
Start Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
NS EXITING...
root@Ubuntazi:~/pract/AODV/TwoRayGround# █

```

**Gambar 4.13 Routing Protocol AODV pada Model Propagasi TwoRayGround**

```

root@Ubuntazi: ~/pract/AODV/Nakagami
root@Ubuntazi:~/pract/AODV/Nakagami# ns Aodv-nak.tcl
num_nodes is set 50
INITIALIZE THE LIST xListHead
Loading connection pattern...
Loading scenario file...
Start Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 17976931348623157081452742373170435679807056
75258449965989174768031572607800285387605895586327668781715404589535143824642343
21326889464182768467546703537516986049910576551282076245490090389328944075868508
45513394230458323690322294816580855933212334827479782620414472316873817718091929
9881250404026184124858368.0
SORTING LISTS ...DONE!
NS EXITING...
root@Ubuntazi:~/pract/AODV/Nakagami# █

```

**Gambar 4.14** Eksekusi *Routing Protocol AODV* pada Model Propagasi Nakagami

Hasil yang diperoleh dari eksekusi *file Tcl* berupa *trace file* berbentuk *.tr* dan *file animasi* pengiriman paket data berbentuk *.nam*. *Trace file* kemudian akan dianalisis parameter-parameternya sehingga menghasilkan data yang diperlukan untuk menghitung PDR, E2D dan RO. Gambar 4.15 menunjukkan tampilan hasil *file .nam*.

```

n -t * -s 0 -x 28.031252774060999 -y
738.84433257151898 -z 0 -z 50 -v circle -
c black
n -t * -s 1 -x 424.96330326943797 -y
467.77793799941702 -z 0 -z 50 -v circle -
c black
n -t * -s 2 -x 372.87750540033898 -y
749.24382166767396 -z 0 -z 50 -v circle -
c black
n -t * -s 3 -x 745.36056364537205 -y
332.72358658874998 -z 0 -z 50 -v circle -
c black
n -t * -s 4 -x 197.771568046065 -y
193.70963192582201 -z 0 -z 50 -v circle -
c black

```

```
n -t * -s 5 -x 281.975876097796 -y
159.25743509445999 -z 0 -z 50 -v circle -
c black
n -t * -s 6 -x 844.18541219564304 -y
729.66748096999595 -z 0 -z 50 -v circle -
c black
n -t * -s 7 -x 539.31845320114803 -y
295.53993726668602 -z 0 -z 50 -v circle -
c black
n -t * -s 8 -x 302.18931498363997 -y
748.591623643194 -z 0 -z 50 -v circle -c
black
n -t * -s 9 -x 61.258648997591997 -y
817.65536631511395 -z 0 -z 50 -v circle -
c black
n -t * -s 10 -x 809.55228272054205 -y
798.12728421773704 -z 0 -z 50 -v circle -
c black
n -t * -s 11 -x 445.82459752715999 -y
168.56121948471599 -z 0 -z 50 -v circle -
c black
n -t * -s 12 -x 713.07610527608301 -y
253.41971016286399 -z 0 -z 50 -v circle -
c black
n -t * -s 13 -x 79.396945509259993 -y
158.03453873870501 -z 0 -z 50 -v circle -
c black
n -t * -s 14 -x 304.41652856968301 -y
70.720431886113005 -z 0 -z 50 -v circle -
c black
n -t * -s 15 -x 357.16075771242299 -y
228.272636358325 -z 0 -z 50 -v circle -c
black
```

**Gambar 4.15 Menentukan posisi awal *node* pada simulasi**



**Table 4.1 Parameter Nam File**

No.	Parameter	Keterangan
1	<i>n node</i>	<i>node</i>
2	<i>-t time</i>	Waktu
3	<i>-s node id</i>	Id pada <i>node</i>
4	<i>-x x location</i>	Posisi <i>x node</i>
5	<i>-y y location</i>	Posisi <i>y node</i>
6	<i>-Z node start</i>	Id <i>node</i> awal
7	<i>-z size of node</i>	Banyaknya jumlah <i>node</i>
8	<i>-v shape</i>	Bentuk <i>node</i> ( <i>Circle</i> , <i>Box</i> dan <i>Hexagon</i> )
9	<i>-c color</i>	Warna pada <i>node</i>

#### 4.4. Implementasi Metrik Analisis

Hasil menjalankan skenario MANET (DSR dan AODV) dalam NS-2 dalam bentuk *Trace File* berekstensi .tr dianalisis dengan 3(tiga) metrik yaitu *Packet delivery ratio* (PDR), *End-to-end delay* (E2D) dan *Routing Overhead* (RO). Implementasi dari masing-masing metrik dengan menggunakan bahasa AWK dijelaskan sebagai berikut.

##### 4.4.1. *Packet delivery ratio* (PDR)

Menghitung PDR dapat dilakukan dengan cara menghitung jumlah paket data yang dikirimkan oleh *node* 1 dan jumlah paket data yang diterima oleh *node* 2 pada suatu *trace file*. Penambahan jumlah paket data yang dikirim akan dilakukan apabila pada baris *trace* yang bersangkutan mengandung kondisi kolom pertama menunjukkan huruf “s” yang menandakan *send packet*, kolom ketiga menunjukkan bahwa *node* yang melakukan pengiriman adalah *node* 1, kolom ke-4 dan kolom ke-7 menunjukkan kode “AGT” dan “cbr” yang menandakan bahwa pengiriman yang dilakukan adalah pengiriman paket data. Penambahan jumlah paket data yang diterima akan dilakukan apabila pada baris *trace* yang bersangkutan mengandung kondisi kolom pertama menunjukkan huruf “r” yang menandakan *send*

*packet*, kolom ketiga menunjukkan bahwa *node* yang melakukan pengiriman adalah *node* 1, kolom ke-4 dan kolom ke-7 menunjukkan kode “AGT” dan “cbr” yang menandakan bahwa pengiriman yang dilakukan adalah pengiriman paket data.

Pengecekan dilakukan hingga akhir baris *trace file* dan hasilnya merupakan nilai PDR dari simulasi skenario. Berikut merupakan *pseudocode* PDR ditunjukkan pada Gambar 4.16 dan implementasinya dapat dilihat pada Lampiran.

```
BEGIN(
sent ← 0
recv ← 0
recv_id ← 0
pdr ← 0 )
(if ($1 == "s" and $3 == "_1_" and $4 ==
"AGT" and $7 == "cbr")
    sent + 1;
    if ($1 == "s" and $3 == "_1_" and $4 ==
"AGT" and $7 == "cbr")
        recv + 1;
)
END(pdr ← (recv/sent)*100
Print sent
Print recv
Print pdr)
```

**Gambar 4.16 Pseudocode PDR**

Contoh perintah untuk analisis PDR dari *trace file* model transmisi *TwoRayGround* dengan protokol AODV dari 3 kecepatan maksimal perpindahan *node* yaitu 5 m/s, 10 m/s dan 15 m/s beserta hasilnya seperti Gambar 4.17.

```

root@Ubuntazi: ~/pract/AODV/TwoRayGround
root@Ubuntazi:~/pract/AODV/TwoRayGround# gawk -f pdr.awk A-Trg.tr
cbr s:24 r:14, r/s Ratio:0.5833, f:24
root@Ubuntazi:~/pract/AODV/TwoRayGround# gawk -f pdr.awk A-Trg.tr
cbr s:25 r:6, r/s Ratio:0.2400, f:17
root@Ubuntazi:~/pract/AODV/TwoRayGround# gawk -f pdr.awk A-Trg.tr
cbr s:26 r:13, r/s Ratio:0.5000, f:58
root@Ubuntazi:~/pract/AODV/TwoRayGround# █

```

**Gambar 4.17 Perintah Menjalankan Beserta Hasil PDR**

#### 4.4.2. *End-to-end delay (E2D)*

```

(if ($1 == "s" and $3 == "_1_" and $4 ==
"AGT" and $7 == "cbr" )
    pkt_sent [$6] ← $2

