TUGAS AKHIR – KI141502

Analisis Kinerja Model Propagasi TwoRayGround pada Dynamic Source Routing (DSR) di lingkungan MANET

Admiral Budi Arviansyah Wilarsani
NRP 05111340000189

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

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – KI141502

Analisis Kinerja Model Propagasi TwoRayGround pada Dynamic Source Routing (DSR) di lingkungan MANET

Admiral Budi Arviansyah Wilarsani
NRP 05111340000189

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

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

UNDERGRADUATE THESES – KI141502

Analysis Tworayground Propagation Model with Dynamic Source Routing (DSR) on MANET

Admiral Budi Arviansyah Wilarsani
NRP 05111340000189

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

DEPARTMENT OF INFORMATICS
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Analisis Kinerja Model Propagasi TwoRayGround pada Dynamic Source Routing (DSR) di lingkungan 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 :
Admiral Budi Arviansyah Wilarsani
NRP : 05111340000189

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.
NIP: 198410162008121002 ..... (pembimbing 1)

SURABAYA
JULI 2018

[Halaman ini sengaja dikosongkan]

Analisis Kinerja Model Propagasi TwoRayGround pada Dynamic Source Routing (DSR) di lingkungan MANET

Nama Mahasiswa : Admiral Budi Arviansyah Wilarsani
NRP : 05111340000189
Jurusan : Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
Dosen Pembimbing 2 : -

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.

Dynamic Source Routing atau disingkat DSR adalah salah satu metode routing yang menggunakan algoritma Reactive. Algoritma ini digunakan apabila kita ingin mendistribusikan data yang jalurnya belum di temui maka dia akan mencari dan menyimoan jalur tersebut

Implementasi pada lingkungan MANET dapat dilakukan dengan menggunakan simulasi. Simulasi dilakukan dengan menggunakan aplikasi Network Simulator 2 (NS-2). Pada penelitian ini akan dilakukan analisa kinerja dari model propagasi TwoRayGround pada protokol routing DSR di lingkungan MANET berdasarkan Packet Delivery Ratio(PDR), Routing Overhead(RO), dan End-to-End Delay.

***Kata kunci**: MANET, DSR, NS-2,TwoRayGround*

[Halaman ini sengaja dikosongkan]

Analysis Tworayground Propagation Model with Dynamic Source Routing (DSR) on MANET

Student Name	: Admiral Budi Arviansyah Wilarsani
Student ID	: 05111340000189
Major	: Informatics FTIK-ITS
Advisor 1	: Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
Advisor 2	: -

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.

Dynamic Source Routing (DSR) is a protocol that we use on MANET. With this protocol we can send a data without knowing the path from sender to reciever.

Implementation on MANET was base on simulation. Simulation will be using Network Simulator 2 with TwoRayGround propagation model on DSR Protocol and we will know the Packet Delivery Ratio(PDR), Routing Overhead(RO), and End to End Delay.

Keyword : MANET,DSR,NS-2,TwoRayGround

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

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

"Analisis Kinerja Model Propagasi TwoRayGround pada Dynamic Source Routing (DSR) di lingkungan 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. Allah SWT atas segala nikmat dan rizki yang telah di berikan kepada penulis.
2. Keluarga yang senantiasa memberikan dukungan penuh baik secara moriil maupun materiil.
3. Bapak Dr.Eng Radityo Anggoro S.Kom., M,Sc. selaku dosen pembimbing pertama yang telah bersedia meluangkan waktu untuk membimbing dan memotivasi penulis serta memberi arahan dalam mengerjakan tugas akhir.
4. Bapak Dr.Eng. Darlis Herumurti, S.Kom, M.Sc. selaku dosen wali dari penulis yang memberikan arahan selama menjalani perkuliahan.
5. Bapak/Ibu dosen, staf dan karyawan Jurusan Teknik Informatika ITS yang telah banyak memberikan dukungan, ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
6. Bagus Putra Mayani dan Kevin Arditya yang selalu membantu penulis dalam mengerjakan tugas akhir ini.

7. Teman Teman BPH SCHEMATICS HMTC 2015, departemen Dalam Negeri HMTC 2014/2015 yang telah memberikan pengalaman berarti selama kuliah.
8. Teman Teman angkatan 2013 dan angkatan 2014 yang sudah membantu penulis selama di kampus.
9. Teman-teman penghuni laboraturium Algoritma Pemrograman yang dukungan dan pembelajaran baru selama perkuliahan.
10. 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, 4 Juli 2018
Penulis

Admiral Budi Arviansyah Wilarsani

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

Pada Zaman era digital saat ini kebutuhan akan teknologi sangat amat dibutuhkan untuk kebutuhan masyarakat sehingga pertumbuhannya pun sangat cepat. Komunikasi merupakan alat terpenting dalam menunjang kehidupan manusia sehingga teknologi di bidang komunikasi pun sangat dibutuhkan. Salah satu teknologi komunikasi yang di gunakan manusia untuk membangun infrastruktur jaringan adalah *Mobile Ad-hoc Network* (MANET).

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. Sebagai contoh, di masa lalu untuk memancarkan paket data dalam jaringan, dibutuhkan infrastruktur seperti tower radio yang mahal dan cukup lama dalam pembangunannya namun, hal ini dapat di modernisasi dengan ketersediaan router sebagai node pada MANET yang memungkinkan jalur komunikasi yang lebih efektif dan efisien. Hal terpenting dalam MANET adalah proses pengiriman data. Proses ini lebih dikenal dengan istilah *routing protocol* dan salah satu metode routing yang ada pada jaringan MANET adalah *Dynamic Source Routing*(DSR).

Dynamic Source Routing atau disingkat DSR adalah salah satu metode *routing* yang menggunakan algoritma *Reactive*. Algoritma ini digunakan apabila kita ingin mendistribusikan data yang jalurnya belum di temui maka dia akan mencari dan menyimpan jalur tersebut.

