Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – IF184802

# MODIFIKASI PEMILIHAN RUTE PROTOKOL ROUTING AODV BERDASARKAN TOTAL RESIDU ENERGI PADA NODE DI LINGKUNGAN VANETs

DICKY KAISAR UTOMO
NRP 05111540000077

Dosen Pembimbing I
Ir. F.X. Arunanto, M.Sc.

Dosen Pembimbing II
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Surabaya 2019

*(Halaman ini sengaja dikosongkan)*

**TUGAS AKHIR – IF184802**

# MODIFIKASI PEMILIHAN RUTE PROTOKOL ROUTING AODV BERDASARKAN TOTAL RESIDU ENERGI PADA NODE DI LINGKUNGAN VANETs

DICKY KAISAR UTOMO
NRP 05111540000077

Dosen Pembimbing I
Ir. F.X. Arunanto, M.Sc.

Dosen Pembimbing II
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Surabaya 2019

*(Halaman ini sengaja dikosongkan)*

UNDERGRADUATE THESIS – IF184802

# MODIFICATION OF ROUTE SELECTION IN AODV ROUTING PROTOCOL BASED ON TOTAL ENERGY RESIDUES NODE IN VANETs ENVIRONMENT

**DICKY KAISAR UTOMO**
**NRP 05111540000077**

**First Advisor**
**Ir. F.X. Arunanto, M.Sc.**

**Second Advisor**
**Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.**

**INFORMATICS DEPARTMENT**
**Faculty of Information Communication and Technology**
**Institut Teknologi Sepuluh Nopember**

**Surabaya 2019**

*(Halaman ini sengaja dikosongkan)*

## LEMBAR PENGESAHAN

# MODIFIKASI PEMILIHAN RUTE PROTOKOL ROUTING AODV BERDASARKAN TOTAL RESIDU ENERGI PADA NODE DI LINGKUNGAN VANETs

## TUGAS AKHIR

Diajukan untuk 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:
**DICKY KAISAR UTOMO**
**NRP: 05111540000077**

Disetujui oleh Pembimbing tugas akhir:

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

**SURABAYA**
**Juli, 2019**

*(Halaman ini sengaja dikosongkan)*

**MODIFIKASI PEMILIHAN RUTE PROTOKOL ROUTING AODV BERDASARKAN TOTAL RESIDU ENERGI PADA LINGKUNGAN VANETs**


| | | |
|---|---|---|
| **Nama Mahasiswa** | **:** | **Dicky Kaisar Utomo** |
| **NRP** | **:** | **05111540000077** |
| **Departemen** | **:** | **Informatika FTIK ITS** |
| **Dosen Pembimbing 1** | **:** | **Ir. F.X. Arunanto, M.Sc.** |
| **Dosen Pembimbing 2** | **:** | **Dr.Eng. Radityo Anggoro, S.Kom., M.Sc** |


## Abstrak


*Vehicular Ad Hoc Networks (VANETs)* adalah jaringan nirkabel *ad hoc* yang bersifat dinamis yang mengalami perubahan topologi secara berkala dan tidak memiliki infrastruktur tetap. VANETs memiliki beberapa algoritma *routing*, salah satunya adalah *Ad Hoc On Demand Distance Vector Routing (AODV).* AODV termasuk dalam algoritma *routing* reaktif, AODV merupakan algoritma *routing* untuk menentukan rute terbaik dalam proses pengiriman dan penerimaan paket data melalui *node* sumber menuju *node* tujuan.

Terdapat dua aktifitas utama pada *routing protocol* AODV yakni pencarian rute dan perawatan rute, dimana pada pencarian rute akan melibatkan *route request (*RREQ) dan *route reply* (RREP) dalam pengiriman paket ke *node* tujuan, sedangkan untuk proses perarawatan rute melibatkan *Hello Message*. Topik yang akan dibahas pada tugas akhir ini lebih mengarah pada proses pencarian rute AODV, dalam penelitian ini dilakukan modifikasi dengan tujuan untuk menentukan rute terbaik berdasarkan total residu energi tertinggi pada *node,* energi pada *node* perlu diperhatikan karna dengan berkurangnya jumlah residu energi pada node dapat meningkatkan besarnya kemungkinan kegagalan rute (*link failure*) yang mengakibatkan penurunan kinerja *routing protocol* AODV, hal

ini dilakukan melalui beberapa tahap yakni penerapan model energi, perhitungan residu energi, modifikasi routing protocol aodv dalam pencarian rute dan seleksi rute serta penerapan simulasi pada NS-2.

Hasil uji yang didapat dari implementasi menggunakan skenario simulasi NS-2 menunjukkan hasil yang signifikan yakni diantaranya *Packet Delivery Ratio* yang mengalami kenaikan sebesar 8.30% untuk skenario *grid*, 14.45% untuk skenario *real*. Penurunan rata-rata *Routing Overhead* sebesar 21.78% untuk skenario *grid*, sedangkan untuk skenario *real* sebesar 39.19%. Penurunan rata-rata *Average End-to-End Delay* sebesar 58.88% untuk skenario *grid*, sedangkan untuk skenario *real* sebesar 58.78% untuk skenario *real*. Penurunan rata-rata *Routing Overhead* sebesar 21.78% untuk skenario *grid*, sedangkan untuk skenario *real* sebesar 39.19%.

Dari hasil di atas, dapat diyakini bahwa modifikasi AODV ini dapat memengaruhi kinerja algoritma *routing* AODV pada lingkungan VANETs.

**Kata kunci : AODV, VANETs , NS-2.**

**MODIFICATION OF ROUTE SELECTION IN AODV ROUTING PROTOCOL BASED ON TOTAL ENERGY RESIDUES NODE IN VANETs ENVIRONMENT**

**Student's Name : Dicky Kaisar Utomo**
**Student's ID : 05111540000077**
**Department : Informatika FTIK-ITS**
**First Advisor : Ir. F.X. Arunanto, M.Sc.**
**Second Advisor : Dr.Eng. Radityo Anggoro, S.Kom, M.Sc.**

## Abstract

*Vehicular Ad Hoc Networks (VANETs) is a dynamic ad hoc wireless network that experiences peroiodic topological changes and does not have a fixed infrastructure.*

*VANETs has several routing algorithm, one of the routing algorithm in VANETs is Ad Hoc On Demand Distance Vector Routing (AODV), AODV is one of the routing algorithm that classify as reactive routing algorithm, AODV is routing algorithm that has main goal to determine the best route in the process of transmitting and delivery data packets through the source node to the destination node.*

*There are two main activities in routing protocol AODV, they are route discovery and route maintenance, which is route discovery involving route request (RREQ) and route reply (RREP) for package delivery to destiny node, while in route maintenance involving Hello Message. The topic that will be discussed in this thesis is about the process of route discovery AODV, In this research modifications were made in order to determine the best route based on the highest total residual energy in node, the energy in node need to be concerned because the lack of energy in node may caused the increase of possibility in link failure which is also caused the decrease of routing protocol AODV performance, this is done in several stages, they are implementation of energy model, calculation*

*of total residual energy, modifications of routing protocol AODV in route discovery and node selection, overall total residual energy in node and implementation of simulation in NS-2.*

*The results of the implementation using NS-2 simulation shows compromising result such as the increasing of Packet Delivery Ratio to 8.30% for grid scenario, while in real scenario was 14.45%. The decreasing of Delivery Delay to 58.88% for grid scenario, while in real scenario was 58.78%. the decreasing of Routing Overhead to 21.78% for grid scenario, while in real scenario was 39.19%.*

*From the results of the implementations it is believed that the modification of AODV is able to influence the performance of AODV routing algorithm in VANET.*

***Keyword: AODV, VANETs, NS-2***

## KATAPENGANTAR

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

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

**"MODIFIKASI PEMILIHAN RUTE PROTOKOL *ROUTING AODV* BERDASARKAN TOTAL RESIDU ENERGI PADA LINGKUNGAN VANETs".**

Harapan dari penulis, semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini dan ke depannya, 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. dan Nabi Muhammad SAW. yang telah membimbing penulis selama hidup.
2. Keluarga penulis (Ibu dan Mbak Yossy dan keluarga penulis yang lain) yang selalu memberikan dukungan baik berupa doa, moral, dan material yang tak terhingga kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini.
3. Bapak Ir. F.X. Arunanto, M.Sc. dan Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. selaku Dosen Pembimbing penulis yang telah membimbing, memberikan nasihat, dan memotivasi penulis sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Bapak Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. selaku kepala Departemen Informatika ITS.
5. Bapak dan Ibu Dosen yang telah memberikan ilmunya selama penulis berkuliah di Informatika ITS.

6. Sahabat Terbaik penulis DEMIGODS (R. Sidqi Tri Priwi dan Neny Lukitasari) yang selalu memberikan semangat secara tidak langsung kepada penulis sehingga penulis tidak pernah merasa bosan dan kesepian dikala penulisan proposal tugas akhir dan pembuatan buku tugas akhir ini.
7. Teman penulis yang berhati mulia Annisa Putri Diana yang senantiasa membantu penulis dengan mengajarkan banyak hal pada kuliah Semester 7 yang lalu.
8. Teman Penulis Hafara Firdausi yang bermurah hati karna sudah mengajari penulis banyak hal terkait Tugas Akhir ini.
9. Teman penulis yang baik hati Hania Maghfira yang kerap berbagi informasi dan memberikan saran serta kritik pada penulis terkait pengerjaan Proposal Tugas Akhir dan Penulisan Buku Tugas Akhir ini.
10. Teman Penulis Mas Aldo yang sudah mengajarkan banyak hal dalam rancang bangun program Tugas Akhir ini.
11. Teman teman HMTC optimasi, HMTC Inspirasi dan HMTC Kreasi yang memberikan wadah untuk berkembang dan belajar banyak hal selama berorganisasi.
12. Untuk orang-orang yang tidak dapat disebutkan satu persatu oleh penulis dan pembaca buku tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini. Namun, penulis memohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juli 2019

Dicky Kaisar Utomo

*(Halaman ini sengaja dikosongkan)*

## DAFTAR ISI

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

*(Halaman ini sengaja dikosongkan)*

# DAFTAR TABEL

*(Halaman ini sengaja dikosongkan)*

## KODE SUMBER

*(Halaman ini sengaja dikosongkan)*

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi saat ini sudah semakin maju, seiring dengan bertambahnya jaman, teknologi kini bukan hal yang abstrak seperti anggapan jaman dahulu, Teknologi yang ada saat ini dapat menyelesaikan permasalahan apapun. Salah satunya permasalahan komunikasi jaringan, teknologi yang ada saat ini memungkinkan manusia untuk tidak melakukan segala hal dengan manual. Manusia dapat mengurangi campur tangan terhadap penggunaan teknlogi atau biasa disebut *less human intervention*. Salah satu teknologi yang mendukung konsep tersebut adalah Vehicular Ad-hoc Networks (VANETs). VANETs digunakan untuk mempermudah komunikasi antar jaringan yang menghubungkan kendaraan satu dengan lainnya menggunakan jaringan nirkabel. Konsep *less human intervention* sesuai dengan VANETs dikarenakan VANETs merupakan teknologi jaringan yang cerdas serta bersifat otomatis, salah satu kegunaan VANETs antara lain adalah *self-driven car, traffic's accident prevention*, *navigation* dan lainnya. [1]

VANETs merupakan jaringan nirkabel atau tanpa kabel yang diterapkan terhadap kendaraan yang saling terhubung dan berkomunikasi satu sama lain, kemampuan teknologi VANETs dalam bebas bergerak menjadikan VANETs menjadi teknologi yang lebih unggul dibandingan jaringan LAN yang tidak memiliki kemampuan tersebut, VANETs juga tidak memerlukan infrastruktur jaringan atau server sebagai pusat dari seluruh administrasi jaringan. pada VANETs terdapat dua algoritma *routing*, yakni proaktif dan reaktif. *Routing* proaktif dalam pencarian rutenya akan mendistribusikan *routing table* keseluruh jaringan dengan memperbarui rute dan dan tujuan yang bersifat berkala atau periodik sehingga algoritma *routing* ini bersifat *table-driven*. *Routing* reaktif dalam pencarian rutenya akan bergantung pada permintaan yang

berarti pencarian rute hanya akan dilakukan ketika terdapat permintaan pengiriman data sehingga algoritma *routing* ini bersifat *on-demand*. Algoritma *routing* yang akan digunakan pada Tugas Akhir ini adalah algoritma *routing* reaktif, yaitu protokol *Ad Hoc On Demand Distance Vector Routing* (AODV).