    if ($1 == "r" and $3 == "_2_" and $4 ==
"AGT" and $7 == "cbr" and recv_id != $6)
        recv + 1
        recv_id ← $6
        pkt_recv[i] ← $2;
    )
END (
for i in pkt_recv
    delay += pkt_recv[i] - pkt_sent[i]
avg_delay ← delay / recv;
print delay
print recv
print avg_delay)

```

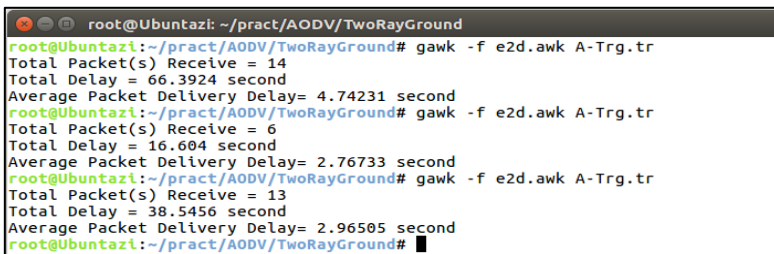
**Gambar 4.18 Pseudocode E2D**

Perhitungan E2D dilakukan dengan menghitung selisih waktu paket data terkirim yang dilakukan oleh *node* 1 dan waktu paket data terkirim yang dilakukan *node* 2 didalam satu *trace file*. Pencatatan waktu paket data yang dikirim akan dilakukan apabila pada baris *trace* yang bersangkutan mengandung kondisi kolom

pertama menunjukkan huruf “s” yang menandakan *send packet*, kolom ketiga menunjukkan bahwa *node* yang melakukan pengiriman adalah *node* 1, kolom ke-4 dan kolom ke-7 menunjukkan kode “AGT” dan “cbr” yang menandakan bahwa pengiriman yang dilakukan adalah pengiriman paket data.

Pencatatan waktu dan ID serta jumlah paket data yang diterima akan dilakukan apabila pada baris *trace* yang bersangkutan mengandung kondisi kolom pertama menunjukkan huruf “r” yang menandakan *send packet*, kolom ketiga menunjukkan bahwa *node* yang melakukan pengiriman adalah *node* 1, kolom ke-4 dan kolom ke-7 menunjukkan kode “AGT” dan “cbr” yang menandakan bahwa pengiriman yang dilakukan adalah pengiriman paket data. Pengecekan dilakukan hingga akhir baris *trace file* dan hasilnya merupakan nilai E2D dari simulasi skenario.

Contoh perintah untuk analisis E2D dari *trace file* model transmisi *TwoRayGround* dengan protokol AODV dari 3 kecepatan maksimal perpindahan *node* yaitu 5 m/s, 10 m/s dan 15 m/s beserta hasilnya seperti Gambar 4.19.



```

root@Ubuntazi: ~/pract/AODV/TwoRayGround
root@Ubuntazi:~/pract/AODV/TwoRayGround# gawk -f e2d.awk A-Trg.tr
Total Packet(s) Receive = 14
Total Delay = 66.3924 second
Average Packet Delivery Delay= 4.74231 second
root@Ubuntazi:~/pract/AODV/TwoRayGround# gawk -f e2d.awk A-Trg.tr
Total Packet(s) Receive = 6
Total Delay = 16.604 second
Average Packet Delivery Delay= 2.76733 second
root@Ubuntazi:~/pract/AODV/TwoRayGround# gawk -f e2d.awk A-Trg.tr
Total Packet(s) Receive = 13
Total Delay = 38.5456 second
Average Packet Delivery Delay= 2.96505 second
root@Ubuntazi:~/pract/AODV/TwoRayGround# █

```

**Gambar 4.19** Perintah menjalankan beserta hasil E2D

#### 4.4.3. Routing Overhead (RO)

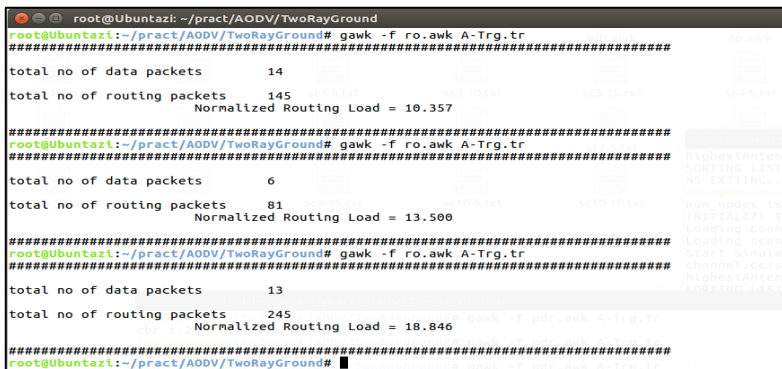
Implementasi perhitungan metrik *routing overhead* AODV maupun DSR dihitung apabila kolom pertama menunjukkan huruf “s” yang berarti *sent packet* atau huruf “f” yang berarti *forward packet*, kolom ke-4 mengandung kode “RTR” yang berarti paket *routing* dan kolom ke -7 menunjukkan “DSR” atau “AODV” yang

menandakan masing-masing paket *routing*. Gambar 4.20 menunjukkan *pseudocode* algoritma RO pada DSR. Implementasi dapat dilihat pada Lampiran.

```
//Input: trace file simulasi skenario
//Output: jumlah packet sent, packet
received dan PDR
BEGIN (
  rt_pkts ← 0
)
(if (($1 == "s" or $1 == "f") and $4 ==
"RTR" and $7 == "DSR" )
  rt_pkts + 1
)
END(
```

**Gambar 4.20 Pseudocode RO**

Contoh perintah untuk analisis RO dari *trace file* model transmisi *TwoRayGround* dengan protokol AODV dari 3 kecepatan maksimal perpindahan *node* yaitu 5 m/s, 10 m/s dan 15 m/s beserta hasilnya seperti Gambar 4.21.



