TUGAS AKHIR – IF4802

Analisis Kinerja Model Propagasi *Nakagami* pada *Ad hoc On Demand Multipath Distance Vector* (AOMDV) *Routing* pada MANET

Paul Aldy Sarumaha
NRP 05111440000072

Dosen Pembimbing I
Ir. MUCHAMMAD HUSNI, M.Kom.

Dosen Pembimbing II
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – IF4802

Analisis Kinerja Model Propagasi *Nakagami* pada *Ad hoc On Demand Multipath Distance Vector* (AOMDV) *Routing* pada MANET

Paul Aldy Sarumaha
NRP 05111440000072

Dosen Pembimbing I
Ir. MUCHAMMAD HUSNI, M.Kom.

Dosen Pembimbing II
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

UNDERGRADUATE THESES – IF4802

Analysis Nakagami Propagation Model with Ad hoc On Demand Multipath Distance Vector (AOMDV) Routing on MANET

Paul Aldy Sarumaha
NRP 05111440000072

Advisor I
Ir. MUCHAMMAD HUSNI, M.Kom.

Advisor II
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Analisis Kinerja Model Propagasi Nakagami pada Ad hoc On Demand Multipath Distance Vector (AOMDV) Routing pada MANET

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur Jaringan Komputer
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :
Paul Aldy Sarumaha
NRP : 05111440000072

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Ir. MUCHAMMAD HUSNI, M.Kom. .....
NIP: 196002211984031001 (pembimbing 1)

Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc. .....
NIP: 198410162008121002 (pembimbing 2)

SURABAYA
JANUARI 2019

[Halaman ini sengaja dikosongkan]

Analisis Kinerja Model Propagasi *Nakagami* pada *Ad hoc On Demand Multipath Distance Vector* (AOMDV) *Routing* pada MANET

Nama Mahasiswa : Paul Aldy Sarumaha
NRP : 05111440000072
Jurusan : Informatika FTIK-ITS
Dosen Pembimbing 1 : Ir. Muchammad Husni, M.Kom.
Dosen Pembimbing 2 : Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

ABSTRAK

MANET merupakan jaringan wireless yang terdiri dari kumpulan node yang bergerak yang memungkinkan untuk melakukan komunikasi secara langsung. Hal ini memungkinkan terjadinya komunikasi jaringan tanpa bergantung pada ketersediaan infrastruktur jaringan yang tetap.

Ad hoc On Demand Multipath Distance Vector atau disingkat AOMDV adalah salah satu routing protokol reaktif pengembangan dari routing protokol unipath AODV. AOMDV mempunyai karakteristik yang mirip dengan AODV, AOMDV berbasis vektor dan menggunakan pendekatan hop by hop. Perbedaannya pada AOMDV bisa ditemukan beberapa path dalam satu pencarian rute.

Implementasi pada lingkungan MANET dapat dilakukan dengan menggunakan simulasi. Simulasi dilakukan dengan menggunakan aplikasi Network Simulator 2 (NS-2). Pada penelitian ini menghasilkan kinerja dari model propagasi Nakagami pada protokol routing AOMDV di lingkungan MANET berdasarkan parameter kecepatan maksimum dengan nilai Packet Delivery Ratio(PDR) terbaik sebesar 87,463% , Routing Overhead(RO) terbaik sebesar 5044,4 paket, dan End-to-End Delay terbaik sebesar 0,0149 detik.

***Kata kunci**: MANET, AOMDV, NS-2,Nakagami*

[Halaman ini sengaja dikosongkan]

Analysis Nakagami Propagation Model with Ad hoc On Demand Multipath Distance Vector (AOMDV) on MANET

Student Name	: Paul Aldy Sarumaha
Student ID	: 05111440000072
Major	: Informatics FTIK-ITS
Advisor 1	: Ir. Muchammad Husni, M.Kom.
Advisor 2	: Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

ABSTRACT

MANET is a Wireless Network base on walking nodes for direct communication. With this technology we can have a communication without a big infrastructure.

Ad hoc On Demand Multipath Distance Vector or AOMDV is a routing protocol development of AODV. AOMDV has similar characteristics to AODV. AOMDV is vector based and using hop by hop approach. The difference between AOMDV and AODV lies in the number of routes found in each route search. In AOMDV several paths can be found in one route search.

Implementation on MANET was base on simulation. Simulation will be using Network Simulator 2 with Nakagami propagation model on AOMDV Protocol and we got the best Packet Delivery Ratio(PDR) is 87,463%, best Routing Overhead(RO) is 5044,4 packet, and best End to End Delay is 0,0149 second.

Keyword : MANET,AOMDV,NS-2,Nakagami

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul :

"Analisis Kinerja Model Propagasi *Nakagami* pada *Ad hoc On Demand Multipath Distance Vector* (AOMDV) *Routing* pada MANET"

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa atas segala nikmat dan rezeki yang telah di berikan kepada penulis.
2. Orangtua penulis yang selalu memberikan doa dan dukungan baik moriil maupun materiil selama penulis menjalani kehidupan.
3. Keluarga yang senantiasa memberikan dukungan penuh baik secara moriil maupun materiil.
4. Bapak Ir. Muchammad Husni, M.Kom selaku dosen pembimbing pertama yang telah bersedia meluangkan waktu untuk membimbing dan memotivasi penulis serta memberi arahan dalam mengerjakan tugas akhir.
5. Bapak Dr.Eng Radityo Anggoro S.Kom., M,Sc. selaku dosen pembimbing kedua yang juga telah bersedia meluangkan waktu untuk memberikan arahan dan memotivasi penulis dalam mengerjakan tugas akhir.
6. Bapak Arya Yudhi Wijaya, S.Kom, M.Kom. selaku dosen wali dari penulis yang memberikan arahan selama menjalani perkuliahan.
7. Bapak/Ibu dosen, staf dan karyawan Jurusan Teknik Informatika ITS yang telah banyak memberikan

dukungan, ilmu dan bimbingan yang tak ternilai harganya bagi penulis.

8. Abdul Majid Hasani sebagai teman pertama penulis dari luar daerah asal selama melangsungkan perkuliahan yang selalu membantu penulis beradaptasi dengan lingkungan perkuliahan.
9. Rizky Fenaldo Maulana sebagai teman dan pembimbing ke “3” penulis yang selalu memberikan arahan dalam pengerjaan Tugas Akhir
10. Buyung Abiyoso, Faris Salbari, Muchammad Baruno, Brilian WM, Hamka Aminullah, M. Luqmanul Hakim, Hanendyo Indira, Aldi Febriansyah, Anandi Jaya, Bayu Aji dan Naufan Arifie sebagai sesama pejuang Tugas Akhir yang selalu membantu penulis dalam mengerjakan tugas akhir ini.
11. Teman-teman Warkop x Jojoran 2014 yang telah memberikan pengalaman berarti selama kuliah.
12. Teman-teman angkatan 2014 yang sudah membantu penulis selama di kampus.
13. Teman-teman grup Korek Api yang selalu memberikan semangat dan canda tawa kepada penulis untuk tetap melakukan yang terbaik.
14. Teman-teman dan pemilik Warung Kopi Pak Toyo yang selalu memberi tempat, dukungan dan pembelajaran baru selama perkuliahan.
15. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, 23 Januari 2019
Penulis

Paul Aldy Sarumaha

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

[Halaman ini sengaja dikosongkan]

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

Dengan kemajuan ilmu pengetahuan dan teknologi yang semakin pesat di era globalisasi saat ini, maka mengharuskan kita untuk turut serta dalam mengikuti perkembangan tersebut dan kebutuhan masyarakat akan komunikasi dalam mengakses informasi pun sudah semakin mudah. Beberapa perangkat komunikasi yang bersifat *mobile* seperti *handphone, tablet*, laptop dan lainnya semakin ikut berkembang karena adanya teknologi nirkabel (*wireless)*.

Teknologi jaringan nirkabel (*wireless)* memungkinkan perangkat komunikasi seperti *handphone* atau *laptop* bisa berhubungan langsung dengan perangkat komunikasi yang lainnya tanpa dibatasi oleh jaringan infrastruktur yang bersifat tetap. Teknologi *wireless* dapat membuat beberapa perangkat komunikasi *mobile* yang saling terhubung di dalam suatu area untuk membentuk sebuah jaringan yang sifatnya tidak permanen atau sementara. Jaringan ini bisa disebut *Mobile Ad Hoc Network* (MANET).

Mobile Ad-Hoc Network (MANET) merupakan suatu jaringan yang terorganisir secara mandiri tanpa adanya dukungan infrastruktur. Dalam MANET, setiap *node* bergerak secara bebas, sehingga jaringan dapat mengalami perubahan topologi dengan cepat. Karena *node* dalam MANET memiliki jarak propagasi yang terbatas, mengakibatkan beberapa *node* tidak bisa berkomunikasi secara langsung dengan *node* lainnya. Pada MANET, terdapat beberapa jenis protokol routing yang terbagi dalam tiga macam yaitu protokol routing *proactive, reactive* dan *hybrid.*

Implementasi pada lingkungan MANET dapat dilakukan dengan menggunakan sebuah simulasi sehingga penelitian ini dapat dilakukan untuk mempelajari sistem dengan baik. Simulasi dilakukan dengan menggunakan *Network simulator 2* (NS-2). Implementasi ini akan dilakukan analisa performa protokol *routing reactive* yaitu *Ad-hoc On Demand Multipath Distance Vector* (AOMDV) yang menggunakan model propagasi *Nakagami* yang ada pada NS-2

Hasil yang diharapkan dari Tugas Akhir ini adalah suatu hasil analisa kinerja model propagasi *Nakagami* pada protokol *routing* AOMDV di lingkungan MANET dengan menggunakan aplikasi. Performa protokol *routing* tersebut diukur berdasarkan *routing overhead*, *packet delivery ratio*, dan *delay* pengiriman paket dari *node* ke *node* lainnya.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana kinerja protokol *routing* AOMDV pada Manet?
2. Bagaimana kinerja model *Nakagami* pada protokol *routing* AOMDV di lingkungan MANET?

1.3. Batasan Permasalahan

Beberapa batasan masalah yang terdapat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Protokol *routing* hanya dijalankan dan diujicobakan pada aplikasi *Network Simulator 2* (NS-2)
2. Protokol *routing* yang diujicobakan adalah AOMDV
3. Lingkungan jaringan yang digunakan untuk uji coba adalah *Mobile Ad hoc Network* (MANET)
4. Model transmisi yang akan dianalisis dalam Tugas Ahir ini adalah *Nakagami*
5. Analisis Kinerja Didasarkan pada *Packet Dilivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay*.

1.4. Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah menganalisis model propagasi *Nakagami* pada protokol AOMDV di lingkungan MANET menggunakan aplikasi *Network Simulator 2* (NS-2) yang efisien dengan parameter *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay*.

1.5. Manfaat

Manfaat Tugas Akhir ini diharapkan dapat menghasilkan hasil analisis kinerja model propagasi *Nakagami* pada protokol AOMDV yang efisien dengan parameter *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), *End-to-End Delay*.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1.6.1. Penyusunan proposal Tugas Akhir

Proposal tugas akhir ini berisi mengenai rencana Melakukan analisa pada model Propagasi *Nakagami* di lingkungan MANET. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.6.2. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan

pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik Tugas Akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. MANET
2. NS-2
3. AOMDV
4. Nakagami

1.6.3. Implementasi

Pada tahap ini dilakukan implementasi berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada serta pengumpulan informasi yang telah dilakukan. Tahap ini suatu bentuk awal aplikasi yang akan diimplementasikan didefinisikan. Pada tahapan ini dibuat prototype sistem atau simulasi dari protocol routing AOMDV yang menggunakan model *Nakagami* pada MANET

1.6.4. Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap system yang sudah dibuat. Pengujian dan evaluasi akan diketahui berdasarkan nilai dari *Paket Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay*.