Pada AODV seluruh informasi *routing* akan disimpan ditiap *node*, dimana setiap *node* akan menyimpan tabel *routing next-hop* yang berisi informasi tujuan dan rute pada *node* sehingga pencarian rute dapat dilakukan, yang nantinya rute terbaik akan dipilih berdasarkan jumlah hop yang paling sedikit. Sedangkan Pada tugas akhir ini menggunakan protokol *routing* AODV pada VANETs yang dimodifikasi untuk meningkatkan efisiensi energi yaitu dengan memilih rute terbaik bersarkan total residu energi pada *node*, energi pada *node* perlu diperhatikan karna dengan berkurangnya jumlah residu energi pada node dapat meningkatkan besarnya kemungkinan kegagalan rute (*link failure*) yang mengakibatkan penurunan kinerja *routing protocol* AODV. [2] terdapat tiga parameter yang harus diperhatikan yakni kenaikan Packet Delivery Ratio (PDR), Routing Overhead (RO) dan Delivery Delay.

## 1.2 Rumusan Masalah

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

1. Bagaimana melakukan pemilihan rute berdasarkan nilai total residu energi pada node dari seluruh rute yang ada?
2. Bagaimana perbandingan performa antara AODV dan modifikasi AODV berdasarkan Packet Delivery Ratio (PDR), Routing Overhead (RO) dan Average End-to-End Delay?

## 1.3 Batasan Permasalahan

Berdasarkan masalah yang diuraikan oleh penulis, maka batasan masalah pada tugas akhir ini adalah:

1. Jaringan yang digunakan adalah VANETs.
2. Routing protocol yang digunakan adalah Ad Hoc On Demand Distance Vector Routing (AODV).
3. Uji coba menggunakan Network Simulator 2 (NS-2).

## 1.4 Tujuan

Tujuan dari tugas akhir ini adalah untuk menentukan rute terbaik berdasarkan Total Residu Energi pada node.

## 1.5 Manfaat

Manfaat yang diperoleh dari pengerjaan Tugas Akhir ini adalah dapat memberikan informasi tentang Analisa performa protocol routing AODV berdasarkan Packet Delivery Ratio (PDR), Routing Overhead (RO) dan Delivery Delay dan untuk Menentukan rute terbaik berdasarkan Total Residu Energi pada node pada lingkungan VANETs.

## 1.6 Metodologi

Pembuatan tugas akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

### 1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi hal – hal terkait tugas akhir, diantaranya adalah Pendahuluan proposal tugas akhir yang mengandung hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah yang dari tugas akhir, tujuan pembuatan tugas akhir, dan manfaat dari hasil tugas akhir. Selain itu terdapat pula tinjauan pustaka yang menjadi referensi dalam pembuatan tugas akhir. Sedangkan Penjelasan mengenai tahapan penyusunan tugas akhir dijelaskan pada sub bab Metodelogi. Jadwal pengerjaan tugas akhir juga akan ditulis pada sub bab Jadwal Kegiatan.

### 1.6.2 Studi Literatur

Pada studi literatur, akan dilakukan pengumpulan informasi dan referensi yang digunakan dalam pengerjaan Tugas Akhir yaitu mengenai Vehicular Ad Hoc Network (VANETs), Ad Hoc On Demand Vector Routing (AODV), dan NS-2 Network Simulator.

### 1.6.3 Implementasi Sistem

Implementasi merupakan tahap untuk mengimplementasikan metode-metode yang sudah diajukan pada proposal tugas akhir. Untuk membangun algoritma yang telah dirancang sebelumnya, implementasi dilakukan dengan menggunakan NS-2 sebagai simulator jaringan, bahasa C/C++ sebagai bahasa pemrograman untuk uji coba mengimplementasikan metode yang sudah diajukan.

### 1.6.4 Pengujian dan Evaluasi

Pengujian dilakukan dengan menggunakan VANETs simulator untuk membuat simulasi keadaan topologi yang diujikan. Simulator yang digunakan adalah NS-2 dan akan menghasilkan keluaran berupa *trace file*. Dari *trace file* tersebut akan dihitung besar *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *Delivery Delay* (DD) untuk mengetahui performa kinerja *routing protocol* yang telah dimodifikasi.

### 1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku sebagai dokumentasi dari pelaksanaan tugas akhir yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

## 1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
   Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, batasan permasalahan, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan tugas akhir.
2. Bab II. Tinjauan Pustaka
   Bab ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan tugas akhir ini. Kajian teori yang dimaksud berisi tentang penjelasan singkat mengenai *Vehicular Ad Hoc Networks* (VANETs), Ad Hoc On Demand Distance Vector Routing (AODV), *Network Simulator* 2 (NS-2), dan AWK.
3. Bab III. Perancangan
   Bab ini berisi pembahasan mengenai perancangan skenario yang akan diimplementasikan dalam tugas akhir. Perancangan skenario berupa perancangan skenario penentuan rute terbaik menggunakan protocol AODV berdasarkan total residu energi, perancangan penghitungan total residu energi, perancangan simulasi pada NS-2, dan perancangan metrik analisis yang terdiri dari *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *Delivery Delay* (DD).
4. Bab IV. Implementasi
   Bab ini menjelaskan implementasi yang berbentuk kode sumber dari proses modifikasi protokol AODV, melakukan simulasi menggunakan NS-2, dan penghitungan metrik analisis.
5. Bab V. Pengujian dan Evaluasi
   Bab ini berisi hasil uji coba dan evaluasi dari implementasi modifikasi pada *routing protocol* AODV yang telah dilakukan untuk menyelesaikan masalah yang dibahas pada tugas akhir. Pengujian dilakukan dengan skenario yang telah dirancang dan dijalankan di NS-2, yang selanjutnya akan didapatkan hasil untuk dianalisis menggunakan skrip AWK yang nantinya akan menghasilkan metrik analisis berupa PDR, RO, dan DD.

6. Bab VI. Kesimpulan dan Saran
   Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan tugas akhir, dan saran untuk pengembangan tugas akhir ke depannya.
7. Bab VII. Daftar Pustaka
   Bab ini berisi daftar pustaka yang dijadikan literatur dalam tugas akhir.
8. Lampiran
   Dalam lampiran terdapat kode sumber program secara keseluruhan.

*(Halaman ini sengaja dikosongkan)*

# BAB II
# TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar atau penjelasan dari metode dan alat 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 VANETs

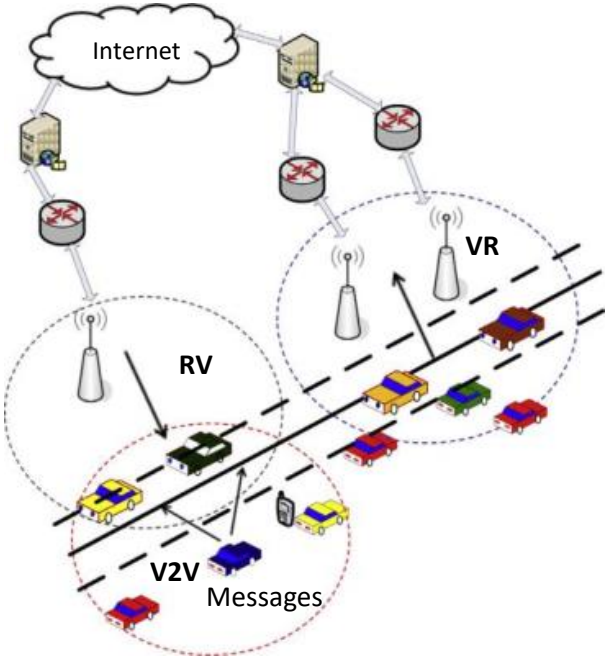
*Vehicular Ad Hoc Networks* (VANETs) adalah jaringan nirkabel *Ad Hoc* yang memiliki konfigurasi pada lapisan data *link* jaringan *Ad Hoc*, namun tidak memiliki infrastruktur tetap dikarenakan VANETs bersifat dinamis yang berarti VANETs dapat bebas secara acak bergerak kemanapun dan mengalami perubahan topologi secara berkala. Setiap *node* jaringan yang ada berperan sebagai router sebagaimana *node* tersebut melakukan *forward* pada *node* jaringan tertentu dan bertanggung jawab secara penuh dalam proses pengiriman dan penerimaan paket data, VANETs kerap digunakan untuk komunikasi antar kendaraan dengan sistem otomatisasi yang canggih seperti penerapan VANETs pada kendaraan listrik yang sumber dayanya berasal dari pasokan listrik yang disimpan ke dalam baterai atau tabung elektrik penyimpan lainnya dimana kendaraan listrik tersebut sudah dilengkapi fitur otomatisasi kendaraan yakni *Intelligent Transport System*, *Inter Vehicle Communication*, *Vehicle to Roadside Communication*. [3]

Teknologi VANETs jika dibandingkan dengan Teknologi MANETs memiliki perbedaan mendasar pada penerapannya, pada MANETs diterapkan pada perangkat bergerak seperti *smartphone, laptop, personal digital assistant* (PDA) dan sebagainya, sedangkan untuk VANETs penerapannya pada kendaraan listrik yang sudah memiliki sistem otomatisasi canggih

Algoritma routing pada VANETs dibagi menjadi dua, yakni

protokol *routing* proaktif dan reaktif. *Node* pada protokol routing proaktif bekerja secara terus menerus mencari rute untuk sampai ke destinasi sehingga tabel *routing* akan selalu melakukan *update*. Pada Algoritma protokol *routing* proaktif terdapat beberapa kategori yakni *Destination-Sequenced Distance-Vector Routing* (DSDV), dan *Optimized Link State Routing Protocol* (OLSR). sedangkan Pada protokol *routing* reaktif terdapat beberapa kategori antara lain *Ad Hoc On Demand Distance Vector Routing* (AODV), Dynamic Source Routing (DSR). Protocol *routing* reaktif melakukan pencarian rute pada node dengan syarat ketika ada permintaan pengiriman paket data.

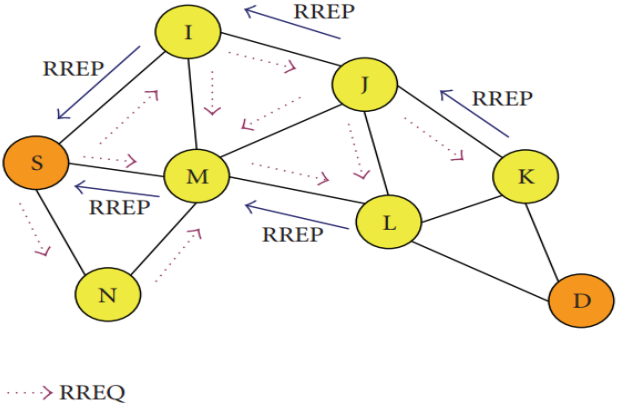
Ilustrasi pada VANETs dapat diamati pada **Gambar 2.1**, algoritma routing aodv akan dimodifikasi sehingga sesuai dengan hasil yang diharapkan yang akan diujikan pada lingkungan VANETs.




**Gambar 2.1** Ilustrasi pada VANETs *[4]*

## 2.2 *Ad Hoc On Demand Distance Vector Routing* (AODV)

*Ad Hoc On Demand Distanse Vector Routing (AODV)* adalah salah satu protokol routing reaktif, dimana protokol AODV digunakan untuk menemukan rute dari node sumber menuju node tujuan. AODV memiliki routing table yang dapat membedakan rute yang sudah dilewati dan yang belum dilewati. Dua fase utama pada AODV adalah pencarian rute dan perawatan rute, selama pencarian rute node sumber akan mengirimkan paket *Route Request* (RREQ) hingga sampai ke node tujuan, paket RREQ yang diterima akan di kirim ke tetangga node jika node yang menerima paket itu bukan node tujuan. Ketika telah sampai pada node tujuan, node tujuan akan mengirimkan paket *Route Reply* (RREP) ke node sumber melewati rute yang telah dilewati. Pada fase perawatan, rute HELLO packet digunakan untuk menjaga konektivitas antara node dan ketika terjadi kegagalan link akan dikirim pesan *Route Error* (RRER) ke node sumber. Ilustrasi pencarian rute dapat dilihat pada **Gambar 2.2**.