Untuk mempelajari sistem dengan baik, implementasi pada lingkungan MANET dapat dilakukan dengan menggunakan simulasi. Simulasi dilakukan dengan menggunakan aplikasi *Network Simulator 2* (NS-2). Pada penelitian ini akan dilakukan analisa kinerja dari model propagasi *TwoRayGround* pada protokol routing DSR di lingkungan MANET berdasarkan *Packet Delivery Ratio*(PDR), *Routing Overhead*(RO), dan *End-to-End Delay*.

1.2. Rumusan Permasalahan

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

1. Bagaimana kinerja Protokol routing DSR pada Manet?
2. Bagaimana hasil analisis studi kinerja model propagasi *TwoRayGround* pada Protokol routing DSR di lingkungan MANET?

1.3. Batasan Permasalahan

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

1. Protokol Routing yang diuji coba adalah *Dynamic Source Routing*.
2. Model Propagasi yang akan dianalisis dalam Tugas Akhir ini adalah *TwoRayGround*.
3. Skenario Ujicoba dilakukan pada Topologi Manet.
4. Simulasi Pengujian Protocol Routing Menggunakan *Network Simulator 2*.
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 *TwoRayGround* pada protocol DSR 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 *TwoRayGround* pada Protokol DSR 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 *TwoRayGround* 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. DSR
4. TwoRayGround

1.6.3. Implementasi

Pada tahap ini dilakukan implementasi berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini merupakan yang paling penting dimana bentuk awal aplikasi yang diimplementasikan didefinisikan. Pada tahapan ini dibuat prototype system yang merupakan rancangan dasar dari system yang akan dibuat. Serta dilakukan desain suatu system dan desain proses proses yang ada

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.

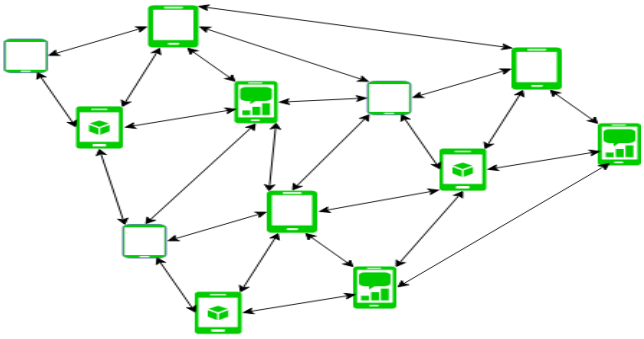
[Halaman ini sengaja dikosongkan]

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 [2]

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

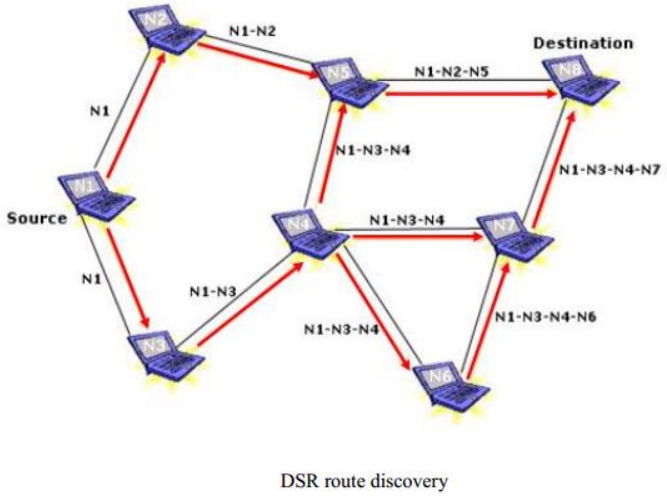
- *Proactive routing*: jenis routing ini bersifat menyebarkan seluruh informasi rute tabel ke setiap *node* yang ada. Apabila terjadi perubahan maka informasi perubahannya akan disebarkan ke seluruh node. Contoh dari tipe routing ini adalah *Optimized Link State Routing* (OLSR).
- *Reactive routing*: cara kerja dari routing ini adalah apabila *node* sumber mau mengirimkan paket maka *node* sumber akan mengirimkan *route request*. setelah sampai di *node* tujuan maka akan di kirim *node reply* ke *node* sumber. Contoh dari routing ini adalah *Dynamic Source Routing* (DSR), *ad-hoc on Demand Vector(*AODV).
- *Hybrid Routing*: jenis routing ini merupakan gabungan dari *Proactive routing* dan *Reactive routing*. Contoh dari routing ini adalah *Zone Routing Protocol*(ZRP)[3].

2.2. *Dynamic Source Routing* (DSR)

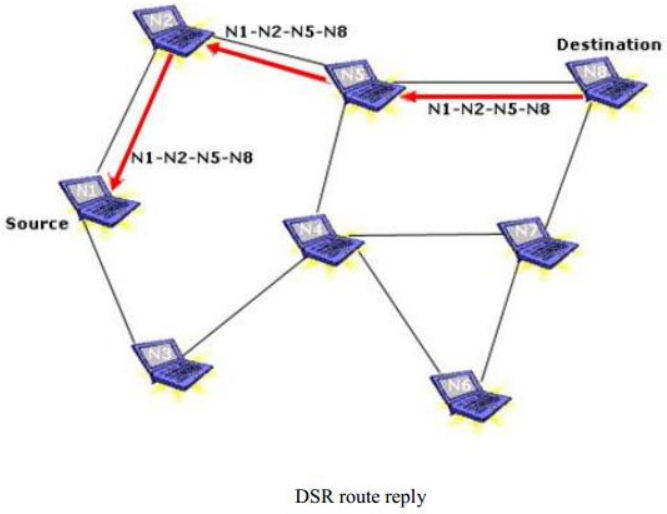
DSR adalah routing protocol yang simpel dan efisien digunakan untuk *multi-hop wireless ad hoc networks*. DSR sendiri merupakan protocol yang dapat mengatur dan menentukan secara mandiri paket data yang akan di kirim. protokolnya terdiri dari dua mekanisme yaitu "*Route Discovery*" dan "*Route Maintenance*" yang mana kedua protokol ini bekerja bersamaan untuk mencari dan memlihara rute yang sudah di temukan. semua aspek dalam protokol ini bergantung pada permintaan sehingga protokol ini dapat memilih sendiri rute mana yang bisa digunakan untuk mencapai tujuan.[4]

