TUGAS AKHIR – KI141502

ANALISIS KINERJA RADIUS TRANSMISI IEEE WLAN 802.11 PADA AD-HOC ON-DEMAND DISTANCE VECTOR (AODV) DI LINGKUNGAN MOBILE AD-HOC NETWORK (MANET)

DIAN ANNISSA'
NRP 05111340000070

Dosen Pembimbing I
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 – KI141502

ANALISIS KINERJA RADIUS TRANSMISI IEEE WLAN 802.11 PADA AD-HOC ON-DEMAND DISTANCE VECTOR (AODV) DI LINGKUNGAN MOBILE AD-HOC NETWORK (MANET)

DIAN ANNISSA'
NRP 05111340000070

Dosen Pembimbing I
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 – KI141502

TRANSMISSON RANGE ANALYSIS OF IEEE WLAN 802.11 WITH AD-HOC ON-DEMAND DISTANCE VECTOR (AODV) ON MOBILE AD-HOC NETWORK (MANET)

DIAN ANNISSA'
NRP 05111340000070

First Advisor
Dr.Eng. Radityo Anggoro, S.Kom, M.Sc.

Department of Informatics
Faculty of Information Technology and Communication
Sepuluh Nopember Institute of Technology
Surabaya 2019

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

ANALISIS KINERJA RADIUS TRANSMISI IEEE WLAN 802.11 PADA *AD-HOC ON-DEMAND DISTANCE VECTOR* (AODV) DI LINGKUNGAN *MOBILE AD-HOC NETWORK* (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 :
DIAN ANNISSA'
NRP : 05111340000070

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. .....
NIP: 198410162008121002 (pembimbing 1)

SURABAYA
JANUARI 2019

[Halaman ini sengaja dikosongkan]

ANALISIS KINERJA RADIUS TRANSMISI IEEE WLAN 802.11 PADA *AD-HOC ON-DEMAND DISTANCE VECTOR* (AODV) DI LINGKUNGAN *MOBILE AD-HOC NETWORK* (MANET)

Nama Mahasiswa : Dian Annissa'
NRP : 05111340000070
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
Dosen Pembimbing 2 : -

Abstrak

Mobile Ad Hoc Network (MANET) merupakan kumpulan beberapa *wireless node* yang dapat diatur secara dinamis tanpa menggunakan infrastruktur jaringan yang ada. Pada MANET, *mobile host* yang terhubung dengan *wireless* dapat bergerak bebas dan juga berperan sebagai *router*. Setiap *node* dalam MANET dapat bergerak secara bebas, sehingga jaringan dapat mengalami perubahan topologi dengan cepat. Selain itu *node* dalam MANET memiliki jarak transmisi yang terbatas, sehingga beberapa *node* tidak bisa berkomunikasi secara langsung dengan *node* lainnya.

Ada banyak *routing protocol* yang dapat diimplementasikan pada MANET, salah satunya adalah *Ad hoc On demand Distance Vector* (AODV). AODV merupakan salah satu *routing protocol* yang termasuk dalam klasifikasi *reactive routing protocol*, sebuah protokol yang hanya akan membuat rute ketika *node* sumber membutuhkannya. AODV memiliki dua fase, yaitu *route discovery* dan *route maintenance.*

Pada Tugas Akhir ini akan dilakukan analisis terhadap pengaruh radius transmisi yang menggunakan standar IEEE WLAN 802.11 terhadap *routing protocol* AODV yang akan di implementasikan dalam lingkungan MANET. Simulasi akan

dijalankan dengan menggunakan aplikasi *Network Simulator* 2 (NS-2) yang nantinya akan menghasilkan file output berupa *trace file* yang kemudian akan digunakan sebagai bahan analisis untuk menghitung *Packet Delivery Ratio*(PDR), *Routing Overhead*(RO), *dan End-to-End Delay*(E2D).

Kata kunci: MANET, AODV, NS2, IEEE, WLAN 802.11, Radius Transmisi.

TRANSMISSON RANGE ANALYSIS OF IEEE WLAN 802.11 WITH AD-HOC ON-DEMAND DISTANCE VECTOR (AODV) ON MOBILE AD-HOC NETWORK (MANET)

Student's Name	**: Dian Annissa'**
Student's ID	**: 05111340000070**
Department	**: Teknik Informatika FTIF-ITS**
First Advisor	**: Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.**
Second Advisor	**: -**

Abstract

Mobile Ad Hoc Network (MANET) is a collection of several wireless nodes that can be arranged dynamically without using existing network infrastructure. In MANET, the mobile host that is connected to wireless can move freely and also acts as a router. Each node in MANET can move freely, so the network can experience topological changes quickly. In addition, nodes in MANET have a limited transmission distance, so some nodes cannot communicate directly with other nodes.

There are many routing protocols that can be implemented on MANET, one of which is Ad hoc On demand Distance Vector (AODV). AODV is an example of reactive routing protocol classification, a protocol that will only create a route when the source node needs it. AODV have 2 phase which are route discovery and route maintenance.

In this final project, an analysis of the influence of the transmission radius that uses the IEEE WLAN 802.11 standard on the AODV routing protocol will be implemented in the MANET environment. The simulation will be carried out using the Network Simulator 2 (NS-2) application which will produce an output file in the form of a trace file which will then be used as analytical

material to calculate the Packet Delivery Ratio (PDR), Overhead Routing (RO), and End-to- End Delay (E2D).

Keyword: MANET, AODV, NS2, IEEE, WLAN 802.11, Transmission Range.

KATA PENGANTAR

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

"Analisis Kinerja Radius Transmisi IEEE WLAN 802.11 Pada *Ad-Hoc On-Demand Distance Vector* (AODV) di Lingkungan *Mobile Ad-Hoc Network* (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 semua rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Bapak Suhadi dan Ibu Yulistiani selaku kedua orangtua penulis serta Aufaa Hayyun Nissak selaku adik penulis atas segala doa, motivasi serta segala dukungan dan kepercayaan penuh kepada penulis sehingga penulis dapat menyelesaikan proses studi hingga menyelesaikan Tugas Akhir ini.
3. Keluarga besar yang selalu memberikan semangat dan dukungan untuk penulis selama selama pengerjaan Tugas Akhir ini.
4. Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc., selaku dosen pembimbing, atas arahan dan bantuannya dalam pengerjaan Tugas Akhir ini.

5. Tim Pemandu LKMM TM ITS 2017, Tim Pemandu Ekspresi yang selalu bisa menjadi sahabat, keluarga, pacar, partner terbaik, dan banyak hal lainnya yang sangat berarti untuk penulis selama menjalani tahun-tahun terakhir sebagai mahasiswa di ITS.
6. Yohana Deasy, Imagina Clara dan Umy Chasanah sebagai sahabat dan teman di angkatan yang selalu ada dan siap sedia membantu penulis dalam suka maupun duka, terutama di masa paling sulit penulis selama di Surabaya.
7. Sahabat Wisma Teratai, Intan, Firda dan Lutfi yang selama ini selalu bisa menjadi teman terbaik dan selalu membantu serta menyemangati penulis dengan kehebohan yang diciptakan di Wisma Teratai.
8. Teman-teman angkatan 2013, 2014, 2015, dan 2016 yang selalu memberikan dukungan untuk penulis dan membantu penulis menyelesaikan Tugas Akhir ini.
9. Teman-teman HMTC Bersahabat, HMTC Berkarya, HMTC Inspirasi dan terutama HMTC Optimasi yang telah memberikan pengalaman yang sangat berharga selama menjadi mahasiswa di Informatika ITS.
10. Teman-teman Pemandu ITS yang telah memberikan kepercayaan dan kesempatan luar biasa kepada penulis sehingga penulis dapat belajar memberikan kebermanfaatan selama menjadi mahasiswa di ITS.
11. Tim Futsal Putri FTIK yang sudah menjadi partner terbaik untuk penulis dalam menjalankan hobi.
12. Sahabat AJK yang sudah membantu dan menyediakan fasilitas selama penulis mengerjakan Tugas Akhir ini.
13. 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, Januari 2019

Dian Annissa’

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini perkembangan teknologi informasi dan komunikasi menjadi salah satu indikator kemajuan peradaban manusia. Kebutuhan manusia akan akses informasi dan komunikasi semakin meningkat setap harinya. Dan salah satu teknologi yang dapat membantu manusia dalam berkomunikasi dengan mudah adalah teknologi jaringan nirkabel (*wireless*). Teknologi nirkabel memungkinkan perangkat komunikasi untuk berkomunikasi secara langsung dengan perangkat lainnya tanpa adanya jaringan infrastruktur yang tetap dan dapat dilakukan dalam posisi bergerak. Teknologi jaringan nirkabel dapat membuat beberapa perangkat yang berada di satu area menjadi terkoneksi dan saling menjangkau serta sangat mungkin membuat sebuah jaringan yang bersifat sementara dan jaringan semacam ini disebut sebagai *Mobile Ad Hoc Network* (MANET)

Dalam jaringan MANET, setiap *node* bergerak secara bebas, sehingga jaringan dapat mengalami perubahan topologi dengan cepat. Setiap perangkat dalam MANET bebas bergerak secara independen ke segala arah, dan karena itu akan sering mengubah tautannya ke perangkat lain. Masing-masing harus meneruskan lalu lintas yang tidak terkait dengan penggunaannya sendiri, dan oleh karena itu menjadi router[1]. Terdapat beberapa metode *routing* pada jaringan MANET salah satunya *Ad-hoc On-Deman Distance Vector* (AODV). Protokol *routing* AODV bersifat reaktif yang artinya hanya melakukan proses pembentukan rute pada saat dibutuhkan. Setiap kali pencarian rute, AODV hanya menemukan satu pilihan rute atau *multipath*. Apabila terjadi kegagalan rute, maka AODV akan mengulang proses pencarian rute.

Topologi MANET sangatlah dinamis dan tidak dapat diprediksi maka tantangan utama dalam membangun MANET

adalah melengkapi setiap perangkat untuk terus menjaga informasi yang diperlukan untuk mengarahkan lalu lintas dengan benar. Dan salah satu faktor yang mempengaruhi jalannya informasi tersebut adalah radius transmisi. Maka dari itu disusunlah tugas akhir ini dalam rangka menganalisis pengaruh variasi radius pada transmisi IEEE WLAN 802.11 dalam protokol *routing AODV*. Berbagai implementasi pada lingkungan MANET dapat dilakukan dengan menggunakan simulasi sehingga sistem dapat dipelajari dengan baik untuk penelitian ini. Simulasi dilakukan dengan menggunakan *Network Simulator* 2 (NS-2). Implementasi ini akan dilakukan analisa kinerja dari IEEE 802.11 pada protokol routing AODV berdasarkan *Packet Delivery Ratio* (PDR), *End-to-End Delay* dan *Routing Overhead* (RO).