**Gambar 2.2** Ilustrasi pencarian rute *[8]*

## 2.3 *Network Simulator 2* (NS-2)

Network simulator 2 (NS-2) adalah alat simulasi untuk merepresentasikan simulasi jaringan, pada tugas akhir ini NS-2 dimanfaatkan untuk pemodelan dan pengujian VANETs pada protokol routing AODV.

*Network simulator* terdapat beberapa versi, namun yang saat ini digunakan untuk tugas akhir ini adalah *Network Simulator* 2 (NS-2). NS-2 menerapkan dua bahasa utama, yakni C++ dan *Object-oriented Tool Command Language* (OTCL). Bahasa C++ digunakan untuk menggambarkan mekanisme internal (*backend*) pada objek simulasi, sedangkan OTCL digunakan untuk menggambarkan lingkungan simulasi eksternal (*frontend*) pada perakitan dan konfigurasi objek simulasi.[4]

Pada tugas akhir ini digunakan NS-2 versi 2.35 untuk melakukan simulasi protocol routing aodv pada lingkungan VANETs yang sudah dimodifikasi. Sebagai output yang dihasilkan dari simulasi ini berupa *Trace file* yang dihasilkan oleh NS-2 yang nantinya juga digunakan sebagai pengukur performansi dari *routing protocol* AODV yang sudah dimodifikasi.

### 2.3.1 Instalasi

Sebelum melakukan instalasi NS-2, pastikan terlebih dahulu bahwa *compiler* yang terpasang sudah sesuai dengan standar yang digunakan oleh NS-2, yaitu gcc-4.8. Untuk memasang *compiler* tersebut dapat dilakukan dengan perintah sebagai berikut:

```
sudo apt install gcc-4.8 g++-4.8
```

***Kode Sumber 2.3.1*** *instalasi compiler*

NS-2 membutuhkan beberapa *package* yang harus sudah terpasang pada komputer sebelum memulai instalasi NS-2. Untuk memasang *dependency* yang dibutuhkan dapat dilakukan dengan perintah sebagai berikut :

```
sudo apt update

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

***Kode Sumber 2.3.2*** *instalasi depedency*

Kemudian, unduh NS-2 versi 2.35 yang dapat dilakukan melalui tautan berikut :

```
https://sourceforge.net/projects/nsnam/fi
les/allinone/ns-allinone-2.35/ns-
allinone-2.35.tar.gz/download
```

***Kode Sumber 2.3.3*** *pengunduhan NS-2*

lalu ekstrak berkas unduhan NS-2 pada direktori yang diinginkan. Selanjutnya, masuk ke dalam direktori ns-allinone-2.35/ns-2.35, lalu ubah baris kode ke-137 pada *file* ls.h di folder linkstate menjadi seperti perintah berikut :

```
gedit Makefile.in
```
ubah baris kode ke-36 dan 37 menjadi:
```
CC    = gcc-4.8
CPP   = g++-4.8

gedit linkstate/ls.h
```
ubah baris kode ke-137 menjadi:
```
void eraseAll(){this->erase
(baseMap::begin(),baseMap::end());}
```

***Kode Sumber 2.3.4*** *pengeditan baris kode*

Setelah mengubah baris kode, kembali ke direktori ns-allinone-2.35 dan instal NS-2 dengan menjalankan perintah berikut:

```
./install
```

***Kode Sumber 2.3.5*** *instalasi NS-2*

Kemudian, *environment* sistem pada NS-2 diatur agar NS-2 dapat dijalankan. Pengaturan tersebut dapat dilakukan melalui perintah sebagai berikut:

```
sudo gedit ~/.bashrc
```

Copy skrip berikut diakhir file:

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/dicky/ns-allinone-
2.35/otcl-1.14
NS2_LIB=/home/dicky/ns-allinone-
2.35/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL
_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/home/dicky/ns-allinone-
2.35/tcl8.5.10/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/home/dicky/ns-allinone-
2.35/bin:/home/dicky/ns-allinone-
2.35/tcl8.5.10/unix:/home/dicky/ns-
allinone-2.35/tk8.5.10/unix
NS=/home/dicky/ns-allinone-2.35/ns-
2.35/
NAM=/home/dicky/ns-allinone-2.35/nam-
1.15/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Lalu masukkan perintah source ~/.bashrc dan ns pada terminal.

### 2.3.2 *Trace File*

*Trace file* merupakan *output* simulasi NS-2 yang berisi informasi detail terkait pengiriman paket data. *Trace file* digunakan untuk menganalisa performa *routing protocol* yang disimulasikan. Penjelasan mengenai *trace file* ditunjukkan pada **Tabel 2.1** .

**Tabel 2.1** Penjelasan *Trace File*

| Kolom ke- | Penjelasan | Isi |
|---|---|---|
| 1 | *Event* | s: *sent*<br>r: *received*<br>f: *forwarded*<br>D: *dropped* |
| 2 | *Time* | Waktu terjadinya *event* |
| 3 | ID *Node* | _x_: dari 0 hingga banyak *node* pada topologi |
| 4 | *Layer* | AGT: *application*<br>RTR: *routing*<br>LL: *link layer*<br>IFQ: *packet queue*<br>MAC: MAC<br>PHY: *physical* |
| 5 | *Flag* | ---: Tidak ada |
| 6 | *Sequence Number* | Nomor paket |
| 7 | Tipe Paket | AODV : paket *routing* AODV<br>cbr : berkas paket CBR (*Constant Bit Rate*)<br>RTS : *Request To Send* yang dihasilkan MAC 802.11<br>CTS : *Clear To Send* yang dihasilkan MAC 802.11<br>ACK : MAC ACK |

| Kolom ke- | Penjelasan | Isi |
|---|---|---|
| | | ARP : Paket *link layer address resolution protocol* ARP: Paket *link layer Address Resolution Protocol* |
| 8 | Ukuran | Ukuran paket pada *layer* saat itu |
| 9 | Detail MAC | [a b c d]<br>a: perkiraan waktu paket<br>b: alamat penerima<br>c: alamat asal<br>d: IP *header* |
| 10 | *Flag* | ------: Tidak ada |
| 11 | Detail IP *source*, *destination*, dan *nexthop* | [a:b c:d e f]<br>a: IP *source node*<br>b: *port source node*<br>c: IP *destination node* (jika -1 berarti *broadcast*)<br>d: *port destination node*<br>e: IP *header* ttl<br>f: IP *nexthop* (jika 0 berarti *node* 0 atau *broadcast)* |

## 2.4 AWK

AWK adalah sebuah program filter untuk teks berupa bahasa pemrograman pada *shell* atau C yang memiliki karakteristik sebagai alat untuk melakukan filter data. AWK bersifat *data-driven* yang berisikan kumpulan perintah yang dijalankan pada *file*. [6] Secara umum AWK dapat digunakan untuk mengelola *database* sederhana, membuat laporan, memvalidasi data, dan membuat algoritma pengubah bahasa komputer ke bahasa lainnya. Dengan kata lain, AWK menyediakan fasilitas yang dapat memecah bagian data untuk proses berikuttnya. Pada tugas akhir ini, AWK digunakan untuk membuat skrip perhitungan metrik analisis berupa *Packet*

*Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *Average End-to-End Delay (E2E)* dari hasil simulasi menggunakan NS-2.

## 2.5 Simulation of Urban Mobility (SUMO)

*Simulation of urban mobility* (SUMO) adalah simulasi lalu lintas berupa paket aplikasi yang bersifat *open source*. SUMO memiliki beberapa fitur untuk pemodelannya, pemodelan sumo memodelkan jaringan dengan menggambarkan jalannya paket data pada arus lalu lintas, SUMO memiliki banyak *tools* yang dapat membantu pemodelan jaringan pada Tugas Akhir ini diantaranya adalah :

- *Netgenerate*, tool untuk membuat peta grid, spider dan *random network.*
- *Netconvert*, tool untuk mengkonversi data peta berbasi program CLI.
- *RandomTrips.py, tool* untuk membuat rute secara acak yang akan dilalui kendaraan di dalam simulasi.
- *Duarouter*, *tool* untuk melakukan analisis rute berdasarkan simulasi.
- *Sumo,* tool untuk melakukan simulasi lalu lintas berdasarkan posisi, rute dan karakteristik kendaraan.
- *Sumo-gui*, *tool* untuk melihat jalannya simulasi yang dibuat sebelumnya secara grafis, pergerakan kendaraan berupa animasi yang dapat dilihat dengan berbagai mode.
- *traceExporter.py, tool* untuk melakukan konversi output dari sumo kedalam format lain.

## 2.6 OpenStreetMap (OSM)

*OpenStreetMap (*OSM*)* adalah sistem aplikasi berbasis web yang dapat menampilkan peta dunia secara spesifik yang bersifat open source, situs web resmi OSM (www.openstreetmap.org) memiliki bagian utama yakni halaman utama dengan tampilan *Google Maps,* dengan mengijinkan pengguna untuk menggunakan fitur *pan*, *zoom*

dan *search* untuk menemukan peta geografis untuk area tertentu, tak hanya itu OSM juga menyediakan fitur *export* untuk mengunduh peta beserta informasi di dalamnya. [5]

## 2.7 Java OpenStreetMap Editor (JOSM)

*Java OpenStreetMap Editor* (JOSM) adalah aplikasi untuk menyunting data yang didapatkan dari *OpenStreetMap* [7]. Menyunting dalam artian melakukan proses penghilangan objek-objek yang dapat mengganggu simulasi berjalan, misalnya saja pada peta terdapat objek pohon atau bangunan yeng tersisa sehingga aplikasi dapat membantu untuk menghilangkan objek yang dapat mengganggu tersebut.

Pada Tugas Akhir ini, penulis menggunakan aplikasi ini untuk menyunting dan merapikan peta yang diunduh dari *OpenStreetMap* yaitu dengan menghilangkan dan menyambungkan jalan yang ada. Penyuntingan juga dilakukan dengan menghilangkan gedung – gedung yang ada di peta, sehingga simulasi yang dibuat dapat berjalan sesuai keinginan.

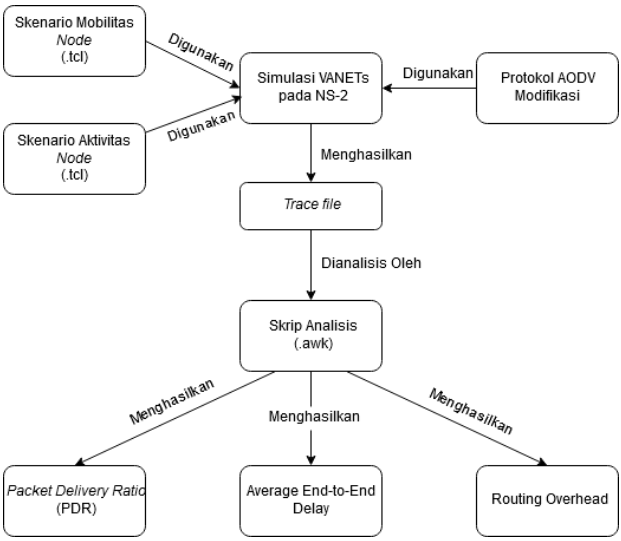
*(Halaman ini sengaja dikosongkan)*

# BAB III
# PERANCANGAN

Pada Bab ini akan membahas segala hal terkait perancangan implememementasi sistem dalam Tugas Akhir ini, hal-hal yang akan dijelaskan pada bab ini berawal dari deskripsi umum sistem, perancangan skenario, alur hingga implementasi sistem yang akan diterapkan pada *Network Simulator* 2 (NS-2).

## 3.1 Deskripsi Umum

Pada Tugas Akhir ini akan dilakukan implementasi routing protocol AODV dengan membandingkan hasil dari routing protocol AODV asli dengan routing protocol AODV yang dimodifikasi. dijalankan dalam bentuk simulasi NS-2 pada lingkungan VANETs. Diagram dari perancangan sistem dapat dilihat pada **Gambar 3.1**.




**Gambar 3.1** Diagram Rancangan Simulasi AODV Modifikasi

Pada Tugas Akhir ini proses pencarian rute melibatkan pengiriman dan penerimaan paket *Route Request* (RREQ) serta pengiriman dan penerimaan paket *Route Reply (*RREP*)* dimana ketika terjadi pengiriman dan penerimaan paket data hal ini juga ikut melibatkan energi yang dibawa sebagai parameter yang akan dijadikan inti dari Tugas Akhir ini. Modifikasi yang dilakukan berfokus pada pencarian rute berdasarkan nilai residu energi pada *node* di dalam satu jaringan. Modifikasi diawali dengan menambahkan *energy model* dan medefinisikan 5 nilai energi yaitu *initial energy, transmission energy, reception energy, idle mode energy* dan *sleep mode energy.* Setelah nilai-nilai energi didefinisikan, diperlukan algoritma untuk melakukan seleksi rute dalam penentuan rute terbaik berdasarkan nilai residu *node* terbesar yakni dengan cara memberitahu *routing protocol* AODV agar proses pencarian rute dan seleksi rute berdasarkan nilai energi yang ada, kemudian dibuat algoritma kondisi perhitungan jumlah residu total energi. Algoritma kondisi selanjutnya yang dibuat adalah seleksi rute dimana hal ini dilakukan agar pengiriman paket data dapat berjalan sebaik mungkin dengan mempertimbangkan nilai residu energi pada *node*.

Modifikasi yang sudah dilakukan akan diterapkan ke dalam simulasi NS-2 dengan meletakkan *node-node* yang tersebar ke dalam peta *grid* dan peta *real* pada lingkungan lalu lintas di Kota Surabaya. Peta tersebut akan dibuat menggunakan bantuan *tools* SUMO. Simulasi yang akan dijalankan memberikan hasil keluaran *trace file* yang kemudian dianalisis menggunakan skrip AWK untuk mendapatkan *Packet Delivery Ratio* (PDR), *End-to-end Delay (*E2E), *Routing Overhead* (RO). Analisis tersebut dapat mengukur performa *routing protocol* AODV yang telah dimodifikasi dibandingkan dengan AODV asli. Analisis ini digunakan untuk mengukur tingkat reliabilitas pengiriman data antara protokol AODV dengan protokol AODV yang dimodifikasi. Daftar Istilah yang sering digunakan pada buku Tugas Akhir ini dapat dilihat pada **Tabel 3.1**.

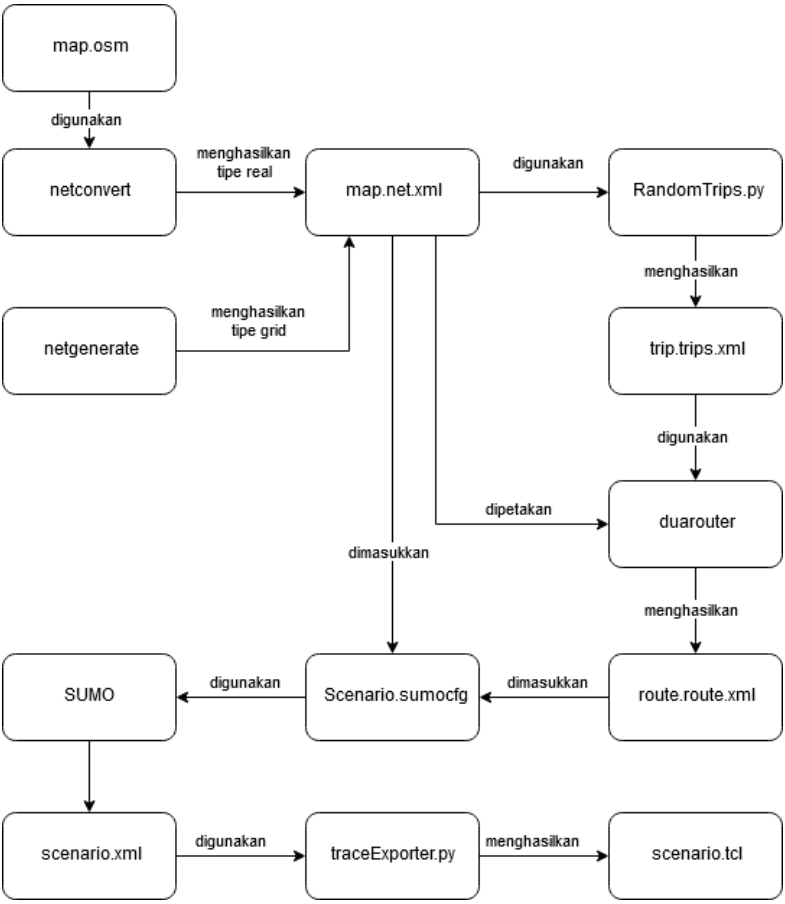
**Tabel 3.1** Daftar Istilah

| No. | Istilah | Penjelasan |
|---|---|---|
| 1 | *Vehicular Ad hoc Networks* (VANETs) | *Vehicular Ad Hoc Networks* (VANETs) merupakan jaringan nirkabel ad hoc yang bersifat dinamis yang berarti pergerakannya bebas dan tidak memiliki infrastruktur yang tetap. |
| 2 | *Ad Hoc On-demand Distance Vector* (AODV) | *Ad Hoc On-demand Distance Vector* (AODV) merupakan salah satu algoritma *routing* reaktif yang digunakan pada Tugas Akhir ini. |
| 3 | *Network Simulator* 2 (NS-2) | NS-2 merupakan alat bantu simulasi berbasis aktivitas pada penelitian jaringan kabel maupun nirkabel. |
| 4 | *Route Request (*RREQ*)* | *Route Request (*RREQ*)* merupakan paket *Request* pada *routing protocol* AODV yang dikirim untuk menemukan rute. |
| 5 | *Route Reply (RREP)* | *Route Reply (RREP)* merupakan paket *Reply* pada *routing protocol* AODV yang dikirim ke node sumber melalui rute yang sudah ditemukan. |
| 6 | *Packet Delivery Ratio* (PDR) | PDR merupakan teknik penghitungan perbandingan jumlah paket yang diterima oleh *node* tujuan dengan jumlah paket yang dikirim oleh *node* sumber. |
| 7 | *Routing Overhead* (RO) | RO merupakan teknik penghitungan jumlah *routing* paket kontrol yang ditransmisikan ke *node* tujuan selama simulasi terjadi. |
| 8 | *Average End-to-End Delay* | *Average End-to-End Delay* merupakan teknik penghitungan waktu yang diperlukan mulai dari paket dikirimkan oleh *node* sumber sampai paket data |

| | | |
|---|---|---|
| | | tersebut berhasil diterima *node* tujuan. |

## 3.2 Perancangan Skenario Mobilitas

Perancangan skenario mobilitas mencakup pembuatan area simulasi, pergerakan *node*, dan implementasi pergerakan *node*. Pada Tugas Akhir ini, terdapat dua macam area simulasi yang akan digunakan yaitu peta *grid* dan *real*. Peta *grid* adalah bentuk area jalan berpetak–petak yang terlihat sebagai jalan berpotongan yang sederhana. Peta *grid* digunakan sebagai simulasi awal VANETs karena lebih stabil. Peta *grid* didapatkan dengan menentukan panjang dan jumlah petak area menggunakan *tools* bantuan SUMO. Sedangkan peta *real* adalah peta asli atau nyata yang digunakan sebagai area simulasi. Peta *real* dibuat dengan mengambil daerah yang diinginkan sebagai area simulasi menggunakan *tools* bantuan OpenStreetMap. Pada Tugas Akhir ini, peta *real* yang diambil penulis adalah salah satu area di kota Surabaya. Perancangan scenario mobilitas dapat dilihat pada **Gambar 3.2**.




**Gambar 3.2** Diagram Perancangan Skenario

## 3.3 Perancangan dan Pendefinisian Model Energi

Modifikasi AODV diawali dengan menambahkan *energy model* dan medefinisikan 5 nilai energi yaitu *initial energy*, *transmission energy*, *reception energy*, *idle mode energy* dan *sleep mode energy*. Nilai-nilai ini didefinisikan untuk membatasi

pengurangan energi. Initial energy merupakan nilai awal energi yang diberikan untuk *node-node* di dalam jaringan, *transmission energy* adalah nilai energi yang berkurang ketika terjadi transmisi paket data, *reception energy* adalah nilai energi yang berkurang ketika terjadi penerimaan paket data, *idle mode* adalah nilai energi yang berkurang ketika node sedang dalam posisi *idle* atau tidak melakukan apa-apa, sleep mode adalah nilai energi yang berkurang ketika node sedang dalam mode *sleep.*

## 3.4 Perancangan Modifikasi AODV Terhadap Seleksi Rute Berdasarkan Residu Energi

Pada Modifikasi AODV perlu adanya perhitungan jumlah energi yang tersisa pada *node-node* yang tersebar diseluruh jaringan dimana perhitungan jumah residu energi didapat dari penambahan nilai energi *node* sekarang dengan nilai energi *node* sebelumnya, perhitungan ini terjadi ketika proses pencarian rute atau pembentukan rute sedang terjadi dan berada di *node* selain *node* sumber dengan kata lain perhitungan hanya terjadi pada *node intermediate* hingga *node* destinasi. Pada dasarnya *routing protocol* AODV memang sudah bisa melakukan reduksi energi tanpa harus dibuatkan algoritma baru, namun terkait dengan jumlah energi yang tersisa diperlukan adanya algoritma baru.

Pembentukan rute pada tugas akhir ini didasarkan oleh pemilihan rute dimana rute yang akan dipilih adalah rute dengan jumlah sisa energi *node* terbesar dimana Algoritma kondisi selanjutnya yang dibuat adalah algoritma seleksi rute yaitu ketika sedang berada di *node* sumber terjadi proses seleksi rute dimana hal ini dilakukan agar pengiriman paket data dapat berjalan sebaik mungkin dengan mempertimbangkan nilai residu energi pada node dengan cara membandingan nilai energi *node* setelahnya dengan sebelumnya jika nilai energi *current node* lebih kecil dibandingan nilai energi *previous node*, maka paket akan di *drop* atau dibebaskan, namun jika terjadi sebaliknya yakni nilai energi *current node* lebih besar dibandingan *previous node* maka

paketnya tidak di *drop* atau tidak dibebaskan dan rute terbaik untuk pengiriman paket data sudah terpilih. Algoritma kondisi diatas dapat dilihat pada **Kode Sumber 3.4.1**.