Route Discovery merupakan fase dalam *Dynamic Source Routing* yang digunakan dalam melakukan pengiriman paket dari *node* sumber menuju *node* tujuan. Proses yang dilakukan *Route*

Discovery adalah mem-*broadcast* paket *request route*. Paket *request route* berisi alamat tujuan, alamat *node* sumber dan unik ID. Setiap *node* yang menerima paket *request route* akan memeriksa *route cache* masing-masing. Jika *node* tidak mengetahui *node* tujuan, maka *node* tersebut akan menambahkan alamat *node* tersebut ke dalam paket dan kemudian mem-*forward*-nya. *Route reply* di-*generate* ketika *route request* mencapai *node* tujuan atau node perantara yang mempunyai informasi *node* tujuan pada *route cachenya*. *Node* tersebut akan menyampaikan paket *route reply* ke *node* selanjutnya hingga sampai *node* sumbernya. Sedangkan *Route Maintenance* digunakan apabila terdapat *route error* yang berguna untuk menghapus informasi pada *route cache* ketika ada hubungan antar *node* yang berubah. *Route Acknowledgement* berguna untuk mengganti informasi hubungan antar node pada *route cache*.[5]



Gambar 2.2 Contoh *Route Discovery*[6]



Gambar 2.3 Contoh *Route Reply*[6]

2.3. 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*[7]. Model ini memiliki keakuratan yang tinggi dalam memperkirakan sinyal dalam skala yang luas dan jauh. Power yang diterima dengan jarak di diberikan oleh persaman 1.

$$\Pr(d) = \frac{PtGtGrht^2hr^2}{d^4L} \quad (1)$$

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[8].

2.4. *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

2.5. 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.[9]

2.6. *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 pada Kode Sumber 2.1 dan keterangannya ditunjukan pada tabel 2.1.

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

Kode Sumber 2.1 Format C*ommand Line* ‘setdest’

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 pengiriman paket.
-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.[10]

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 pada Kode Sumber 2.2.

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

Kode Sumber 2.2 Contoh C*ommand Line* 'setdest'

2.7. *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" diigunakan sesuai dengan *command line* pada gambar 2.4 dan dengan keterangan yang diunjukan pada tabel 2.2.[11]

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

Kode Sumber 2.3 Format *Command Line* 'cbrgen.tcl'

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.

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

Kode Sumber 2.4 Contoh C*ommand Line* 'cbrgen.tcl'

BAB III
ANALISIS DAN PERANCANGAN SISTEM

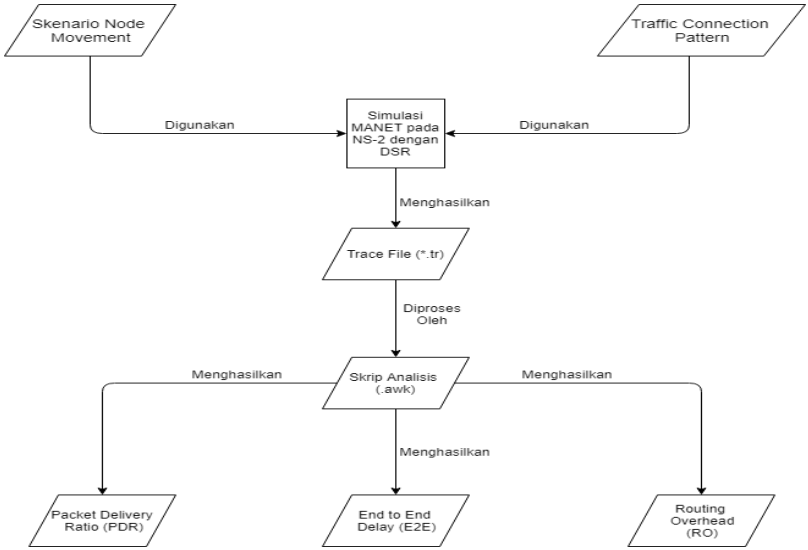
Pada bab ini akan dijelaskan mengenai dasar perancangan dari perangkat lunak yang akan dibangun dalam tugas akhir ini. Perancangan tersebut mencakup deskripsi umum aplikasi , arsitektur sistem, model fungsional dan diagram alur aplikasi.

3.1. Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dilakukan analisis tentang performa model propagasi *TwoRayGround* 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 men-*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 *TwoRayGround* pada simulasi skenario yang dijalankan pada NS-2 menggunakan *routing protocol* DSR pada system operasi Linux.

Pada tiap skenario, kecepatan maksimum dari satu *node* ke *node* lainnya dibuat bervariasi 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 *TwoRayGround.*



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	800m x 800m
4	Kecepatan Maksimal	-5m/s -10m/s -15m/s
5	Sumber *Traffic*	CBR
6	Waktu Jeda (Dalam Detik)	1
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*	2

No.	Parameter	Spesifikasi
3	-s *seed*	1.0
4	-mc *connection*	1
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 3.2

Table 3.3 Penjelasan simulasi NS-2

No	Parameter	Spesifikasi
1	*Network Simulator*	NS-2, 2.35
2	Routing Protocol	DSR
3	Waktu Simulasi	100 detik
4	Area Simulasi	800m x 800m
5	Banyak *node*	50
6	Radius Transmisi	250m
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	*TwoRayGround*
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 merupakan perbandingan dari jumlah paket komunikasi yang dikirimkan dengan paket komunikasi yang diterima. *Packet delivery ratio* dihitung menggunakan persamaan 2, dimana *received* adalah jumlah paket komunikasi yang diterima dan *sent* adalah jumlah paket komunikasi yang dikirimkan. Semakin tinggi *packet delivery ratio* semakin berhasil pengiriman paket yang dilakukan.