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah bagaimana pengaruh variasi radius pada transmisi IEEE WLAN 802.11 dapat mempengaruhi kinerja protokol AODV di lingkungan MANET?

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada Tugas Akhir ini memiliki batasan sebagai berikut:

1. Topologi jaringan yang digunakan adalah jaringan *Mobile Ad-hoc Network* (MANET).
2. *Routing Protocol* yang diujicobakan yaitu *Ad-hoc On-Demand Distance Vector* (AODV).
3. Simulasi pengujian jaringan menggunakan *Network Simulator 2* (NS-2).
4. MAC *protocol* yang akan dianalisis adalah IEEE 802.11.
5. Analisis Kinerja Didasarkan pada *Packet Dilivery Ratio* (PDR), *End-to-End Delay* dan *Routing Overhead* (RO).

1.4 Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah untuk menganalisa kinerja *Ad-hoc On-Demand Distance Vector* (AODV) pada transmisi IEEE WLAN 802.11 di lingkungan *Mobile Ad-Hoc Network* (MANET) dengan variasi radius transmisi yang beragam.

1.5 Manfaat

Manfaat yang diperoleh dari pengerjaan Tugas Akhir ini adalah sebagai acuan dalam penelitian yang berkaitan dengan faktor radius dalam transmisi terhadap performa routing protocol.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahapan awal dari Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir berisi pendahuluan, deskripsi dan gagasan metode-metode yang dibuat dalam Tugas Akhir ini. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir ini. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

1.6.2 Studi Literatur

Pada tahap ini, dipelajari sejumlah referensi yang diperlukan dalam melakukan implementasi yaitu mengenai AODV, NS-2, MANET, IEEE 802.11.

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 sistem 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 akan dilakukan uji coba terhadap sistem yang sudah dibuat. Pengujian dilakukan dengan pembuatan skenario *node-movement* yang menggunakan protokol *routing* AODV serta *Traffic-Connection Pattern Generation*. Kemudian simulasi tersebut dijalankan pada NS-2 *Network Simulator* dan akan menghasilkan *trace file*. Dari *trace file* tersebut akan dianalisis *1*. *Packet Dilivery Ratio* (PDR), *End-to-End Delay dan Routing Overhead* (RO) untuk mengetahui efek perubahan radius transmisi yang telah dimodifikasi.

1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
   Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.
2. Bab II. Tinjauan Pustaka
   Bab ini membahas teori oenunjang yang berhubungan dengan pokok pembahasan dan yang mendasari pembuatan Tugas Akhir ini.
3. Bab III. Perancangan
   Bab ini berisi pembahasan mengenai perancangan metode yang akan diimplementasikan dan dilakukan pengujian dari aplikasi yang akan dibangun.
4. Bab IV. Implementasi
   Bab ini menjelaskan implementasi rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan berupa implementasi skenario yang dirancang menggunakan *file node-movement* dan *traffic-pattern* yang sudah ada pada *network simulator*, konfigurasi sistem dan kode sumber analisis yang digunakan sebagai pengujian performa protokol *routing*.
5. Bab V. Pengujian dan Evaluasi
   Bab ini berisi hasil uji coba serta evaluasi dari implementasi skenario pada *routing protocol* AODV yang telah dilakukan untuk menyelesaikan masalah yang dibahas pada Tugas Akhir. Pengujian dilakukan dengan skenario yang dijalankan di NS-2 untuk mendapatkan data uji PDR, E2D serta RO.
6. Bab VI. Kesimpulan dan Saran
   Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan terhadap rumusan masalah Tugas Akhir serta saran untuk pengembangan selanjutnya.
7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

8. Lampiran
   Bagian ini mencantumkan kode sumber program secara keseluruhan.

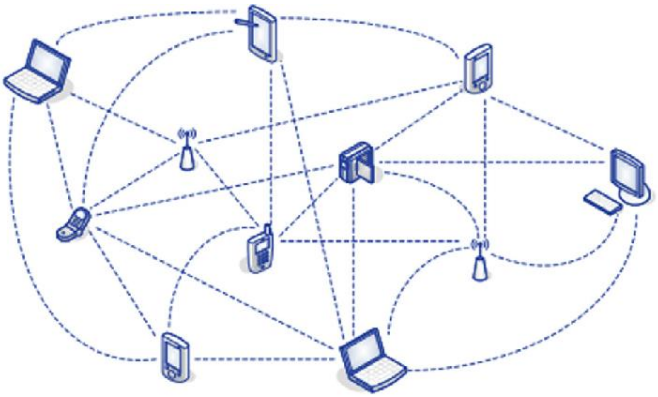
BAB II
TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai dasar teori atau penjelasan dari metode dan *tools* yang digunakan dalam Tugas Akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan riset yang berkaitan.

2.1 *Mobile Ad Hoc Network* (MANET)

Mobile Ad Hoc Network (MANET) merupakan kumpulan beberapa *wireless node* yang dapat diatur secara dinamis tanpa menggunakan infrastruktur jaringan yang ada. Pada MANET, *mobile host* yang terhubung dengan *wireless* dapat bergerak bebas dan juga berperan sebagai *router*.[2] Setiap perangkat pada MANET bergerak bebas secara independen ke segala arah, dan hal ini akan sering membuat perubahan *link* ke perangkat lainnya. Masing-masingnya harus meneruskan *traffic* yang tidak terkait dengan penggunaannya sendiri, dan oleh karena itu menjadikan hal tersebut sebagai *router*. Tantangan pertama dalam pembuatan MANET adalah melengkapi setiap *device* untuk terys menjaga informasi yang dibutuhkan untuk *route traffic* dengan benar. Beberapa jaringan dapat beroperasi sendiri atau harus terhubung pada internet yang lebih besar. Hal ini mungkin berisi satu atau beberapa dan *transceiver* yang berbeda antar *node* sehingga menghasilkan topologi yang otonom dan sangat dinamis.[1] Ilustrasi MANET dapat dilihat pada Gambar 2.1.

Tujuan utama dari *routing protocol* pada MANET adalah untuk memaksimalkan *throughput* jaringan, memaksimalkan efisiensi energi, memaksimalkan *lifetime* jaringan (*network lifetime*), dan meminimalkan *delay*. *Throughput* jaringan biasanya diukur dengan *packet delivery ratio* (PDR) sedangkan konsumsi energi diukur oleh besarnya *routing overhead* yang merupakan jumlah atau ukuran dari *routing control packets*.[3]



Gambar 2.1 Skema MANET

Pada MANET terdapat berbagai jenis protokol *routing* yang secara keseluruhan dapat dibagi menjadi beberapa kelompok, antara lain sebagai berikut:

1. *Proactive Routing*
   Tipe ini akan mengelola daftar tujuan dan rute terbaru masing-masing dengan cara mendistribusikan *routing table* ke seluruh jaringan, sehingga jalur lalu lintas (*traffic*) akan sering dilalui oleh *routing table* tersebut. Hal ini akan memperlambat aliran data jika terjadi restrukturisasi *routing table*. Beberapa contoh dari *proactive routing* adalah OLSR (*Optimized Link State Routing*) *protocol*, DSDV (*Highly Dinamic Destination Sequenced Distance Vector*) *routing protocol*, HSR (*Hierarchial State Routing*) *protocol, Babel* dan lainnya.
2. *Reactive Routing*
   Tipe ini akan mencari rute (*on demand*) dengan cara membanjiri jaringan dengan paket *router request*. Sehingga dapat menyebabkan jaringan akan penuh (*clogging*). Contoh *reactive routing* adalah *Ad Hoc on Demand Distance Vector* (AODV), *Dynamic Source Routing* (DSR), *Associativity Based Routing* dan lainnya.

3. *Flow Oriented Routing*
   Tipe ini mencari rute dengan mengikuti aliran yang disediakan. Salah satu pilihan adalah *unicast* secara terus-menerus ketika meneruskan data saat mempromosikan *link* baru. Beberapa kekurangan tipe protokol ini adalah membutuhkan waktu yang lama untuk mencari rute yang baru. Beberapa contohnya adalah *Signal Stability Routing* (SSR), *Interzone Routing Protocol* (IERP), dan *Lightweight Underlay Network Ad Hoc Routing* (LUNAR).
4. *Hybrid Routing*
   Tipe protokol ini menggabungkan antara *proactive routing* dengan *reactive routing*. Protokol yang termasuk pada tipe ini adalah *Hybrid Wireless Mesh Protocol* (HWMP), *Zone Routing Protocol* (ZRP) dan *Hybrid Routing Protocol for Large Scale* MANET (HRPLS).[2]

2.2 *Ad-hoc On demand Distance Vector* (AODV)

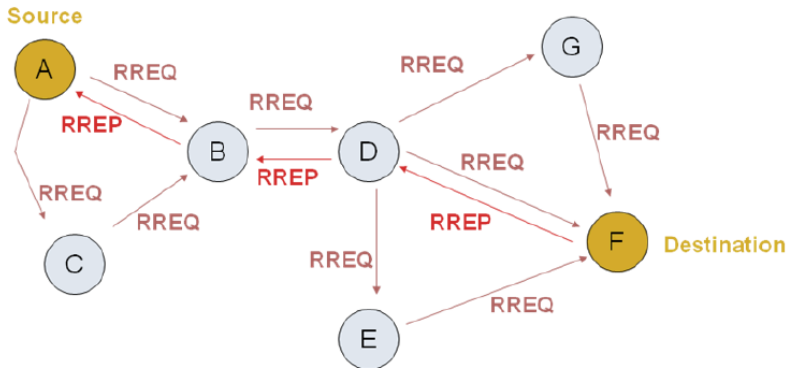
Ad Hoc on Demand Distance Vector (AODV) adalah protokol ruting yang dirancang untuk jaringan *ad hoc* nirkabel dan seluler. Protokol ini menetapkan rute ke tujuan sesuai permintaan dan mendukung *routing unicast* dan *multicast.* Protokol AODV dikembangkan oleh Nokia Research Center, Universitas California, Santa Barbara dan Universitas Cincinnati pada tahun 1991. [4]

Ad-hoc On demand Distance Vector (AODV) adalah salah satu *routing* protokol yang termasuk dalam klasifikasi *reactive routing protocol*. Sebuah protokol yang hanya me-*request* sebuah rute saat dibutuhkan. AODV dikembangkan oleh C. E. Perkins, E.M. Belding-Royer dan S. Das pada RFC 3561.