```
if (I am not the source)
then
      calculate_residual_energy
endif
else if (I am the source)
      if ( current_energy < previous_energy)
      then
            drop packet
      else
            previous_energy = current_energy
endif
```

**Kode Sumber 3.4.1** Pseudocode Algoritma Seleksi Rute

## 3.5 Perancangan Simulasi pada NS-2

Simulasi VANETs pada NS-2 dilakukan dengan menggabungkan *file* skenario yang telah dibuat menggunakan SUMO dan *file* skrip yang berekstensi .tcl dimana file tersebut berisikan konfigurasi lingkungan simulasi.

Terdapat penambahan baris kode diantaranya adalah penambahan dan pendefinisian 5 nilai energi pada *file* scen-en.tcl, penambahan dan penyesuaian variabel - variabel baru pada *file* aodv.cc, *file* aodv.h dan *file* aodv_packet.h sedangkan terkait proses seleksi rute diberikan tambahan berupa baris kode yang mengandung algoritma kondisi pada *file* aodv.cc.

## 3.6 Perancangan Metrik Analisis

Berikut ini merupakan parameter yang akan dilihat pada tugas akhir ini untuk mengetahui pengaruh dari implementasi modifikasi AODV yang telah dibuat.

### 3.6.1 *Packet Delivery Ratio (PDR)*

*Packet Delivery Ratio* (PDR) merupakan teknik penghitungan perbandingan antara jumlah paket yang diterima oleh *node* tujuan dengan jumlah paket yang dikirim oleh *node* sumber. PDR dihitung dengan persamaan 3.7.1.

$$PDR = \frac{packet\ received}{packet\ sent} x\ 100\ \% \qquad \textbf{(3.7.1)}$$

Keterangan :
***Packet Received*** : jumlah total paket yang diterima oleh *node* tujuan.
***Packet Sent*** : jumlah total paket yang dikirim oleh *node* sumber.

### 3.6.2 *Routing Overhead* (RO)

*Routing Overhead* (RO) merupakan teknik penghitungan jumlah *routing* paket kontrol yang ditransmisikan per data paket ke *node* tujuan selama simulasi terjadi. *Routing* paket kontrol yang ditransmisikan berupa paket *route request* (RREQ), *route reply* (RREP), dan *route error* (RERR). Penghitungan RO digunakan untuk melihat banyaknya rute yang dicari dalam jaringan selama simulasi dijalankan. RO dihitung dengan persamaan 3.7.2.

$$RO = \sum_{m=1}^{sent\ and\ forwarded\ num} packet\ sent \qquad \textbf{(3.7.2)}$$

### 3.6.3 *Average End-to-End Delay* (E2E)

*Average End-to-End Delay* dihitung berdasarkan rata-rata *delay* antara waktu paket data diterima dan waktu paket dikirimkan dalam satuan detik. Delay tiap paket didapat dari rentang waktu antara *node* asal saat mengirimkan paket dan *node*

tujuan menerima paket. Delay tiap paket tersebut semua dijumlahkan dan dibagi dengan jumlah paket yang berhasil diterima, maka akan didapatkan rata – rata E2E, yang dapat dihitung dengan persamaan 3.7.3.

$$E2E = \frac{\sum_{m=1}^{recvnum} CBRRecvTime - CBRSentTime}{recvnum} \qquad \textbf{(3.7.3)}$$

Keterangan:

*E2E* = Average *End-to-End Delay*

*CBRRecvTime* = Waktu *node* asal mengirimkan paket

*CBRSentTime* = Waktu *node* tujuan menerima paket

*recvnum* = Jumlah paket yang berhasil diterima

*(Halaman ini sengaja dikosongkan*

# BAB IV
# IMPLEMENTASI

Bab ini akan membahas segala hal terkait dengan implementasi dari perancangan yang telah dijelaskan pada bab sebelumnya. Implementasi ini berupa kode sumber untuk membangun program. Implementasi yang akan dijelaskan pada bab ini, meliputi lingkungan pembangunan sistem, implementasi skenario mobilitas yang berupa skenario *grid* dan skenario *real*, implementasi model energi, implementasi perhitungan residu energi, implementasi modifikasi *routing protocol* AODV dalam pencarian rute dan seleksi rute, implemetasi simulasi pada NS-2, implementasi perhitungan total residu energi pada node secara keseluruhan serta implementasi metrik analisis yang akan dibandingkan untuk mengetahui performa AODV yang telah dimodifikasi.

## 4.1 Lingkungan Pembangunan Sistem

Lingkungan sistem yang digunakan untuk membangun implementasi skenario dapat dilihat pada **Tabel 4.1**.

**Tabel 4.1** Tabel Lingkungan Pembangunan Sistem

| **Perangkat** | **Komponen** | **Spesifikasi** |
|---|---|---|
| **Perangkat Keras** | *Processor* | Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59 GHz |
| | Memori | RAM 8,00 GB |
| | Penyimpanan | 928 GB |
| **Perangkat Lunak** | Sistem Operasi | Ubuntu Bionic 18.04 LTS 64 bit |
| | Simulator | Network Simulator 2 versi 2.35 |

## 4.2 Implementasi Skenario Mobilitas

Implementasi skenario mobilitas pada tugas akhir ini dilakukan di dalam lingkungan VANETs yang dibagi kedalam dua bagian yakni skenario *grid* yang menggunakan peta berpetak yang menggambarkan kondisi jalan yang dipenuhi node yang tersebar dan skenario *real* yang menggunakan peta jalan sesugguhnya yang diambil dari sebagian area di Kota Surabaya.

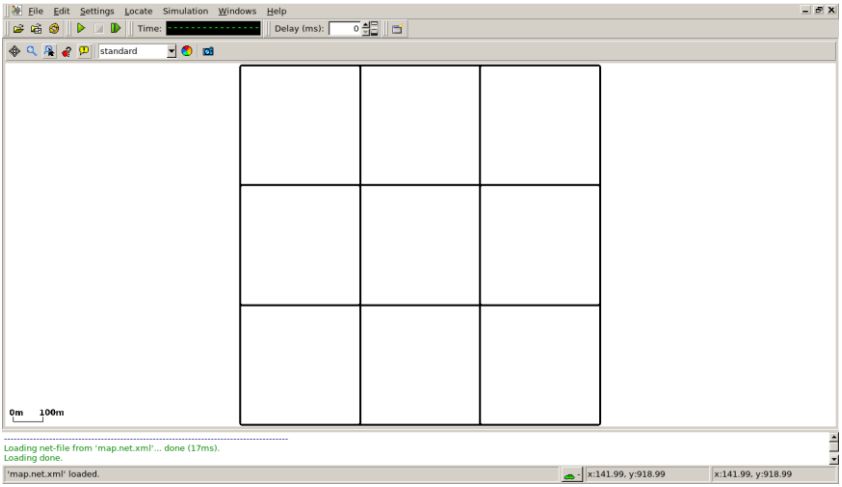
### 4.2.1 Skenario Grid

Implementasi Skenario Grid merupakan implementasi skenario mobilitas pertama yang dilakukan, pada skenario grid penulis menggunakan *tools* yang sudah disediakan oleh SUMO. Dengan menggunakan *netgenerate* penulis membuat peta *grid* berukuran 650 m x 650 m yang membentuk persimpangan titik potong jalan vertikal dan horisontal sebanyak 4 titik x 4 titik, titik persimpangan ini akan membentuk 9 petak area yang memiliki luas per petak sebesar 200 m x 200 m, selain itu dengan *tool netgenerate* dapat juga menentukan kecepatan sebesar 20 m/s. perintah *netgenerate* dapat dilihat pada ***Kode Sumber 4.2.1***.

```
net netgenerate --grid --grid.number=4 --
grid.length=200 --default.speed=20 --
tls.gueseret=1 --output-file=map.net.xml
```

**Kode Sumber 4.2.1** Perintah netgenerate

Ketika perintah *netgenerate* sudah dijalankan maka akan terbenetuk *file* berekstensi (.xml) yang menunjukkan bentuk peta, *file* ini dapat dilihat pada **Gambar 4.1.**




**Gambar 4.1** Peta Grid

Ketika peta sudah terbentuk maka hal selanjutnya yang dilakukan adalah membuat *node* serta pergerakan *node* dengan mengunakan *tool* randomTrips, randomTrips akan membuat node dengan posisi dan pergerakannya yang secara acak di dalam peta, perintah randomTrips dapat dilihat pada **Kode Sumber 4.2.2**.

```
python $SUMO_HOME/tools/randomTrips.py -n
map.net.xml -e 48 -l --trip-
attributes="departLane=\"best\"
departSpeed=\"max\"
departPos=\"random_free\"" -o trip.trips.xml
```

**Kode Sumber 4.2.2** Perintah randomTrips

Selanjutnya yang dilakukan adalah membuat rute untuk node yang sudah terbentuk sebelumnya menggunakan *tool*

duarouter, perintah duarouter dapat dilihat pada **Kode Sumber 4.2.3**.

```
duarouter -n map.net.xml -t trip.trips.xml
-o route.rou.xml --ignore-errors --repair
```

**Kode Sumber 4.2.3** Perintah duarouter

Ketika telah melakukan pembuatan node, pergerakan node serta rute untuk masing – masing node, maka hal yang dilakukan selanjutnya adalah melakukan *generate* file agar membentuk skenario seutuhnya, pembentukan skenario ini membutuhkan file skrip berekstensi (.sumocfg), isi dari file ini dapat dilihat pada
**Kode Sumber *4.2.4***.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"
xsi:noNamespaceSchemaLocation="http://sumo.
dlr.de/xsd/sumoConfiguration.xsd">
   <input>
      <net-file value="map.net.xml"/>
      <route-files value="routes.rou.xml"/>
   </input>
   <time>
      <begin value="0"/>
      <end value="200"/>
   </time>
</configuration>
```

**Kode Sumber 4.2.4** File Skrip.sumocfg

*File* .sumocfg nantinya akan disimpan ke dalam direktori yang sama dengan *file* peta dan *file* rute pergerakan *node*. Kemudian penulis akan membuat *file* skenario dalam bentuk *file* (.xml) dari sebuah *file* skrip berekstensi (.sumocfg) menggunakan *tool* SUMO. Perintah untuk menggunakan *tool* sumo dapat dilihat pada **Kode Sumber 4.2.5**.

```
sumo -c file.sumocfg --fcd-output
scenario.xml
```

**Kode Sumber 4.2.5** Perintah sumo

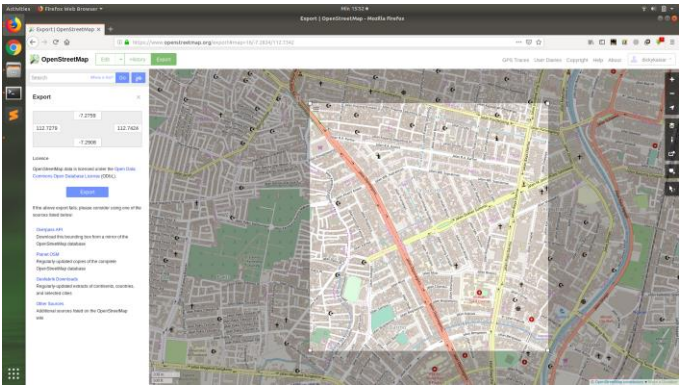
*File* yang telah dibuat kemudian dikonversi ke dalam format file dengan ekstensi (.tcl) agar dapat disimulasikan menggunakan NS-2, tool yang dapat digunakan untuk melakukan konversi file tersebut adalah traceExporter yang dapat dilihat pada **Kode Sumber 4.2.6** .

```
python $SUMO_HOME/tools/traceExporter.py --
fcd-input=scenario.xml --ns2mobility-
output=scenario.tcl
```

**Kode Sumber 4.2.6** Perintah traceExporter

### 4.2.2 Skenario Real

Implementasi Skenario *Real* merupakan implementasi skenario mobilitas selanjutnya yang dilakukan setelah melakukan implementasi skenario grid, pada skenario real penulis menggunakan *OpenStreetMap* yang sudah disediakan oleh SUMO, pada *OpenStreetMap* dilakukan *export* peta di area sekitar JL. Dr. Soetomo, Surabaya. Area yang diambil ditandai dengan bujur sangkar pada **Gambar 4.2**.



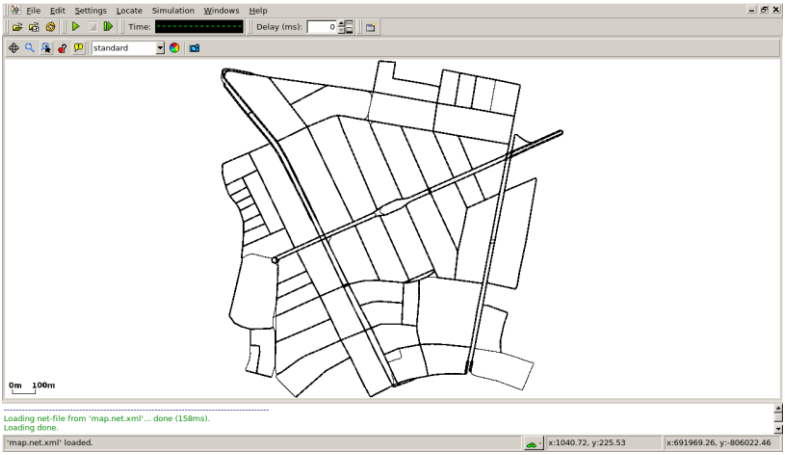
**Gambar 4.2** Peta Skenario Real OpenStreetMap

Ketika sudah terbentuk file hasil dari *export* pada *OpenStreetMap*, langkah selanjutnya yang dilakukan adalah konversi file dengan ekstensi (.osm) menjadi file dengan ekstensi (.xml) menggunakan *tool* dari SUMO yakni netconvert. Perintah netconvert dapat dilihat pada ***Kode Sumber 4.2.7***.

```
netconvert --osm-files map.osm --output-
file map.net.xml
```

**Kode Sumber 4.2.7** Perintah neconvert

Hasil konversi file dengan ekstensi (.osm) menjadi file dengan ekstensi (.xml) dapat dilihat pada **Gambar 4.3**.