```
root@Ubuntu21: ~/pract/AODV/TwoRayGround
root@Ubuntu21:~/pract/AODV/TwoRayGround# gawk -f ro.awk A-Trg.tr
#####
total no of data packets      14
total no of routing packets  145
Normalized Routing Load = 10.357
#####
root@Ubuntu21:~/pract/AODV/TwoRayGround# gawk -f ro.awk A-Trg.tr
#####
total no of data packets      6
total no of routing packets   81
Normalized Routing Load = 13.500
#####
root@Ubuntu21:~/pract/AODV/TwoRayGround# gawk -f ro.awk A-Trg.tr
#####
total no of data packets      13
total no of routing packets   245
Normalized Routing Load = 18.846
#####
root@Ubuntu21:~/pract/AODV/TwoRayGround#
```

**Gambar 4.21 Perintah Menjalankan Beserta Hasil RO**

*[Halaman ini sengaja dikosongkan]*

## BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari skenario NS-2 yang telah dibuat. Pengujian menghasilkan perbandingan dari skenario-skenario yang telah dijalankan pada simulasi.

### 5.1. Lingkungan Pengujian

Pengujian dilakukan pada laptop yang menggunakan dua sistem operasi yaitu Windows dan Linux. Spesifikasi laptop yang digunakan ditunjukkan pada Table 5.1.

**Table 5.1 Spesifikasi Laptop yang Digunakan**

Komponen	Spesifikasi
CPU	Intel (R) Core(TM) i7-4700HQ CPU @2.40GHz
Sistem Operasi	Windows 10 64-bit, Linux Ubuntu 16.4 LTS 64-bit (NS-2, AODV, DSR, Nakagami, <i>TwoRayGround</i> , <i>Mobilty generation</i> , <i>TwoRayGround</i> , <i>Traffic-connection Generation</i> )
Memori	8 GB DDR3
Media Penyimpanan	750 GB

### 5.2. Kriteria Pengujian

Pengujian pada skenario yang dihasilkan melalui *mobility generator default* dari NS-2 menggunakan beberapa kriteria. Table 5.2 menunjukkan kriteria-kriteria yang ditentukan dalam skenario.

**Table 5.2 Kriteria Pengujian**

Kriteria	Spesifikasi
Skenario	Manet ( <i>Random way point</i> )
Jumlah <i>Node</i>	50
Jumlah Percobaan	10 kali
Posisi Awal	Acak

Pergerakan	Acak
Protokol <i>Routing</i>	AODV dan DSR
Jarak <i>Node 1</i> dan <i>Node 2</i>	Acak

### 5.3. Analisis *Packet delivery ratio*

*Trace file* hasil dari menjalankan program skenario *node-movement* menggunakan model transmisi *TwoRayGround* dan Nakagami pada kedua *routing* AODV dan DSR dianalisis nilai PDR menggunakan *script* pdr.awk. Hasil tiap perhitungan PDR dari setiap skenario ditabulasikan menjadi seperti Table 5.3, Table 5.4, Table 5.5 dan Table 5.6.

**Table 5.3 Hasil ujicoba selama 10 kali pada AODV dengan model Nakagami (PDR)**

AODV NAKAGAMI	PDR		
	5 m/s	10 m/s	15 m/s
SC1	0,9583	0,7917	0,7600
SC2	0,8800	0,8846	0,6667
SC3	0,8929	0,8333	0,6538
SC4	0,7200	0,5926	0,7083
SC5	0,9615	0,3200	0,6154
SC6	0,6400	0,7692	0,7143
SC7	0,4400	0,7308	0,4800
SC8	0,9600	0,8000	0,5618
SC9	0,6538	0,6923	0,5600
SC10	0,8800	0,7778	0,7037
Rata-rata	0,7986	0,7192	0,6424

**Table 5.4 Hasil ujicoba selama 10 kali pada AODV dengan model *TwoRayGround* (PDR)**

AODV TwoRayGround	PDR		
	5 m/s	10 m/s	15 m/s
SC1	0,9290	0,2000	0,0769
SC2	0,1852	0,6250	0,1250



SC3	0,6538	0,4231	0,1600
SC4	0,5000	0,5417	0,3846
SC5	0,4231	0,2609	0,1481
SC6	0,1481	0,1923	0,4167
SC7	0,2143	0,3077	0,2069
SC8	0,6296	0,2000	0,2500
SC9	0,3462	0,5200	0,5000
SC10	0,5833	0,2400	0,5000
Rata-rata	0,3684	0,33107	0,2691

**Table 5.5 Hasil ujicoba selama 10 kali pada DSR dengan model Nakagami (PDR)**

DSR NAKAGAMI	PDR		
	5 m/s	10 m/s	15 m/s
SC1	0,8800	0,3333	0,7600
SC2	0,8696	1,0000	0,4615
SC3	0,9200	0,5200	0,3750
SC4	0,3846	0,7917	0,3750
SC5	1,0000	0,4348	0,1818
SC6	0,9600	0,8077	0,5217
SC7	0,9615	0,5833	0,5200
SC8	1,0000	0,7200	0,7200
SC9	0,8800	0,5417	0,8519
SC10	1,0000	0,7308	0,8000
Rata-rata	0,9138	0,6463	0,5566

**Table 5.6 Hasil ujicoba selama 10 kali pada DSR dengan model TwoRayGround (PDR)**

DSR TwoRayGround	PDR		
	5 m/s	10 m/s	15 m/s
SC1	0,8800	0,3333	0,7600
SC2	0,8696	1,0000	0,3913
SC3	0,9200	0,7826	0,5833
SC4	0,3846	0,7917	0,6087

SC5	1,0000	0,4348	0,2917
SC6	0,2800	0,8077	0,5217
SC7	0,4000	0,7500	0,5200
SC8	1,0000	0,7200	0,7200
SC9	0,6818	0,5417	0,8519
SC10	1,0000	0,7308	0,8000
Rata-rata	0,7416	0,6893	0,6049

**Table 5.7 Rata-rata PDR Skenario *Node-Movement* AODV**

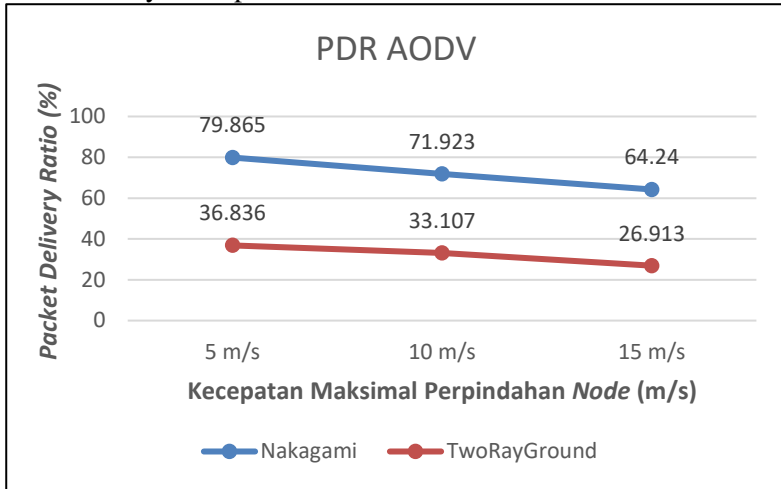
Kecepatan Maksimal Perpindahan <i>Node</i> (m/s)	PDR (%) AODV	
	Nakagami	<i>TwoRayGround</i>
5	79,8650	36,8360
10	71,9230	33,1070
15	64,2400	26,9130

**Table 5.8 Rata-rata PDR Skenario *Node-Movement* DSR**

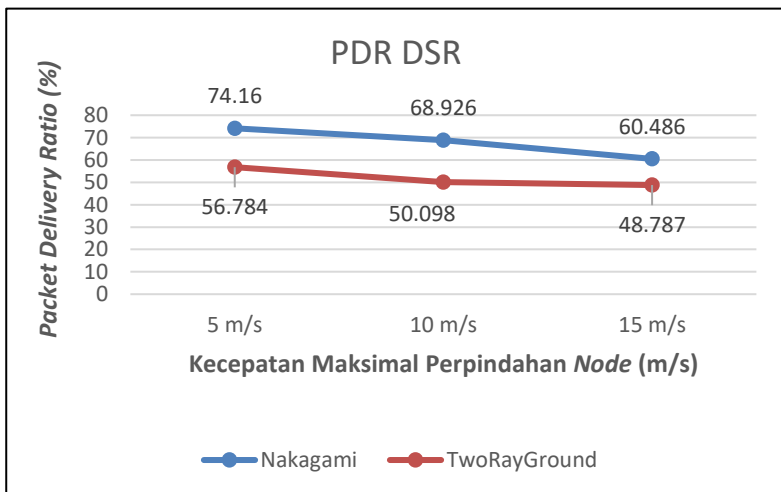
Kecepatan Maksimal Perpindahan <i>Node</i> (m/s)	PDR (%) DSR	
	Nakagami	<i>TwoRayGround</i>
5	74,1600	56,7840
10	68,9260	50,0980
15	60,4860	48,7870

Pada Table 5.7 dan Gambar 5.1 menggambarkan performa PDR yang dihasilkan model transmisi *TwoRayGround* dan Nakagami pada jaringan MANET dengan menggunakan skenario *node-movement* (*mobilty generation*) yang memiliki sifat *Random way point*. Dapat dilihat bahwa PDR AODV yang dihasilkan semakin menurun dengan bertambahnya kecepatan maksimal perpindahan *node*. Table 5.8 dan Gambar 5.2 juga menunjukkan

bahwa performa PDR DSR mengalami penurunan dengan bertambahnya kecepatan maksimal *node*.



**Gambar 5.2** Grafik PDR *Routing Protocol AODV*



**Gambar 5.1** Grafik PDR pada *Routing Protocol DSR*

Apabila diperhatikan dari kedua grafik PDR tiap *routing protocol* menunjukkan sifat yang sama. Pada model transmisi *TwoWayGround* performa PDR dari DSR kurang baik khususnya pada AODV. Dengan kecepatan maksimal pergerakan 5 m/s, PDR yang dihasilkan sebesar 36.8360% dan semakin menurun ketika kecepatan maksimal pergerakan *node* ditingkatkan. Ketika mencapai 15 m/s nilai PDR AODV pada model transmisi *TwoRayGround* tersisa 26.9130%. PDR pada *routing protocol* DSR menunjukkan hal yang sama, akan tetapi nilai PDR-nya lebih baik dari AODV. Nakagami memiliki nilai PDR yang tinggi namun apabila diperhatikan perubahan yang sangat signifikan, pada kecepatan maksimal 5 m/s ke 10 m/s turun hingga 13.6740% pada DSR dan 15.6250% pada AODV. Meskipun memiliki nilai yang kurang bagus, *TwoRayGround* adalah model propagasi yang stabil.

#### 5.4. Analisis *End-to-end delay*

*Trace file* hasil dari menjalankan program skenario *node-movement* menggunakan model transmisi *TwoRayGround* dan Nakagami pada kedua *routing* AODV dan DSR dianalisis nilai PDR menggunakan *script* e2d.awk. Hasil tiap perhitungan *End-to-end delay* dari setiap skenario ditabulasikan seperti Table 5.9, Table 5.10, Table 5.11 dan Table 5.12.

**Table 5.9 Hasil ujicoba selama 10 kali AODV dengan model Nakagami (E2D)**

AODV NAKAGAMI	E2D		
	5 m/s	10 m/s	15 m/s
SC1	0,0139	0,2408	0,3049
SC2	0,0541	0,6929	2,8665
SC3	0,1246	1,0174	0,0844
SC4	0,0370	0,6802	0,0382
SC5	0,0178	4,0823	1,3081
SC6	1,2006	0,0911	0,5298
SC7	0,0362	0,2185	0,0626
SC8	0,0201	0,1637	1,5487

SC9	0,0645	0,0907	0,8141
SC10	0,0146	2,3980	0,6029
Rata-rata	0,1583	0,9676	0,8160

**Table 5.10 Hasil ujicoba selama 10 kali AODV dengan model  
*TwoRayGround (E2D)***