Ciri utama dari AODV adalah menjaga timer-based state pada setiap *node* sesuai dengan penggunaan tabel routing. Tabel routing akan kadaluarsa jika jarang digunakan. Ada dua tahapan dalam AODV yaitu route discovery dan route maintenance. Route discovery memiliki dua pesan yaitu berupa Route Request (RREQ) dan Route

Reply (RREP). Sedangkan Route maintenance berupa Route Error (RERR).

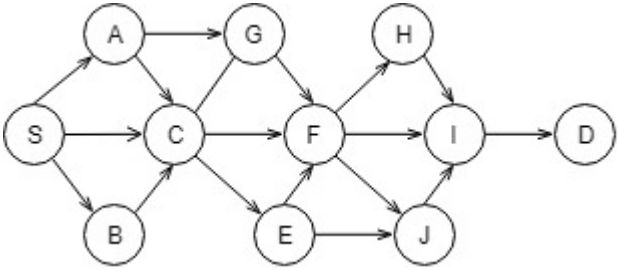
AODV adalah sebuah metode *routing* pesan antar *node* yang memungkinkan *node-node* tersebut untuk melewatkan pesan melalui lingkungannya ke *node* yang tidak dapat dihubungi secara langsung. AODV melakukan ini dengan cara menemukan rute yang bisa dilalui oleh pesan. Selain itu AODV juga memastikan rute ini tidak mengandung perulangan (*loop*), menangani perubahan rute, dan membuat rute baru apabila terjadi *error* [5]. Ilustrasi pencarian rute oleh AODV dapat dilihat pada **Error! Reference source not found.**



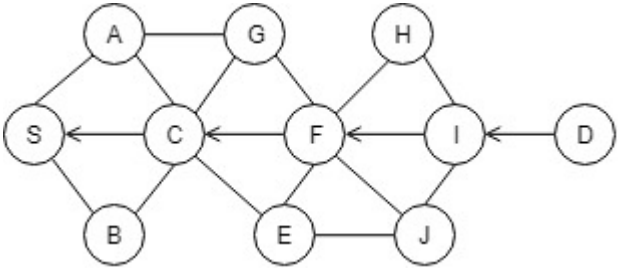
Gambar 2.2 *Routing discovery* pada AODV [5]

Didalam proses *route discovery*, *node* asal mem-*broadcast route request* (RREQ) *packet*s dimana disetakan nomor *sequence* tujuan. Ketika *node* tujuan atau *node* yang memiliki *route* ke tujuan menerima *packet* RREQ, *node* tersebut akan memeriksa nomor *sequence* tujuan yang sampai pada *node*-nya ketika *packet* tiba dan nomor *sequence* sama didalam RREQ. Untuk memastikan bahwa *packet* tersebut masih bersifat baru, *node* tujuan membalas paket RREQ dengan *route reply* (RREP) *packet*. RREP dibuat dan dikirim kembali ke *node* asal hanya jika nomor sequence tujuan sama dengan atau lebih besar dari yang dispesifikasikan di RREQ. AODV hanya menggunakan link yang bersifat simetris dan RREP mengikuti *path*

sebaliknya dari *path* yang dihasilkan oleh RREQ. Ketika menerima RREP, setiap *node* diantara asal dan tujuan memperbarui *routing table next-hop* dengan RREP ke tujuannya. *Node* asal kemudian memilih *route* dengan jumlah *hop* paling sedikit untuk mengirimkan *packet* tujuannya.[12] Untuk ilustrasi RREQ dan RREP pada AODV dapat dilihat masing-masing pada Gambar 2.3 dan 2.4.



Gambar 2.3 Mekanisme RREQ pada AODV



Gambar 2.4 Mekanisme RREP pada AODV

AODV memerlukan setiap *node* untuk menjaga tabel *routing* yang berisi *field*:

- *Destination IP Address*: berisi alamat IP dari *node* tujuan yang digunakan untuk menentukan rute.
- *Destination Sequence Number*: *destination sequence number* bekerjasama untuk menentukan rute.
- *Next Hop*: 'Loncatan' (*hop*) berikutnya, bisa berupa tujuan atau *node* tengah, *field* ini dirancang untuk meneruskan paket ke *node* tujuan.

- *Hop Count*: Jumlah *hop* dari alamat IP sumber sampai ke alamat IP tujuan.
- *Lifetime*: Waktu dalam milidetik yang digunakan untuk *node* menerima RREP.
- *Routing Flags*: Status sebuah rute yang berupa *down* (tidak valid), *up* (valid) atau sedang diperbaiki. [6]

2.3 *Network Simulator-2* (NS2)

Network Simulator (Versi 2), yang banyak dikenal dengan NS-2, adalah sebuah alat simulasi berbasis aktivitas yang berguna dalam mempelajari sifat dinamis jaringan komunikasi. Simulasi fungsi jaringan kabel, nirkabel, dan protokol dapat dilakukan dengan menggunakan NS-2. NS-2 menggunakan dua bahasa utama yaitu Bahasa C++ dan *Object-oriented Tool Command Language* (OTCL). Di NS-2 mendefinisikan mekanisme internal (*backend*) dari objek simulasi, dan OTCL mendefinisikan lingkungan simulasi eksternal (*frontend*) untuk perakitan dan konfigurasi objek. Setelah simulasi, NS-2 memberikan output simulasi baik dalam bentuk file NAM atau *trace file*. [7]

Ada beberapa keuntungan menggunakan NS sebagai perangkat lunak simulasi pembantu analisis dalam riset, antara lain adalah NS dilengkapi dengan *tool* validasi yang digunakan untuk menguji kebenaran pemodelan yang ada pada NS. Secara *default*, semua pemodelan NS akan dapat melewati proses validasi ini. Pemodelan media, protokol dan komponen jaringan yang lengkap dengan perilaku trafiknya sudah disediakan pada *library* NS.

NS juga bersifat open source dibawah *Gnu Public License* (GPL), sehingga NS dapat diunduh dan digunakan secara gratis melalui web site NS yaitu http://www.isi.edu/nsnam/. Sifat *open source* juga mengakibatkan pengembangan NS menjadi lebih dinamis.[6]

Pada Tugas Akhir ini digunakan NS-2 versi 2.35 sebagai aplikasi simulasi jaringan skenario MANET yang dihasilkan oleh program *default* dari NS-2 yaitu *generator file node-movement* dan

traffic-connection pattern menggunakan protokol AODV. NS-2 dijalankan pada sistem operasi Linux.

2.3.1 Instalasi

NS-2 membutuhkan beberapa *package* yang harus sudah ter*install* sebelum memulai instalasi NS-2. Untuk meng*install dependency* yang dibutuhkan dapat dilakukan dengan *command* yang ditunjukkan pada Gambar 2.5

```
sudo apt-get install build-essential autoconf
automake libxmu-dev
```

Gambar 0.5 Perintah untuk menginstall dependency NS-2

Setelah menginstall dependency yang dibutuhkan, ekstrak *package* NS-2 dan ubah baris kode ke-137 pada file ls.h di folder linkstate menjadi seperti pada Gambar **2.6** 2.6. Lalu *Install* NS-2 dengan menjalankan perintah ./install pada folder NS-2.

```
void eraseAll(){this->erase(baseMap::begin(),
baseMap::end()); }
```

Gambar 0.6 *Baris kode yang diubah pada file ls.h*

2.4 *Generator File Node-Movement*

Untuk menghasilkan *random movement* dari *node* dalam jaringan nirkabel digunakan *tools* yang disebut ‘setdest’ yang dikembangkan oleh CMU (*Carnegie Mellon University*). Kecepatan gerak yang spesifik menuju lokasi acak atau lokasi spesifik yang berada dalam kawasan yang telah ditentukan menghasilkan sebuah *node movement*. Ketika *node* tiba di lokasi pergerakan, *node* tersebut bisa diatur untuk berhenti sementara waktu. Selanjutnya *node* akan

terus bergerak menuju lokasi berikutnya. Lokasi ‘setdest’ berada pada direktori ‘`~ns/indep-utils.cmu-scen-gen/setdest/`’.

Pengguna harus menjalankan program ‘setdest’ sebelum menjalankan program simulasi. Format *command line* ‘setdest’ ditunjukan pada Gambar 2.7 dan keterangannya ditunjukkan pada Tabel 2.1.

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

Gambar 0.7 Format *Command Line* ‘setdest’

Tabel 2.1 Keterangan pada *Command Line* ‘setdest’

Parameter	Keterangan
-v version	Versi ‘setdest’ simulator yang digunakan.
-n num	Jumlah *node* dalam skenario.
-s speedtype	Tipe kecepatan *node* yang digunakan dalam simulasi
-m minspeed	Kecepatan minimum sebuah *node*.
-M maxspeed	Kecepatan maksimum sebuah *node*. *Node* akan bergerak pada kecepatan acak dalam rentang [0, maxspeed].
-t simtime	Waktu pengerjaan simulasi.
-P pausetype	Tipe waktu jeda yang dimiliki *node* dalam simulasi.
-p pausetime	Durasi ketika sebuah *node* tetap diam setelah tiba di lokasi pergerakan. Jika nilai ini diatur ke 0 maka *node* tidak akan berhenti ketika tiba di lokasi pergerakan dan akan terus bergerak.
-x maxx	Panjang maksimum area simulasi.
-y maxy	Lebar maksimum area simulasi.

File output dihasilkan oleh c*ommand line* 'setdest' yang berisi jumlah *node* dan mobilitas yang akan digunakan dalam *file* tcl selama simulasi. Selain mengandung skrip pergerakan *file output* juga mengandung beberapa statistik lain tentang perubahan *link* dan rute.[8]

Untuk membuat skenario *node movement* yang terdiri dari 50 *node*, bergerak dengan kecepatan maksimum 10 m/s dengan jeda rata-rata antar gerakan sebesar 2 detik, simulasi akan berhenti setelah 100 detik dengan batas topologi yang diartikan sebagai 500 x 500 meter^2, maka *command line*-nya seperti pada Gambar 2.8.

```
./setdest -v 2 -n 20 -s 1 -m 20 -M 60 -t 200 -P
1 -p 1 -x 500 -y 500 > scenario.txt
```

Gambar 0.8 Contoh *Command Line* 'setdest'

2.5 *File Traffic Connection Pattern*

Skrip *traffic-scenario pattern generator* digunakan sebagai konfigurasi antara mobilitas *node Random traffic connection* pada TCP dan CBR. Skrip Tcl yang disebut "cbrgrn" dapat menghasilkan alur *traffic* yang acak. Skrip ini membantu untuk menghasilkan *traffic load*. *Traffic load* dapat berupa TCP atau CBR. Skrip ini disimpan dalam file 'CMU-scen-gen' yang terletak dalam direktori ~ns/indep-utils/cmu-scen-gen. Program "cbrgen.tcl" digunakan sesuai dengan *command line* pada Gambar 2.9 dan dengan keterangan yang ditunjukkan pada tabel 2.2.