1.6.5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Tinjauan Pustaka

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan metode yang nantinya akan diimplementasikan dan dilakukan pengujian dari aplikasi yang dibangun.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan sistem yang sudah dilakukan pada tahap perancangan. Penjelasan berupa scenario *node node* pada jaringan *wireless*, konfigurasi sistem dan skrip analisis yang digunakan untuk menguji performa protocol *routing*

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari sistem dan performa dalam skenario mobilitas *ad hoc* yang dibuat dalam *network simulator*.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

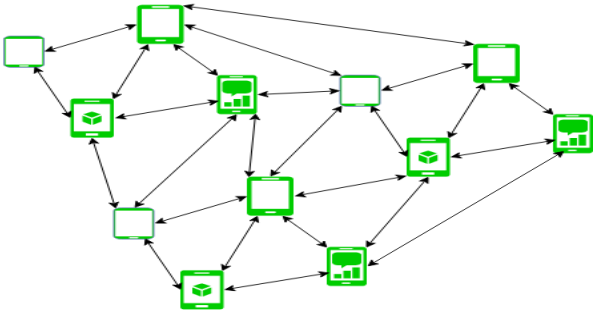
Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II
TINJAUAN PUSTAKA

Bab ini menjelaskan tentang tinjauan pustaka yang menjadi dasar pembuatan Tugas Akhir. Beberapa teori, pustaka, dan teknologi yang mendasari pengerjaan Tugas Akhir ini. Serta memberi gambaran terhadap alat, protokol *routing* serta definisi yang digunakan dalam pembuatan Tugas Akhir.

2.1. *Mobile Ad Hoc Network* (MANET)

Mobile Ad-Hoc Network (MANET) merupakan kumpulan dari perangkat bergerak atau *mobile device* yang dapat digunakan menjadi sekumpulan *node node* tanpa infrastuktur yang terpusat. Sifat dinamis yang dimiliki MANET sendiri membuat setiap *node* yang ada di dalam jaringan tersebut berperilaku sebagai *router* yang mana memiliki peran dalam penemuan dan pemeliharaan rute pengiriman paket ke *node* lain dalam jaringan. Keuntungan utama dalam penggunaan MANET sendiri adalah bersifat fleksibel, biaya yang rendah, dan bersifat tangguh. Keunggulan MANET ini memiliki implementasi yang penting terhadap operasi militer di wilayah perang, operasi penyelamatan sandera perang dan menyebarkan informasi secara cepat dan menyeluruh ketika perang[1].



Gambar 2.1 Ilustrasi skenario MANET [7]

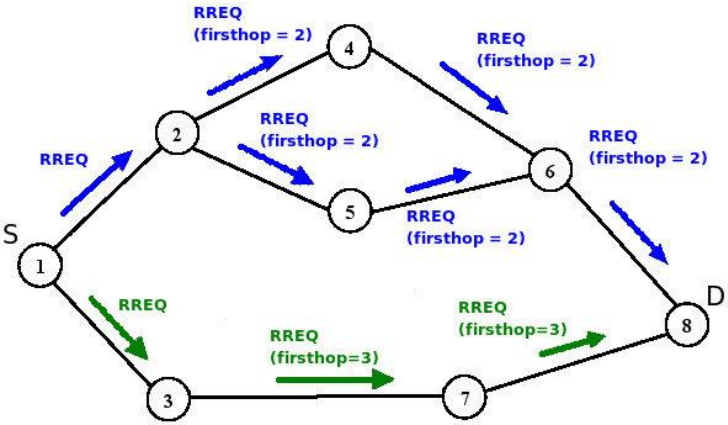
Protokol routing yang tersedia dalam MANET terbagi menjadi tiga kategori yaitu.

- *Proactive routing*: merupakan sebuah protokol berdasar pada sebuah *routing table* terus di-*update* dalam waktu berkala. Contoh dari tipe routing ini adalah *Optimized Link State Routing* (OLSR), *Destination Sequenced Distance-Vector* (DSDV) dan *Better Approach To Mobile Ad-hoc Network* (B.A.T.M.A.N).
- *Reactive routing*: jenis protokol ini akan mencari rute (*on demand*) dengan cara membanjiri jaringan dengan paket *router request*. Sehingga dapat menyebabkan jaringan akan penuh (*clogging*). Contoh dari protokol routing ini adalah *Dynamic Source Routing* (DSR), *Ad-hoc on Demand Distance Vector(*AODV).
- *Hybrid Routing*: jenis routing ini menggabungkan kelebihan dari *proactive routing* dan *reactive routing* untuk mendapatkan sebuah protokol yang paling efektif. Contoh dari *hybrid routing* ini adalah *Zone Routing Protocol*(ZRP) dan *Zone-based Hierarchical Link State Routing Protocol* (ZHLS).

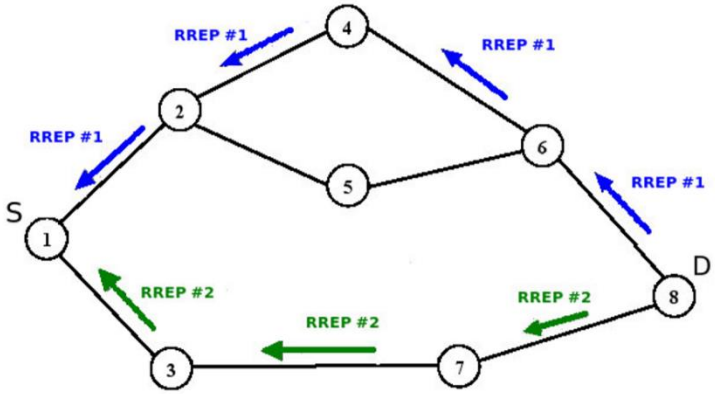
2.2. *Ad-hoc On Demand Multipath Distance Network* (AOMDV)

AOMDV merupakan sebuah *routing protocol* reaktif pengembangan dari protokol *routing unipath* AODV. Dari namanya, *On-Demand* yang berarti pada saat diminta atau diperlukan, atau bisa diartikan rute akan dicari saat diperlukan. AOMDV menyediakan dua layanan utama yaitu *route discovery* dan *maintenance*. AOMDV mempunyai karakteristik yang mirip dengan AODV, AOMDV berbasis vektor dan menggunakan pendekatan *hop-by-hop*. Perbedaan antara AOMDV dan AODV terletak pada jumlah rute yang ditemukan dalam tiap kali pencarian rute atau *route discovery*. Pada AOMDV bisa ditemukan beberapa

path dalam satu pencarian rute, sehingga apabila rute pilihan pertama terjadi kegagalan maka bisa dialihkan ke rute yang lain[2].



Gambar 2.2.1 Contoh *Route Request*[6]



Gambar 2.2.2 Contoh *Route Reply*[6]

2.3. Model Propagasi *Nakagami*

Model Nakagami bersifat lebih umum dan bisa diterapkan untuk berbagai kondisi *fading*, tergantung pada parameter m yang digunakan. Nakagami memiliki *probability density function* (PDF) yang dinyatakan pada persamaan (1) dibawah ini [3].

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

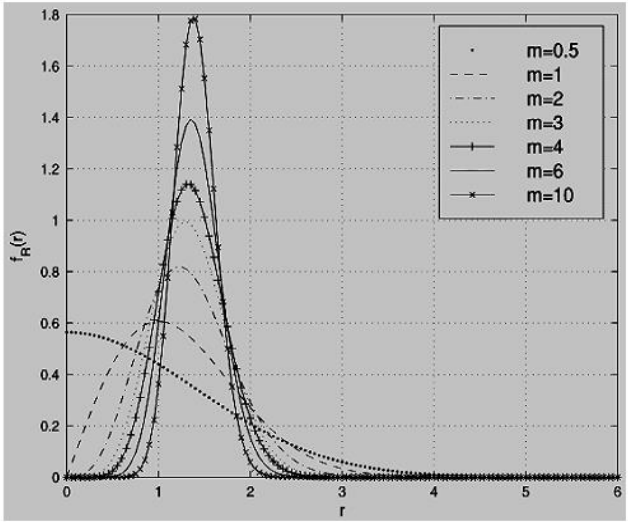
dimana :
m = parameter *fading*
Γ = fungsi gamma
R = amplitude *fading*
Ω = E[R^2]

Pengaruh parameter-m pada Nakagami :

1. Apabila m ≤ 1, maka persamaan (2) menjadi *probabilitas density function* dari kanal fading Rayleigh.
2. Apabila m > 1, merujuk ke kanal fading Rician. Dimana kanal fading Rician memiliki factor K yang memiliki pengaruh disini. Sehingga nilai m adalah sebagai persamaan (2) [4].

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

Pengaruh dari parameter-m dapat dilihat pada grafik *Probability Density Function (PDF)* Nakagami-m-m seperti pada Gambar 2.3.



Gambar 2.3.1 Kanal *Nakagami*-m-m untuk parameter-m yang berbeda [3]

2.4. Model Propagasi *TwoRayGround*

Dalam mobile radio channel, *TwoRayGround* merupakan model propagasi yang memprediksi *path loss* ketika sinyal yang diterima terdiri dari komponen *Line of Sight* dan komponen *Multi path* yang dibentuk oleh *ground reflected wave*[5]. Model ini memiliki keakuratan yang tinggi dalam memperkirakan sinyal dalam skala yang luas dan jauh. Power yang diterima dengan jarak di diberikan oleh persaman 3.

$$\Pr(d) = \frac{PtGtGrht^2hr^2}{d^4L} \quad (3)$$

dimana ht dan hr adalah tinggi dari antena *transmitter* dan *receiver*, Gt dan Gr adalah tegangan dari antenna *transmitter* dan *receiver*. nilai L diasumsikan sama dengan nilai L pada propagasi *free space*, L = 1[7].

2.5. *Network Simulator* 2 (NS-2)

NS2 adalah alat simulasi jaringan *open source* 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 dengan simulator ini. Karena kefleksibelannya, NS2 menjadi popular dikalangan komunitas peneliti sejak awal kemunculannya pada tahun 1989.

NS2 terdiri dari dua bahasa pemrograman yaitu C++ dan OTcl(*Objek-oriented Tool Command Language*). C++ mendefinisikan mekanisme internal dari simulasi objek dan OTcl berfungsi untuk menset simulasi dengan assembly dan mengkonfigurasi objek sebagai penjadwalan diskrit. C++ dan OTcl saling berhubungan menggunakan TclCL. Setelah simulasi, output NS2 dapat berupa basis teks atau animasi berdasarkan simulasi. Untuk menginterprestasikan output ini secara grafik dan interaktif maka dibutuhkan NAM (Network Animator) dan XGraph. Untuk menganalisa tingkah laku dari jaringan user dapat mengekstrak subset dari data teks dan mentransformasikannya agar menjadi lebih atraktif.

Pada prakteknya, NS2 merupakan simulasi yang berjalan pada sistem UNIX. Oleh sebab itu NS2 dapat berjalan dengan baik di sistem operasi Linux, OpenBSD, FreeBSD, dan sistem operasi berbasis unix lainnya. Walaupun demikian NS2 dapat juga berjalan pada Windows dengan menggunakan tool tambahan yaitu Cygwin. Cygwin adalah *port* dari *tool* pengembangan GNU (*GNU's Not UNIX*) untuk Microsoft Windows

2.6. AWK

AWK merupakan sebuah pemrograman seperti pada shell atau C yang memiliki karakteristik yaitu sebagai alat yang cocok untuk memanipulasi sebuah text yang dapat digunakan sebagai ekstraksi dari sebuah *dataset*. AWK ditulis menggunakan Bahasa pemrogramannya sendiri yaitu awk *programming language*.

Pada tugas akhir ini penulis menggunakan AWK sebagai alat untuk membuat script dalam perhitungan *Packet Delivery Ratio* (PDR), *End-to-End Delay* (E2D), dan *Routing Overhead* (RO) dari hasil *trace* NS-2.[7]

2.7. *Generator File Node Movement*

CMU atau *Carnegie Mellon University* mengembangkan sebuah alat bernama 'setdest' yang digunakan untuk men-*generate random movement* dari *node* di jaringan *wireless*. *Node movement* dihasilkan dengan kecepatan maksimal yang spesifik serta penempatan setiap *node* nya yang acak ataupun yang telah ditentukan. Apabila *node* yang sudah ditentukan sampai pada lokasi tujuan maka *node* tersebut akan berpindah ke lokasi selanjutnya. Pergerakan *node* tersebut dapat dihentikan sementara.

Sebelum menjalankan program simulasi pengguna harus menjalankan program 'setdest'. Format *command line* 'setdest' ditunjukan seperti dibawah ini dan keterangannya ditunjukan pada tabel 2.1.