AODV TwoRayGround	<i>E2D</i>		
	5 m/s	10 m/s	15 m/s
SC1	0,9197	3,9540	4,9119
SC2	4,7035	2,3736	3,8241
SC3	1,8676	1,6220	2,0105
SC4	1,0707	5,8002	1,6691
SC5	4,1427	5,4074	0,0058
SC6	5,8324	4,3812	2,3055
SC7	0,8679	2,7014	3,8870
SC8	1,0643	2,8110	3,2953
SC9	0,0391	1,9729	2,7365
SC10	4,7423	2,7673	2,9650
Rata-rata	2,5250	3,3791	2,7611

**Table 5.11 Hasil ujicoba selama 10 kali DSR dengan model  
*Nakagami (E2D)***

DSR NAKAGAMI	<i>E2D</i>		
	5 m/s	10 m/s	15 m/s
SC1	0,0123	0,5818	0,8760
SC2	0,0599	0,2380	1,3655
SC3	0,0327	0,8847	3,8938
SC4	0,5258	5,8259	0,0522
SC5	0,0169	10,8446	0,2666
SC6	1,5278	0,1548	1,0483
SC7	0,2995	0,2115	0,9867
SC8	0,0180	0,1675	2,1177

SC9	0,2722	0,0898	1,8263
SC10	0,0178	1,8282	2,2208
Rata-rata	0,2783	2,0827	1,4654

**Table 5.12 Hasil ujicoba selama 10 kali AODV dengan model *TwoRayGround* (E2D)**

DSR <i>TwoRayGround</i>	E2D		
	5 m/s	10 m/s	15 m/s
SC1	0,2725	12,9898	9,3637
SC2	11,3319	5,8268	10,0718
SC3	6,1584	2,8753	6,5406
SC4	5,9202	7,9998	5,2063
SC5	9,0834	14,6355	10,5559
SC6	12,3940	8,3800	2,0908
SC7	0,0277	7,9648	8,3667
SC8	3,8399	8,4022	4,1496
SC9	0,1088	0,0908	11,5437
SC10	6,6320	10,2981	8,4939
Rata-rata	5,5769	7,9463	7,6383

**Table 5.13 Rata-rata Rata-rata E2D Skenario *Node-Movement* AODV**

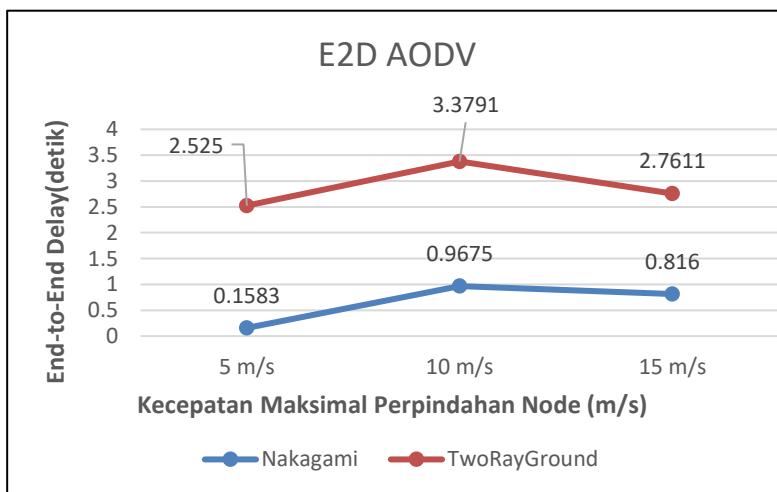
Kecepatan Maksimal Perpindahan <i>Node</i> (m/s)	E2D (detik) AODV	
	Nakagami	<i>TwoRayGround</i>
5	0.1583	2.5250
10	0.9676	3.3791
15	0.8160	2.7611

**Table 5.14 Rata-rata Rata-rata E2D Skenario *Node-Movement* DSR**

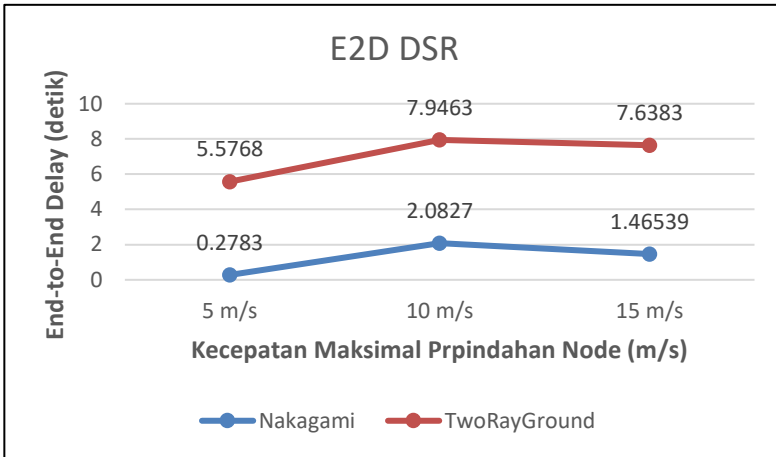
Kecepatan Maksimal	E2D (detik) DSR	
	Nakagami	<i>TwoRayGround</i>

Perpindahan Node (m/s)		
5	0.2783	5.5769
10	2.0827	7.9463
15	1.4654	7.6383

Peforma kedua *reactive routing protocol* pada *End-to-end delay* terbilang cukup buruk khususnya DSR. Di kedua model propagasi AODV dan DSR menunjukkan bahwa terdapat *delay* yang amat besar. Pada Table 5.13 *routing protocol* AODV pada model transmisi *TwoRayGround* mengalami perubahan yang sangat signifikan ketika kecepatan maksimal dinaikkan hingga 15 m/s. Pada Table 5.14 DSR menunjukkan peforma yang buruk pada model transmisi *TwoRayGround* yaitu *delay* sebesar 7.9463 detik pada kecepatan pergerakan maksimal 10 m/s. Dengan menggunakan model transmisi Nakagami dengan kecepatan maksimal pergerakan *node* 10 m/s, nilai E2D mencapai 2.0827 detik dengan waktu simulasi 100 detik.



**Gambar 5.3 Grafik E2D pada Routing Protocol AODV**



**Gambar 5.4** Grafik E2D pada *Routing Protocol AODV*

Pada Gambar 5.3 dan Gambar 5.4 memperlihatkan performa *reactive routing protocol* pada model transmisi Nakagami dan *TwoRayGround*. Sifat acak pada tiap *node* yang telah dibuat menghasilkan situasi yang beragam dan memungkinkan terjadi penundaan pengiriman paket karena rute terputus.

### 5.5. Analisis Routing Overhead

*Trace file* hasil dari menjalankan program skenario *node-movement* menggunakan model transmisi *TwoRayGround* dan Nakagami pada kedua *routing AODV* dan DSR dianalisis nilai *Routing Overhead* menggunakan *script ro.awk*. Hasil tiap perhitungan PDR dari setiap skenario ditabulasikan menjadi seperti Table 5.15, Table 5.16, Table 5.17 dan Table 5.18.

**Table 5.15** Hasil ujicoba selama 10 kali AODV dengan model Nakagami (RO)

AODV NAKAGAMI	RO		
	5 m/s	10 m/s	15 m/s



SC1	83	464	191
SC2	308	136	948
SC3	222	161	527
SC4	214	453	305
SC5	66	683	400
SC6	566	436	644
SC7	296	499	357
SC8	96	270	289
SC9	355	565	258
SC10	131	397	205
Rata-rata	233,7	406,4	412,4

**Table 5.16 Hasil ujicoba selama 10 kali AODV dengan model *TwoRayGround* (RO)**

AODV TwoRayGround	RO		
	5 m/s	10 m/s	15 m/s
SC1	55	250	61
SC2	286	184	130
SC3	464	130	189
SC4	184	227	219
SC5	348	399	165
SC6	181	290	416
SC7	141	243	248
SC8	113	140	145
SC9	121	590	378
SC10	145	81	245
Rata-rata	203,8	253,4	219,6

**Table 5.17 Hasil ujicoba selama 10 kali DSR dengan model *Nakagami* (RO)**

DSR NAKAGAMI	RO		
	5 m/s	10 m/s	15 m/s
SC1	73	324	219
SC2	274	136	429

SC3	151	155	787
SC4	274	492	422
SC5	37	616	256
SC6	424	572	274
SC7	356	295	395
SC8	69	222	273
SC9	425	270	262
SC10	24	250	222
Rata-rata	210,7	333,2	353,9

**Table 5.18 Hasil ujicoba selama 10 kali DSR dengan model *TwoRayGround* (RO)**

DSR <i>TwoRayGround</i>	RO		
	5 m/s	10 m/s	15 m/s
SC1	27	287	133
SC2	302	45	174
SC3	328	242	324
SC4	134	293	270
SC5	204	229	165
SC6	94	409	294
SC7	60	243	249
SC8	86	190	175
SC9	133	473	254
SC10	164	77	145
Rata-rata	153,2	248,8	218,3

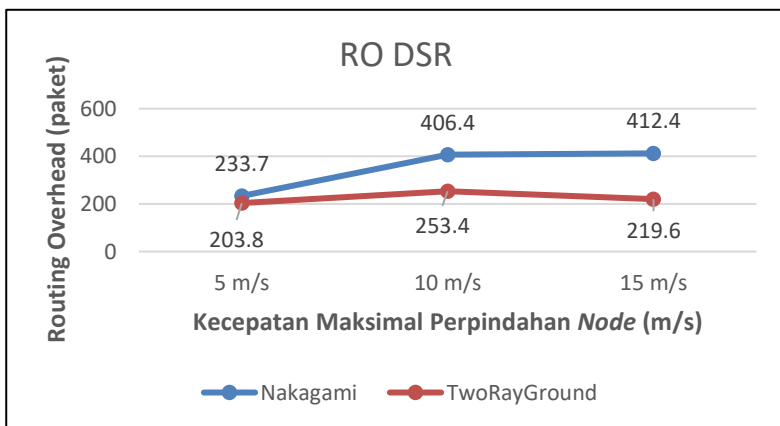
**Table 5.19 Rata-rata RO Skenario *Node-Movement* AODV**

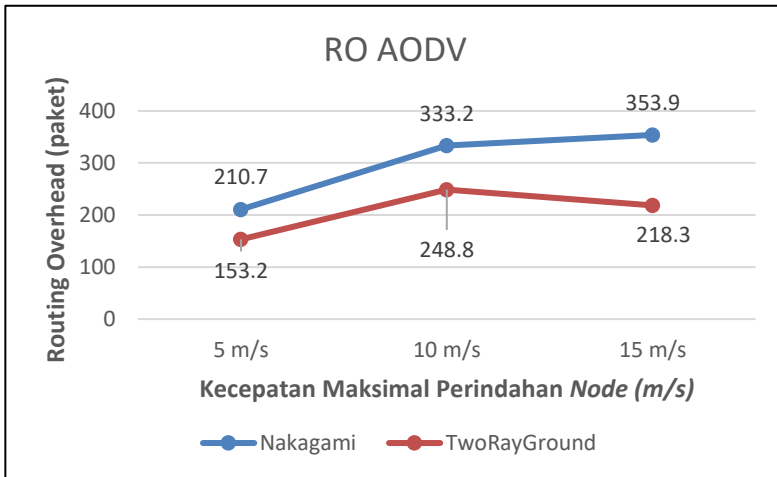
Kecepatan Maksimal Perpindahan <i>Node</i> (m/s)	RO (paket) AODV	
	Nakagami	<i>TwoRayGround</i>
5	233.7	203.8
10	406.4	253.4
15	412.4	219.6

**Table 5.20 Rata-rata RO Skenario *Node-Movement***

Kecepatan Maksimal Perpindahan <i>Node</i> (m/s)	RO (paket) DSR	
	Nakagami	<i>TwoRayGround</i>
5	210.7	153.2
10	333.2	248.8
15	353.9	218.3

Pada Table 5.17 dan Table 5.18 merupakan hasil pengujian model transmisi Nakagami dan *TwoRayGround*. Nilai pada analisis *routing overhead* pada *routing protocol* AODV dan DSR bersifat fluktuatif. Pada Gambar 5.5 dan Gambar 5.6 grafik menunjukkan bahwa baik AODV dan DSR mengalami peningkatan paket *routing* seiring dengan penambahan kecepatan pergerakan *node* pada model transmisi Nakagami.

**Gambar 5.5 Grafik RO pada *Routing Protocol* DSR**



**Gambar 5.6 Grafik RO pada *Routing Protocol* AODV**

Hasil analisis metrik-metrik diatas menyebabkan perbedaan antara model transmisi terhadap masing-masing *routing protocol* yang digunakan. Beberapa faktor yang menyebabkan terjadinya perbedaan antara lain adalah lokasi dan pergerakan *node* yang dikonfigurasi secara acak sehingga dapat menjadi *worst case* pada model transmisi dan atau *routing protocol* tertentu.

## BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

### 6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Skenario MANET yang dihasilkan oleh *node-movement (mobilty generation)* dan dijalankan menggunakan model transmisi *TwoRayGround* dengan penambahan kecepatan maksimal perpindahan *node* memiliki performa sebagai berikut:
  - *Packet delivery ratio* pada *routing protocol AODV* kecil dan cenderung menurun pada saat kecepatan pergerakan *node* pada simulasi bertambah. Pada DSR performa PDR turun stabil namun hasilnya lebih baik daripada AODV.
  - *End-to-end delay* yang dihasilkan oleh DSR mengalami peningkatan ketika kecepatan maksimal perpindahan *node* ditambah. Pada kecepatan 5 m/s ke 10 m/s mengalami kenaikan dari 5.57688 detik menjadi 7.94631 detik.
  - *Routing Overhead* yang dihasilkan oleh kedua *routing protocol reactive* bersifat fluktuatif pada model transmisi *TwoRayGround*.