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

Gambar 0.9 Format *Command Line* cbrgen.tcl

Tabel 2.2 Keterangan *Command Line* cbrgen.tcl

Parameter	Keterangan
-type cbr|tcp	Jenis *traffic* yang digunakan berupa TCP atau CBR.
-nn *node*s	Jumlah total *node.*
-s seed	*Random seed.*
-mc connections	Jumlah koneksi.
-rate rate	Jumlah paket per detik. Pada CBR, panjang paket adalah tetap yaitu sebesar 512 bytes selama simulasi.

Command line pada Gambar 2.10 merupakan cara membuat sebuah *file* koneksi CBR antara 2 *node* yang memiliki maksimal 1 koneksi dengan nilai *seed* 1.0 dan jumlah paket per detik sebanyak 0.25.

```
ns cbrgen.tcl -type cbr -nn 20 -seed seed 1.0 -
mc 1 -rate 1.0 > cbr.txt
```

Gambar 0.10 Contoh *command line* cbrgen.tcl

2.6 AWK

AWK merupakan sebuah bahasa pemrograman yang didesain untuk *text-processing* dan biasanya digunakan sebagai alat ekstrasi data dan pelaporan. AWK bersifat *data-driven* yang berisikan kumpulan perintah yang akan dijalankan pada data tekstural baik secara langsung pada *file* atau digunakan sebagai bagian dari *pipeline*.

AWK memiliki beberapa fitur, antara lain:

- AWK menjadikan *text file* sebagai *records* dan *fields*
- Seperti bahasa pemrograman lainnya, AWK mengandung variabel, kondisi, dan *looping*
- AWK mempunyai operator aritmatika dan *string*
- AWK bisa di*generate* menjadi laporan yang berformat

- AWK membaca dari sebuah *file* atau dari input *standard*, dan menjadikannya output standard. AWK tidak bisa digunakan pada *file* yang tidak mengandung *text*.[9]

Pada pengerjaan Tugas Akhir ini AWK digunakan sebagai *tools* untuk membuat *script* dalam perhitungan *Packet Delivery Ratio* (PDR), *End-to-End Delay* (E2E) dan *Routing Overhead* (RO) dari hasil *trace* NS-2. Kode sumber PDR, E2E dan RO tercantum pada lampiran.

2.7 IEEE WLAN 802.11

IEEE WLAN 802.11 adalah serangkaian spesifikasi kendali akses medium dan lapisan fisik untuk mengimplementasikan komunikasi komputer wireless local area network di frekuensi 2.4, 3.6, 5 dan 60 GHz. Mereka diciptakan dan dioperasikan oleh Institute of Electrical and Electronics Engineers. Versi dasar dirilis tahun 1997 dan telah melalui serangkaian pembaruan dan menyediakan dasar bagi produk jaringan nirkabel Wi-Fi[10]. Terdapat beberapa jenis kode WLAN IEEE 802.11, yaitu 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, dan 802.11ad [11].

2.7.1 IEEE 802.11a

Dirilis pada Oktober 1999 yang merupakan standar wireless network dengan maksimum data transfer rate 54 Mbps dan bekerja pada frekuensi 5GHz dengan menggunakan metode transmisi Orthogonal Frequency Division Multiplexing (OFDM, yang mengizinkan pentransmisian data secara paralel di dalam sub-frekuensi (resisten terhadap interferensi dengan gelombang lain).

Range maksimal untuk indoor hanya sekitar 15 meter / ± 50 ft, sedangkan untuk outdoor ± 100 ft / 30 meter, dan standar 802.11a tidak kompatibel dengan 802.11 b,g.

2.7.2 IEEE 802.11b

Dirilis pada awal tahun 2000 yang merupakan standar wireless network dengan maksimum data transfer rate 5.5Mbps atau 11Mbps dan bekerja pada frekuensi 2,4GHz, dan dikenal juga dengan IEEE 802.11 HR. Namun setelah dilakukan uji coba lapangan kecepatan maksimum yang didapat hanya mencapai 5.9Mbps pada protokol TCP, dan 7.1Mbps pada protokol UDP.

Metode transmisi yang digunakannya adalah DSSS, dan memiliki range area yang lebih panjang yaitu 150 feet / 45 meters di dalam indoor dan 300 feet / 90 meter dalam outdoor.

2.7.3 IEEE 802.11g

Dirilis pada bulan Juni 2003 yang merupakan standar wireless network dengan maksimum data transfer rate 54 Mbps pada pita frekuensi 2,4 GHz yang hampir sama dengan 802.11b, tetapi metode transmisi yang digunakan adalah OFDM atau sama dengan metode yang digunakan oleh IEEE 802.11a. Range area 150 feet / 45 meter untuk indoor dan 300 feet / 90 meter untuk outdoor.

2.7.4 IEEE 802.11n

Dirilis pada 11 September 2009 yang merupakan standar wireless network dengan maksimum data transfer rate 600 Mbps pada pita frekuensi 2,4 GHz atau 5 GHz.Namun setelah dilakukan uji coba oleh Wi-Fi Alliance kecepatan maksimum yang di dapat hanya 450 Mbps dan bekerja pada frekuensi 2,4 GHz atau 5 GHz, sama seperti teknologi MIMO (multiple-input multiple-output).

Range maksimal untuk indoor 70 meter sedangkan outdoor bisa mencapai 250 meter, dan kemungkinan besar Wi-Fi 802.11n akan diaplikasikan di perangkat router dan adapter.

2.7.5 IEEE 802.11ac

Standar wireless memiliki throughput multi-stasiun minimal 1 gigabit per detik dan throughput single-link minimal 500 megabit per detik (500 Mbit/s).Dengan memperluas konsep antarmuka udara yang dipeluk oleh 802.11n = bandwidth RF yang lebih luas (hingga 160 MHz), lebih banyak aliran spasial MIMO (hingga delapan), pengguna multi-user MIMO downlink (hingga empat klien), dan high- Modulasi densitas (sampai 256-QAM).

2.7.6 IEEE 802.11ad

Dirilis pada 2012 oleh WiGig yang merupakan standar wireless network berkemampuan WiGig tri-band yang beroperasi pada pita 2,4, 5 dan 60 GHz dan mampu memberikan tingkat transfer data hingga 7 Gbit / s atau kira-kira secepat transmisi 8-band 802.11ac, dan lebih dari 11 kali lebih cepat dari pada tingkat 802.11n.

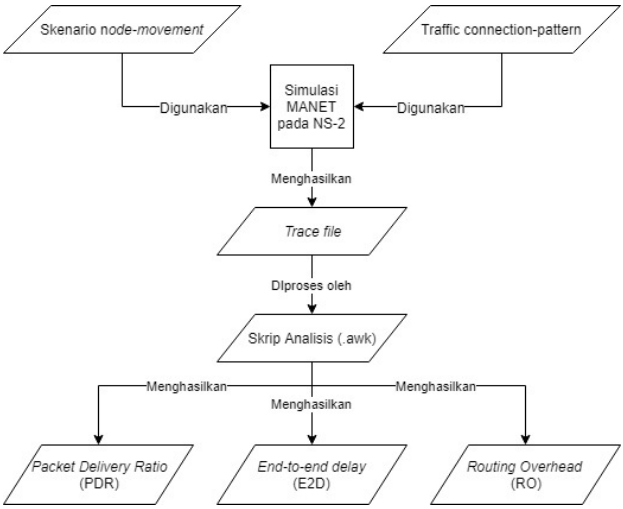
[Halaman ini sengaja dikosongkan]

BAB III
PERANCANGAN

Bab ini membahas mengenai perancangan implementasi sistem yang dibuat pada Tugas Akhir. Perancangan tersebut mencakup deskripsi umum, perancangan skenario, hingga alur dan implementasinya.

3.1 Deskripsi Umum

Pada Tugas Akhir ini akan dilakukan analisis radius transmisi IEEE WLAN 802.11 pada *routing protocol* AODV di lingkungan MANET. Dalam pembuatan skenario MANET digunakan *Mobility Generator* yang bersifat *Random Way Point* dan sudah tersedia pada *Network Simulator*-2 (NS-2) yaitu dengan cara men-*generate file node-movement (mobility generation)* dan membuat koneksi antar *node* menggunakan *file traffic-connection pattern*. Rancangan simulasi yang dibuat dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Rancangan Simulasi

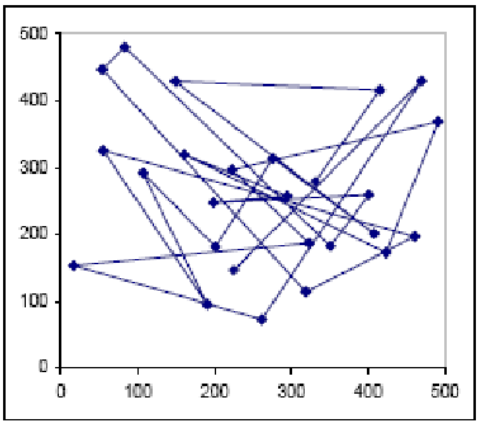
Dalam penelitian ini penulis akan menganalisis performa dari IEEE WLAN 802.11 dengan berberapa variasi jarak transmisi pada simulasi skenario yang dijalankan pada NS-2 menggunakan *routing protocol* AODV pada sistem operasi Linux. Hasil analisis tersebut dinilai berdasarkan *Packet Delivery Ratio* (PDR), *End-to-End Delay* (E2D) dan *Routing Overhead* (RO).

3.2 Perancangan Skenario

Perancangan skenario mobilitas pada Tugas Akhir ini diawali dengan melakukan pembuatan skenario yang bersifat *Random Waypoint* pada *mobility generation yang* kemudian membuat koneksi dengan menggunakan *file traffic-connection* yang sudah tersedia pada NS-2. Pada tugas akhir ini skenario dibuat untuk melihat pergerakan dari *node* yang sudah dibuat berdasarkan pada tiga jenis jumlah *node* yaitu 20, 50, 100 dan tiga variasi radius transmisi yaitu 150 m, 200 m, 250 m. Sedangkan untuk koneksinya digunakan dua *node* untuk menentukan *node* pengirim dan *node* penerima paket. Penjelasan perancangan skenario pada *mobility generator* dan pembuatan koneksi antar *node* adalah sebagai berikut:

3.2.1 *Node-Movement (Mobility Generation)* dalam MANET

Perancangan skenario *mobility generation* yang bersifat *random waypoint* dibuat dengan men-*generate file node-movement* yang telah ada pada NS-2 atau *tools*-nya biasa disebut ‘setdest’ yang nantinya akan menghasilkan *output* dalam bentuk .txt dan digunakan dalam *file* Tcl selama simulasi pada NS-2 sebagai bentuk pergerakan *node* yang berpindah-pindah.