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

Table 2.1 Penjelasan *Command Line* 'setdest'

Parameter	Keterangan
-v *version*	Menentukan versi dari 'Setdest' yang digunakan

-n *num*	Menentukan jumlah node yang dibuat pada skenario
-p *pausetime*	Menentukan durasi berhenti pada sebuah paket apabila sudah sampai di tujuan
-M *maxspeed*	Menentukan kecepatan maksimum dari sebuah *node*.
-t *simtime*	Menentukan lama waktu jalannya simulasi.
-x *max* x	Menentukan panjang maksimum dari area simulasi.
-y *max* y	Menentukan lebar maksimum dari area simulasi

Command line 'setdest'yang sudah di-*generate* menghasilkan *file* yang berisi jumlah *node* dan pergerakan dari *node* yang sudah dibuat dalam *file* berbentuk .tcl. Selain pergerakan *node file* tersebut juga berisi tentang perpindahan rute.

Untuk membuat skenario *node-movement* yang terdiri dari 50 *node*, bergerak dengan kecepatan maksimum 5 m/s, simulasi akan berhenti setelah 100 detik dengan batas topologi yang diartikan sebagai 800 x 800 meter, contoh *command line* seperti perintah dibawah ini

```
./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x 800 -y 800 > scenario.txt
```

2.8. *File Traffic Connection Pattern*

Random Traffic Connection memiliki dua buah tipe *traffic* yaitu TCP dan CBR. Keduanya dapat dibuat *menggunakan script traffic scenario generator*. *Script* ini dapat membantu kita untuk men-*generate* beban *traffic*. *Script* yang dimaksud di dalam sini adalah 'cbrgen.tcl'. program "cbrgen.tcl" digunakan sesuai dengan *command line* dibawah ini dan dengan keterangan yang ditunjukan pada tabel 2.2.[4]

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

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

Table 2.2 Penjelasan *Command Line* 'cbrgen.tcl'

Parameter	Keterangan
-type cbr|tcp	Menentukan jenis *traffic* yang digunakan.
-nn *nodes*	Menentukan jumlah total *node*.
-s *seed*	Menentukan nilai *random seed*
-mc *connection*	Menentukan jumlah koneksi antar node.
-rate *rate*	Menentukan jumlah paket per detik yang terkirim.

[Halaman ini sengaja dikosongkan]

BAB III
ANALISIS DAN PERANCANGAN SISTEM

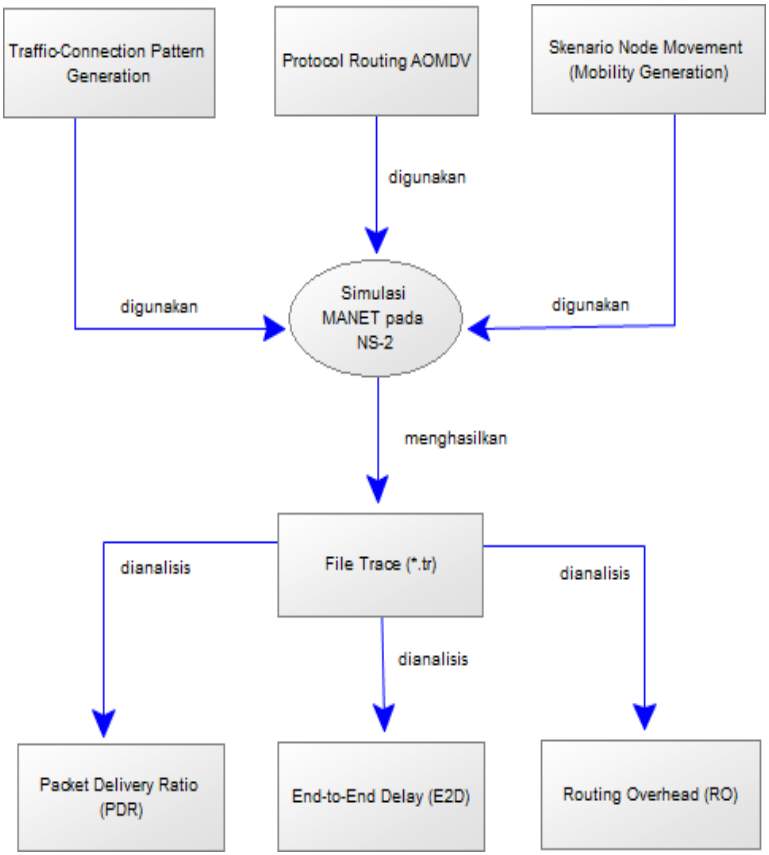
Pada bab ini akan dijelaskan mengenai dasar perancangan dari perangkat lunak yang akan dibangun dalam tugas akhir ini. Perancangan tersebut mencakup deskripsi umum aplikasi, arsitektur sistem, model fungsional dan diagram alur aplikasi.

3.1. Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dilakukan analisis tentang performa model propagasi *Nakagami* pada MANET. Pada pembuatan skenario penulis menggunakan *mobility generator* yang bersifat random way point dan telah ada pada *Network Simulator-2* (NS-2) yaitu dengan cara melakukan *generate file node movement* dan membuat koneksi antar *node* menggunakan *file traffic connection pattern*. Rancangan Simulasi yang dibuat dapat dilihat pada Gambar 3.1.

Dalam penelitian ini penulis akan menganalisa performa dari model Propagasi *Nakagami* pada simulasi skenario yang dijalankan pada NS-2 menggunakan *routing protocol* AOMDV pada sistem operasi Linux.

Pada tiap skenario, kecepatan maksimum dari satu *node* ke *node* lainnya dibuat berbagai variasi yaitu 5, 10 dan 15 m/s. Hasil proses uji coba dari tiap skenario akan menghasilkan sebuah *trace file* yang nantinya akan dilakukan analisis perhitungan metrik *packet delivery Ratio* (PDR), *End-to-End Delay* (E2E), dan *Routing Overhead* (RO). Dari hasil metrik tersebut dianalisis performa propagasi *Nakagami*.



Gambar 3.1 Skema umum sistem

3.2. Perancangan Skenario

Skenario uji coba dalam Tugas Akhir ini dibuat dengan menggunakan *mobility generation* setelah itu penguji akan membuat koneksi dari skenario yang sudah ada dengan *file traffic connection* yang sudah ada pada NS-2. Pada tugas akhir ini skenario dibuat untuk melihat pergerakan dari *node* yang sudah dibuat berdasarkan pada tiga kecepatan maksimal yaitu 5m/s,

10m/s, 15m/s. Sedangkan untuk koneksinya digunakan dua *node* untuk menentukan *node* pengirim dan *node* penerima paket.

3.2.1. Perancangan Skenario Node Movement (Mobility Generation)

Perancangan skenario dibuat dengan men-*generate file node movement* yang sudah disediakan oleh NS-2 biasa kita kenal dengan *tools* bernama'setdest' yang akan digunakan dalam *file* tcl selama simulasi pada NS-2 sebagai bentuk pergerakan *node* yang berpindah-pindah.

Table 3.1 Penjelasan Skenario *node movement*

No.	Parameter	Spesifikasi
1	Jumlah *Node*	50
2	Waktu Simulasi	100 detik
3	Area	510m x 510m
4	Kecepatan Maksimal	-5m/s -10m/s -15m/s
5	Sumber *Traffic*	CBR
6	Waktu Jeda (Dalam Detik)	10
7	Ukuran Paket	512 bytes
8	*Rate* Paket	0.25 paket per detik
9	Jumlah maksimal koneksi	1
10	Model mobilitas yang digunakan	*Random way point*

3.2.2. *Traffic-Connection Pattern*

Traffic-Connection dibuat dengan menjalankan program *cbrgren.tcl* yang telah ada pada NS-2 yang digunakan untuk memberikan koneksi dari *node node* yang sudah dibuat dalam skenario diatas selama melakukan simulasi pada NS-2.

Table 3.2 Penjelasan *Traffic-connection pattern*

No.	Parameter	Spesifikasi
1	-type cbr|tcp	CBR
2	-nn *nodes*	10
3	-s *seed*	1.0
4	-mc *connection*	8
5	-rate *rate*	0.25

3.3. Perancangan Simulasi NS-2

Pada perancangan kode NS-2 dengan konfigurasi MANET. Dilakukan dengan menggunakan skenario mobilitas dan digabungkan dengan *script* TCL yang berisikan konfigurasi mengenai lingkungan simulasi. Konfigurasi lingkungan simulasi MANET pada NS-2 dapat dilihat di tabel 3.3

Table 3.3 Penjelasan simulasi NS-2

No	Parameter	Spesifikasi
1	*Network Simulator*	NS-2, 2.35
2	Routing Protocol	AOMDV
3	Waktu Simulasi	100 detik
4	Area Simulasi	510m x 510m
5	Banyak *node*	50
6	Radius Transmisi	100m
7	Agen Pengirim	*Constant Bit Rate* (CBR)
8	*Source/Destination*	Statis
9	*Packet Rate*	512 bytes
10	Ukuran Paket	32
11	Protokol MAC	IEEE 802.11
12	Propagasi Sinyal	*Nakagami*
13	Tipe Kanal	*Wireless Channel*

3.4. Perancangan Metriks Analisis

Berikut ini merupakan beberapa parameter yang dianalisis dalam Tugas Akhir ini:

3.4.1. *Packet Delivery Ratio* (PDR)

Packet delivery ratio dihitung dengan membandingkan jumlah paket komunikasi yang dikirimkan dengan paket komunikasi yang diterima. *Packet delivery ratio* dihitung menggunakan persamaan 4, dimana *received* adalah jumlah paket komunikasi yang diterima dan *sent* adalah jumlah paket komunikasi yang dikirimkan.

$$Packet\ Delivery\ Ratio = \frac{Received}{Sent} x\ 100 \quad (4)$$

3.4.2. *End-to-End Delay*(E2E)

End-to-end delay mengindikasikan interupsi transmisi paket dari *node* asal ke tujuan. Total interupsi didapatkan dari akumulasi *delay-delay* kecil yang ada dalam jaringan. Rata-rata *end to end delay* pada paket yang diterima bisa dihitung berdasarkan selisih waktu antara transmisi dan respon paket pada *Constant Bit Rate* (CBR) dan membaginya dengan jumlah total transimi CBR menggunakan persamaan 5

$$End\ to\ End\ Delay = \sum_{i<sent}^{i=0} \frac{t^{received(i)} - t^{sent(i)}}{sent} \quad (5)$$

3.4.3. *Routing Overhead* (RO)

Routing overhead adalah jumlah paket *routing* yang ada didalam sebuah jaringan dibagi dengan jumlah keseluruhan paket data yang diterima, perhitungan menggunakan persamaan 6.

$$Routing\ Overhead = \sum_{i<sent}^{i=0} \frac{packet\ routing}{received} \quad (6)$$

BAB IV
IMPLEMENTASI

Bab ini akan membahas tentang implementasi Tugas Akhir berdasarkan rancangan perangkat lunak yang meliputi lingkungan pembangunan perangkat lunak, implementasi skenario, implementasi simulasi NS-2, dan implementasi matrik analisis.

4.1. Lingkungan Implementasi Protokol

Sub bab ini menjelaskan tentang lingkungan implementasi Protokol yang dilakukan pada lingkungan:

Tabel 4.1 Spesifikasi Lingkungan Implementasi

Perangkat Keras	- Laptop Asus X450CA o Prosesor Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz o RAM 6 GB o HDD 1 TB
Perangkat Lunak	- Laptop Asus X450CA o Sistem Operasi Ubuntu 16.04 o Network Simulator 2, 2.35

4.2. Implementasi Skenario

Implementasi skenario mobilitas MANET dimulai dengan melakukan pembuatan skenario oleh *Mobility Generation* untuk menempatkan dan menentukan *node node* yang akan di simulasikan. Setelah membuat skenario maka kita membuat jalur jalur yang menghubungkan antar node dengan *traffic connection pattern*. implementasi skenarionya adalah sebagai berikut.