$$Packet\ Delivery\ Ratio = \frac{Received}{Sent} \quad (2)$$

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. Total interupsi terdiri dari *delay* yang mungkin karena *buffer* pada *route discovery latency*, *delay* pada antrian *interface*, retransmisi. 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 3

$$End\ to\ End\ Delay = \sum_{i<sent}^{i=0} \frac{t^{received(i)} - t^{sent(i)}}{sent} \quad (3)$$

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 4.

$$Routing\ Overhead = \sum_{i<sent}^{i=0} \frac{packet\ routing}{received} \quad (4)$$

BAB IV
IMPLEMENTASI

Bab ini akan menjelaskan tentang implementasi Tugas Akhir berdasarkan rancangan perangkat lunak yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi skenario, implementasi simulasi NS-2, dan implementasi matiks 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 lenovo E431 o Prosesor Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz o RAM 12 GB o HDD 1 TB
Perangkat Lunak	- Laptop Lenovo E431 o Sistem Operasi Ubuntu 14.04 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* pada kode sumber 4.1 yang digunakan untuk menghasilkan gerakan acak pada node adalah sebagai berikut:

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

Kode Sumber 4.1 Format kode 'setdest'

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 4.2 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 800 -
    y 800 > scenario.txt
```

```
2.  ./setdest -v 1 -n 50 -p 10 -M 10 -t 100 -x 800 -
    y 800 > scenario.txt
3.  ./setdest -v 1 -n 50 -p 10 -M 15 -t 100 -x 800 -
    y 800 > scenario.txt
```

Kode Sumber 4.2 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 dule 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 dimau, dan nilai *random seed*. *Format command line* pada kode sumber 4.7 yang digunakan untuk menghasilkan gerakan acak pada *node* adalah sebagai berikut.

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

Kode Sumber 4.3 Format kode 'cbrgen.tcl'

Pada kode sumber 4.4 merupakan bentuk implementasi untuk menjalankan cbrgen.tcl untuk membuat file koneksi CBR diantara 2 node memiliki maksimal 1 koneksi dengan nilai seed 1.0 dan jumlah paket per detik sebanyak 0.25 yang disimpan dalam cbr.txt yang nantinya akan digunakan pada saat simulasi NS-2.

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

Kode Sumber 4.4 Implementasi kode 'cbrgen.tcl'

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

Table 4.1 Penjelasan simulasi NS-2

No.	Parameter	Spesifikasi
1	Channel/	Wireless Channel
2	Propagation/	TwoRayGround
3	Phy/ WirelessPhy	WirelessPhy
4	Mac/ 802.11	802.11
5	Queue Drotail/PriQueue	PriQueue
6	Antenna/OmniAntena	OmniAntena
7	Nilai X	800
8	Nilai Y	800
9	Nilai seed	0.0
10	Routing Protocol	DSR
11	File traffic connection	“cbrtest.txt”
12	File node movement	“scenario.txt”

```
1.  set val(chan)           Channel/WirelessChannel;
2.  set val(prop)           Propagation/TwoRayGround;
3.  set val(netif)          Phy/WirelessPhy;
4.  set val(mac)            Mac/802_11;
5.  set val(ifq)            CMUPriQueue;
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)             DSR;
11. set opt(x)              800;
12. set opt(y)              800;
13. set val(stop)           100;
14. set val(seed)           0;
```

```
set val(cp)             "cbr.txt";
set val(sc)             "scenario.txt";
```

Kode Sumber 4.5 Konfigurasi parameter pada NS-2

Pada kode sumber 4.11 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;
```

Kode Sumber 4.6 Pengaturan *Transmission range*

```
set ns_       [new Simulator]
set tracefd       [open trace.tr w]
#set windowVsTime2 [open win.tr w]
set namtrace      [open simwrls.nam w]


$ns_ trace-all $tracefd
#$ns use-newtrace
$ns_ namtrace-all-
wireless $namtrace $opt(x) $opt(y)

# set up topography object
set topo       [new Topography]

$topo load_flatgrid $opt(x) $opt(y)

set god_ [create-god $val(nn)]

#
#  Create nn mobilenodes [$val(nn)] and attach them
 to the channel.
#

# configure the nodes
        $ns_ node-config -adhocRouting $val(rp) \
```

```
24.             -llType $val(ll) \
25.             -macType $val(mac) \
26.             -ifqType $val(ifq) \
27.             -ifqLen $val(ifqlen) \
28.             -antType $val(ant) \
29.             -propType $val(prop) \
30.             -phyType $val(netif) \
31.             -channelType $val(chan) \
32.             -topoInstance $topo \
33.             -agentTrace ON \
34.             -routerTrace ON \
35.             -macTrace OFF \
36.             -movementTrace ON
37.     for {set i 0} {$i < $val(nn) } { incr i } {
38.         set node_($i) [$ns_ node]
39.         $node_($i) random-motion 0;
40.     }
```

Kode Sumber 4.7 Pengaturan variabel global pada NS-2

Skrip yang ditunjukan pada kode sumber 4.7 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 4.8 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 4.8 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.

```
1.  awk -f PDR.awk trace.tr
```

Kode Sumber 4.9 Menjalankan file PDR.awk

Hasil dari perintah yang dijalankan adalah *packet delivery ratio* dari simulasi yang telah dijalankan dapat dilihat pada gambar 4.1.

Gambar 4.1 Contoh hasil nilai PDR

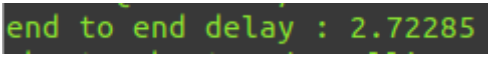
4.4.2. Implementasi *End-to-End Delay*

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.

```
1.  awk –f E2E.awk trace.tr
```

Kode Sumber 4.10 Menjalankan file E2E.awk

Hasil dari perintah yang dijalankan adalah *end-to-end delay* dari simulasi yang telah dijalankan dapat dilihat pada gambar 4.2.