Gambar 3.2 *Random Waypoint* dalam *Node-movement*

Pada Gambar 3.2 menunjukkan pola dari *Random Waypoint* dalam skenario *node-movement (mobility generation)* pada MANET dan parameter skenario yang diterapkan dalam *node-movement* tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Parameter Skenario *Node-Movement*

No.	Parameter	Spesifikasi
1.	Versi ‘setdest’	2
2.	Jumlah *Node*	20, 50, 100
3.	Waktu Simulasi	200 detik
4.	Area	500 m x 500 m
5.	Kecepatan Minimal	20 m/s
6.	Kecepatan Maksimal	60 m/s
7.	Sumber *Traffic*	CBR
8	Tipe jeda	1
9.	Waktu Jeda (dalam detik)	1
11.	Model mobilitas yang digunakan	*Random Waypoint*

3.2.2 *Traffic-Connection Pattern*

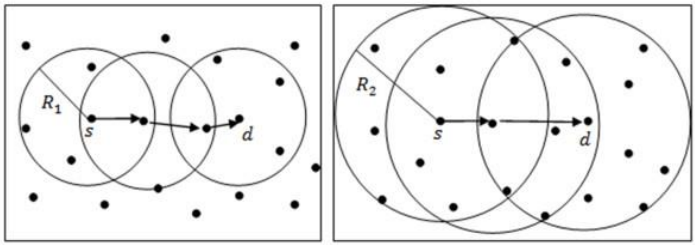
Traffic-Connection dihasilkan dengan menjalankan program cbrgen.tcl yang telah ada pada NS-2 yang nantinya akan menghasilkan *output* dalam bentuk .txt sehingga dapat digunakan sebagai koneksi untuk menghubungkan antar *node* yang ada pada skenario selama simulasi pada NS-2.

Tabel 3.2 Parameter *Traffic-Connection Pattern*

No.	Parameter	Spesifikasi
1.	-type cbr|tcp	CBR
2.	-nn *nodes*	20, 50, 100
3.	-s seed	1.0
4.	-mc connections	1
5.	-rate rate	1.0

3.3 Perancangan Simulasi pada NS-2

Pada perancangan kode NS-2 dengan konfigurasi MANET, dilakukan penggabungan skenario mobilitas dan skrip TCL yang berisi beberapa parameter untuk membangun percobaan simulasi. Salah satu parameter yang akan dikonfigurasi adalah radius transmisi yang memiliki default parameter WLAN IEEE 802.11.



Gambar 3.3 Efek Perubahan Radius Transmisi

Gambaran perancangan simulasi radius transmisi WLAN IEEE 802.11 dapat dilihat pada Gambar 3.3 yang menampilkan dua jenis ukuran radius transmisi yang berbeda serta ilustrasi dari efek perubahan yang dihasilkan karena radius transmisi yang memiliki perbedaan ukuran. Pengaturan simulasi rancangan sistem MANET yang digunakan pada NS-2 dapat dilihat pada Tabel 3.3.

Tabel 3.3 Parameter Lingkungan Simulasi pada NS-2

No.	Parameter	Spesifikasi
1.	*Network Simulator*	NS-2, 2.35
2.	*Routing Protocol*	AODV
3.	Waktu Simulasi	200 detik
4.	Area Simulasi	500 m x 500 m
5.	Banyak *Node*	20, 50, 100
6.	Radius Transmisi	150m, 200m, 250m
7.	Tipe Data	*Constant Bit Rate* (CBR)
8.	*Source/Destination*	Statis
9.	*Packet Rate*	512 bytes
12.	Protokol MAC	IEEE 802.11
13.	Mode Propagasi	*TwoRayGround*
14.	Tipe Antena	OmniAntenna
15.	Tipe Peta	MANET (*random way point*)
16.	Tipe Interface Queue	Droptail/PriQueue
17.	Tipe kanal	Wireless channel

3.4 Perancangan Metrik Analisis

Beberapa hal yang akan dianalisis pada Tugas akhir ini adalah sebagai berikut:

3.4.1 *Packet Delivery Ratio* (PDR)

Packet Delivery Ratio (PDR) merupakan perbandingan antara jumlah paket yang diterima dengan jumlah paket yang dikirimkan. PDR dihitung dengan persamaan 3.1.

$$PDR = \frac{received}{sent} x\ 100\ \% \qquad (3.1)$$

Keterangan:

PDR = *Packet Delivery Ratio*

received = banyak paket data yang diterima

sent = banyak paket data yang dikirimkan

PDR dapat menunjukkan keberhasilan paket yang dikirimkan. Semakin tinggi PDR, artinya semakin berhasil pengiriman paket yang dilakukan.

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

Rata-rata *End to End Delay* merupakan rata-rata dari *delay* atau waktu yang dibutuhkan tiap paket untuk sampai ke *node* tujuan dalam satuan detik. E2D dihitung dengan persamaan 3.2, dimana d*elay* tiap paket didapatkan dari rentang waktu antara *node* asal mengirimkan paket (`CBRSentTime`) dan *node* tujuan menerima paket (`CBRRecvTime`). Dari *delay* tiap paket tersebut semua dijumlahkan dan dibagi dengan jumlah paket yang berhasil diterima (`recvnum`).

$$E2E = \frac{\sum_{m=1}^{recvnum} CBRRecvTime - CBRSentTime}{recvnum} \quad (3.2)$$

Keterangan:

E2E = *End-to-End Delay*

CBRRecvTime = Waktu *node* asal mengirimkan paket

CBRSentTime = Waktu *node* tujuan menerima paket

recvnum = Jumlah paket yang berhasil diterima

3.4.3 *Routing Overhead* (RO)

Routing Overhead merupakan jumlah paket kontrol *routing* yang ditransmisikan ke *node* tujuan selama simulasi terjadi. *Routing Overhead* didapatkan berdasarkan jumlah semua paket kontrol *routing* yang ditransmisikan, baik itu paket *route request* (RREQ), *route reply* (RREP), maupun *route error* (RERR). Perhitungan *Routing Overhead* dapat dilihat dengan persamaan 3.3.

$$RO = \sum_{m=1}^{sent\ and\ forwarded\ num} packet\ sent \quad (3.3)$$

[Halaman ini sengaja dikosongkan]

BAB IV
IMPLEMENTASI

Pada bab ini akan membahas implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi meliputi lingkungan pembangunan perangkat lunak, implementasi skenario, implementasi simulasi pada NS-2 dan implementasi metrik analisis.

4.1 Lingkungan Implementasi Protokol

Pada sub bab ini menjelaskan tentang lingkungan implementasi protokol yang penjelasannya tercantum pada Tabel 4.1.

Tabel 4.1 Spesifikasi Lingkungan Implementasi

Perangkat Keras	Personal Komputer ACER ASPIRE M3970 - Prosesor Intel® Core™ i3-2120 CPU @ 3.30GHz x 4 - RAM 8 GB - HDD 500GB
Perangkat Lunak	Personal Komputer ACER ASPIRE M3970 - Sistem Operasi Ubuntu 16.04 - *Network Simulator*-2, 2.35 - Patch AODV - Microsoft Excel 2016

4.2 Implementasi Skenario

Pada sub bab ini implementasi skenario mobilitas MANET dipelajari dalam kondisi yang berbeda pada beban *traffic*-nya dan mobilitas/pergerakan *node*-nya. Pada studi simulasi jaringan MANET, digunakan dua model yaitu *mobility generation*, digunakan untuk mempelajari pengaruh mobilitas dari *node* pada kinerja keseluruhan jaringan serta *traffic-conection* yang digunakan untuk mempelajari pengaruh beban *traffic* pada jaringan. Implementasi skenarionya adalah sebagai berikut:

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

Implementasi skenario pada *mobility generation* menggunakan *tools generate default* yang dimiliki oleh NS-2 yaitu 'setdest'. Setiap simulasi digunakan *file* skenario *node movement (mobility generation)* yang ditandai dengan jeda waktu. Simulasi dilakukan dengan pola gerakan yang dihasilkan dari kecepatan maksimal yang berbeda yang bertujuan untuk mempelajari efek mobilitas. Pengaturan *file* skenario pergerakan *node* disesuaikan dengan mobilitas yang berbeda, sehingga dibuat dengan memvariasikan kecepatan maksimal. Ppada NS-2, 'setdest' digunakan untuk menghasilkan *file node-movement* dengan mengunakan algoritma *Random Way Point*. Format *command line* pada Gambar 4.1 yang digunakan untuk menghasilkan gerakan acak pada *node* adalah sebagai berikut:

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

Gambar 4.1 Format *Command Line* 'setdest'

File mobilitas yang dihasilkan pada simulasi disimpan dalam direktori " ~ ns/indep-utils/CMU-scen-cen/setdest". Gambar 4.2 merupakan implementasi *command line* pada 'setdest' dengan berbagai kecepatan maksimal yang berbeda sesuai dan *node* sebanyak 50. Serta untuk setiap kecepatan maksimal tersebut dibuat 10 buah *file* untuk satu protokol *routing* dan satu model propagasi.

```
./setdest -v 2 -n 20 –s 1 –m 20 -M 60 -t 200 –P
1 -p 1 -x 500 -y 500 > scenario.txt

./setdest -v 2 -n 50 –s 1 –m 20 -M 60 -t 200 –P
1 -p 1 -x 500 -y 500 > scenario.txt

./setdest -v 2 -n 100 –s 1 –m 20 -M 60 -t 200 –
P 1 -p 1 -x 500 -y 500 > scenario.txt
```

Gambar 4.2 Implementasi pada 'setdest'

4.2.2 *File Traffic-Connection Pattern Generation*

Untuk TCP dan CBR, implementasi *random traffic-connection* diatur dengan pergerakan antar *node* menggunakan skrip *traffic-scenario generator*. Skrip *traffic generator* terdapat pada direktori ~ns/indep-utils/cmu-scen-gen yang disimpan dalam bentuk *file* cbrgen.tcl. *File* ini dapat digunakan untuk membuat *traffic connection* CBR ataupun TCP pada jaringan pergerakan antar *node*. Pada menjalankan perintah pada *file traffic-connection* cbrgen.tcl, tipe *traffic connection*-nya (CBR atau TCP) harus dideklarasikan terlebih dahulu, kemudian koneksi maksimal dan banyaknya *node* yang ada pada jaringan tersebut, *random seed* dan misalkan pada koneksi CBR, *rate* yang nilai kebalikannya digunakan untuk menghitung rentang waktu antar paket CBR yang kemudian disimpan dalam sebuah *file traffic*. Format *command line* pada Gambar 4.3 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-
```

Gambar 4.3 Format *Command Line* cbrgen.tcl

Gambar 4.4 merupakan implementasi untuk menjalankan cbrgen.tcl untuk membuat *file* koneksi CBR diantara 20, 50, 100 *node* memiliki maksimal 1 koneksi dengan nilai *seed* 1.0 dan jumlah paket per detik sebanyak 1.0 yang disimpan dalam cbr.txt yang akan digunakan pada saat simulasi.