4.2.1. Skenario *File Node-Movement*(*Mobility Generation*)

Dalam iimplementasi yang dilakukan pada pembuatan skenario dengan *mobility generation* menggunakan *tools generate default* yang sudah disediakan oleh NS-2 ketika kita melakukan installasi sebelumnya yaitu 'setdest'. Skenario yang sudah di-*generate* ini akan digunakan untuk setiap simulasi yang dilakukan beradasarkan kecepatan maksimal yang berbeda beda. Algoritma yang digunakan untuk membuat skenario ini adalah *Random Way Point* sehinnga penempatan *node node* yang ada akan bersifat acak. . *Format command line* yang digunakan untuk menghasilkan gerakan acak pada node adalah sebagai berikut:

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

Ketentuan-ketentuan yang diujicobakan pada skenario ini adalah versi ‘setdest’ simulator yaitu 1, jumlah node dalam skenario yaitu 50, waktu jeda (*pause time*) yaitu 10 detik,kecepatan maksimalnya yaitu skenario A sebesar 5m/s, skenario B sebesar 10m/s dan skenario C sebesar 15m/s, waktu simulasi yaitu 100 detik. Kemudian file mobilitas yang dihasilkan disimpan dalam direktori “ ~ns/indep-utils/cmu-scen-gen/setdest/”. Pada kode sumber 4.1 dapat dilihat implementasi *command line* pada ‘setdest’ dengan berbagai kecepatan maksimal yang berbeda sesuai dan node sebanyak 50. Dan untuk setiap kecepatan maksimal tersebut dibuat 10 buah file untuk satu protokol routing dan satu model propagasi.

```
1. ./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x 510 -
   y 510 > scenario1-5.txt
2. ./setdest -v 1 -n 50 -p 10 -M 10 -t 100 -x 510 -
   y 510 > scenario1-10.txt
```

```
3.  ./setdest -v 1 -n 50 -p 10 -M 15 -t 100 -x 510 -
    y 510 > scenario1-15.txt
```

Kode Sumber 4.1 Implementasi 'setdest'

4.2.2. File *traffic-connection pattern*

Dalam implementasi *random traffic connection* yang menggunakan tipe CBR ini di-setting dengan menggunakan skrip *traffic scenario generator*. skrip *traffic scenario generator* akan menghasilkan file bernama 'cbrgen.tcl' yang mana akan digunakan unruk membuat jaringan atau hubungan antar *node node* yang sudah dibuat pada skenario sebelumnya. ketika kira akan menggunakan file 'cbrgen.tcl' ini kita haris menentukan tipe koneksi yang digunakan(apakah CBR atau TCP), banyaknya *node* yang ada, koneksi maksimal yang diinginkan, dan nilai *random seed*. *Format command line* yang digunakan untuk menghasilkan gerakan acak pada *node* adalah sebagai berikut:

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

Dibawah ini merupakan bentuk implementasi cbrgen.tcl untuk membuat file koneksi CBR diantara 10 node memiliki maksimal 8 koneksi dengan nilai seed 1.0 dan jumlah paket per detik sebanyak 0.25 yang disimpan dalam cbrtest.txt yang nantinya akan digunakan pada saat simulasi NS-2.

```
   ns cbrgen.tcl -type cbr -nn 10 -seed 1.0 -mc 8 -rate 0.25
> cbrtest.txt
```

4.3. Implementasi Simulasi pada NS-2

Implementasi simulasi NS-2 dilakukan dengan cara pendeskripsian lingkungan simulasi pada sebuah file dengan

ekstensi .tcl, dan .txt file-file ini berisi konfigurasi setiap *node* dan proses yang dilakukan selama simulasi berjalan.

Pada kode sumber 4.2 menunjukan skrip konfigurasi awal parameter paremeter yang diberikan untuk menjalakan simulasi MANET pada NS-2. Isi dari parameternya adalah sebegai berikut.

Table 4.3 Penjelasan simulasi NS-2

No.	Parameter	Spesifikasi
1	Channel/	Wireless Channel
2	Propagation/	Nakagami
3	Phy/ WirelessPhy	WirelessPhy
4	Mac/ 802.11	802.11
5	Queue/Droptail/PriQueue	PriQueue
6	Antenna/OmniAntena	OmniAntena
7	Nilai X	510
8	Nilai Y	510
9	Nilai seed	0.0
10	Routing Protocol	AOMDV
11	File traffic connection	“cbrtest.txt”
12	File node movement	“scenario.txt”

```
1.  set val(chan)           Channel/WirelessChannel;
2.  set val(prop)           Propagation/Nakagami;
3.  set val(netif)          Phy/WirelessPhy;
4.  set val(mac)            Mac/802_11;
5.  set val(ifq)            Queue/Droptail/PriQueue;
6.  set val(ll)             LL;
7.  set val(ant)            Antenna/OmniAntenna;
8.  set val(ifqlen)         50;
9.  set val(nn)             50;
10. set val(rp)             AOMDV;
11. set opt(x)              510;
12. set opt(y)              510;
13. set val(stop)           100;
14. set val(seed)           0.0;
```

```
15. set val(cp)             "cbrtest.txt";
16. set val(sc)             "scenario.txt";
```

Kode Sumber 4.2 Konfigurasi parameter pada NS-2

Dibawah ini merupakan pengaturan dari *transmission range* yang digunakan pada simulasi. Nilai yang diubah ialah *RXThresh_(receiver Sensitivity Threshold)*. Nilai 1.42681e-08 pada variabel tersebut memiliki artian bahwa *range* atau jangkauan yang dapat dicapai ialah sejauh 100 meter.

```
Phy/WirelessPhy set RXThresh_ 1.42681e-08;
```

```
1.  set ns_       [new Simulator]
2.  set tracefd       [open Naka-M5_1.tr w]
3.  #set windowVsTime2 [open win.tr w]
4.  set namtrace      [open Naka-M5_1.nam w]
5.  $ns_ trace-all $tracefd
6.  #$ns use-newtrace
7.  $ns_ namtrace-all-
    wireless $namtrace $opt(x) $opt(y)
8.
9.  # set up topography object
10. set topo       [new Topography]
11.
12. $topo load_flatgrid $opt(x) $opt(y)
13.
14. set god_ [create-god $val(nn)]
15. #  Create nn mobilenodes [$val(nn)] and attach them
     to the channel.
16. # configure the nodes
17. $ns_ node-config -
    adhocRouting $val(adhocRouting) \
18.              -llType $val(ll) \
19.              -macType $val(mac) \
20.              -ifqType $val(ifq) \
21.              -ifqLen $val(ifqlen) \
22.              -antType $val(ant) \
23.              -propType $val(prop) \
24.              -phyType $val(netif) \
```

```
25.             -channelType $val(chan) \
26.             -topoInstance $topo \
27.             -agentTrace ON \
28.             -routerTrace ON \
29.             -macTrace OFF \
30.             -movementTrace ON
31.     for {set i 0} {$i < $val(nn) } { incr i } {
32.         set node_($i) [$ns_ node]
33.         $node_($i) random-motion 0;
34.     }
```

Kode Sumber 4.3 Pengaturan variabel global pada NS-2

Skrip yang ditunjukan pada kode sumber 4.3 merupakan skrip untuk pengaturan variabel global yang diawali dengan set ns merupakan kode untuk pembuatan simulator baru. Set *tracefd* dan *set namtrace* merupakan pengaturan untuk menentukan nama dari *trace file* dan *file network* yang akan dihasilkan dan disimpan setelah simulasi selesai dilaksanakan. *Set topo* merupakan pengaturan untuk objek topografi berdasarkan pada luas koordinat yang telah dikonfigurasi sebelumnya. *Set god* dan *node config – channelType* merupakan konfigurasi yang dilakukan pada *node-node* yang akan dibuat. Terakhir dilakukan perulangan untuk membuat pergerakan dari *node node*. *Node-node* yang dibuat tidak dapat melakukan pergerakan secara acak karena pergerakan *node* merupakan *trace file* yang dihasilkan oleh *mobility generator*.

Skrip pada kode sumber 4.4 merupakan bagian akhir dari keseluruhan skrip yang digunakan untuk menginisiasi penempatan awal *node-node* yang dibuat pada skenario *node-movement* (*mobility generation*), pergerakan *node* tersebut selama waktu simulasi dilakukan dan melakukan konfigurasi pengiriman paket data yang dilakukan nantinya dihasilkan pada *file output.tr* pada potongan skrip tersebut, akan dipanggil file skenario *node-movement* (*mobility generation*) dan *traffic-connection pattern* kemudian pengiriman paket data dimulai pada detik ke-0 dan

diberhentikan pada detik ke 100 seperti yang telah di konfigurasi sebelumnya.

```
puts "Loading Conneciton Pattern ...."
source $val(cp)

#Define traffice mode
puts "Loading scenarion file...."
source $val(sc)

#define node initial position in nam

    for {set i 0} {$i < $val(nn) } { incr i } {
        $ns_ initial_node_pos $node_($i) 20
    }


#tell nodes when the simulation ends
    for {set i 0} {$i < $val(nn) } { incr i } {
        $ns_ at $val(stop).0 "$node_($i) reset";
    }

$ns_ at $val(stop).0002 "puts \"NS EXITING....\";
$ns_ halt"

puts $tracefd "M 0.0 nn $val(nn) x $opt(x) y $opt(y
) rp $val(rp)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $
val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)

puts "Starting Simulation...."
$ns_ run
```

Kode Sumber 4.4 Menginisiasi penempatan awal node

4.4. *Implementasi Metriks Analisis*

Setelah selesai melakukan simulasi dengan NS-2 maka akan tercipta sebuah *trace file* yang berisi nilai nilai yang dapat kita

gunakan untuk melakukan analisis dari simulasi yang sudah dilaksanakan. Dalam Tugas Akhir ini penulis akan melakukan analisis dari nilai rata rata *Packet Delivery Ratio*(PDR), rata rata *Routing OverheadI*(RO), dan rata rata dari *End to End Delay*(E2E).

4.4.1. Implementasi *Packet Delivery Ratio*

Packet Delivery Ratio didapatkan dengan membandingkan pengiriman dan penerimaan data yang dikirimkan oleh agen. Pada Tugas Akhir ini, penulis menggunakan agen dengan pengiriman data CBR. Kemudian, pada persamaan 2 telah dijelaskan rumus untuk menghitung *packet delivery ratio*. Skrip awk untuk menghitung *packet delivery ratio* berdasarkan kedua informasi tersebut dapat dilihat pada lampiran. Cara menjalankan skrip awk dapat dilihat pada perintah awk di bawah ini.

```
awk –f PDR.awk Naka-M5_1.tr
```

4.4.2. Implementasi *End-to-End Delay*

Perhitungan *end to end delay* didasarkan pada Persamaan 3 dan sudah dijelaskan pada bagian 3.4.2 Skrip awk untuk menghitung *end to end delay* berdasarkan kedua informasi tersebut dapat dilihat pada lampiran. Contoh perintah untuk memanggil skrip tcl untuk menganilisis *trace file* dan *outputnya* dapat dilihat pada perintah awk di bawah ini.

```
awk –f E2E.awk Naka-M5_1.tr
```

4.4.3. Implementasi *Routing Overhead*

Perhitungan RO didasarkan pada Persamaan 4 dan sudah dijelaskan pada bagian 3.4.3 Skrip awk untuk menghitung RO berdasarkan kedua informasi tersebut dapat dilihat pada lampiran.

Contoh perintah untuk memanggil skrip tcl untuk menganilisis *trace file* dan *outputnya* dapat dilihat pada perintah awk di bawah ini.

```
awk –f ro.awk Naka-M5_1.tr
```

[Halaman ini sengaja dikosongkan]

BAB V
PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada dari skenario NS-2 yang telah dibuat. Pengujian fungsionalitas akan dibagi ke dalam beberapa skenario pengujian.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Sistem

Perangkat Keras	- Laptop Asus X450CA o Prosesor Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz o RAM 6 GB o HDD 1 TB
Perangkat Lunak	- Laptop Asus X450CA o Sistem Operasi Windows 10 Education, Linux Ubunru 16.04 LTS 64 bit (NS-2, AOMDV, *Mobility Generation, Traffic Connection Genneration, Nakagami.*)

5.2. Kriteria Pengujian

Pengujian pada skenario yang dihasilkan oleh *mobility generator* default dari NS-2 menggunakan beberapa kriteria. Pada tabel 5.2 menunjukan kriteria-kriteria yang ditentukan didalam skenario.

.