Gambar 4.2 Contoh hasil nilai E2E

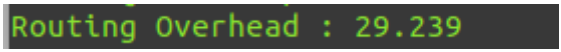
4.4.3. Implementasi *Routing Overhead*

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.

```
1.  awk –f ro.awk trace.tr
```

Kode Sumber 4.11 Menjalankan file ro.awk

Hasil dari perintah yang dijalankan adalah *end-to-end delay* dari simulasi yang telah dijalankan dapat dilihat pada Gambar 4.12.



Gambar 4.3 Contoh hasil nilai RO

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 lenovo E431 o Prosesor Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz o RAM 12 GB o HDD 1 TB
Perangkat Lunak	- Laptop Lenovo E431 o Sistem Operasi Windows 10 pro, Linux Ubunru 14.04 LTS 64 bit (NS-2, DSR, *Mobility Generation, Traffic Connection Genneration, TwoRayGround*.)

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.

.

Table 5.1 Penjelasan skenario pengujian

Kriteria	Spesifikasi
Skenario	MANET

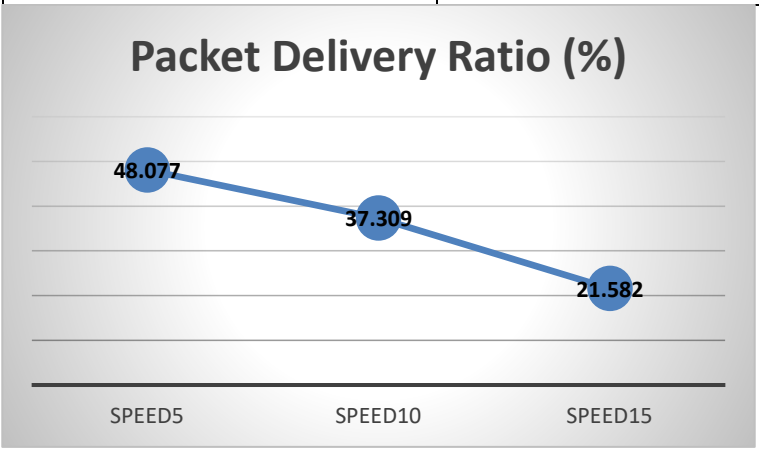
Kecepatan Maksimal Perpindahan node (m/s)	5, 10, 15
Jumlah percobaan	10
Jarak node 1 dan node 2	Acak
Posisi awal node	Acak
Pergerakan	Acak
Protokol Routing	DSR
Pengiriman Paket Data	0 – 100 Detik

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 *TwoRayGround* seperti pada tabel 5.2.

Table 5.2 Nilai hasil PDR

Kecepatan Maksimal Perpindahan Node (m/s)	PDR (%)
5	48.077
10	37.309
15	21.582



Gambar 5.1 Grafik nilai PDR

Pada tabel 5.2 dan gambar 5.1 menunjukan performa PDR yang dihasilkan oleh model propagasi *TwoRayGround* pada jaringan MANET dengan menggunakan skenario *node-Movement* (*mobility generation*) yang bersifat *Random Way Point*.

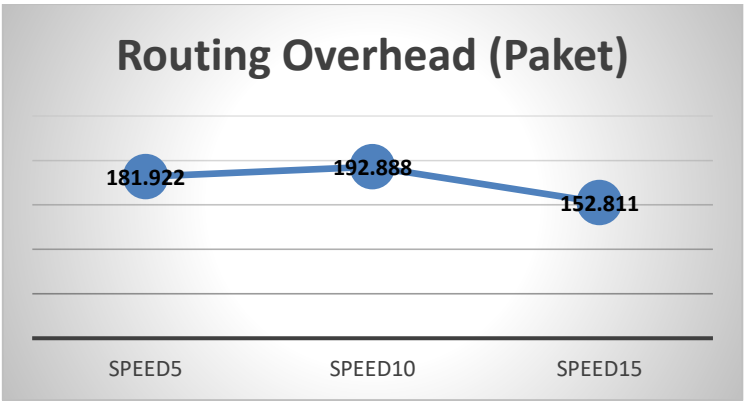
Bisa dilihat bahwa nilai rata rata dari *Packet Delivery Ratio* menunjukan nilai yang relatif menurun. Bisa dilihat bahwa nilai rata rata pada kecepatan Maksimal 5m/s adalah 48.077% sementara pada kecepatan maksimal 10m/s memiliki nilai 37.309% dan pada kecepatan maksimal 15m/s bernilai 21.582%. hal ini menunjukan bahwa semakin tinggi kecepatan maksimal yang dipunya maka semakin rendah pula paket data yang dikirimkan. Pergerakan *node* yang lebih dinamis inilah yang memungkinkan terjadinya banyak rute putus saat pengiriman paket data ataupun kegagalan pembentukan tabel routing yang dibuat oleh DSR sehingga menyebabkan paket data yang dikirimkan tidak sampai ke tujuan. 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 *TwoRayGround* menjadi seperti pada tabel 5.3

Table 5.3 Nilai hasil RO

Kecepatan Maksimal Perpindahan Node (m/s)	RO (paket)
5	181.922
10	192.888
15	152.811