**Gambar 4.3** Hasil Konversi Peta Real

Langkah selanjutnya yang dilakukan sama dengan ketika membuat skenario mobilitas *grid*, yaitu membuat *node* asal dan *node* tujuan menggunakan *tool* randomTrips. lalu membuat jalur rute *node* untuk sampai ke tujuan menggunakan *tool* duarouter. Kemudian membuat *file* skenario dengan ekstensi (.xml) menggunakan *tool* SUMO dengan bantuan *file* skrip dengan ekstensi (.sumocfg). Selanjutnya dilakukan konversi *file* skenario dengan ekstensi (.tcl) untuk dapat dilakukan simulasi pada NS-2 menggunakan *tool* traceExporter. Perintah untuk menggunakan *tools* tersebut sama dengan ketika membuat skenario *grid* yang telah dibahas sebelumnya.

## 4.3 Implementasi Model Energi

Pada AODV yang dimodifikasi dilakukan implementasi model energi untuk dapat mengambil dan menampilkan nilai energi pada *node*, nilai energi digunakan untuk menghitung residu energi pada *node* dan untuk menghitung total residu energi secara keseluruhan. model energi ini diimplementasikan

pada file dengan ekstensi (.tcl), pada file tersebut ditambahkan 2 bagian pendefinisian energi yakni pada deklarasi variable dan pada bagian konfigurasi node (*global node setting*). Deklarasi variable yang dimaksud adalah deklarasi model energi yang mengandung 5 nilai energi yaitu *initial energy*, *transmission energy*, *reception energy*, *idle mode energy* dan *sleep mode energy*. Detail model energi dapat dilihat pada **Kode Sumber 4.3.1**, sedangkan untuk konfigurasi node dapat dilihat pada **Kode Sumber 4.3.2**.

```
set val(energy_mod)        EnergyModel
set val(energy_init)       50
set val(tx_power)          0.33
set val(rx_power)          0.1
set val(rx_power)          0.1
set val(sleep_power)       0.03
```

**Kode Sumber 4.3.1** Deklarasi Variable Pada File (.tcl)

```
-energyModel $val(energy_mod) \
-initialEnergy $val(energy_init) \
-txPower $val(tx_power) \
-rxPower $val(rx_power) \
-idlePower $val(idle_power) \
-sleepPower $val(sleep_power) \
```

**Kode Sumber 4.3.2** Konfigurasi Node Pada File (.tcl)

## 4.4 Modifikasi AODV Terhadap Seleksi Rute Berdasarkan Residu Energi

Pada *file* aodv.cc di bagian fungsi AODV::recvReply ditambahkan beberapa baris kode yang mengandung algoritma kondisi yang dapat melakukan seleksi rute berdasarkan residu

energi, implementasi ini dilakukan Ketika simulasi *node* pada NS-2 sudah dapat menghasilkan energi. ketika proses pencarian rute sedang berada di *node* sumber maka terjadi proses seleksi rute dengan cara membandingan nilai energi *node* setelahnya dengan sebelumnya jika nilai energi *current node* lebih kecil dibandingan nilai energi *previous node*, maka paket akan di *drop* atau dibebaskan, namun jika terjadi sebaliknya yakni nilai energi *current node* lebih besar dibandingan *previous node* maka paketnya tidak di *drop* atau tidak dibebaskan, sedangkan ketika proses pencarian rute berada pada *node intermediate* hingga *node* destinasi dilakukan perhitungan residu energi dengan menjumlahkan nilai residu energi *node* sekarang dengan sebelumnya dimana Masing masing *node* akan berkurang nilai energinya selama simulasi berjalan berdasarkan model energi yang telah dibuat sebelumnya. Implementasi seleksi rute berdasarkan residu energi dapat dilihat pada **Kode Sumber 4.4.1**.

```
if(ih->daddr() == index)
  {
    if(rp->rp_resenergy < iEnergy
      Packet::free(p);
    else
      iEnergy = rp->rp_resenergy;
  }
  else
    rp->rp_resenergy+=iEnergy;
```

**Kode Sumber 4.4.1** Implementasi Seleksi Rute Berdasarkan Residu Energi

Sebelum itu penulis juga menambahkan modifikasi agar algoritma routing aodv diatur berdasarkan energi yang ada, modifikasi ini dapat dilihat pada **Kode Sumber 4.4.2**.

```
iNode =(MobileNode *)
(Node::get_node_by_address (index) );
  ((MobileNode *)iNode)-
>getLoc(&xpos,&ypos,&zpos);
  iEnergy =  iNode->energy_model()->energy();
```

**Kode Sumber 4.4.2** Modifikasi Algoritma Routing Aodv Berdasarkan Residu Energi

## 4.5 Implementasi Simulasi Pada NS-2

Implementasi simulasi pada NS-2 di lingkungan VANETs dijalankan menggunakan *file* dengan ekstensi (.tcl). *file* ini berisi konfigurasi yang dibutuhkan untuk setiap *node* dan langkah-langkah yang dilakukan selama simulasi. konfigurasi lingkungan simulasi dapat dilihat pada **Kode Sumber 4.5.1.**

```
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)       1500
set opt(y)       1500
set val(ifqlen)  1000
set val(nn)      200
set val(seed)    1.0
set val(adhocRouting) AODV
set val(stop)    200
set val(cp)      "cbr200.tcl"
set val(sc)      "scenario.tcl"
```

**Kode Sumber 4.5.1** Konfigurasi Lingkungan Simulasi

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* oleh SUMO. Konfigurasi *file traffic* dapat dilihat pada **Kode Sumber 4.5.2.**

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(198) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(199) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 1000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 0 "$cbr_(0) start"
```

**Kode Sumber 4.5.2** Konfigurasi *File Traffic*

## 4.6 Implementasi Metrik Analisis

Simulasi yang dijalankan menggunakan NS-2 menghasilkan keluaran berupa *trace file* yang berisikan data mengenai aktivitas yang terjadi selama simulasi dalam bentuk file dengan ekstensi (.tr). Dari data tersebut, dapat dilakukan analisis performa *routing protocol* AODV dengan mengukur beberapa metrik. Pada tugas akhir ini, metrik yang akan dianalisis adalah *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay* (E2E).

### 4.6.1 *Packet Delivery Ratio (PDR)*

*Packet Delivery Ratio* (PDR) merupakan teknik penghitungan perbandingan antara jumlah paket yang diterima oleh *node* tujuan dengan jumlah paket yang dikirim oleh *node* sumber. Pada *file* dengan ekstensi (.tr) terdapat beberapa *event* yang menunjukkan aktifitas selama simulasi dijalankan, Dalam menghitung PDR, dapat ditemukan pada *file* dengan ekstensi (.tr) *event* bernama "AGT", *event* ini menunjukkan paket komunikasi data. Pada *event* AGT juga terdapat jumlah paket yang dikirimkan dan paket yang diterima. Setelah itu, nilai PDR dapat dihitung menggunakan file dengan ekstensi (.awk), berikut kode sumber berupa pseudocode terkait perhitungan PDR yang dapat dilihat pada **Kode Sumber 4.6.1**, detail kode dapat dilihat pada **lampiran D.1**.

```
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
     eeceived++
  end if
pdr = received / sent
```

**Kode Sumber 4.6.1** Pseudocode Perhitungan PDR

### 4.6.2 *Routing Overhead* (RO)

*Routing Overhead* (RO) merupakan teknik penghitungan jumlah *routing* paket kontrol yang ditransmisikan per data paket ke *node* tujuan selama simulasi terjadi. *Routing* paket kontrol yang ditransmisikan berupa paket *route request* (RREQ), *route reply* (RREP), dan *route error* (RERR). Penghitungan RO

dilakukan dengan cara menjumlahkan paket dengan event “sent” pada kolom ke-1 dengan event “RTR” pada kolom ke-4. berikut kode sumber berupa pseudocode terkait perhitungan RO yang dapat dilihat pada **Kode Sumber 4.6.2**, detail kode dapat dilihat pada **lampiran E.1**.

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

**Kode Sumber 4.6.2** Pseudocode Perhitungan RO

### 4.6.3 *Routing Overhead* (RO)

Dalam menghitung *Average End-to-End Delay* (E2E), dapat ditemukan pada *file* dengan ekstensi (.tr) *event* bernama “AGT” pada kolom ke-2 dan *event* bernama “r” pada kolom ke-4, *event* ini menunjukkan perbedaaan paket dikirim atau diterima. Perhitungan *delay* didapat dari paket dengan mengurangi waktu dari paket diterima dengan waktu dari paket dikirim dengan kondisi harus memiliki *id* paket yang sama. Setelah itu untuk menghitung rata-ratanya hanya perlu membagi total *delay* paket dengan jumlah paket yang ada, berikut kode sumber berupa pseudocode terkait perhitungan *Average End-to-End Delay* (E2E) yang dapat dilihat pada **Kode Sumber 4.6.2**, detail kode dapat dilihat pada **lampiran F.1**.