Tabel 5.2 Penjelasan skenario pengujian

Kriteria	Spesifikasi
Skenario	MANET

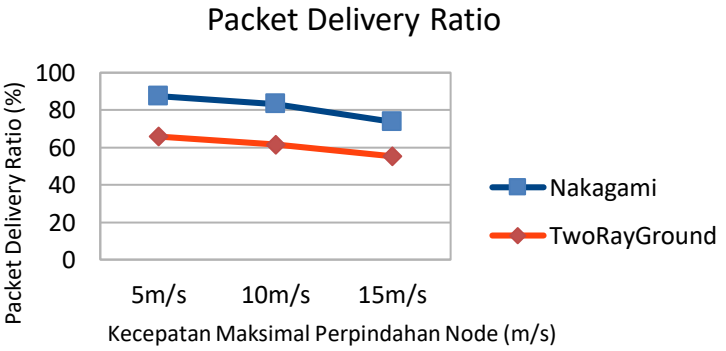
Kecepatan Maksimal Perpindahan node (m/s)	5, 10, 15
Jumlah node (node)	20, 50, 100
Radius (m)	50, 75, 100
Jumlah percobaan	10
Posisi awal node	Acak
Pergerakan	Acak
Protokol Routing	AOMDV
Pengiriman Paket Data	0 – 100 Detik

5.3. Analisis *Packet Delivery Ratio (PDR)*

Dari hasil analisis yang dilakukan setelah melakukan uji coba berulang kali pada akhir penulis mendapatkan nilai rataan pada PDR dengan menggunakan model propagasi *Nakagami* dan *TwoRayGround* seperti pada tabel 5.3.1, 5.3.2 dan 5.3.3

Tabel 5.3.1 Nilai hasil PDR (kecepatan maksimal)

Kecepatan Maksimal Perpindahan Node (m/s)	PDR (%)	
	Nakagami	TwoRayGround
5	87,463	65,833
10	83,268	61,57
15	73,739	55,287



Gambar 5.3.1 Grafik nilai PDR (kecepatan maksimal)

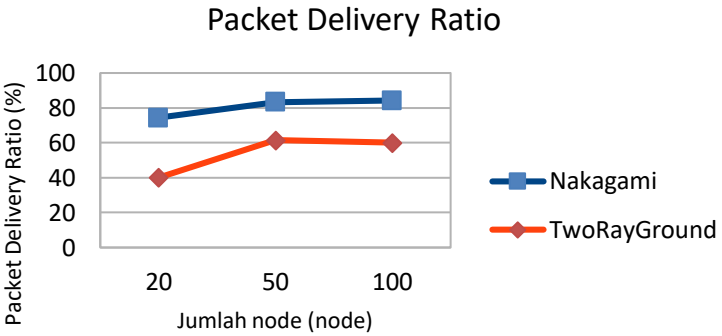
Pada tabel 5.3.1 dan gambar 5.3.1 menunjukan performa PDR yang dihasilkan oleh model propagasi *Nakagami* dan *TwoRayGround* pada jaringan MANET dengan menggunakan skenario *node-Movement* (*mobility generation*) yang bersifat *Random Way Point*.

Dapat dilihat bahwa nilai rata rata dari *Packet Delivery Ratio* model *Nakagami* menunjukan nilai yang relatif menurun. Dapat dilihat bahwa nilai rata rata pada kecepatan maksimal 5m/s adalah 87.463% sementara pada kecepatan maksimal 10m/s memiliki nilai 83.268% dan pada kecepatan maksimal 15m/s bernilai 73.739%. Nilai rata-rata dari *Packet Delivery Ratio* pada model *TwoRayGround* menunjukkan nilai yang relatif menurun pula. Nilai rata-rata pada kecepatan maksimal 5m/s adalah 65,833% sementara pada kecepatan maksimal 10m/s memiliki nilai 61,57% dan pada kecepatan maksimal 15m/s bernilai 55,287%. Hal ini menunjukan bahwa semakin tinggi kecepatan maksimal maka semakin rendah pula paket data yang dikirimkan.

Terlihat bahwa nilai PDR yang dihasilkan oleh model *Nakagami* lebih baik daripada nilai PDR yang dihasilkan oleh model *TwoRayGround*. Pergerakan *node* yang lebih dinamis inilah yang memungkinkan terjadinya banyak rute putus saat pengiriman paket data ataupun kegagalan pembentukan tabel routing yang dibuat oleh AOMDV sehingga menyebabkan paket data yang dikirimkan tidak sampai ke tujuan.

Tabel 5.3.2 Nilai hasil PDR (jumlah node)

Jumlah node (node)	PDR (%)	
	Nakagami	TwoRayGround
20	74,435	40,066
50	83,268	61,57
100	84,244	60,075

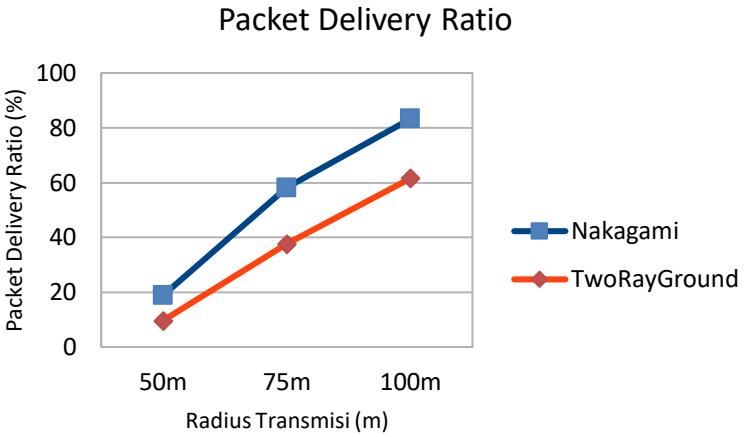


Gambar 5.3.2 Grafik nilai PDR (jumlah node)

Pada tabel 5.3.2 dan gambar 5.3.2 menunjukan performa PDR yang dihasilkan oleh model propagasi *Nakagami* dan *TwoRayGround* pada jaringan MANET dengan menggunakan skenario *node-Movement* (*mobility generation*) yang bersifat *Random Way Point* berdasarkan jumlah node. Dapat dilihat bahwa model *Nakagami* mengalami peningkatan nilai PDR seiring dengan bertambahnya jumlah *node*. Pada jumlah *node* 20 nilai PDR yang dihasilkan adalah 74,4%, pada jumlah *node* 50 nilai PDR yang dihasilkan adalah 83,2%, dan pada jumlah node 100 nilai PDR yang dihasilkan adalah 84,2%. Sedangkan pada model *TwoRayGround* grafik PDR yang dihasilkan bersifat fluktuatif dimana pada jumlah *node* 20 nilai PDR yang dihasilkan adalah 40%, pada jumlah *node* 50 nilai PDR yang dihasilkan adalah 61,5%, dan pada jumlah *node* 100 nilai PDR yang dihasilkan adalah 60%.

Tabel 5.3.3 Nilai hasil PDR (radius)

Radius (m)	PDR (%)	
	Nakagami	TwoRayGround
50	18,826	9,662
75	58,187	37,631
100	83,268	61,57



Gambar 5.3.3 Grafik nilai PDR (radius)

Pada tabel 5.3.3 dan gambar 5.3.3 menunjukan performa PDR yang dihasilkan oleh model propagasi *Nakagami* dan *TwoRayGround* pada jaringan MANET dengan menggunakan skenario *node-Movement* (*mobility generation*) yang bersifat *Random Way Point* berdasarkan radius transmisi. Dapat dilihat bahwa model *Nakagami* mengalami peningkatan nilai PDR seiring dengan bertambahnya radius transmisi. Pada radius 50 meter nilai PDR yang dihasilkan adalah 18,8%, pada radius 75 meter nilai PDR yang dihasilkan adalah 58,1%, dan pada radius 100 meter nilai PDR yang dihasilkan adalah 83,2%. Sedangkan pada model *TwoRayGround* juga mengalami peningkatan nilai PDR seiring dengan bertambahnya radius transmisi dimana pada radius 50 meter nilai PDR yang dihasilkan adalah 9,6%, pada radius 75 meter nilai PDR yang dihasilkan adalah 37,6%, dan pada radius transmisi 100 meter nilai PDR yang dihasilkan adalah 61,5%.

Dari ketiga percobaan dengan parameter yang berbeda diatas nilai PDR yang dihasilkan oleh model *Nakagami* lebih baik dibandingkan dengan model *TwoRayGround*, hal ini disebabkan karena model *Nakagami* memiliki kemampuan untuk mengidentifikasi/memperhatikan lingkungan sekitar area simulasi

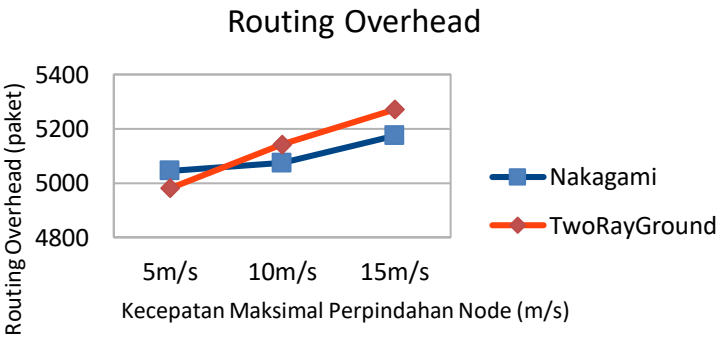
dan pengaruh ketinggian *antenna* pada *node* yang ada. Hal lain yang dapat mempengaruhi penurunan ataupun peningkatan nilai PDR yang dihasilkan adalah penempatan dan pergerakan secara acak yang di implementasikan pada skenario yang dihasilkan oleh file *node-movement* (*mobility generation*).

5.4. Analisis *Routing Overhead(RO)*

Dari hasil analisis yang dilakukan setelah melakukan uji coba berulang kali pada akhir penulis mendapatkan nilai rataan pada RO dengan menggunakan model propagasi *Nakagami* dan *TwoRayGround* menjadi seperti pada tabel 5.4.1, 5.4.2, dan 5.4.3

Tabel 5.4.1 Nilai hasil RO (kecepatan maksimal)

Kecepatan Maksimal Perpindahan Node (m/s)	RO (paket)	
	Nakagami	TwoRayGround
5	5044,4	4980,4
10	5073,8	5142,9
15	5173,9	5270,7



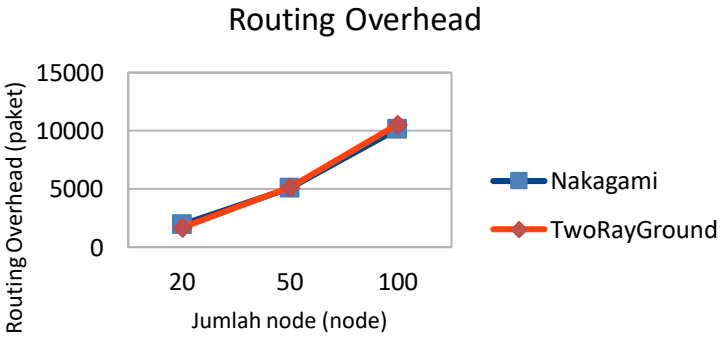
Gambar 5.4.1 Grafik nilai RO (kecepatan maksimal)

Dari hasil skenario yang sudah dibuat dengan parameter kecepatan maksimal perpindahan node bisa dilihat bahwa nilai rata rata dari *Routing Overhead* yang dihasilkan oleh model *Nakagami*

relatif meningkat. Pada kecepatan maksimal 5m/s nilai yang dihasilkan 5044,4 paket sementara pada kecepatan maksimal 10m/s nilai yang dihasilkan 5073,8 paket dan pada kecepatan maksimal 15m/s adalah 5173,9 paket. Pada model *TwoRayGround* nilai *Routing Overhead* juga relatif meningkat, saat kecepatan maksimal 5m/s nilai yang dihasilkan 4980,4 paket sementara pada kecepatan maksimal 10m/s nilai yang dihasilkan meningkat menjadi 5142,9 paket lalu pada kecepatan maksimal 15m/s nilai nya meningkat lagi menjadi 5270,7 paket. Dapat dilihat pada saat kecepatan maksimal perpindahan *node* 10m/s dan 15m/s terjadi peningkatan nilai *Routing Overhead* pada model *Nakagami* yaitu sebanyak 29,4 paket dan 100,1 paket sedangkan pada model *TwoRayGround* mengalami peningkatan nilai *Routing Overhead* sebanyak 162,5 paket dan 127,8 paket.