2. Skenario MANET yang dihasilkan oleh *node-movement (mobilty generation)* dan dijalankan menggunakan model transmisi Nakagami dengan penambahan kecepatan maksimal perpindahan *node* memiliki performa sebagai berikut:

- *Packet delivery ratio* pada *routing protocol* AODV dan DSR cenderung menurun stabil pada saat kecepatan pergerakan *node* pada simulasi bertambah.
  - *End-to-end delay* yang dihasilkan oleh DSR dan AODV cenderung bersifat fluktuatif tidak terdapat perbedaan yang terlalu signifikan.
  - *Routing Overhead* yang dihasilkan di kedua *routing protocol reactive* pada model transmisi Nakagami cenderung mengalami peningkatan.
3. Hasil analisis metrik PDR menunjukkan model transmisi Nakagami memiliki keunggulan 39.72% dibanding model transmisi *TwoRayGround* pada *routing protocol* AODV. Pada DSR model Nakagami lebih unggul 15.96% daripada model *TwoRayGround*.
  4. Hal-hal yang dapat mempengaruhi hasil analisis metrik PDR, E2D dan RO yang dihasilkan dari model transmisi *TwoRayGround* dan Nakagami adalah:
    - Posisi awal *node* yang dibuat secara acak
    - Pergerakan *node* yang dibuat secara acak
    - Keadaan sekitar simulasi
    - Lingkungan jaringan yang digunakan
    - Ketinggian antenna
  5. Dari hasil percobaan pada skenario MANET, didapatkan nilai PDR, E2D dan RO pada model transmisi Nakagami lebih bagus daripada nilai PDR, E2D dan RO pada model transmisi *TwoRayGround*, baik menggunakan *routing protocol* AODV maupun DSR sehingga:
    - Model transmisi *TwoRayGround* tidak terpengaruh dengan keadaan sekitar simulasi sehingga dapat digunakan di lingkungan MANET yang bentuk jaringannya lebih besar dan bersifat global. Model ini juga menunjukkan nilai yang kurang bagus apabila jarak yang digunakan terlalu dekat.
    - Model transmisi Nakagami terpengaruh dengan keadaan sekitar simulasi sehingga dapat digunakan

dilingkungan VANET yang bentuk jaringannya lebih detail daripada MANET. Meskipun demikian hasil yang ditunjukkan pada pengaplikasian pada *reactive routing protocol* menampilkan performa yang baik.

## 6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Di antaranya adalah sebagai berikut:

1. Dapat dilakukan percobaan terhadap analisis metrik lainnya seperti *jitter* dan *throughput*.
2. Dapat dilakukan percobaan dengan bermacam-macam variasi kombinasi antara kecepatan maksimal pergerakan *node*, jumlah *node*, waktu simulasi dan parameter lainnya.
3. Modifikasi pada skrip *traffic-connection file* dengan mengganti jumlah koneksi dan banyak paket yang dikirim.
4. Menggunakan fasilitas lain dari NS-2 seperti *xgraph* dan lain-lain.

*[Halaman ini sengaja dikosongkan]*



## DAFTAR PUSTAKA

- [1] M. Affandes, "3. Mobile Ad Hoc Network (MANET)," 31 Oktober 2011. [Online]. Available: <https://affandezone.wordpress.com/2011/10/30/3-routing-pada-manet/>. [Accessed 2016 November 6].
- [2] R. F. Sari, A. Syarif and B. Budiarmo, "Analisis Kinerja Protokol Routing Ad Hoc On-Demand Distance Vector (AODV) Pada Jaringan Ad Hoc Hybrid Perbandingan Hasil Simulasi Dengan NS-2 dan Implementasi Pada Testbed Dengan PDA," *Makara Teknologi*, vol. 112, p. 8, 2008.
- [3] F. A. F. I. M. Sony Sumaryo and I. M. Yudha Purwanto, "Performansi Dynamic Source Routing (DSR) Dengan Sumber Trafik CBR, Pareto dan Exponential," *Seminar Nasional Sistem dan Informatika 2007*, p. 56, 2007.
- [4] A. Saputra, "Pengetahuan Tentang Jaringan," 25 Maret 2011. [Online]. [Accessed 10 November 2016].
- [5] Pajala, Elina, Isotalo and Tero, "An improved simulation model for Nakagami-m-m fading channels for satellite positioning applications," Finlandia, Institute of Communications Engineering Tampere University of Technology, p. 82.
- [6] R. Baumann, *Engineering and simulation of mobile ad hoc routing protocols for VANET on Highways and in cities*, Institute of Technology Zurich, 2014.
- [7] D. A. Maltz, "Simulation and Implementation" in *On Demand Routing in Multi-Hop Wireless Mobile Ad Hoc Network*, School of Computer Science Carnegie Mellon, 2001.
- [8] 8 November 2014. [Online]. Available: <http://bengkelubuntu.org/teks/awk/Praktikum%2001%20%2>

0Berkenalan%20dengan%20AWK.pdf. [Accessed 12 November 2016].

- [9] I. M. B. D. P. D. Jeroen Hoebeke, "An overview of mobile ad hoc networks: Applications and challenges," *Communications Network*, vol. 4, 2004.

## LAMPIRAN

1	$\$node_{(0)}$	set X_	723.656598657969
2	$\$node_{(0)}$	set Y_	594.184588731824
3	$\$node_{(0)}$	set Z_	0.000000000000
4	$\$node_{(1)}$	set X_	562.161488267498
5	$\$node_{(1)}$	set Y_	469.709067698449
6	$\$node_{(1)}$	set Z_	0.000000000000
7	$\$node_{(2)}$	set X_	714.242451757293
8	$\$node_{(2)}$	set Y_	699.171528133489
9	$\$node_{(2)}$	set Z_	0.000000000000
10	$\$node_{(3)}$	set X_	592.912269494439
11	$\$node_{(3)}$	set Y_	96.205280665471
12	$\$node_{(3)}$	set Z_	0.000000000000
13	$\$node_{(4)}$	set X_	325.315750533640
14	$\$node_{(4)}$	set Y_	616.314720267770
15	$\$node_{(4)}$	set Z_	0.000000000000
16	$\$node_{(5)}$	set X_	611.079339567689
17	$\$node_{(5)}$	set Y_	365.007335266919
18	$\$node_{(5)}$	set Z_	0.000000000000
19	$\$node_{(6)}$	set X_	831.736517814267
20	$\$node_{(6)}$	set Y_	558.866158277765
21	$\$node_{(6)}$	set Z_	0.000000000000
22	$\$node_{(7)}$	set X_	651.119910060182
23	$\$node_{(7)}$	set Y_	105.171257624315
24	$\$node_{(7)}$	set Z_	0.000000000000
25	$\$node_{(8)}$	set X_	399.132395365634
26	$\$node_{(8)}$	set Y_	628.122068522356
27	$\$node_{(8)}$	set Z_	0.000000000000
28	$\$node_{(9)}$	set X_	558.339964664707

29	\$node_(9)	set Y_	586.733829227890
30	\$node_(9)	set Z_	0.000000000000
31	\$node_(10)	set X_	312.608465254903
32	\$node_(10)	set Y_	625.549026966141
33	\$node_(10)	set Z_	0.000000000000
34	\$node_(11)	set X_	47.957996832035
35	\$node_(11)	set Y_	426.183289812441
36	\$node_(11)	set Z_	0.000000000000
37	\$node_(12)	set X_	765.462888571499
38	\$node_(12)	set Y_	594.956298859536
39	\$node_(12)	set Z_	0.000000000000
40	\$node_(13)	set X_	425.678485885990
41	\$node_(13)	set Y_	302.840045930522
42	\$node_(13)	set Z_	0.000000000000
43	\$node_(14)	set X_	116.650135259430
44	\$node_(14)	set Y_	674.781124783790
45	\$node_(14)	set Z_	0.000000000000
46	\$node_(15)	set X_	402.539698029463
47	\$node_(15)	set Y_	722.064190914238
48	\$node_(15)	set Z_	0.000000000000
49	\$node_(16)	set X_	220.075326919478
50	\$node_(16)	set Y_	232.781089113203
51	\$node_(16)	set Z_	0.000000000000
52	\$node_(17)	set X_	781.141415058499
53	\$node_(17)	set Y_	660.201470489126
54	\$node_(17)	set Z_	0.000000000000
55	\$node_(18)	set X_	793.613611888520
56	\$node_(18)	set Y_	538.249478083079
57	\$node_(18)	set Z_	0.000000000000
58	\$node_(19)	set X_	69.352696399521

59	<code>\$node_(19) set Y_ 588.905174527847</code>
60	<code>\$node_(19) set Z_ 0.000000000000</code>
61	<code>\$node_(20) set X_ 288.641168232393</code>
62	<code>\$node_(20) set Y_ 166.512988574035</code>
63	<code>\$node_(20) set Z_ 0.000000000000</code>