```
total_delay = 0
count = 0

for i = 1 to the number of rows
     count++
     if in a row contains “s” and AGT then
          start_time[packet_id] = time
     else in a row contains “r” and AGT then
          end_time[packet_id] = time
     end if
     delay[packet_id] = end_time[packet_id]
     - start_time[packet_id]
     total_delay += delay[packet_id]
e2e = total_delay / count
```

**Kode Sumber 4.6.2** Pseudocode Perhitungan E2E

*(Halaman ini sengaja dikosongkan)*

# BAB V
# UJI COBA DAN EVALUASI

Bab ini membahas mengenai uji coba yang dilakukan dan evaluasi sesuai dengan rancangan dan implementasi. Dari hasil yang didapatkan setelah melakukan uji coba, akan dilakukan evaluasi sehingga dapat diambil kesimpulan untuk bab selanjutnya.

## 5.1 Lingkungan Uji Coba

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(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59 GHz |
| **Sistem Operasi** | Ubuntu Bionic 18.04 LTS 64 bit |
| **Linux Kernel** | Linux kernel 4.4 |
| **Memori** | RAM 8 GB |
| **Penyimpanan** | 928 GB |

Adapun versi perangkat lunak yang digunakan dalam Tugas Akhir ini adalah sebagai berikut:

- SUMO versi 0.25.0 untuk pembuatan skenario mobilitas VANETs.
- JOSM versi 10301 untuk penyuntingan peta OpenStreetMap.
- NS-2 versi 2.35 untuk simulasi skenario VANETs.

Pengujian dilakukan dengan menjalankan skenario yang disimulasikan menggunakan NS-2. Dari simulasi tersebut dihasilkan sebuah *trace file* dengan ekstensi (.tr) yang akan dianalisis menggunakan *file* dengan ekstensi (.awk) untuk mendapatkan *Packet Delivery Ratio*, *Routing Overhead*, dan *Average End-to-End Delay*.

## 5.2 Hasil Uji Coba

Simulasi yang telah dijalankan oleh NS-2 hingga proses analisis menggunakan skrip AWK menghasilkan keluaran berupa nilai metrik analisis, Hasil uji coba menggunakan *node* sumber dan *node* tujuan yang diletakkan secara statis, sedangkan untuk *node* diantarnya (*node intermediate*) diletakkan secara acak. Hasil dari scenario *grid* dan scenario *real* untuk Tugas Akhir ini dapat dilihat pada **Tabel 5.2**.

**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 | 1500 m x 1500 m |
| 5 | Jumlah *Node* | 50, 100, 150, 200 |
| 6 | Jumlah Paket Maksimum | 50 |
| 7 | Kecepatan Maksimum | 20 m/s |
| 8 | *Initial Energy* | 50 J |
| 9 | Protokol MAC | IEEE 802.11p |
| 10 | *Transmission Range* | 200 m |

### 5.2.1 Hasil Uji Coba Skenario *Grid*

Uji coba skenario *grid* dilakukan sebanyak 10 kali pada 50 *node*, 100 *node*, 150 *node* dan 200 *node*. dengan skenario mobilitas *random* pada skenario peta *grid* dengan luas area peta sebesar 1500 m x 1500 m dimana jumlah *node* sebanyak 50 untuk lingkungan yang jarang, 100 dan 150 *node* untuk lingkungan yang sedang, dan 200 *node* untuk lingkungan yang padat dilakukan pada kecepatan standar yaitu 20 m/s. Hasil analisis dapat dilihat pada **Tabel 5.3**.

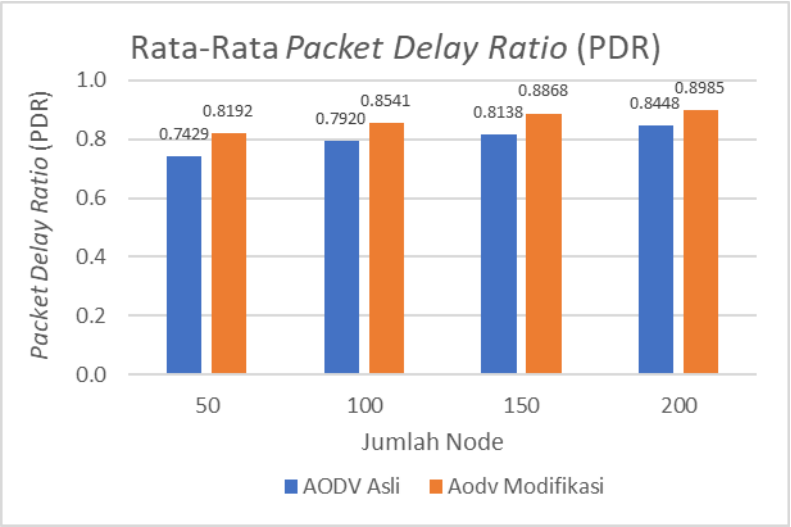
**Tabel 5.3** Hasil Rata-Rata PDR Skenario Grid

| **Packet Delay Ratio (PDR)** | | | | |
|---|---|---|---|---|
| **Node** | **AODV Asli** | **AODV Modifikasi** | **Kenaikan** | **Prosentase** |
| 50 | 0,7429 | 0,8192 | 0,0763 | 10,27% |
| 100 | 0,7920 | 0,8541 | 0,0621 | 7,84% |
| 150 | 0,8138 | 0,8868 | 0,0730 | 8,97% |
| 200 | 0,8448 | 0,8985 | 0,0537 | 6,36% |

Berdasarkan **Tabel 5.3** menunjukkan bahwa *routing protocol* AODV Asli dan *routing protocol* AODV yang sudah dimodifikasi mengalami kenaikan yang signifikan pada nilai *Packet Delay Ratio* (PDR) seiring bertambahnya jumlah *node*. Pada lingkungan yang jarang dengan jumlah 50 *node*, menghasilkan kenaikan nilai PDR sebesar 0,0763, atau naik menjadi sekitar 10,27% dimana *routing protocol* AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 100 node, menghasilkan kenaikan nilai PDR sebesar 0,0621, atau naik menjadi sekitar 7,84% dimana routing protocol AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 150 node, menghasilkan kenaikan nilai

PDR sebesar 0,0730, atau naik menjadi sekitar 8,97% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang padat dengan jumlah 200 node, menghasilkan kenaikan nilai PDR sebesar 0,0537, atau naik menjadi sekitar 6,36% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut.

Dapat dilihat rata-rata kenaikan yang terjadi untuk nilai PDR adalah 8,30%. Jika ketiga lingkungan tersebut dibandingkan dapat dilihat bahwa dengan jumlah *node* yang lebih sedikit atau pada lingkungan dengan *node* yang jarang, menghasilkan nilai PDR yang lebih bagus daripada di lingkungan dengan jumlah *node* yang sedang maupun yang padat baik untuk *routing protocol* AODV asli maupun *routing protocol* AODV yang telah dimodifikasi. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan nilai PDR yang lebih bagus dibandingkan AODV asli dengan kenaikan nilai PDR yang cukup signifikan, untuk lebih jelasnya dapat dilihat pada **Gambar 5.1**.




**Gambar 5.1** Grafik *Packet Delay Ratio* Skenario *Grid*

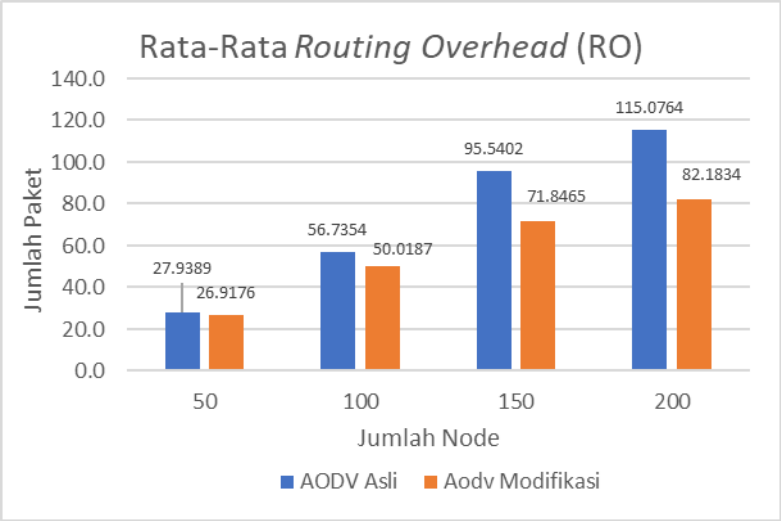
Hasil uji coba skenario *grid* untuk nilai *Routing Overhead* (RO) 50 *node*, 100 *node*, 150 *node* dan 200 *node* dapat dilihat pada **Tabel 5.4** .

**Tabel 5.4** Hasil Rata-Rata RO Skenario Grid

| ***Routing Overhead* (*RO*)** | | | | |
|---|---|---|---|---|
| **Node** | **AODV Asli** | **AODV Modifikasi** | **Penurunan** | **Prosentase** |
| 50 | 27,9389 | 26,9176 | 1,0212 | 3,66% |
| 100 | 56,7354 | 50,0187 | 6,7167 | 11,84% |
| 150 | 95,5402 | 71,8465 | 23,6937 | 24,80% |
| 200 | 115,0764 | 82,1834 | 32,8930 | 28,58% |

Berdasarkan **Tabel 5.4** menunjukkan bahwa *routing protocol* AODV Asli dan *routing protocol* AODV yang sudah dimodifikasi mengalami kenaikan yang signifikan pada nilai *Routing Overhead* (RO) seiring bertambahnya jumlah *node*. Pada lingkungan yang jarang dengan jumlah 50 *node*, menghasilkan nilai RO sebesar 27,9389, atau turun menjadi sekitar 3,66% dimana *routing protocol* AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 100 node, menghasilkan penurunan nilai RO sebesar 6,7167, atau turun menjadi sekitar 11,84% dimana routing protocol AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 150 node, menghasilkan penurunan nilai RO sebesar 23,6937, atau turun menjadi sekitar 24,80% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang padat dengan jumlah 200 node, menghasilkan penurunan nilai RO sebesar 32,8930, atau turun menjadi sekitar 28,58% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut.

Dapat dilihat rata-rata penurunan yang terjadi untuk nilai RO adalah 21,78%. Jika ketiga lingkungan tersebut dibandingkan dapat dilihat bahwa dengan jumlah *node* yang lebih sedikit atau pada lingkungan dengan *node* yang jarang, menghasilkan nilai RO yang lebih bagus daripada di lingkungan dengan jumlah *node* yang sedang maupun yang padat baik untuk *routing protocol* AODV asli maupun *routing protocol* AODV yang telah dimodifikasi. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan nilai RO yang lebih bagus dibandingkan AODV asli dengan penurunan nilai RO yang cukup signifikan , walaupun demikian semakin besarnya nilai RO menyebabkan akan semakin besar kemungkinan *failure link* atau kegagalan rute, sehingga kinerja terbaik AODV akan lebih bagus jika nilai RO yang dihasilkan semakin turun, untuk lebih jelasnya dapat dilihat pada **Gambar 5.2** .




**Gambar 5.2** Grafik *Routing Overhead* Skenario *Grid*

Hasil uji coba skenario *grid* untuk nilai *Average End-to-End Delay* (E2E) 50 *node*, 100 *node*, 150 *node* dan 200 *node* dapat dilihat pada **Tabel 5.5**.

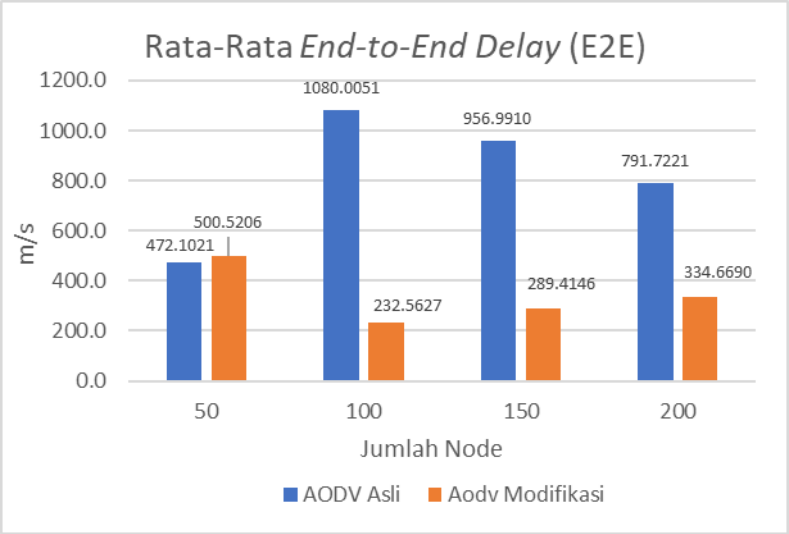
**Tabel 5.5** Hasil Rata-Rata E2E Skenario *Grid*

| ***Average End-to-End Delay* (E2E)** | | | | |
|---|---|---|---|---|
| **Node** | **AODV Asli** | **AODV Modifikasi** | **Penurunan** | **Prosentase** |
| 50 | 472,1021 | 500,5206 | -28,4185 | -6,02% |
| 100 | 1080,0051 | 232,5627 | 847,4424 | 78,47% |
| 150 | 956,9910 | 289,4146 | 667,5764 | 69,76% |
| 200 | 791,7221 | 334,6690 | 457,0531 | 57,73% |