Tabel 5.4.2 Nilai hasil RO (jumlah node)

Jumlah node (node)	RO (paket)	
	Nakagami	TwoRayGround
20	1942,3	1675,1
50	5073,8	5142,9
100	10159,2	10544



Gambar 5.4.2 Grafik nilai RO (jumlah node)

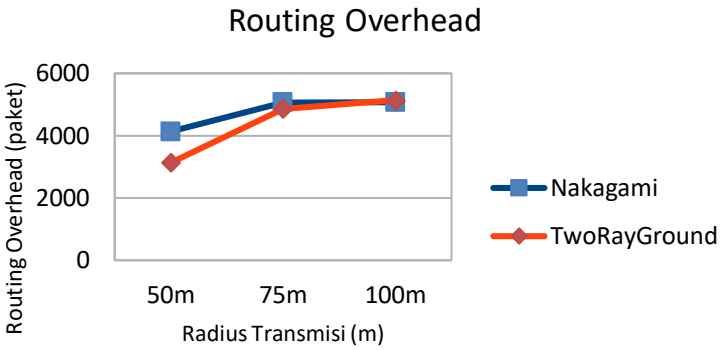
Pada tabel 5.4.2 dan gambar 5.4.2 menunjukkan performa *Routing Overhead* pada jaringan MANET dengan menggunakan skenario *node-Movement* (*mobility generation*) yang bersifat *Random Way Point* berdasarkan jumlah *node*. Dapat dilihat bahwa model *Nakagami* mengalami peningkatan nilai *Routing Overhead* seiring dengan bertambahnya jumlah *node*. Pada jumlah node 20 nilai RO yang dihasilkan adalah 1942,3 paket, pada jumlah node 50 nilai RO yang dihasilkan adalah 5073,8 paket, dan pada jumlah node 100 nilai RO yang dihasilkan adalah 10159,2 paket. Pada model *TwoRayGround* grafik RO yang dihasilkan juga relatif meningkat dimana pada jumlah *node* 20 nilai RO yang dihasilkan adalah 1675,1 paket, pada jumlah *node* 50 nilai RO yang dihasilkan adalah 5142,9 paket, dan pada jumlah node 100 nilai RO yang dihasilkan adalah 10544 paket. Dapat dilihat pada saat perpindahan jumlah *node* 50 dan 100 terjadi peningkatan nilai *Routing Overhead* pada model *Nakagami* yaitu sebanyak 3131,5 paket dan 5085,4 paket sedangkan pada model *TwoRayGround* mengalami peningkatan nilai *Routing Overhead* sebanyak 3467,8 paket dan 5401,1 paket. Semakin banyak jumlah node maka semakin banyak pula jumlah paket yang akan dikirimkan.

Tabel 5.4.3 Nilai hasil RO (radius)

Radius (m)	RO (paket)	
	Nakagami	TwoRayGround
50	4134,3	3133
75	5066,3	4857,8
100	5073,8	5142,9

Pada tabel 5.4.3 dan gambar 5.4.3 menunjukan performa RO yang dihasilkan oleh model propagasi *Nakagami* dan *TwoRayGround* pada jaringan MANET dengan menggunakan skenario *node-Movement* (*mobility generation*) yang bersifat *Random Way Point* berdasarkan radius transmisi. Dapat dilihat bahwa model *Nakagami* mengalami peningkatan nilai RO seiring dengan bertambahnya radius transmisi. Pada radius 50 meter nilai

RO yang dihasilkan adalah 4134,3 paket, pada radius 75 meter nilai RO yang dihasilkan adalah 5066,3 paket, dan pada radius 100 meter nilai RO yang dihasilkan adalah 5073,8 paket. Sedangkan pada model *TwoRayGround* juga mengalami peningkatan nilai RO seiring dengan bertambahnya radius transmisi dimana pada radius 50 meter nilai RO yang dihasilkan adalah 3133 paket, pada radius 75 meter nilai RO yang dihasilkan adalah 4857,8 paket, dan pada radius transmisi 100 meter nilai RO yang dihasilkan adalah 5142,9 paket.



Gambar 5.4.3 Grafik nilai RO (radius)

Dapat dilihat saat perpindahan radius 50 meter dan 75 meter terjadi peningkatan nilai *Routing Overhead* pada model *Nakagami* yaitu sebanyak 932 paket dan 7,5 paket sedangkan pada model *TwoRayGround* mengalami peningkatan nilai *Routing Overhead* sebanyak 1724,8 paket dan 285,1 paket.

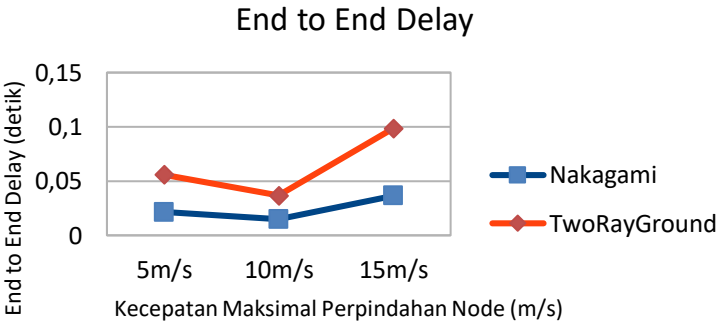
Faktor yang menyebabkan meningkatnya nilai rata-rata *Routing Overhead* adalah terjadinya rute putus saat pengiriman paket, sehingga paket RERR bertambah untuk melakukan maintenance link komunikasi. Selain itu faktor lainnya yang menyebabkan peningkatan pada nilai rata-rata *Routing Overhead* adalah lokasi dan pergerakan pada *node* yang bersifat dinamis

5.5. Analisis *End-to-End Delay (E2E)*

Dari hasil analisis yang dilakukan setelah melakukan uji coba berulang kali pada akhir penulis mendapatkan nilai rataan pada E2E dengan menggunakan model propagasi *Nakagami* menjadi tabel 5.5.1, 5.5.2 dan 5.5.3

Tabel 5.5.1 Hasil nilai E2E (kecepatan maksimal)

Kecepatan Maksimal Perpindahan Node (m/s)	End to End delay (detik)	
	Nakagami	TwoRayGround
5	0,0214	0,0559
10	0,0149	0,0368
15	0,0364	0,0986



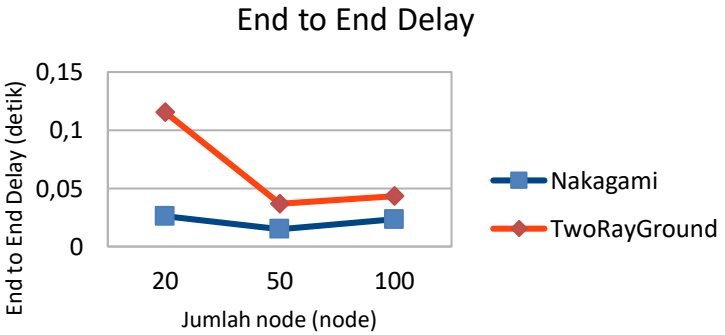
Gambar 5.5.1 Grafik nilai E2E (kecepatan maksimal)

Pada tabel 5.5.1 dan gambar 5.5.1, pengujian model propagasi *Nakagami* dengan parameter kecepatan maksimal menunjukan nilai rata rata dari *End to End Delay* yang bersifat fluktuatif. Pada kecepatan maksimal 5m/s rata rata nilainya adalah 0,0214 detik sementara pada kecepatan maksimal 10m/s adalah 0,0149 detik dan pada kecepatan maksimal 15m/s adalah 0,0364

detik. Pada model *TwoRayGround* nilai rata-rata dari *End to End Delay* juga bersifat fluktuatif dimana pada kecepatan 5m/s nilai yang dihasilkan 0,0559 detik, lalu sedikit turun pada saat kecepatan 10m/s menjadi 0,0368 detik dan meningkat tajam saat kecepatan 15m/s menjadi 0,0986 detik.

Tabel 5.5.2 Hasil nilai E2E (jumlah node)

Jumlah node (node)	End to End delay (detik)	
	Nakagami	TwoRayGround
20	0,0261	0,1155
50	0,0149	0,0368
100	0,0234	0,0433



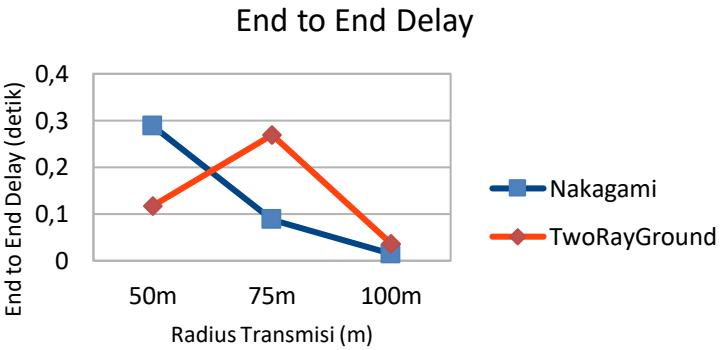
Gambar 5.5.2 Grafik nilai E2E (jumlah node)

Pada tabel 5.5.2 dan gambar 5.5.2, pengujian model propagasi *Nakagami* dengan parameter jumlah *node* menunjukan nilai rata rata dari *End to End Delay* yang bersifat fluktuatif. Pada jumlah *node* 20 rata rata nilainya adalah 0,0261 detik sementara pada jumlah *node* 50 menurun menjadi 0,0149 detik dan pada jumlah *node* 100 meningkat menjadi 0,0234 detik. Pada model *TwoRayGround* nilai rata-rata dari *End to End Delay* juga bersifat fluktuatif dimana pada jumlah *node* 20 nilai yang dihasilkan 0,1155 detik, lalu menurun dengan tajam pada saat jumlah *node* 50

menjadi 0,0368 detik dan meningkat sedikit saat jumlah *node* 100 menjadi 0,0433 detik.

Tabel 5.5.3 Hasil nilai E2E (radius)

Radius (m)	End to End delay (detik)	
	Nakagami	TwoRayGround
50	0,28801	0,1175
75	0,0878	0,2692
100	0,0149	0,0368



Gambar 5.5.3 Grafik nilai E2E (radius)

Pada tabel 5.5.3 dan gambar 5.5.3, pengujian model propagasi *Nakagami* dengan parameter radius transmisi menunjukan penurunan nilai rata rata dari *End to End Delay*. Pada radius transmisi 50 meter rata rata nilainya adalah 0,28801 detik sementara pada radius 75 meter menurun menjadi 0,0878 detik dan pada radius 100 meter menurun hingga menjadi 0,0149 detik. Pada model *TwoRayGround* nilai rata-rata dari *End to End Delay* bersifat fluktuatif dimana pada radius 50 meter nilai yang dihasilkan 0,1175 detik, lalu meningkat pada saat radius 75 meter menjadi 0,2692 detik dan menurun saat radius 100 meter menjadi 0,0368 detik.

Hal ini dapat terjadi akibat mobilitas pengiriman paket antar *node* yang sangat dinamis pada *skenario node-movement*(*mobility generation*) yang dihasilkan oleh model propagasi *Nakagami* dan *TwoRayGround*. Mobilitas yang sangat dinamis ini memungkinkan terjadinya rute putus saat pengiriman paket data sehinga pengiriman paket dimasukkan ke dalam antrian dan menunggu rute baru terbentuk sebelum pengiriman paket data dilanjutkan kembali.

[Halaman ini sengaja dikosongkan]

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

Kesimpulan yang dapat diambil dari tugas akhir ini adalah sebagai berikut:

1. Skenario Manet yang dihasilkan oleh *node-movement*(*mobility generation*) dan dijalankan menggunakan model propagasi *Nakagami* dengan parameter kecepatan maksimal perpindahan *node* memiliki performa sebagai berikut:
   - Performa *Packet Delivery Ratio* yang dihasilkan semakin menurun secara stabil mulai dari 87,463% hingga 73,739% dari kecepatan maksimal perpindahan *node* sebesar 5m/s hingga 15m/s.
   - Performa *Routing Overhead* yang dihasilkan memiliki nilai semakin meningkat mulai dari 5.044,4 paket pada kecepatan maksimal 5m/s menjadi 5.073,8 paket pada kecepatan maksimal 10m/s dan berubah menjadi 5.173,9 paket pada kecepatan maksimal 15m/s.
   - Performa *End to End Delay* yang dihasilkan memiliki nilai *delay* fluktuatif, yaitu sedikit menurun pada saat kecepatan maksimal 10m/s namun meningkat secara tajam pada saat kecepatan maksimal 15m/s.