**Gambar 7.1** Posisi awal tiap *node*

1	<code>\$god_ set-dist 0 1 1</code>
2	<code>\$god_ set-dist 0 2 1</code>
3	<code>\$god_ set-dist 0 3 3</code>
4	<code>\$god_ set-dist 0 4 2</code>
5	<code>\$god_ set-dist 0 5 2</code>
6	<code>\$god_ set-dist 0 6 1</code>
7	<code>\$god_ set-dist 0 7 3</code>
8	<code>\$god_ set-dist 0 8 2</code>
9	<code>\$god_ set-dist 0 9 1</code>
10	<code>\$god_ set-dist 0 10 2</code>
11	<code>\$god_ set-dist 0 11 4</code>
12	<code>\$god_ set-dist 0 12 1</code>
13	<code>\$god_ set-dist 0 13 2</code>
14	<code>\$god_ set-dist 0 14 3</code>
15	<code>\$god_ set-dist 0 15 2</code>
16	<code>\$god_ set-dist 0 16 3</code>
17	<code>\$god_ set-dist 0 17 1</code>
18	<code>\$god_ set-dist 0 18 1</code>
19	<code>\$god_ set-dist 0 19 3</code>
20	<code>\$god_ set-dist 0 20 3</code>
21	<code>\$god_ set-dist 0 21 2</code>
22	<code>\$god_ set-dist 0 22 1</code>

23	\$god_	set-dist	0	23	2
24	\$god_	set-dist	0	24	1
25	\$god_	set-dist	0	25	3
26	\$god_	set-dist	0	26	1
27	\$god_	set-dist	0	27	3
28	\$god_	set-dist	0	28	3
29	\$god_	set-dist	0	29	3
30	\$god_	set-dist	0	30	1
31	\$god_	set-dist	0	31	3
32	\$god_	set-dist	0	32	3
33	\$god_	set-dist	0	33	1
34	\$god_	set-dist	0	34	3
35	\$god_	set-dist	0	35	3
36	\$god_	set-dist	0	36	1
37	\$god_	set-dist	0	37	3
38	\$god_	set-dist	0	38	3
39	\$god_	set-dist	0	39	3
40	\$god_	set-dist	0	40	2
41	\$god_	set-dist	0	41	2
42	\$god_	set-dist	0	42	4
43	\$god_	set-dist	0	43	1
44	\$god_	set-dist	0	44	2
45	\$god_	set-dist	0	45	1
46	\$god_	set-dist	0	46	4
47	\$god_	set-dist	0	47	3
48	\$god_	set-dist	0	48	1
49	\$god_	set-dist	0	49	2
50	\$god	set-dist	1	2	2

51	\$god_ set-dist 1 3 2
52	\$god_ set-dist 1 4 2
53	\$god_ set-dist 1 5 1
54	\$god_ set-dist 1 6 2
55	\$god_ set-dist 1 7 2
56	\$god_ set-dist 1 8 1
57	\$god_ set-dist 1 9 1

**Gambar 7.2 Konfigurasi GOD pada *file mobility generator***

1	\$ns_ at 10.000000000000 "\$node_(0) setdest 168.574799996232 706.402109826831 0.700841752294"
2	\$ns_ at 10.000000000000 "\$node_(1) setdest 668.824966069216 655.912329903749 3.236041741468"
3	\$ns_ at 10.000000000000 "\$node_(2) setdest 64.590689571310 178.566081726864 1.563163197166"
4	\$ns_ at 10.000000000000 "\$node_(3) setdest 811.835928031214 432.726846171353 3.864688293276"
5	\$ns_ at 10.000000000000 "\$node_(4) setdest 554.458882421592 103.770397413579 1.091984130240"
6	\$ns_ at 10.000000000000 "\$node_(5) setdest 775.858057378436 721.114158721581 0.387282536960"
7	\$ns_ at 10.000000000000 "\$node_(6) setdest 20.719100735051 75.107117794054 3.930667560217"
8	\$ns_ at 10.000000000000 "\$node_(7) setdest 663.509983792919 82.396439786176 0.675806830537"

9	\$ns_ at 10.000000000000 "\$node_(8) setdest 676.856913519147 17.106029998524 0.661936606672"
10	\$ns_ at 10.000000000000 "\$node_(9) setdest 67.408854204938 708.755844359104 4.657842898469"
11	\$ns_ at 10.000000000000 "\$node_(10) setdest 767.441825645021 194.148869971504 1.499301656581"
12	\$ns_ at 10.000000000000 "\$node_(11) setdest 772.422569923544 433.161400176485 2.780711986913"
13	\$ns_ at 10.000000000000 "\$node_(12) setdest 329.428326459856 106.394510606783 2.955018819693"
14	\$ns_ at 10.000000000000 "\$node_(13) setdest 532.144994320385 91.787532610308 0.522275161611"
15	\$ns_ at 10.000000000000 "\$node_(14) setdest 794.748518617752 275.996622340124 1.274571900747"
16	\$ns_ at 10.000000000000 "\$node_(15) setdest 341.590737179125 116.485048755745 2.568550191181"
17	\$ns_ at 10.000000000000 "\$node_(16) setdest 326.639653193077 167.011531148664 1.011595849509"
18	\$ns_ at 10.000000000000 "\$node_(17) setdest 840.183981498300 749.877821042805 4.961098558714"
19	\$ns_ at 10.000000000000 "\$node_(18) setdest 366.558859577459 212.213709692110 0.558080037842"
20	\$ns_ at 10.000000000000 "\$node_(19) setdest 221.749718830441 513.727691235337 0.248606998173"



21	\$ns_ at 10.000000000000 "\$node_(20) setdest 176.858166508958 518.361647501407 3.595126788781"
22	\$ns_ at 10.000000000000 "\$node_(21) setdest 161.667721480337 319.783355392268 1.772379626905"
23	\$ns_ at 10.000000000000 "\$node_(22) setdest 599.372506611860 740.158382873736 0.110591023230"
24	\$ns_ at 10.000000000000 "\$node_(23) setdest 758.344166943707 797.557310024724 2.337177222626"
25	\$ns_ at 10.000000000000 "\$node_(24) setdest 754.839041888835 316.952229890987 1.141050104596"
26	\$ns_ at 10.000000000000 "\$node_(25) setdest 254.350649566607 368.779542461537 4.613072308600"
27	\$ns_ at 10.000000000000 "\$node_(26) setdest 122.174347776981 791.115316085514 0.769774463287"
28	\$ns_ at 10.000000000000 "\$node_(27) setdest 580.435458286799 393.406230602063 2.824138355772"
29	\$ns_ at 10.000000000000 "\$node_(28) setdest 268.008197994368 425.764211467224 0.400002242113"
30	\$ns_ at 10.000000000000 "\$node_(29) setdest 826.510481644464 773.481433427459 2.029035871858"
31	\$ns_ at 10.000000000000 "\$node_(30) setdest 128.997880844762 445.106320253600 2.125553079442"

**Gambar 7.3** Pergerakan setiap *node* dari potongan Skenario

1	ns_ at 10.193533064858 "\$god_ set-dist 15 25 1"
2	\$ns_ at 10.193533064858 "\$god_ set- dist 25 30 2"
3	\$ns_ at 10.552875494245 "\$god_ set- dist 11 26 4"
4	\$ns_ at 10.552875494245 "\$god_ set- dist 21 26 2"
5	\$ns_ at 10.552875494245 "\$god_ set- dist 26 35 3"
6	\$ns_ at 10.844518098321 "\$god_ set- dist 38 41 2"
7	\$ns_ at 11.631237186258 "\$god_ set- dist 3 4 3"
8	\$ns_ at 11.631237186258 "\$god_ set- dist 3 10 3"
9	\$ns_ at 11.631237186258 "\$god_ set- dist 3 14 4"
10	\$ns_ at 11.631237186258 "\$god_ set- dist 3 23 2"
11	\$ns_ at 11.631237186258 "\$god_ set- dist 3 37 3"
12	\$ns_ at 11.631237186258 "\$god_ set- dist 4 7 3"
13	\$ns_ at 11.631237186258 "\$god_ set- dist 4 40 2"
14	\$ns_ at 11.631237186258 "\$god_ set- dist 7 10 3"
15	\$ns_ at 11.631237186258 "\$god_ set- dist 7 14 4"
16	\$ns_ at 11.631237186258 "\$god_ set- dist 7 19 4"
17	\$ns_ at 11.631237186258 "\$god_ set- dist 7 23 2"
18	\$ns_ at 11.631237186258 "\$god_ set- dist 7 29 4"

19	\$ns_ at 11.631237186258 "\$god_ set- dist 7 37 3"
20	\$ns_ at 11.631237186258 "\$god_ set- dist 10 40 2"
21	\$ns_ at 11.631237186258 "\$god_ set- dist 14 40 3"
22	\$ns_ at 11.631237186258 "\$god_ set- dist 19 40 3"
23	\$ns_ at 11.631237186258 "\$god_ set- dist 23 40 1"
24	\$ns_ at 11.631237186258 "\$god_ set- dist 29 40 3"
25	\$ns_ at 11.631237186258 "\$god_ set- dist 37 40 2"
26	\$ns_ at 11.812148258609 "\$god_ set- dist 5 18 1"
27	\$ns_ at 11.812148258609 "\$god_ set- dist 18 32 2"
28	\$ns_ at 11.935267100098 "\$god_ set- dist 4 31 3"
29	\$ns_ at 11.935267100098 "\$god_ set- dist 10 31 3"
30	\$ns_ at 11.935267100098 "\$god_ set- dist 14 31 4"
31	\$ns_ at 11.935267100098 "\$god_ set- dist 23 31 2"

**Gambar 7.4 Informasi pada GOD dari potongan Skenario**

1	#
2	# nodes: 10, max conn: 8, send rate: 0.25,
3	seed: 1.0
4	#
5	#