Gambar 5.2 Grafik nilai RO

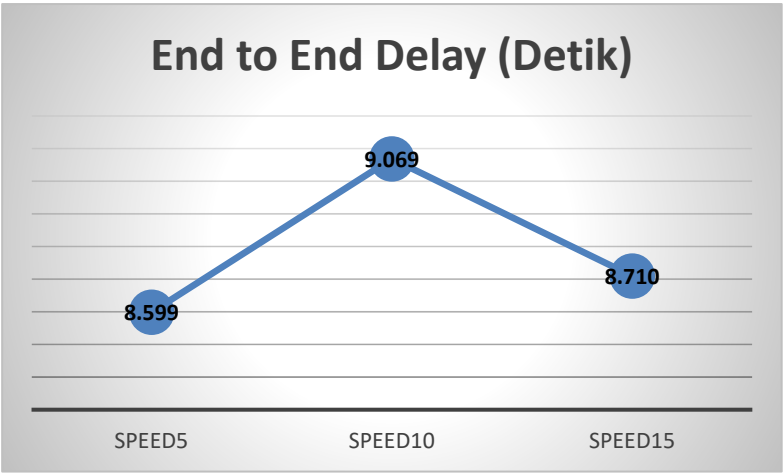
Dari hasil skenario yang sudah dibuat bisa dilihat bahwa nilai rata rata dari *Routing Overhead* yang dihasilkan bersifat fluktuatif. Pada kecepatan maksimal 5m/s adalah 181.922 paket sementara pada kecepatan maksimal 10m/s adalah 192.888 paket dan pada kecepatan maksimal 15m/s adalah 152.811 paket. Hal ini dapat terjadi akibat mobilitas pengiriman paket antar *node* yang sangat dinamis pada *skenario node-movement*(*mobility generation*) yang dihasilkan oleh model propagasi *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.

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 *TwoRayGround* menjadi tabel 5.4.

Table 5.4 Hasil nilai E2E

Kecepatan Maksimal Perpindahan Node (m/s)	End to End delay (Detik)
5	8.599
10	9.069
15	8.710



Gambar 5.3 Grafik hasil nilai E2E

Pada tabel 5.4 dan gambar 5.3 menunjukan pengujian model propagasi *TwoRayGround* untuk nilai rata rata dari *End to End Delay* bersifat fluktuatif pada jumlah waktu yang dibutuhkan untuk mengirimkan paket data yang dikirimkan. Pada kecepatan maksimal 5m/s rata rata nilainya adalah 8.599 detik sementara pada kecepatan maksimal 10m/s adalah 9.069 detik dan pada kecepatan maksimal 15m/s adalah 8.710 detik. Hal ini terjadi dikarenakan kecepatan yang berubah menjadi semakin cepat mulai dari kecepatan maksimum 5m/s menjadi 10m/s dan 15m/s.

[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 dalam tugas akhir ini adalah sebagai berikut:

1. Berikut ini merupakan nilai rataan dari PDR, RO,E2E pada MANET menggunakan protokol routing DSR.
    - Nilai rataan PDR yang didapat adalah 35.656%.
    - Nilai rataan RO yang didapat adalah 175.873 paket yang terkirim.
    - Nilai rataan E2E yang didapat adalah 8.793 detik untuk mengirimkan paket.
2. Skenario Manet yang dihasilkan oleh *node-movement*(*mobility generation*) dan dijalankan menggunakan model propagasi *TwoRayGround* dengan penambahan kecepatan maksimal perpindahan *node* memiliki performa sebagai berikut.
    - Performa *Packet Delivery Ratio* yang dihasilkan fluktuatif mulai dari 45.454% pada kecepatan maksimal 5m/s menjadi 17.391% pada kecepatan maksimal 10m/s dan menjadi 28% pada kecepatan maksimal 15m/s.
    - Performa *Routing Overhead* yang dihasilkan memiliki nilai semakin naik mulai dari 102.799 paket pada kecepatan maksimal 5m/s menjadi 174.032 paket pada kecepatan maksimal 10m/s dan berubah menjadi 181.442 paket pada kecepatan maksimal 15m/s.

- Performa *End to End Delay* yang dihasilkan memiliki nilai semakin turun yang berbanding lurus dengan kecepatan maksimalnya mulai dari 13.240 detik pada kecepatan maksimal 5m/s menjadi 9.395 detik pada kecepatan maksimal 10m/s dan berubah menjadi 6.633 detik pada kecepatan maksimal 15m/s.

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 DSR 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.

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] "MANET: Mobile Ad hoc Network," *GeeksforGeeks*, 08-Mar-2018. .

[3] N. H. Saeed, M. F. Abbod, and H. S. Al-Raweshidy, "MANET routing protocols taxonomy," 2012, pp. 123–128.

[4] "DSR, Dynamic Source Routing Protocol." [Online]. Available: http://www.networksorcery.com/enp/protocol/dsr.htm. [Accessed: 05-Jun-2018].

[5] "Dynamic Source Routing (DSR)," *MTI*. [Online]. Available: https://mti.binus.ac.id/2014/10/27/dynamic-source-routing-dsr/. [Accessed: 13-Jul-2018].

[6] "Introduction of Mobile ad hoc network (MANET)," *BINUS Malang*. [Online]. Available: http://binus.ac.id/malang/2017/09/introduction-of-mobile-ad-hoc-network-manet/. [Accessed: 13-Jul-2018].

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

[8] "18.2 Two-ray ground reflection model." [Online]. Available: https://www.isi.edu/nsnam/ns/doc/node218.html. [Accessed: 05-Jun-2018].

[9] "AWK, oh AWK - Mufid's Code blog." [Online]. Available: https://mufid.github.io/blog/2012/awk/. [Accessed: 05-Jul-2018].

[10] D. A. Maltz, "On-Demand Routing in Multi-hop Wireless Mobile Ad Hoc Networks:," Defense Technical Information Center, Fort Belvoir, VA, May 2001.