```
ns cbrgen.tcl -type cbr -nn 20 -seed 1.0 -mc 1
-rate 1.0 > cbr.txt

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

ns cbrgen.tcl -type cbr -nn 100 -seed 1.0 -mc 1
-rate 1.0 > cbr.txt
```

Gambar 4.4 Implementasi Koneksi cbrgen.tcl

Gambar 4.5 menunjukkan *output* yang disimpan dalam cbr.txt sehingga menghasilkan koneksi CBR dan menggunakan *Agent* UDP. Koneksi UDP disini merupakan konfigurasi antara *node* ke-1 dan ke-2. Pengiriman paket data dilakukan setiap satu detik dengan besar paket data 512byte dan maksimal pengiriman paket 10.000.

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1.0
$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"
```

Gambar 4.5 Output *cbr.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* ini berisi konfigurasi setiap *node* dan proses yang dilakukan selama simulasi berjalan. Potongan konfigurasi lingkungan simulasi dapat dilihat pada Gambar 4.6.

```
set val(chan)   Channel/WirelessChannel;
set val(prop)   Propagation/TwoRayGround;
set val(netif)  Phy/WirelessPhy;
set val(mac)    Mac/802_11;
set val(ifq)    Queue/Droptail/PriQueue;
set val(ll)     LL;
set val(ant)    Antenna/OmniAntenna;
set opt(x)      500;
set opt(y)      500;
set val(ifqlen) 50;
set val(nn)     20;
set val(seed)   1.0;
set val(rp)     AODV;
set val(stop)   200;
set val(cp)     "cbr.txt"
set val(sc)     "scen.txt"
```

Gambar 4.6 Implementasi Simulasi NS-2

Pada konfigurasi dilakukan pemanggilan terhadap *file traffic* yang berisikan konfigurasi *node* asal, *node* tujuan, pengiriman paket, serta *file* skenario yang berisi pergerakan *node* yang telah di*generate*. Kode implementasi pada NS-2 dapat dilihat pada lampiran A.1 Kode Skenario NS-2.

Konfigurasi untuk *file traffic* bisa dilakukan dengan membuat *file* berekstensi .txt untuk menyimpan konfigurasi tersebut. Pada *file* konfigurasi lingkungan simulasi, *file traffic* tersebut dimasukkan agar dibaca sebagai *file traffic*. Potongan konfigurasi *file traffic* dapat dilihat pada Gambar 4.7.

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(18) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(19) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1.0
$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"
```

***Ga*mbar 4.7 Implementasi Simulasi NS-2**

Pada konfigurasi tersebut, ditentukan *node* sumber dan *node* tujuan pengiriman paket. Pengiriman dimulai pada detik ke- 2.55. Implementasi konfigurasi file traffic untuk simulasi pada NS-2 dapat dilihat pada lampiran A.2 Kode Konfigurasi Traffic

Gambar 4.8 merupakan pengaturan dari *transmission range* yang digunakan pada simulasi. Nilai yang diubah adalah *RXThresh_* (*Receiver Sensitivity Threshold*). Nilai 2.81838e-09 pada variabel memiliki arti bahwa jangkauan atau *range* yang dapat dicapai adalah sejauh 150 meter, nilai 8.91754e-10 adalah 200 meter dan 3.65262e-10 adalah 250 meter.

```
Phy/WirelessPhy set RXThresh_ 2.81838e-09; #150m
Phy/WirelessPhy set RXThresh_ 8.91754e-10; #200m
Phy/WirelessPhy set RXThresh_ 3.65262e-10; #250m
```

Gambar 4.8 Konfigurasi *Transmission Range* pada NS-2

4.4 Implementasi Metrik Analisis

Simulasi yang telah dijalankan oleh NS-2 menghasilkan sebuah *trace file* yang berisikan data mengenai apa saja yang terjadi selama simulasi dalam bentuk *plain text* berekstensi .tr. Dari data *trace file* tersebut dapat dilakukan analisis performa *routing protocol* yang dijalankan. Pada Tugas Akhir ini, metrik yang akan dianalisis adalah *Packet Delivery Ratio* (PDR), Rata-rata *End-to-End Delay* (E2D), dan *Routing Overhead* (RO).

4.4.1 Implementasi *Packet Delivery Ratio* (PDR)

Packet Delivery Ratio (PDR) merupakan perbandingan antara jumlah paket data yang diterima oleh *destination node* dengan jumlah paket data yang dikirim oleh *source node*. Pada persamaan 3.1 telah dijelaskan bagaimana menghitung PDR. Skrip AWK untuk menghitung PDR berdasarkan kedua informasi tersebut dapat dilihat pada lampiran A.3 Kode Skrip AWK Packet Delivery Ratio. Cara menjalankan skrip AWK untuk menghitung PDR adalah `awk -f PDR.awk trace.tr`.

PDR didapatkan dengan cara menghitung setiap baris terjadinya event pengiriman dan penerimaan paket data yang dikirim melalui agen pada trace file. Skrip menyaring setiap baris yang mengandung string AGT karena kata kunci tersebut menunjukan event yang berhubungan dengan paket komunikasi data. Penghitungan dilakukan dengan menjumlahkan paket yang dikirimkan dan paket yang diterima dengan menggunakan karakter pada kolom pertama sebagai filter. Kolom pertama menunjukan event yang terjadi dari sebuah paket. Setelah itu nilai PDR dihitung dengan cara persamaan 3.1. Pseudocode untuk menghitung PDR dapat dilihat pada Gambar 4.9

```
sent = 0
received = 0
for i = 1 to the number of rows
     if in a row contains "s" and AGT then
          sent++
     else if in a row contains "r" and AGT then
          received++
     end if
pdr = received / sent
```

Gambar 4.9 Pseudocode untuk menghitung PDR

4.4.2 Implementasi Rata-Rata *End-to-End Delay*

End-to-End Delay merupakan waktu rata-rata yang dibutuhkan oleh paket saat dikirimkan oleh *source node* dengan penerimaan paket data oleh *destination node*. Perhitungan E2E telah dijelaskan melalui persamaan 3.2. Skrip AWK untuk menghitung E2E dapat dilihat pada lampiran A.4 Kode Skrip AWK Rata-Rata End-to-End Delay **Error! Reference source not found.**. Hasil dari E2D dinyatakan dalam *millisecond* (ms).

Pada Tugas Akhir ini, perhitungan *end-to-end delay* juga dilakukan saat sesi kedua seperti perhitungan *packet delivery ratio*. Contoh perintah pengeksekusian skrip awk untuk menganalisis *trace file* adalah `awk -f e2e.awk trace.tr`. *Pseudocode* untuk menghitung rata-rata E2E dapat dilihat pada Gambar 4.10.

```
sum_delay = 0
counter = 0

for i = 1 to the number of rows
     counter++
     if layer == AGT and event == s then
      start_time[packet_id] = time
     else if layer == AGT and event == r then
```

```
      end_time[packet_id] = time
     end if
     delay[packet_id] = end_time[packet_id] -
start_time[packet_id]
     sum_delay += delay[packet_id]
e2e = sum_delay / counter
```

Gambar 4.10 Pseudocode untuk perhitungan rata-rata E2E

4.4.3 Implementasi *Routing Overhead*

Routing Overhead merupakan perbandingan antara total jumlah paket *routing* yang dikirim dengan total jumlah paket data yang diterima. Perhitungan *routing overhead* telah dijelaskan pada persamaan 3.3. Skrip AWK untuk menghitung *routing overhead* dapat dilihat pada lampiran A.5 Kode Skrip AWK Routing Overhead.

Pada Tugas Akhir ini, perhitungan *routing overhead* juga dilakukan pada sesi kedua seperti perhitungan *packet delivery ratio* dan *end-to-end delay*. Contoh perintah untuk mengeksekusi skrip awk untuk menganalisis *trace file* adalah `awk -f ro.awk trace.tr`.

Pseudocode untuk perhitungan *routing overhed* dapat dilihat pada Gambar 4.11.