```

6 # 1 connecting to 2 at time
  2.5568388786897245
7 #
8 set udp_(0) [new Agent/UDP]
9 $ns_ attach-agent $node_(1) $udp_(0)
10 set null_(0) [new Agent/Null]
11 $ns_ attach-agent $node_(2) $null_(0)
12 set cbr_(0) [new
  Application/Traffic/CBR]
13 $cbr_(0) set packetSize_ 512
14 $cbr_(0) set interval_ 1
15 $cbr_(0) set random_ 1
16 $cbr_(0) set maxpkts_ 10000
17 $cbr_(0) attach-agent $udp_(0)
18 $ns_ connect $udp_(0) $null_(0)
19 $ns_ at 2.5568388786897245 "$cbr_(0)
  start"
20 #

```

**Gambar 7.5 Koneksi yang digunakan pada cbr-2-0.25.txt**

```

1 set
  val(chan) Channel/WirelessC
  hannel ;
2 set
  val(prop) Propagation/TwoRa
  yGround ;
3 set
  val(netif) Phy/WirelessPhy
  ;
4 set
  val(mac) Mac/802_11
  ;

```

```

5  set
   val(ifq)                Queue/DropTail/Pr
   iQueue      ;
6  set
   val(ll)                LL
                        ;
7  set
   val(ant)              Antenna/OmniAnten
   na      ;
8  set
   val(ifqlen)          100
                        ;
9  set
   val(rp)              DSR
                        ;
10 set val(x)            850
11 set val(seed)        0.0
12 set val(y)            850
13 set val(tr)          D-Trg.tr
14 set val(nn)          50
15 set val(cp)          "sc10-
16                       15.txt"
17 set val(sc)          "cbr-2-
18                       0.25"
19 set val(stop)        100.0
20 set ns_              [new Simulator]
21 #
22 # open traces
23 #
24 if { $val(rp) == "DSR" } {
   set val(ifq)          CMUPriQueue

```



49		-ifqType
	\$val(ifq) \	
50		-ifqLen
	\$val(ifqlen) \	
51		-antType
	\$val(ant) \	
52		-propType
	\$val(prop) \	
53		-
	phyType \$val(netif) \	
54		-channel
	\$chan_1_ \	
55		-
	topoInstance \$topo \	
56		-agentTrace
	ON \	
57		-routerTrace
	ON \	
58		-movementTrace ON \
59		-macTrace
	OFF	
60		
61	#Phy/WirelessPhy set CPTresh_ 10.0	
62	#Phy/WirelessPhy set CSTresh_	
	2.78831e-9 ;#100m	
63	#Phy/WirelessPhy set RXThresh_	
	1.11532e-8 ;#50m	
64	#Phy/WirelessPhy set bandwidth_ 1Mb	
65	#Phy/WirelessPhy set Pt_ 0.2818	
66	#Phy/WirelessPhy set freq_ 2.4e+9	
67	#Phy/WirelessPhy set L_ 1.0	
68		
69		

```

70
71         for {set i 0} {$i < $val(nn)
} {incr i} {
72             set node_($i) [$ns_
node]
73             $node_($i) random-
motion 0
74                 ;
75         }
76 puts "Loading connection pattern..."
77 source $val(cp)
78
79 puts "Loading skenario file..."
80 source $val(sc)
81
82 for {set i 0} {$i < $val(nn) } {incr
i} {
83     $ns_ initial_node_pos $node_($i)
20
84 }
85
86 for {set i 0} {$i < $val(nn) } {incr
i} {
87     $ns_ at $val(stop).0 "$node_($i)
reset";
88 }
89
90 $ns_ at $val(stop).0002 "puts \"NS
EXITING...\"; $ns_ halt"
91
92 #proc stop {} {
93     # global ns_ tracefd

```



```

94     # namtrace
95     # $ns_ flush-trace
96     #close $tracefd
97     #close $namtrace
98 #}
99
100 puts $tracefd "M 0.0 nn $val(nn) x
    $val(x) y $val(y) rp $val(rp)"
101 puts $tracefd "M 0.0 sc $val(sc) cp
    $val(cp) seed $val(seed)"
102 puts $tracefd "M 0.0 prop $val(prop)
    ant $val(ant)"
103
104 puts "Start Simulation..."
105 $ns run

```

**Gambar 7.6** File .tcl untuk Protokol Routing DSR dengan Model *TwoRayGround*

```

    set
    val(chan)           Channel/WirelessCh
1  annel      ;
    set
    val(prop)          Propagation/TwoRay
2  Ground      ;
    set
    val(netif)         Phy/WirelessPhy
3                ;
    set
    val(mac)           Mac/802_11
4                ;
    set
    val(ifq)           Queue/DropTail/Pri
5  Queue      ;

```

```

6   set
   val(ll)                LL
   ;
7   set
   val(ant)              Antenna/OmniAntenn
   a                    ;
8   set
   val(ifqlen)          100
   ;
9   set
   val(rp)              AODV
   ;
10  set val(x)           850
11  set val(y)           850
12  set val(seed)       0.0
13  set val(tr)         A-Trg.tr
14  set val(nn)         50
   set val(cp)          "sc5-
15  5.txt"              ;
   set val(sc)           "cbr-2-
16  0.25"              ;
17  set val(stop)       100.0
18  set ns_              [new Simulator]
19
20  #
21  # open traces
22  #
23
24  set tracefd [open A-Trg.tr w]
25
26  $ns_ trace-all $tracefd
27

```

```

28 set namtrace [open A-Trg.nam w]
29
30     $ns_ namtrace-all-wireless
31 $namtrace $val(x) $val(y)
32
33 Phy/WirelessPhy set RXThresh_
34 1.42681e-08 ;#100m
35
36 set topo [new Topography]
37 $topo load_flatgrid $val(x) $val(y)
38
39 set god_ [create-god $val(nn)]
40
41 set chan_1_ [new $val(chan)]
42
43     $ns_ node-config -adhocRouting
44 $val(rp) \
45     -llType
46 $val(ll) \
47     -macType
48 $val(mac) \
49     -ifqType
50 $val(ifq) \
51     -ifqLen
52 $val(ifqlen) \
53     -antType
54 $val(ant) \
55     -propType
56 $val(prop) \
57     -
58 phyType $val(netif) \

```