2. Skenario Manet yang dihasilkan oleh *node-movement*(*mobility generation*) dan dijalankan menggunakan model propagasi *Nakagami* dengan parameter jumlah *node* memiliki performa sebagai berikut:

- Performa *Packet Delivery Ratio* yang dihasilkan semakin meningkat secara stabil mulai dari 74,435% hingga 83,268% dari jumlah *node* sebesar 20 *node* hingga 100 *node*.
- Performa *Routing Overhead* yang dihasilkan memiliki nilai semakin meningkat mulai dari 1942,3 paket pada jumlah *node* 20 menjadi 5.073,8 paket pada jumlah *node* 50 dan berubah menjadi 10.159,2 paket pada jumlah *node* 100.
- Performa *End to End Delay* yang dihasilkan memiliki nilai *delay* fluktuatif, yaitu menurun secara tajam pada saat jumlah *node* 50 namun sedikit meningkat pada saat jumlah *node* 100.

3. Skenario Manet yang dihasilkan oleh *node-movement*(*mobility generation*) dan dijalankan menggunakan model propagasi *Nakagami* dengan parameter radius transmisi memiliki performa sebagai berikut:
   - Performa *Packet Delivery Ratio* yang dihasilkan semakin meningkat secara stabil mulai dari 18,826% hingga 83,268% dari radius transmisi sebesar 50 meter hingga 100 meter.
   - Performa *Routing Overhead* yang dihasilkan memiliki nilai semakin meningkat mulai dari 4.134,43 paket pada radius 50 meter menjadi 5.066,3 paket pada radius 75 meter dan berubah menjadi 5.073,8 paket pada radius 100 meter.
   - Performa *End to End Delay* yang dihasilkan memiliki nilai *delay* fluktuatif, yaitu meningkat pada saat radius 50 meter lalu menurun secara tajam pada saat radius 100 meter.

4. Hal – hal yang dapat mempengaruhi nilai PDR, E2E Delay dan RO yang dihasilkan dari model transmisi *Nakagami* adalah :
    - Posisi awal *node* yang dibuat secara acak.
    - Pergerakan *node* yang dibuat secara acak.

6.2. Saran

Dalam pengerjaan Tugas Akhir ini terdapat beberapa saran untuk perbaikan serta pengembangan sistem yang telah dikerjakan sebagai berikut:

1. Dapat dilakukan dengan penambahan atau pengurangan jumlah *node* dan penambahan jumlah percobaan untuk skenario *node-movement*(*mobility generation*).
2. Dapat dilakukan modifikasi pada parameter parameter lain yang berhubungan dengan simulasi AOMDV di NS2.
3. Dapat melakukan perbandingan dengan protokol routing yang lain sehingga mendapatkan hasil yang lebih baik.
4. Dapat dilakukan perbandingan dengan model propagasi yang lain agar mendapatkan nilai yang lebih baik.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] P. C. Sekhar, M. R. P. Kumar, B. P. Kumar, and C. Koteswararao, “Impact of Routing Overhead in A Real-Time MANET Environment,” vol. 4, p. 7, 2013.

[2] Rianda Anisia, Rendy Munadi, Ridha Muldina Negara. 2016. "Analisis performansi routing protocol OLSR dan AOMDV pada Vehicular Ad Hoc Network (VANET)," Maret 2016, vol:5, No.1.

[3] Pajala, Elina, Isotalo, Tero and dkk, "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.

[4] R. Kumar, P. Verma, and Y. Singh, “Mobile Ad Hoc Networks and It’s Routing Protocols,” vol. 7, no. 8, p. 10, 2013.

[5] A. Goldsmith, *Wireless communications*, 1. publ. Cambridge, U.K.: Cambridge University Press, 2004.

[6] Phillip Hurni.2007."Unsynchronized Energy-Efficient Medium Access Control and Routing in Wireless Sensor Networks," Philosophisch-Naturwissenschaftlichen Fakult¨at der Universit¨at Bern

[7] Admiral Budi Arviansyah Wilarsani, Radityo Anggoro, "Analisis Kinerja Model Propagasi *TwoRayGround* pada *Dynamic Source Routing* (DSR) di lingkungan MANET," Surabaya, Institut Teknologi Sepuluh Nopember, 2018.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

```
1.  #
2.  # nodes: 50, pause: 10.00, max speed: 5.00, max x:
    510.00, max y: 510.00
3.  #
4.  $node_(0) set X_ 135.750954250386
5.  $node_(0) set Y_ 249.694883657278
6.  $node_(0) set Z_ 0.000000000000
7.  $node_(1) set X_ 182.814657191152
8.  $node_(1) set Y_ 191.562026090664
9.  $node_(1) set Z_ 0.000000000000
10. $node_(2) set X_ 74.711349536656
11. $node_(2) set Y_ 204.677829249526
12. $node_(2) set Z_ 0.000000000000
13. $node_(3) set X_ 498.793769266713
14. $node_(3) set Y_ 290.289597045685
15. $node_(3) set Z_ 0.000000000000
16. $node_(4) set X_ 492.805215151395
17. $node_(4) set Y_ 28.871413975780
18. $node_(4) set Z_ 0.000000000000
19. $node_(5) set X_ 9.951753399792
20. $node_(5) set Y_ 160.764896136033
21. $node_(5) set Z_ 0.000000000000
22. $node_(6) set X_ 454.892554487486
23. $node_(6) set Y_ 415.304532885398
24. $node_(6) set Z_ 0.000000000000
25. $node_(7) set X_ 65.373899109608
26. $node_(7) set Y_ 318.704139664877
27. $node_(7) set Z_ 0.000000000000
28. $node_(8) set X_ 20.033257849728
29. $node_(8) set Y_ 445.693230692808
30. $node_(8) set Z_ 0.000000000000
31. $node_(9) set X_ 123.399070330664
32. $node_(9) set Y_ 387.464792694123
33. $node_(9) set Z_ 0.000000000000
34. $node_(10) set X_ 418.695669632568
35. $node_(10) set Y_ 96.836002439235
36. $node_(10) set Z_ 0.000000000000
37. $node_(11) set X_ 9.932581646363
38. $node_(11) set Y_ 194.284431464309
39. $node_(11) set Z_ 0.000000000000
```

```
40. $node_(12) set X_ 173.644322042358
41. $node_(12) set Y_ 284.975200769222
42. $node_(12) set Z_ 0.000000000000
43. $node_(13) set X_ 199.604589181895
44. $node_(13) set Y_ 460.436798185309
45. $node_(13) set Z_ 0.000000000000
46. $node_(14) set X_ 231.498877660317
47. $node_(14) set Y_ 249.078008578677
48. $node_(14) set Z_ 0.000000000000
49. $node_(15) set X_ 73.802255755493
50. $node_(15) set Y_ 289.599801382692
51. $node_(15) set Z_ 0.000000000000
52. $node_(16) set X_ 427.986335661532
53. $node_(16) set Y_ 285.494592074043
54. $node_(16) set Z_ 0.000000000000
55. $node_(17) set X_ 165.398612342056
56. $node_(17) set Y_ 26.498884938603
57. $node_(17) set Z_ 0.000000000000
58. $node_(18) set X_ 354.711998191219
59. $node_(18) set Y_ 401.430845513850
60. $node_(18) set Z_ 0.000000000000
61. $node_(19) set X_ 401.636635949933
62. $node_(19) set Y_ 166.016710717575
63. $node_(19) set Z_ 0.000000000000
64. $node_(20) set X_ 424.945723947303
65. $node_(20) set Y_ 493.107151832964
66. $node_(20) set Z_ 0.000000000000
67. $node_(21) set X_ 150.094300955444
68. $node_(21) set Y_ 60.295996691931
69. $node_(21) set Z_ 0.000000000000
70. $node_(22) set X_ 464.882394195427
71. $node_(22) set Y_ 131.697105719475
72. $node_(22) set Z_ 0.000000000000
73. $node_(23) set X_ 310.138259604284
74. $node_(23) set Y_ 330.972045711751
75. $node_(23) set Z_ 0.000000000000
76. $node_(24) set X_ 224.186319557660
77. $node_(24) set Y_ 55.935298105362
78. $node_(24) set Z_ 0.000000000000
79. $node_(25) set X_ 68.265767679382
80. $node_(25) set Y_ 133.963275371204
81. $node_(25) set Z_ 0.000000000000
82. $node_(26) set X_ 477.145613906041
```

```
$node_(26) set Y_ 110.473191202251
$node_(26) set Z_ 0.000000000000
$node_(27) set X_ 260.771651412478
$node_(27) set Y_ 482.963389383253
$node_(27) set Z_ 0.000000000000
$node_(28) set X_ 68.952128445274
$node_(28) set Y_ 295.548883214166
$node_(28) set Z_ 0.000000000000
$node_(29) set X_ 490.361537778564
$node_(29) set Y_ 367.359902390945
$node_(29) set Z_ 0.000000000000
$node_(30) set X_ 330.822193026779
$node_(30) set Y_ 423.774416317483
$node_(30) set Z_ 0.000000000000
$node_(31) set X_ 26.963546946276
$node_(31) set Y_ 210.659694654685
$node_(31) set Z_ 0.000000000000
        $node_(32) set X_ 345.748218424998
        $node_(32) set Y_ 167.614025264913
        $node_(32) set Z_ 0.000000000000
        $node_(33) set X_ 494.738326483772
        $node_(33) set Y_ 31.094867148840
        $node_(33) set Z_ 0.000000000000
        $node_(34) set X_ 84.646449854239
        $node_(34) set Y_ 54.907826776809
        $node_(34) set Z_ 0.000000000000
        $node_(35) set X_ 480.506987917575
        $node_(35) set Y_ 299.198264447795
        $node_(35) set Z_ 0.000000000000
        $node_(36) set X_ 149.680806173386
        $node_(36) set Y_ 20.026842962388
        $node_(36) set Z_ 0.000000000000
        $node_(37) set X_ 147.672159046822
        $node_(37) set Y_ 321.895680593863
        $node_(37) set Z_ 0.000000000000
        $node_(38) set X_ 217.011749394807
        $node_(38) set Y_ 487.116652934408
        $node_(38) set Z_ 0.000000000000
        $node_(39) set X_ 335.821853364107
        $node_(39) set Y_ 106.901493003944
        $node_(39) set Z_ 0.000000000000
        $node_(40) set X_ 386.223832304719
        $node_(40) set Y_ 479.547043718813
```

```
126.        $node_(40) set Z_ 0.000000000000
127.        $node_(41) set X_ 31.685296001505
128.        $node_(41) set Y_ 336.397299145031
129.        $node_(41) set Z_ 0.000000000000
130.        $node_(42) set X_ 85.695081535861
131.        $node_(42) set Y_ 166.594112087408
132.        $node_(42) set Z_ 0.000000000000
133.        $node_(43) set X_ 61.475711014808
134.        $node_(43) set Y_ 433.571081269250
135.        $node_(43) set Z_ 0.000000000000
136.        $node_(44) set X_ 437.554667648247
137.        $node_(44) set Y_ 329.046132143493
138.        $node_(44) set Z_ 0.000000000000
139.        $node_(45) set X_ 436.493102647496
140.        $node_(45) set Y_ 3.963881704138
141.        $node_(45) set Z_ 0.000000000000
142.        $node_(46) set X_ 352.417926754087
143.        $node_(46) set Y_ 437.233882670907
144.        $node_(46) set Z_ 0.000000000000
145.        $node_(47) set X_ 57.657417911744
146.        $node_(47) set Y_ 327.082450500026
147.        $node_(47) set Z_ 0.000000000000
148.        $node_(48) set X_ 353.991479631566
149.        $node_(48) set Y_ 156.788900325096
150.        $node_(48) set Z_ 0.000000000000
151.        $node_(49) set X_ 503.331492525281
152.        $node_(49) set Y_ 265.798646653099
153.        $node_(49) set Z_ 0.000000000000
```