```
ro = 0
for i = 1 to the number of rows
     if in a row contains "s" and RTR then
        ro++
     end if
```

Gambar 4.11 *Pseudocode* untuk perhitungan *Routing Overhead*

[Halaman ini sengaja dikosongkan]

BAB V
UJI COBA DAN EVALUASI

Pada Bab ini akan dilakukan tahap ujicoba dan evaluasi sesuai dengan rancangan dan implementasi modul penentuan rute perjalanan aplikasi Clearroute. Dari hasil yang didapatkan setelah melakukan uji coba, akan dilakukan evaluasi sehingga dapat ditarik kesimpulan pada bab selanjutnya.

5.1 Lingkungan Pengujian

Uji coba dilakukan pada perangkat dengan spesifikasi seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat yang Digunakan

Komponen	**Spesifikasi**
CPU	Intel® Core™ i3-2120 CPU @ 3.30GHz x 4
Sistem Operasi	Xubuntu 16.04 LTS
Linux Kernel	Linux kernel 4.4
Memori	8 GB
Penyimpanan	HDD 500GB

Parameter lingkungan uji coba yang digunakan pada NS-2 dapat diihat pada X. Pengujian dilakukan dengan menjalankan skenario yang disimulasikan pada NS-2. Dari simulasi tersebut dihasilkan sebuah *trace file* dengan ekstensi .tr yang akan dianalisis dengan bantuan skrip awk untuk mendapatkan *Packet Delivery Ratio* (PDR), *End-to-End Delay* (E2D) dan *Routing Overhead* (RO).. Kriteria skenario pengujian terdapat pada Tabel 5.2. Setelah perancangan skenario, pengujian dilakukan sebanyak 10 kali dengan radius transmisi dan jumlah *node* yang berbeda sesuai kriteria untuk mendapatkan hasil analisis yang lebih baik.

Tabel 5.2 Lingkungan Uji Coba

No.	Parameter	Spesifikasi
1	Network simulator	NS-2.35
2	Routing protocol	AODV
3	Waktu simulasi	200 detik
4	Area simulasi	500 m x 500 m
5	Jumlah *Node*	20, 50, 100
6	Radius transmisi	150m, 200m, 250m
7	Kecepatan minimum	20 m/s
28	Kecepatan maksimum	60 m/s
9	Protokol MAC	IEEE 802.11
10	Model Propagasi	*Two-ray ground*

5.2 Hasil Uji Coba

Untuk hasil uji coba simulasi didapatkan *file* `trace.tr` yang berisi hasil *Packet Delivery Ratio* (PDR), *End-to-End Delay* dan *Routing Overhead* (RO). Hasil uji coba yang telah dilakukan sebanyak 10 kali akan dirata-ratakan dan didapatkan hasil yang tercantum pada sub bab berikut.

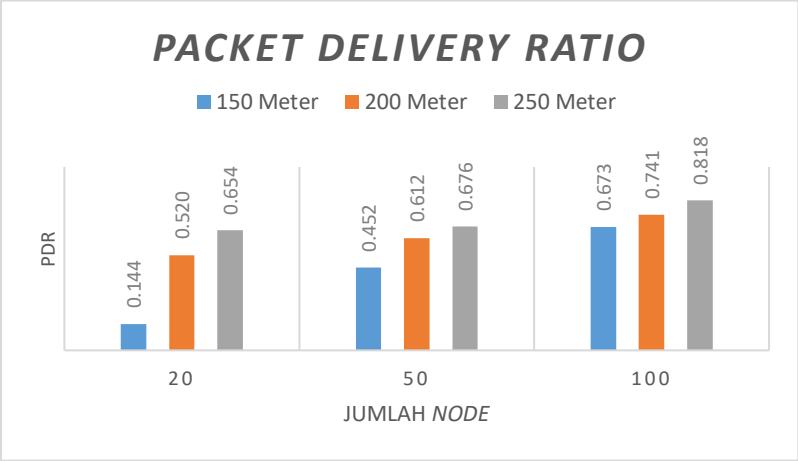
5.2.1 Analisis *Packet Delivery Ratio* (PDR)

Hasil *trace file* didapat dari menjalankan program skenario *node-movement (mobility generation)* yang menghasilkan nilai PDR. Nilai tersebut dianalisis melalui skrip pdr.awk. Hasil setiap perhitungan PDR skenario terdapat pada Tabel 5.3.

Tabel 5.3 PDR Skenario *Node-Movement*

Radius Transmisi (m)	Jumlah *Node*		
	20	**50**	**100**
150	0.144	0.452	0.673
200	0.520	0.612	0.741
250	0.654	0.676	0.818

Pada Tabel 5.3 dan Gambar 5.1 menunjukkan performa PDR yang dihasilkan oleh model propagasi *TwoRayGround* pada jaringan MANET yang bersifat *Random Waypoint* dengan menggunakan skenario *node-movement (mobility generation)*.



***Gambar 5.1 Grafik PDR Skenario* Node-Movement**

Nilai PDR merupakan nilai persentase kesuksesan data yang terkirim sehingga semakin besar nilai PDR maka semakin tinggi kesuksesan pengiriman data. Berdasarkan grafik pada **Error! Reference source not found.**, dapat dilihat bahwa dari semua variasi *node* yang diujikan, semakin besar radius transmisi, nilai PDR akan semakin tinggi. Selisih peningkatan nilai PDR paling besar

seiring dengan bertambahnya radius transmisi terjadi pada 20 *node* dengan rata-rata kenaikan 25,50 %, untuk 50 *node* memiliki rata-rata kenaikan 11,22%, dan untuk 100 *node* memiliki rata-rata kenaikan 7,27% atau yang paling rendah. Nilai rata-rata PDR untuk radius transmisi 150 meter adalah 0,423, sedangkan untuk nilai radius transmisi 200 meter adalah 0,624 dan untuk radius transmisi 250 meter adalah 0,716. Dari keseluruhan data diatas didapatkan nilai PDR terendah dicatatkan oleh 20 *node* dengan radius transmisi 150 meter, yaitu sebesar 0,144 dan nilai PDR tertinggi dicatatkan oleh 100 *node* dengan radius transmisi 250 meter 0,818.

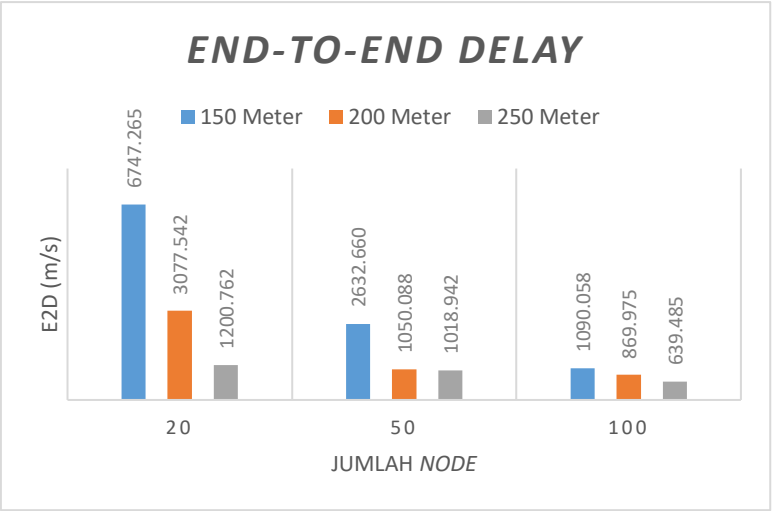
Berdasarkan hasil simulasi pada ketiga lingkungan tersebut dapat dilihat bahwa dengan semakin bertambahnya radius transmisi dan semakin bertambahnya jumlah *node* maka hasil dari PDR juga mengalami peningkatan, dengan rata-rata peningkatan sebesar 14.6% tiap selisih pertambahan radius transmisi sebesar 50 meter. Selain itu semakin sedikit jumlah *node* maka efek variasi radius transmisi akan terasa semakin signifikan terhadap perbedaan hasil PDR yang dihasilkan.

5.2.2 Analisis *End-to-End Delay* (E2D)

Hasil *trace file* didapat dari menjalankan program skenario *node-movement (mobility generation)* yang menghasilkan nilai *End-to-End Delay* dianalisis melalui skrip e2d.awk. Hasil tiap perhitungan E2D skenario ditabulasikan dan dirata-ratakan menjadi Tabel 5.4.

Tabel 5.4 E2D Skenario *Node-Movement*

Radius Transmisi (m)	Jumlah *Node*		
	20	**50**	**100**
150	6747.265	2632.660	1090.058
200	3077.542	1050.088	869.975
250	1200.762	1018.942	639.485



Gambar 5.2 Grafik E2D Skenario *Node-Movement*

Pada Tabel 5.4 dan grafik pada Gambar 5.2 dapat dilihat bahwa pengujian variasi radius transmisi untuk nilai *End-to-End Delay* mengalami penurunan jumlah waktu yang dibutuhkan untuk mengirimkan paket data.

Penurunan hasil E2D paling signifikan karena efek perubahan variasi radius transmisi terjadi pada uji coba 20 *node* dengan selisih rata-rata tiap variasi radius transmisi sebesar 57,69%, sedangkan untuk 50 *node* sebesar 31,54% dan 100 *node* sebesar 23,34%. Nilai rata-rata E2D untuk radius transmisi 150m sebesar 3489.994, kemudian untuk radius transmisi 200m mengalami penurunan sebanyak 47,73% menjadi 1665,868 dan untuk radius transmisi 250m memiliki rata-rata nilai E2D sebesar 953,063 atau menurun sebesar 57,21%. Dari keseluruhan data diatas didapatkan nilai E2D terendah dicatatkan oleh 100 *node* dengan radius transmisi 250 meter, yaitu sebesar 639,485 dan nilai E2D tertinggi dicatatkan oleh 20 *node* dengan radius transmisi 150 meter 6747,265.

Berdasarkan hasil simulasi pada ketiga lingkungan tersebut dapat dilihat bahwa dengan semakin bertambahnya radius transmisi

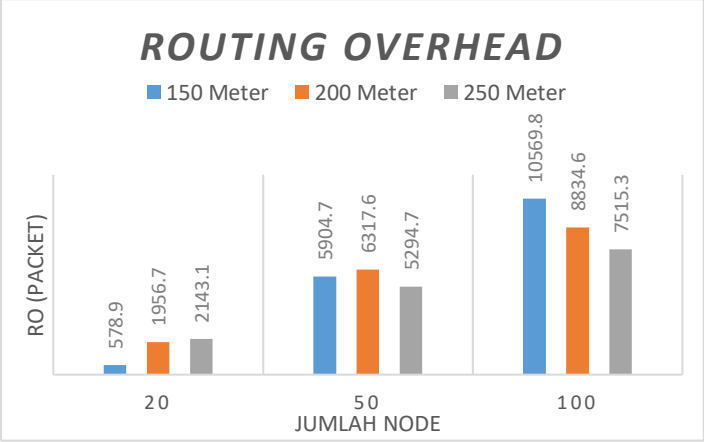
dan bertambahnya jumlah *node* maka hasil dari E2D mengalami penurunan, dengan rata-rata penurunan sebesar 52,47% tiap selisih pertambahan radius transmisi sebesar 50 meter. Selain itu semakin sedikit jumlah *node* maka efek variasi radius transmisi akan terasa semakin signifikan terhadap perbedaan hasil E2D yang dihasilkan.

5.2.3 Analisis *Routing Overhead* (RO)

Hasil *trace file* didapat dari menjalankan program skenario *node-movement (mobility generation)* yang menghasilkan nilai *Routing Overhead* (RO) kemudian dianalisis melalui skrip ro.awk. Hasil tiap perhitungan RO skenario ditabulasikan dan dirata-ratakan menjadi Tabel 5.5.

Tabel 5.5 RO Skenario *Node-Movement*

Radius Transmisi (m)	Jumlah *Node*		
	20	**50**	**100**
150	578.9	5904.7	10569.8
200	1956.7	6317.6	8834.6
250	2143.1	5294.7	7515.3



Gambar 5.3 Grafik RO Skenario *Node-Movement*

Pada Tabel 5.5 dan Gambar 5.3 menunjukkan pengujian variasi radius transmisi pada jaringan MANET untuk nilai *Routin Overhead* yang bersifat *Random Way Point* dengan menggunakan skenario *node-movement (mobility generation)*. Berdasarkan grafik pada Gambar 5.3, dapat dilihat bahwa grafik tersebut memiliki nilai yang fluktuatif. Untuk uji coba dengan 20 *node*, RO yang dihasilkan tiap variasi radius transmisi mengalami kenaikan seiring bertambahnya jarak radius transmisi itu sendiri. Pada 50 *node* menghasilkan nilai RO yang fluktuatif, mengalami peningkatan pada radius 150m namun mengalami penurunan pada radius 250m. Sedangkan untuk 100 *node*, RO yang dihasilkan megalami penurunan tiap bertambahnya radius transmisi. Jika nilai-nilai tersebut dirata-rata berdasarkan besar radius transmisi, untuk radius transmisi 100 m memiliki hasil RO 5684,5, sedangkan radius transmisi 200 m memiliki hasil 5703,0 dan radius 250 m memiliki hasil RO sebesar 4884,4 Grafik RO yang fluktuatif salah satunya terjadi karena pengaruh gerakan acak *node* yang memungkinkan terjadinya banyak rute putus saat pengiriman paket data ataupun kegagalan saat pembentukan *routing table* oleh AODV sehingga menyebabkan paket data yang dikirimkan tidak sampai tujuan.

[Halaman ini sengaja dikosongkan]

BAB VI
KESIMPULAN DAN SARAN

Pada Bab ini akan diberikan kesimpulan yang diperoleh dari Tugas Akhir yang telah dikerjakan dan saran tentang pengembangan dari Tugas Akhir ini yang dapat dilakukan di masa yang akan datang.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil uji coba dan evaluasi Tugas Akhir ini adalah sebagai berikut:

1. Perubahan konfigurasi radius transmisi pada *default* IEEE WLAN 802.11 membawa pengaruh yang signifikan pada kinerja *routing protocol* AODV di lingkungan MANET.
    a. Pengiriman paket menjadi lebih efisien seiring bertambahnya radius transmisi dan jumlah *node* dengan rata-rata peningkatan *Packet Delivery Ratio* (PDR) sebesar 14,66%.
    b. Terjadi penurunan *delay* pengiriman paket seiring bertambahnya radius transmisi dan jumlah *node* dengan rata-rata penurunan *End-to-End Delay* (E2D) sebesar 52,47%..
    c. Hasil dari *Routing Overhead* (RO) memiliki rata-rata yang fluktuatif dikarenakan gerak acak *node* yang di atur menggunakan algoritma *Random Waypoint* dalam *file node-movement* sehingga mempengaruhi kinerja AODV.
2. Dari hasil semua rangkaian simulasi diatas dapat disimpulkan bahwa semakin besar jarak radius transmisi WLAN IEEE 802.11 akan semakin baik pula kinerja *routing protokol* AODV di lingkungan MANET

6.2 Saran

Saran yang diberikan dari hasil uji coba dan evaluasi Tugas Akhir ini adalah sebagai berikut:

1. Untuk mendapatkan hasil uji coba yang lebih baik dan akurat, dapat dilakukan penambahan jumlah pengujian skenario.
2. Melakukan penambahan atau pengurangan jumlah *node* untuk skenario *node-movement mobility generation)*.
3. Melakukan modifikasi pada parameter-parameter lain yang berhubungan dengan simulasi AODV pada NS-2 agar mendapatkan hasil yang lebih beragam.
4. Dapat dilakukan perbandingan dengan *routing protocol* lainnya sehingga mendapatkan hasil yang variatif dan lebih baik.

DAFTAR PUSTAKA

[1] Zanjireh, M. M.; Shahrabi, A.; Larijani, H, "ANCH: A New Clustering Algorithm for Wireless Sensor Networks": 450–455, 2013.

[2] "3. Mobile Ad Hoc Network (MANET)," [Online]. Available: https://affandezone.wordpress.com/2011/10/30/3-routing-pada-manet/. [Diakses 15 Des 2018].

[3] W. Windianto, S. Djanali, dan M. Husni, "OPTIMASI ROUTING PADA PROTOKOL AODV_EXT DENGAN MENGGUNAKAN LINK EXPIRATION TIME (LET)," JUTI J. Ilm. Teknol. Inf., vol. 13, no. 2, hlm. 143, Jul 2015.

[4] "What is an Ad Hoc On-Demand Distance Vector (AODV)? - Definition from Techopedia," [Online]. Tersedia pada: https://www.techopedia.com/definition/2922/ad-hoc-on-demand-distance-vector-aodv. [Diakses 15 Des 2018]

[5] R. F. Sari dan A. Syarif, "Analisis Kinerja Protokol Routing Ad Hoc On-Demand Distance Vector (AODV) pada Jaringan Ad Hoc," 22 October 2010.

[6] R. F. Sari, A. Syarif, dan B. Budiardjo, "ANALISIS KINERJA PROTOKOL ROUTING AD HOC ON-DEMAND DISTANCE VECTOR (AODV) PADA JARINGAN AD HOC HYBRID: PERBANDINGAN HASIL SIMULASI DENGAN NS-2 DAN IMPLEMENTASI PADA TESTBED DENGAN PDA," MAKARA Technol. Ser., vol. 12, no. 1 Okt 2010.

[7] N. M. Mittal dan S. Choudhary, "Comparative Study of Simulators for Vehicular Ad-hoc Networks (VANETs)," vol. 4, no. 4, hlm. 10, 2014

[8] J. Macker <macker@itd.nrl.navy.mil>, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance

Issues and Evaluation Considerations." [Online]. Tersedia pada: https://tools.ietf.org/html/rfc2501. [Diakses 15 Des 2018]

[9] L. Jateng, "Definisi dan 6 Contoh Fungsi Perintah AWK di Linux Terminal," Kumpulan Program Aplikasi, Themes, Icon Linux Mint.

[10] "IEEE 802.11" [online] Tersedia pada : https://id.wikipedia.org/wiki/IEEE_802.11 [Diakses 15 Des 2018]

[11] "Perbedaan Wifi IEEE 802.11 b / g / n / a / ac / ad", [Online] Tersedia pada : https://itwae.blogspot.com/2018/01/perbedaan-wifi-ieee-80211-abgnacad.html [Diakses 15 Des 2018]

[12] "Routing Protocol (AODV, DSR, DSDV)," [Online]. Available: http://menulicious.student.telkomuniversity.ac.id/tag/routing-protocol/ [Diakses 15 Des 2018]

LAMPIRAN

1.1 A.1 Kode Skenario NS-2