[11] R. Kumar, P. Verma, and Y. Singh, "Mobile Ad Hoc Networks and It's Routing Protocols," vol. 7, no. 8, p. 10, 2013.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

```
1.  #
2.  # nodes: 50, pause: 10.00, max speed: 5.00, max x:
    800.00, max y: 800.00
3.  #
4.  $node_(0) set X_ 115.048817342649
5.  $node_(0) set Y_ 13.347820652730
6.  $node_(0) set Z_ 0.000000000000
7.  $node_(1) set X_ 184.404711973895
8.  $node_(1) set Y_ 424.007525340087
9.  $node_(1) set Z_ 0.000000000000
10. $node_(2) set X_ 341.584010759712
11. $node_(2) set Y_ 356.774695918229
12. $node_(2) set Z_ 0.000000000000
13. $node_(3) set X_ 639.953032394459
14. $node_(3) set Y_ 767.892860183892
15. $node_(3) set Z_ 0.000000000000
16. $node_(4) set X_ 56.588324664713
17. $node_(4) set Y_ 736.474657148758
18. $node_(4) set Z_ 0.000000000000
19. $node_(5) set X_ 493.848751373715
20. $node_(5) set Y_ 390.641687170945
21. $node_(5) set Z_ 0.000000000000
22. $node_(6) set X_ 497.415714958326
23. $node_(6) set Y_ 667.731687912762
24. $node_(6) set Z_ 0.000000000000
25. $node_(7) set X_ 669.706783094213
26. $node_(7) set Y_ 195.400849320781
27. $node_(7) set Z_ 0.000000000000
28. $node_(8) set X_ 135.842814914720
29. $node_(8) set Y_ 335.449590201842
30. $node_(8) set Z_ 0.000000000000
31. $node_(9) set X_ 694.340952956797
32. $node_(9) set Y_ 174.211470278908
33. $node_(9) set Z_ 0.000000000000
34. $node_(10) set X_ 724.424217334073
35. $node_(10) set Y_ 748.333184526605
36. $node_(10) set Z_ 0.000000000000
37. $node_(11) set X_ 708.628089398761
38. $node_(11) set Y_ 561.192318966613
39. $node_(11) set Z_ 0.000000000000
```

```
$node_(12) set X_ 494.201434867899
$node_(12) set Y_ 707.271543869861
$node_(12) set Z_ 0.000000000000
$node_(13) set X_ 374.280877469672
$node_(13) set Y_ 697.754289473615
$node_(13) set Z_ 0.000000000000
$node_(14) set X_ 327.819973273798
$node_(14) set Y_ 214.858192366200
$node_(14) set Z_ 0.000000000000
$node_(15) set X_ 468.869029526775
$node_(15) set Y_ 634.212993065897
$node_(15) set Z_ 0.000000000000
$node_(16) set X_ 613.147078346133
$node_(16) set Y_ 474.341496886460
$node_(16) set Z_ 0.000000000000
$node_(17) set X_ 736.633350797868
$node_(17) set Y_ 208.417268505039
$node_(17) set Z_ 0.000000000000
$node_(18) set X_ 632.889131559278
$node_(18) set Y_ 678.318290293730
$node_(18) set Z_ 0.000000000000
$node_(19) set X_ 63.920350115614
$node_(19) set Y_ 363.024289419188
$node_(19) set Z_ 0.000000000000
$node_(20) set X_ 497.788853091207
$node_(20) set Y_ 622.766809709253
$node_(20) set Z_ 0.000000000000
$node_(21) set X_ 195.394585431105
$node_(21) set Y_ 574.751056625559
$node_(21) set Z_ 0.000000000000
$node_(22) set X_ 172.778730638785
$node_(22) set Y_ 628.651370750620
$node_(22) set Z_ 0.000000000000
$node_(23) set X_ 401.304790068277
$node_(23) set Y_ 596.845248561309
$node_(23) set Z_ 0.000000000000
$node_(24) set X_ 629.837437301452
$node_(24) set Y_ 575.448231700163
$node_(24) set Z_ 0.000000000000
$node_(25) set X_ 598.467498943498
$node_(25) set Y_ 797.492506233612
$node_(25) set Z_ 0.000000000000
$node_(26) set X_ 34.103795675000
```

```
83. $node_(26) set Y_ 410.825221392244
```

Kode Sumber 7.1 Posisi *node* dari potongan Skenario

```
1.  $god_ set-dist 0 1 2
2.  $god_ set-dist 0 2 3
3.  $god_ set-dist 0 3 5
4.  $god_ set-dist 0 4 4
5.  $god_ set-dist 0 5 3
6.  $god_ set-dist 0 6 5
7.  $god_ set-dist 0 7 4
8.  $god_ set-dist 0 8 2
9.  $god_ set-dist 0 9 4
10. $god_ set-dist 0 10 5
11. $god_ set-dist 0 11 5
12. $god_ set-dist 0 12 5
13. $god_ set-dist 0 13 4
14. $god_ set-dist 0 14 2
15. $god_ set-dist 0 15 4
16. $god_ set-dist 0 16 4
17. $god_ set-dist 0 17 4
18. $god_ set-dist 0 18 5
19. $god_ set-dist 0 19 2
20. $god_ set-dist 0 20 4
21. $god_ set-dist 0 21 3
22. $god_ set-dist 0 22 3
23. $god_ set-dist 0 23 4
24. $god_ set-dist 0 24 4
25. $god_ set-dist 0 25 5
26. $god_ set-dist 0 26 2
27. $god_ set-dist 0 27 3
28. $god_ set-dist 0 28 5
29. $god_ set-dist 0 29 2
30. $god_ set-dist 0 30 3
31. $god_ set-dist 0 31 3
32. $god_ set-dist 0 32 2
33. $god_ set-dist 0 33 2
34. $god_ set-dist 0 34 5
35. $god_ set-dist 0 35 2
36. $god_ set-dist 0 36 1
37. $god_ set-dist 0 37 3
```

```
38. $god_ set-dist 0 38 4
39. $god_ set-dist 0 39 1
40. $god_ set-dist 0 40 3
41. $god_ set-dist 0 41 6
42. $god_ set-dist 0 42 4
43. $god_ set-dist 0 43 3
44. $god_ set-dist 0 44 3
45. $god_ set-dist 0 45 3
46. $god_ set-dist 0 46 2
47. $god_ set-dist 0 47 1
48. $god_ set-dist 0 48 2
49. $god_ set-dist 0 49 5
50. $god_ set-dist 1 2 1
51. $god_ set-dist 1 3 3
52. $god_ set-dist 1 4 2
53. $god_ set-dist 1 5 2
54. $god_ set-dist 1 6 3
55. $god_ set-dist 1 7 3
56. $god_ set-dist 1 8 1
57. $god_ set-dist 1 9 3
58. $god_ set-dist 1 10 4
59. $god_ set-dist 1 11 3
60. $god_ set-dist 1 12 3
61. $god_ set-dist 1 13 2
62. $god_ set-dist 1 14 2
63. $god_ set-dist 1 15 2
64. $god_ set-dist 1 16 2
65. $god_ set-dist 1 17 3
66. $god_ set-dist 1 18 3
67. $god_ set-dist 1 19 1
68. $god_ set-dist 1 20 3
69. $god_ set-dist 1 21 1
70. $god_ set-dist 1 22 1
71. $god_ set-dist 1 23 2
72. $god_ set-dist 1 24 3
73. $god_ set-dist 1 25 3
74. $god_ set-dist 1 26 1
75. $god_ set-dist 1 27 2
76. $god_ set-dist 1 28 3
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 3
```

```
$god_ set-dist 1 33 1
$god_ set-dist 1 34 4
$god_ set-dist 1 35 1
$god_ set-dist 1 36 1
$god_ set-dist 1 37 1
$god_ set-dist 1 38 3
$god_ set-dist 1 39 2
$god_ set-dist 1 40 1
$god_ set-dist 1 41 4
$god_ set-dist 1 42 2
$god_ set-dist 1 43 1
$god_ set-dist 1 44 1
$god_ set-dist 1 45 1
$god_ set-dist 1 46 1
$god_ set-dist 1 47 2
$god_ set-dist 1 48 1
$god_ set-dist 1 49 3
```