```

51 $chan_1_ \                                -channel
52 $topo \                                    -topoInstance
53 ON \                                        -agentTrace
54 ON \                                        -routerTrace
55 OFF                                         -macTrace
56
57 #Phy/WirelessPhy set CPTHresh_ 10.0
58 #Phy/WirelessPhy set CSTresh_
59 2.78831e-9 ;#100m
60 #Phy/WirelessPhy set RXThresh_
61 1.11532e-8 ;#50m
62 #Phy/WirelessPhy set bandwidth_ 1Mb
63 #Phy/WirelessPhy set Pt_ 0.2818
64 #Phy/WirelessPhy set freq_ 2.4e+9
65 #Phy/WirelessPhy set L_ 1.0
66
67         for {set i 0} {$i < $val(nn) }
68 {incr i} {
69         set node_($i) [$ns_
70         node]
71         $node_($i) random-
72         motion 0
73         ;
74     }
75
76 puts "Loading connection pattern..."
77 source $val(cp)

```

```
74
75 puts "Loading skenario file..."
76 source $val(sc)
77
78 for {set i 0} {$i < $val(nn) } {incr
79 i} {
80     $ns_ initial_node_pos $node_($i)
81     50
82 }
83
84 for {set i 0} {$i < $val(nn) } {incr
85 i} {
86     $ns_ at $val(stop).000000001
87     "$node_($i) reset";
88 }
89
90 $ns_ at $val(stop).000000001 "puts
91 \"NS EXITING...\""; $ns_ halt"
92
93
94 proc stop {} {
95     global ns_ tracefd
96     namtrace
97     $ns_ flush-trace
98     close $tracefd
99     close $namtrace
100 }
101
102 puts "Start Simulation..."
103 $ns run
```

**Gambar 7.7 File .tcl untuk Protokol Routing AODV**

```
1 BEGIN {
2
3     sendLine = 0;
4
5     recvLine = 0;
6
7     fowardLine = 0;
8
9 }
10
11
12
13 $0 ~/^s.* AGT/ {
14
15     sendLine ++ ;
16
17 }
18
19
20
21 $0 ~/^r.* AGT/ {
22     recvLine ++ ;
23
24 }
25
26
27 $0 ~/^f.* RTR/ {
28
29     fowardLine ++ ;
```

```

30
31 }
32
33
34
35 END {
36     printf "cbr s:%d r:%d, r/s
Ratio:%.4f, f:%d \n", sendLine,
recvLine,
(recvLine/sendLine), fowardLine;
37
38 }

```

**Gambar 7.8 Implementasi *Packet delivery ratio***

```

1 BEGIN{
2   recvd = 0;
3   rt_pkts = 0;
4 }
5
6 {
7
8   if (( $1 == "r" ) && ( $7 == "cbr" ||
$7 == "tcp" ) && ( $4 == "AGT" ))
9     recvd++;
10
11  if (($1 == "s" || $1 == "f") && $4 ==
"RTR" && ($7 == "udp" || $7 == "AODV"
|| $7 == "message" || $7 == "DSR"))
12    rt_pkts++;

```

```

13 }
14 END{
15
16 printf("\n");
17 printf("total no of data
18 packets\t%d\n",recvd);
19 printf("\ntotal no of routing
20 packets\t%d\n",rt_pkts);
21 printf("Normalized Routing Load =
22 %.3f\n", rt_pkts/recvd);
23 printf("\n");
24 }

```

**Gambar 7.9 Implementasi Packet delivery ratio**

```

1 BEGIN {
2
3     seqno = -1;
4
5
6     count = 0;
7
8 }
9
10 {
11     if($4 == "AGT" && $1 == "s" &&
12 seqno < $6) {
13
14         seqno = $6;

```



```
15
16     }
17
18     #end-to-end delay
19
20     if($4 == "AGT" && $1 == "s") {
21
22         start_time[$6] = $2;
23
24         } else if(($7 == "tcp" || $7 == "cbr")
25 && ($1 == "r")) {
26
27         end_time[$6] = $2;
28
29         } else if($1 == "D" && ($7 ==
30 "tcp" || $7 == "cbr")) {
31
32         end_time[$6] = -1;
33
34     }
35 }
36
37 END {
38
39     for(i=0; i<=seqno; i++) {
40
41         if(end_time[i] > 0) {
42
```

```

        delay[i] = end_time[i] -
43 start_time[i];
44
45         count++;
46
47     }
48
49     else
50
51     {
52
53         delay[i] = -1;
54
55     }
56
57 }
58
59 for(i=0; i<=seqno; i++) {
60
61     if(delay[i] > 0) {
62
63         n_to_n_delay =
64         n_to_n_delay + delay[i];
65     }
66
67 }
68
69     n_to_n_delay = n_to_n_delay/count;
70
```

```

71
72     print "\n";
73
74     print "Rata-rata End-to-end
75     delay    = " n_to_n_delay * 1000 "
76     ms";
77
78     print "\n";

```

**Gambar 7.10 Implementasi *End-to-end delay***

## Instalasi NS-2

Pada Tugas Akhir ini dijelaskan cara instalasi NS-2, berikut langkah-langkah instalasi NS-2:

- Lakukan *update* komponen terbaru dari Ubuntu dan pastikan Ubuntu sudah ter-*update* seluruhnya.

```

$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get update

```

**Gambar 7.11 Perintah *update* pada Terminal Ubuntu**

- Kemudian lakukan instalasi modul-modul dependensi dari NS-2 yaitu *build-essential*, *autoconf*, *automake*, *tcl8.5-dev*, *tk8.5-dev*, *perl*, *xgraph*, *libxt-dev*, *libx11-dev*, *libxmu-dev* dan *gcc-4.4*.
- Lakukan ekstraksi pada *file* NS-2 yang sudah diunduh dan lakukan penginstalan.
-

- Untuk melakukan pengecekan pada NS-2 dapat dilakukan dengan cara melakukan *command* “ns” pada terminal dan apabila muncul tanda “%” maka NS-2 telah ter-*install*.



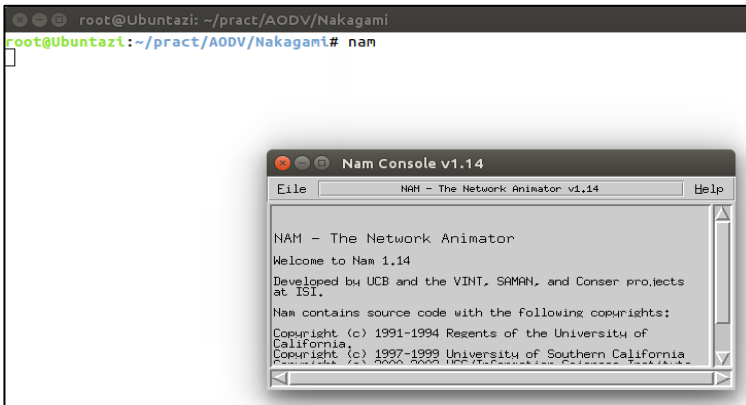
```

root@Ubuntazi: ~/pract/AODV/Nakagami
root@Ubuntazi:~/pract/AODV/Nakagami# ns
% █

```

**Gambar 7.12** Tampilan *command line* ‘ns’ dijalankan

- Untuk melakukan pengecekan pada NAM dapat dilakukan dengan cara melakukan *command* “nam” pada terminal dan akan muncul tampilan seperti berikut:



```

root@Ubuntazi: ~/pract/AODV/Nakagami
root@Ubuntazi:~/pract/AODV/Nakagami# nam

```

Nam Console v1.14

File NAM - The Network Animator v1.14 Help

NAM - The Network Animator  
Welcome to Nam 1.14  
Developed by UCB and the VINT, SAMAN, and Conser projects at ISI.  
Nam contains source code with the following copyrights:  
Copyright (c) 1991-1994 Regents of the University of California,  
Copyright (c) 1997-1999 University of Southern California

**Gambar 7.13** Tampilan *command line* ‘nam’ dijalankan

## BIODATA PENULIS



**Azi Prastyo**, lahir di Pontianak pada 17 Juni 1992. Penulis adalah anak terakhir dari tiga bersaudara. Dengan masa kecil yang *nomaden*, penulis pernah merasakan hidup di beberapa kota besar seperti Pontianak, Manado, dan Surabaya. Penulis menempuh pendidikan formal di SDN 11 Sarapung, Manado hingga kelas IV SD penulis melanjutkan pendidikan di SDN Kertajaya XIII Surabaya atau biasa dikenal SDN PUJA II Surabaya (1998-2004), SMPN 6 Surabaya (2004-2007), SMAN 15 Surabaya (2007-2010). Pada tahun 2010, penulis memulai pendidikan S1 Jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya angkatan 2010 yang terdaftar dengan NRP 5110100225. Di jurusan Teknik Informatika, penulis mengambil bidang minat Arsitektur Jaringan Komputer (AJK) dan memiliki ketertarikan di Jaringan Komputer. Selama menempuh kuliah, penulis juga aktif dalam organisasi kemahasiswaan seperti Himpunan Mahasiswa Teknik Computer (HMTC) dan berkesempatan menjabat sebagai Kepala Departemen Pengembangan Profesi pada tahun 2013. Penulis dapat dihubungi melalui alamat email [aziprastyo.tc@gmail.com](mailto:aziprastyo.tc@gmail.com).