```
set val(chan)   Channel/WirelessChannel;
set val(prop)   Propagation/TwoRayGround;
set val(netif)  Phy/WirelessPhy;
set val(mac)    Mac/802_11;
set val(ifq)    Queue/Droptail/PriQueue;
set val(ll)     LL;
set val(ant)    Antenna/OmniAntenna;
set opt(x)      500;
set opt(y)      500;
set val(ifqlen) 50;
set val(nn)     20;
set val(seed)   1.0;
set val(rp)     AODV;
set val(stop)   200;
set val(cp)     "cbr.txt"
set val(sc)     "scen.txt"

set val(sc)     "scen1modif.txt";

set ns_         [new Simulator]

# setup topography object

set topo   [new Topography]

# create trace object for ns and nam

set tracefd     [open scenario1.tr w]
set namtrace    [open scenario1.nam w]
```

```
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x)
$opt(y)

# Create God
set god_ [create-god $val(nn)]
#global node setting
$ns_ node-config -adhocRouting
$val(adhocRouting) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace ON \
                 -movementTrace ON \

# 802.11p default parameters
Phy/WirelessPhy set   RXThresh_ 2.81838e-09 ;
#150m
#Phy/WirelessPhy  set  RXThresh_  8.91754e-10;
#200m
#Phy/WirelessPhy  set  RXThresh_  3.65262e-10;
#250m
Phy/WirelessPhy set   CSThresh_ 5.57189e-11 ;
#400m
```

```
#  Create the specified number of nodes
[$val(nn)] and "attach" them
#  to the channel.
for {set i 0} {$i < $val(nn)} {incr i} {
           set node_($i) [$ns_ node]
           $node_($i) random-motion 0 ;#
disable random motion
}
# Define node movement model
puts "Loading connection pattern..."
source $val(cp)

# Define traffic model
puts "Loading scenario file..."
source $val(sc)

# Define node initial position in nam

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

    # 20 defines the node size in nam, must
adjust it according to your scenario
    # The function must be called after
mobility model is defined

    $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)  "stop"
$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(adhocRouting)"
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
```

1.2 A.2 Kode Konfigurasi *Traffic*

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(18) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(19) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1.0
$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
```

1.3 A.3 Kode Skrip AWK Packet Deliver Ratio

```
BEGIN {
        sendLine = 0;
        recvLine = 0;
        fowardLine = 0;
}

$0 ~/^s.* AGT/ {
        sendLine ++ ;
}

$0 ~/^r.* AGT/ {
        recvLine ++ ;
}

$0 ~/^f.* RTR/ {
        fowardLine ++ ;
}

END {
        printf "cbr s:%d r:%d, r/s Ratio:%.4f,
f:%d \n", sendLine, recvLine,
(recvLine/sendLine),fowardLine;
}
```

1.4 A.4 Kode Skrip AWK Rata-Rata End-to-End Delay

```
BEGIN{
      sum_delay = 0;
      count = 0;
}
{
      if ($2 >= 101) {

            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 \t= "
n_to_n_delay * 1000 " ms \n";
}
```

1.5 A.5 Kode Skrip AWK Routing Overhead

```
BEGIN {
      rt_pkts = 0;
}
{
            if (($1 == "s" || $1 == "f") &&
($4 == "RTR") && ($7 == "AODV")) {

                  rt_pkts++;
            }
      }
}
END {
      printf "Routing Packets \t= %d \n",
rt_pkts;
}
```

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Dian Annissa' atau yang kerap dipanggil Dian merupakan anak sulung dari 2 bersaudara yang lahir pada 3 Oktober 1995 di Kabupaten Blitar provinsi Jawa Timur. Penulis menempuh pendidikan sekolah dasar di SD Islam Kota Blitar (2001-2007) lalu melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 1 Blitar (2007-2010) dan penulis menempuh pendidikan menengah atas di SMA Negeri 1 Blitar (2010-2013). Selanjutnya penulis melanjutkan pendidikan sarjana di Departemen Informatika angkatan tahun 2013, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Selama Mahasiswa penulis juga terlibat dalam beberapa kepanitiaan dan organisasi di kampus. Mulai dari staff Badan Koordinasi Pemandu BEM Fakultas Teknologi Informasi, staff Kementrian Pengembangan Sumber Daya Mahasiswa BEM ITS, Sekretaris Departemen Kaderisasi dan Pemetaan Mahasiswa di Kabinet Optimasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTC). Penulis juga pernah tergabung dalam Tim Pemandu Latihan Keterampilan Mahasiswa (LKMM) Tingkat Pra-Tingkat Dasar dan Tingkat Dasar di Fakultas Teknologi Informasi serta Tim Pemandu LKMM Tingkat Menengah ITS tahun 2016 dan 2017.

Kritik dan saran sangat diharapkan guna meningkatkan kualitas dan penulisan selanjutnya. Penulis dapat dihubungi melalui nomor *handphone*: 085736232555 atau *email*: diananissa95@gmail.com.