Kode Sumber 7.2 Pembuatan ‘GOD’ dari potongan skenario

```
#
# nodes: 2, max conn: 1, send rate: 4.0, 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_ 4.0
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
```

```
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
#Total sources/connections: 1/1
#
```

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

```
# A 10-node example for ad-
hoc simulation with DSR

# Define options
set val(chan)           Channel/WirelessChannel
;# channel type
set val(prop)           Propagation/TwoRayGround
;# radio-propagation model
set val(netif)          Phy/WirelessPhy
;# network interface type
set val(mac)            Mac/802_11
;# MAC type
set val(ifq)            CMUPriQueue    ;# interface
 queue type
set val(ll)             LL
;# link layer type
set val(ant)            Antenna/OmniAntenna
;# antenna model
set val(ifqlen)         50
;# max packet in ifq
set val(nn)             50
;# number of mobilenodes
set val(rp)             DSR                        ;
# routing protocol
set opt(x)              800                ;# X dim
ension of topography
set opt(y)              800                ;# Y dim
ension of topography
set val(stop)       100            ;# time of simul
ation end
set val(seed)       0                ;
set val(cp)                 "cbr.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 trace.tr w]
25. #set windowVsTime2 [open win.tr w]
26. set namtrace      [open simwrls.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 DSR

```
1.  BEGIN{
2.      recvs = 0;
```

```
3.      besaran_paket = 0;
4.  }
5.  {
6.      if (($1 == "s" || $1 == "r") && $4 == "RTR"){
7.          recvs++;
8.      }
9.      if (($1 == "s" || $1 == "r") && $4 == "RTR")
10.     {
11.         besaran_paket = besaran_paket + $8;
12.     }
13. }
14. END{
15.     printf("Routing Overhead : %.3f\n", besaran_pak
    et/recvs);
16. }
```

Kode Sumber 7.5 Implementasi perhitungan RO

```
1.  BEGIN {
2.      sendLine = 0;
3.      recvLine = 0;
4.      forwardLine = 0;
5.  }
6.
7.  $0~/^s.*AGT/{
8.      sendLine++
9.  }
10.
11. $0~/^r.*AGT/{
12.     recvLine++
13. }
14.
15. $0~/^f.*RTR/{
16.     forwardLine++
17. }
18. END{
19.     printf"Ratio:%.4f\n",(recvLine/sendLine)*100;
20. }
```

Kode Sumber 7.6 implementasi perhitungan PDR

```
BEGIN {
seqno = -1;
# droppedPackets = 0;
# receivedPackets = 0;
count = 0;
}
{
if($4 == "AGT" && $1 == "s" && seqno < $6) {
seqno = $6;
}
if($4 == "AGT" && $1 == "s") {
start_time[$6] = $2;
} else if(($7 == "cbr") && ($1 == "r")) {
end_time[$6] = $2;
} else if($1 == "D" && $7 == "cbr") {
end_time[$6] = -1;
}
}
END {
for(i=0; i<=seqno; i++) {
if(end_time[i] > 0) {
delay[i] = end_time[i] - start_time[i];
count++;
}
else
{
delay[i] = -1;
}
}
for(i=0; i<=seqno; i++) {
if(delay[i] > 0) {
n_to_n_delay = n_to_n_delay + delay[i];
}
}
n_to_n_delay = n_to_n_delay/count;
printf("end to end delay : ")
print n_to_n_delay * 1000 ;
}
```

Kode Sumber 7.7 implementasi perhitungan E2E

Installasi NS-2

Instalasi NS-2 dilakukan pada sistem operasi Ubuntu 14.04. 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

Admiral Budi Arviansyah Wilarsani, lahir pada tanggal 24 Maret 1994 di Jakarta. Saat ini sedang menempuh perguruan tinggi di Institut Teknologi Sepuluh November Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi pada tahun 2013. Terlibat aktif pada organisasi mahasiswa tingkat jurusan antara lain Departemen Dalam Negeri Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS sebagai staff tahun 2014 – 2015, sebagai staff Dana dan Usaha pada Schematics 2014, dan wakil ketua pada Schematics 2015 .

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