Kode Sumber 7.1 Posisi *node* dari potongan Skenario

```
1.  $god_ set-dist 0 1 1
2.  $god_ set-dist 0 2 1
3.  $god_ set-dist 0 3 2
4.  $god_ set-dist 0 4 2
5.  $god_ set-dist 0 5 1
6.  $god_ set-dist 0 6 2
7.  $god_ set-dist 0 7 1
8.  $god_ set-dist 0 8 1
9.  $god_ set-dist 0 9 1
10. $god_ set-dist 0 10 2
```

```
$god_ set-dist 0 11 1
$god_ set-dist 0 12 1
$god_ set-dist 0 13 1
$god_ set-dist 0 14 1
$god_ set-dist 0 15 1
$god_ set-dist 0 16 2
$god_ set-dist 0 17 1
$god_ set-dist 0 18 2
$god_ set-dist 0 19 2
$god_ set-dist 0 20 2
$god_ set-dist 0 21 1
$god_ set-dist 0 22 2
$god_ set-dist 0 23 1
$god_ set-dist 0 24 1
$god_ set-dist 0 25 1
$god_ set-dist 0 26 2
$god_ set-dist 0 27 2
$god_ set-dist 0 28 1
$god_ set-dist 0 29 2
$god_ set-dist 0 30 2
$god_ set-dist 0 31 1
$god_ set-dist 0 32 1
$god_ set-dist 0 33 2
$god_ set-dist 0 34 1
$god_ set-dist 0 35 2
$god_ set-dist 0 36 1
$god_ set-dist 0 37 1
$god_ set-dist 0 38 2
$god_ set-dist 0 39 1
$god_ set-dist 0 40 2
$god_ set-dist 0 41 1
$god_ set-dist 0 42 1
$god_ set-dist 0 43 1
$god_ set-dist 0 44 2
$god_ set-dist 0 45 2
$god_ set-dist 0 46 2
$god_ set-dist 0 47 1
$god_ set-dist 0 48 1
$god_ set-dist 0 49 2
$god_ set-dist 1 2 1
$god_ set-dist 1 3 2
$god_ set-dist 1 4 2
$god_ set-dist 1 5 1
```

```
54. $god_ set-dist 1 6 2
55. $god_ set-dist 1 7 1
56. $god_ set-dist 1 8 2
57. $god_ set-dist 1 9 1
58. $god_ set-dist 1 10 2
59. $god_ set-dist 1 11 1
60. $god_ set-dist 1 12 1
61. $god_ set-dist 1 13 2
62. $god_ set-dist 1 14 1
63. $god_ set-dist 1 15 1
64. $god_ set-dist 1 16 2
65. $god_ set-dist 1 17 1
66. $god_ set-dist 1 18 2
67. $god_ set-dist 1 19 1
68. $god_ set-dist 1 20 2
69. $god_ set-dist 1 21 1
70. $god_ set-dist 1 22 2
71. $god_ set-dist 1 23 1
72. $god_ set-dist 1 24 1
73. $god_ set-dist 1 25 1
74. $god_ set-dist 1 26 2
75. $god_ set-dist 1 27 2
76. $god_ set-dist 1 28 1
77. $god_ set-dist 1 29 2
78. $god_ set-dist 1 30 2
79. $god_ set-dist 1 31 1
80. $god_ set-dist 1 32 1
81. $god_ set-dist 1 33 2
82. $god_ set-dist 1 34 1
83. $god_ set-dist 1 35 2
84. $god_ set-dist 1 36 1
85. $god_ set-dist 1 37 1
86. $god_ set-dist 1 38 2
87. $god_ set-dist 1 39 1
88. $god_ set-dist 1 40 2
89. $god_ set-dist 1 41 1
90. $god_ set-dist 1 42 1
91. $god_ set-dist 1 43 2
92. $god_ set-dist 1 44 2
93. $god_ set-dist 1 45 2
94. $god_ set-dist 1 46 2
95. $god_ set-dist 1 47 1
96. $god_ set-dist 1 48 1
```

```
97. $god_ set-dist 1 49 2
```

Kode Sumber 7.2 Pembuatan 'GOD' dari potongan skenario

```
1.  #
2.  # nodes: 10, max conn: 8, send rate: 0.25, seed: 1.
    0
3.  #
4.  #
5.  # 1 connecting to 2 at time 2.5568388786897245
6.  #
7.  set udp_(0) [new Agent/UDP]
8.  $ns_ attach-agent $node_(1) $udp_(0)
9.  set null_(0) [new Agent/Null]
10. $ns_ attach-agent $node_(2) $null_(0)
11. set cbr_(0) [new Application/Traffic/CBR]
12. $cbr_(0) set packetSize_ 512
13. $cbr_(0) set interval_ 1
14. $cbr_(0) set random_ 1
15. $cbr_(0) set maxpkts_ 10000
16. $cbr_(0) attach-agent $udp_(0)
17. $ns_ connect $udp_(0) $null_(0)
18. $ns_ at 2.5568388786897245 "$cbr_(0) start"
19. #
```

Kode Sumber 7.3 Koneksi yang digunakan pada 'cbrtest.txt'

```
1.  # Simulation with AOMDV Routing Protocol
2.
3.  # Define options
4.  set val(chan)           Channel/WirelessChannel
    ;# channel type
5.  set val(prop)           Propagation/Nakagami   ;# r
    adio-propagation model
```

```
6.  set val(netif)          Phy/WirelessPhy
    ;# network interface type
7.  set val(mac)            Mac/802_11
    ;# MAC type
8.  set val(ifq)            CMUPriQueue    ;# interface
     queue type
9.  set val(ll)             LL
    ;# link layer type
10. set val(ant)            Antenna/OmniAntenna
    ;# antenna model
11. set val(ifqlen)         50
    ;# max packet in ifq
12. set val(nn)             50
    ;# number of mobilenodes
13. set val(rp)             AOMDV
     ;# routing protocol
14. set opt(x)              510                ;# X dim
    ension of topography
15. set opt(y)              510                ;# Y dim
    ension of topography
16. set val(stop)       100            ;# time of simul
    ation end
17. set val(seed)       0                ;
18. set val(cp)                 "cbrtest.txt";
19. set val(sc)                 "scenario.txt";
20.
21. Phy/WirelessPhy set RXThresh_ 1.42681e-08;
22.
23. set ns_       [new Simulator]
24. set tracefd       [open Naka-M5_1.tr w]
25. #set windowVsTime2 [open win.tr w]
26. set namtrace      [open Naka-M5_1.nam w]
27.
28.
29. $ns_ trace-all $tracefd
30. #$ns use-newtrace
31. $ns_ namtrace-all-
    wireless $namtrace $opt(x) $opt(y)
32.
33. # set up topography object
34. set topo       [new Topography]
35.
36. $topo load_flatgrid $opt(x) $opt(y)
```

```
37.
38. set god_ [create-god $val(nn)]
39.
40. #
41. #  Create nn mobilenodes [$val(nn)] and attach them
     to the channel.
42. #
43.
44. # configure the nodes
45.         $ns_ node-config -adhocRouting $val(rp) \
46.              -llType $val(ll) \
47.              -macType $val(mac) \
48.              -ifqType $val(ifq) \
49.              -ifqLen $val(ifqlen) \
50.              -antType $val(ant) \
51.              -propType $val(prop) \
52.              -phyType $val(netif) \
53.              -channelType $val(chan) \
54.              -topoInstance $topo \
55.              -agentTrace ON \
56.              -routerTrace ON \
57.              -macTrace OFF \
58.              -movementTrace ON
59.
60.     for {set i 0} {$i < $val(nn) } { incr i } {
61.         set node_($i) [$ns_ node]
62.         $node_($i) random-motion 0;
63.     }
64.
65. # Define node movement model
66. puts "Loading Conneciton Pattern ...."
67. source $val(cp)
68.
69. #Define traffice mode
70. puts "Loading scenarion file...."
71. source $val(sc)
72.
73. #define node initial position in nam
74.
75.     for {set i 0} {$i < $val(nn) } { incr i } {
76.         $ns_ initial_node_pos $node_($i) 20
77.     }
78. #tell nodes when the simulation ends
```

```
79.     for {set i 0} {$i < $val(nn) } { incr i } {
80.         $ns_ at $val(stop).0 "$node_($i) reset";
81.     }
82. $ns_ at $val(stop).0002 "puts \"NS EXITING....\";
83. $ns_ halt"
84.
85. puts $tracefd "M 0.0 nn $val(nn) x $opt(x) y $opt(y
    ) rp $val(rp)"
86. puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $
    val(seed)"
87. puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

88.
89. puts "Starting Simulation...."
90. $ns_ run
```

Kode Sumber 7.4 file .tcl untuk simulasi AOMDV

```
1.  BEGIN{
2.  rt_pkts = 0;
3.  }
4.  {
5.  if (($1 == "s" || $1 == "f") && ($4 == "RTR"))
6.          rt_pkts++;
7.  }
8.
9.  END {
10. printf ("Total number of routing packets\t%d\n",
    rt_pkts);
11. }
```

Kode Sumber 7.5 Implementasi perhitungan RO

```
1.  BEGIN {
2.  sent=0;
3.  recv=0;
4.  pdr=0;
```

```
5.  }
6.  {
7.  # Count Packet Send
8.  if ($1 == "s" && $3 == "_1_" && $4 == "AGT" && $7
    == "cbr")
9.  {
10. sent++;
11. }
12.
13. # Count Packet Receive
14. if ($1 == "r" && $3 == "_2_" && $4 == "AGT" && $7
    == "cbr")
15. {
16. recv++;
17. }
18. }
19.
20. END {
21. pdr = (recv / sent) * 100
22. print "Transmitted packet (s): ", sent;
23. print "Received packet (s): ", recv;
24. print "Packet Delivery Ratio: ", pdr, "%";
25. }
```

Kode Sumber 7.6 implementasi perhitungan PDR

```
1.  BEGIN {
2.  for (i in pkt_id)
3.          {
4.                  pkt_id[i] = 0;
5.          }
6.  for (i in pkt_send)
7.          {
8.                  pkt_sent[i] = 0;
9.          }
10. for (i in pkt_recv)
11.         {
12.                 pkt_recv[i] = 0;
13.         }
14.         delay = avg_delay = 0;
15.         recv = 0;
```

```
16.         recv_id = 0;
17. }
18.
19. {
20. # Count Packet Send
21. if ($1 == "s" && $3 == "_1_" && $4 == "AGT" && $7
    == "cbr")
22.         {
23.                 pkt_sent[$6] = $2;
24.         }
25.
26. # Count Pakcet Receive
27. if ($1 == "r" && $3 == "_2_" && $4 == "AGT" && $7
    == "cbr" && recv_id != $6)
28.         {
29.                 recv++;
30.                 recv_id = $6;
31.                 pkt_recv[$6] = $2;
32.         }
33. }
34.
35. END{
36. for (i in pkt_recv)
37.         {
38.         delay += pkt_recv[i] - pkt_sent[i];
39.         }
40.
41. avg_delay = delay / recv;
42.
43. print "Total Packet(s) Receive = ", recv;
44. print "Total Delay = ", delay, " second";
45. print "Average Packet Delivery Delay = ",
    avg_delay, "second";
46. }
```

Kode Sumber 7.7 implementasi perhitungan E2E

Installasi NS-2

Instalasi NS-2 dilakukan pada sistem operasi Ubuntu 16.04. Yang diperlukan untuk menggunakan NS-2 adalah melakukan instalasi dependensi dan *source code* ns-2.35. Sebelum melakukan instalasi NS-2 diperlukan beberapa dependensi agar NS-2 dapat dijalankan. Cara instalasi dependensi tersebut ditunjukkan pada perintah di bawah

```
1. sudo apt-get install build-
   essential autoconf automake
```

Kode Sumber 7.8 installasi NS-2

Setelah semua dependensi terpasang selanjutnya adalah mengunduh *source code* ns-2.35 dengan cara yang ditunjukkan pada perintah di bawah

```
1. wget http://jaist.dl.sourceforge.net/project/nsnam/
   allinone/nsallinone-2.35/ns-allinone-2.35.tar.gz
```

Kode Sumber 7.9 mengunduh kode sumber ns-2.35

Lalu mengekstrak file dengan kode dibawah ini

```
1. tar -xvf ns-allinone-2.35.tar.gz
```

Kode Sumber 7.10 Ekstrak file NS-2

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Paul Aldy Sarumaha, lahir pada tanggal 28 Juni 1996 di Banda Aceh. Penulis adalah anak kedua dari tiga bersaudara dan dibesarkan di Pekanbaru, Riau. Penulis menempuh pendidikan formal di SDN 001 Sukajadi (2002-2008), SMPN 1 Pekanbaru (2008-2011), SMAN 8 Pekanbaru (2011-2014). Pada tahun 2014, penulis memulai pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi dan Komunikasi di Institut Teknologi Sepuluh Nopember Surabaya. Penulis pernah mengikuti organisasi mahasiswa tingkat jurusan sebagai Anggota REEVA pada Schematics 2016 .

Apabila ingin bertanya lebih lanjut, silahkan kirim *email* ke paulaldy28@gmail.com.