Berdasarkan **Tabel 5.5** menunjukkan bahwa *routing protocol* AODV Asli dan *routing protocol* AODV yang sudah dimodifikasi mengalami kenaikan dan penurunan pada nilai *end-to-end delay* seiring bertambahnya jumlah *node*. Pada lingkungan yang jarang dengan jumlah 50 *node*, menghasilkan nilai *end-to-end delay* sebesar -28.4185, atau naik menjadi sekitar 6,02% dimana *routing protocol* AODV yang telah dimodifikasi tidak terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 100 node, menghasilkan penurunan nilai *end-to-end delay* sebesar 847,4424, atau turun menjadi sekitar 78,47% dimana routing protocol AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 150 node, menghasilkan penurunan nilai *end-to-end delay* sebesar 667,5764, atau turun menjadi sekitar 69,76% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang padat dengan jumlah 200 node, menghasilkan penurunan nilai *end-to-end delay* sebesar 457,0531, atau turun menjadi sekitar 57,73% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut.

Dapat dilihat rata-rata penurunan yang terjadi untuk nilai *end-to-end delay* adalah 58,88%. Jika ketiga lingkungan tersebut dibandingkan dapat dilihat bahwa dengan jumlah *node* yang lebih sedikit atau pada lingkungan dengan *node* yang jarang, menghasilkan nilai *end-to-end delay* yang lebih bagus daripada di lingkungan dengan jumlah *node* yang sedang maupun yang padat baik untuk *routing protocol* AODV asli maupun *routing protocol* AODV yang telah dimodifikasi. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan nilai *end-to-end delay* yang lebih bagus dibandingkan AODV asli dengan penurunan nilai E2E yang cukup signifikan, walaupun demikian nilai *end-to-end delay* jika semakin besar maka akan menyebabkan tidak sampainya paket pada node atau mengalami delay pengiriman paket , sehingga kinerja terbaik AODV akan lebih bagus jika nilai

*end-to-end delay* yang dihasilkan semakin turun, untuk lebih jelasnya dapat dilihat pada **Gambar 5.3**.




**Gambar 5.3** Grafik End-to-End Delay Skenario Grid

### 5.2.2 Hasil Uji Coba Skenario *Real*

Uji coba skenario *real* dilakukan sebanyak 10 kali pada 50 *node*, 100 *node*, 150 *node* dan 200 *node*. dengan skenario mobilitas *random* pada skenario peta *real* dengan luas area peta sebesar 650 m x 650 m dimana jumlah *node* sebanyak 50 untuk lingkungan yang jarang, 100 dan 150 *node* untuk lingkungan yang sedang, dan 200 *node* untuk lingkungan yang padat dilakukan pada kecepatan standar yaitu 20 m/s. Hasil analisis dapat dilihat pada **Tabel 5.6**.

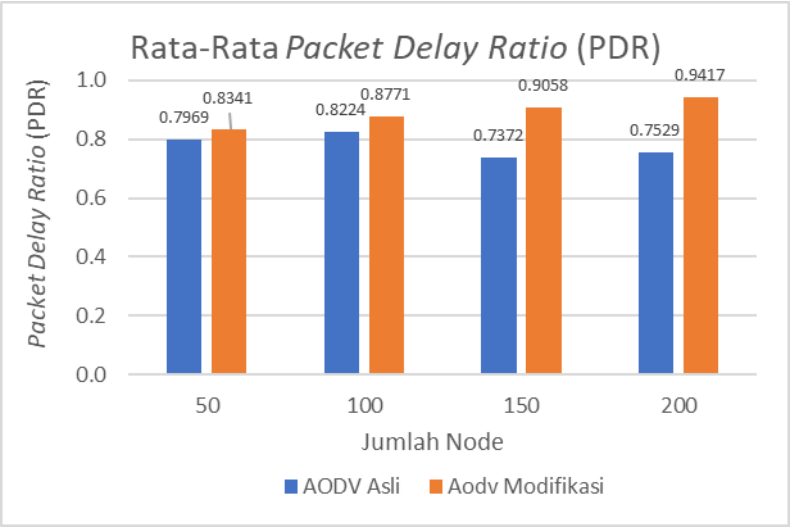
**Tabel 5.6** Hasil Rata-Rata PDR Skenario *Real*

| Packet Delay Ratio (PDR) | | | | |
|---|---|---|---|---|
| **Node** | **AODV Asli** | **AODV Modifikasi** | **Kenaikan** | **Prosentase** |
| 50 | 0,7969 | 0,8341 | 0,0372 | 4,67% |
| 100 | 0,8224 | 0,8771 | 0,0547 | 6,65% |
| 150 | 0,7372 | 0,9058 | 0,1687 | 22,88% |
| 200 | 0,7529 | 0,9417 | 0.1888 | 25,08% |

Berdasarkan **Tabel 5.6** menunjukkan bahwa *routing protocol* AODV Asli dan *routing protocol* AODV yang sudah dimodifikasi mengalami kenaikan yang signifikan pada nilai *Packet Delay Ratio* (PDR) seiring bertambahnya jumlah *node*. Pada lingkungan yang jarang dengan jumlah 50 *node*, menghasilkan kenaikan nilai PDR sebesar 0,0372, atau naik menjadi sekitar 4,67% dimana *routing protocol* AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 100 node, menghasilkan kenaikan nilai PDR sebesar 0,0547, atau naik menjadi sekitar 6,65% dimana routing protocol AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 150 node, menghasilkan kenaikan nilai PDR sebesar 0,1687, atau naik menjadi sekitar 22,88% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang padat dengan jumlah 200 node, menghasilkan kenaikan nilai PDR sebesar 0,1888, atau naik menjadi sekitar 25,08% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut.

Dapat dilihat rata-rata kenaikan yang terjadi untuk nilai PDR adalah 14,45%. Jika ketiga lingkungan tersebut dibandingkan dapat dilihat bahwa dengan jumlah *node* yang lebih sedikit atau pada lingkungan dengan *node* yang jarang,

menghasilkan nilai PDR yang lebih bagus daripada di lingkungan dengan jumlah *node* yang sedang maupun yang padat baik untuk *routing protocol* AODV asli maupun *routing protocol* AODV yang telah dimodifikasi. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan nilai PDR yang lebih bagus dibandingkan AODV asli dengan kenaikan nilai PDR yang cukup signifikan, untuk lebih jelasnya dapat dilihat pada **Gambar 5.4**.




**Gambar 5.4** Grafik *Packet Delay Ratio* Skenario *Real*

Hasil uji coba skenario *real* untuk nilai Routing Overhead (RO) 50 *node*, 100 *node*, 150 *node* dan 200 *node* dapat dilihat pada **Tabel 5.7.**

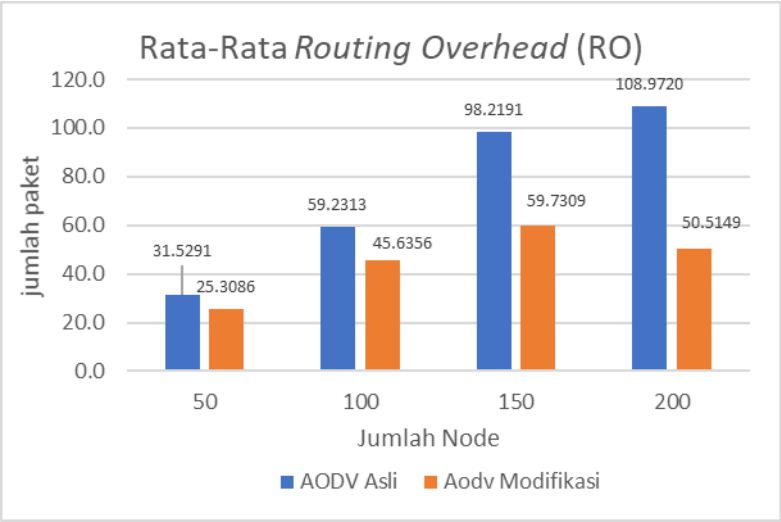
**Tabel 5.7** Hasil Rata-Rata RO Skenario Real

| *Routing Overhead* (*RO*) | | | | |
|---|---|---|---|---|
| **Node** | **AODV Asli** | **AODV Modifikasi** | **Penurunan** | **Prosentase** |
| 50 | 31,5291 | 25,3086 | 6,2205 | 19,73% |
| 100 | 59,2313 | 45,6356 | 13,5958 | 22,95% |
| 150 | 98,2191 | 59,7309 | 38,4882 | 39,19% |
| 200 | 108,9720 | 50,5149 | 58,4571 | 53,64% |

Berdasarkan **Tabel 5.**7 menunjukkan bahwa *routing protocol* AODV Asli dan *routing protocol* AODV yang sudah dimodifikasi mengalami kenaikan yang signifikan pada nilai *Routing Overhead* (RO) seiring bertambahnya jumlah *node*. Pada lingkungan yang jarang dengan jumlah 50 *node*, menghasilkan nilai RO sebesar 6,2205, atau turun menjadi sekitar 19,73%% dimana *routing protocol* AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 100 node, menghasilkan penurunan nilai RO sebesar 13,5958, atau turun menjadi sekitar 22,95% dimana routing protocol AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 150 node, menghasilkan penurunan nilai RO sebesar 38,4882, atau turun menjadi sekitar 39,19% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang padat dengan jumlah 200 node, menghasilkan penurunan nilai RO sebesar 58,4571, atau turun menjadi sekitar 53,64% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut.

Dapat dilihat rata-rata penurunan yang terjadi untuk nilai RO adalah 39,19%. Jika ketiga lingkungan tersebut dibandingkan dapat dilihat bahwa dengan jumlah *node* yang lebih sedikit atau pada lingkungan dengan *node* yang jarang, menghasilkan nilai

RO yang lebih bagus daripada di lingkungan dengan jumlah *node* yang sedang maupun yang padat baik untuk *routing protocol* AODV asli maupun *routing protocol* AODV yang telah dimodifikasi. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan nilai RO yang lebih bagus dibandingkan AODV asli dengan penurunan nilai RO yang cukup signifikan , walaupun demikian semakin besarnya nilai RO menyebabkan akan semakin besar kemungkinan *failure link* atau kegagalan rute, sehingga kinerja terbaik AODV akan lebih bagus jika nilai RO yang dihasilkan semakin turun, untuk lebih jelasnya dapat dilihat pada **Gambar 5.5**.




**Gambar 5.5** Grafik Routing Overhead Skenario *Real*

Hasil uji coba skenario *real* untuk nilai *Average End-to-End Delay* (E2E) 50 *node*, 100 *node*, 150 *node* dan 200 *node* dapat dilihat pada **Tabel 5.8**.

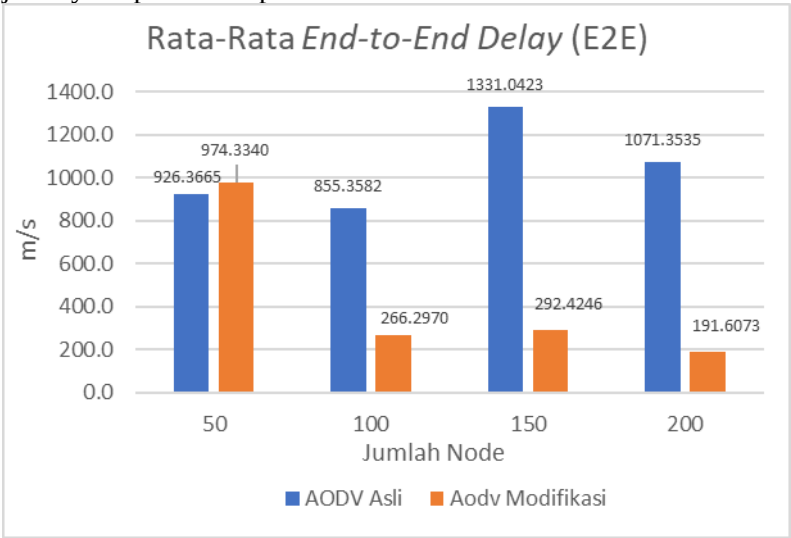
**Tabel 5.8** Hasil Rata-Rata E2E Skenario *Real*

| ***Average End-to-End Delay* (E2E)** | | | | |
|---|---|---|---|---|
| **Node** | **AODV Asli** | **AODV Modifikasi** | **Penurunan** | **Prosentase** |
| 50 | 926,3665 | 974,3340 | -47,9675 | -5,18% |
| 100 | 855,3582 | 266,2970 | 589,0612 | 68,87% |
| 150 | 1331,0423 | 292,4246 | 1038,6177 | 78,03% |
| 200 | 1071,3535 | 191,6073 | 879,7462 | 82,12% |

Berdasarkan **Tabel 5.8** menunjukkan bahwa *routing protocol* AODV Asli dan *routing protocol* AODV yang sudah dimodifikasi mengalami kenaikan dan penurunan pada nilai *end-to-end delay* seiring bertambahnya jumlah *node*. Pada lingkungan yang jarang dengan jumlah 50 *node*, menghasilkan nilai *end-to-end delay* sebesar -47,9675, atau naik menjadi sekitar 5,18% dimana *routing protocol* AODV yang telah dimodifikasi tidak terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 100 node, menghasilkan penurunan nilai *end-to-end delay* sebesar 589,0612, atau turun menjadi sekitar 68,87% dimana routing protocol AODV yang telah dimodifikasi terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang sedang dengan jumlah 150 node, menghasilkan penurunan nilai *end-to-end delay* sebesar 1038,6177, atau turun menjadi sekitar 78,03% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut. Pada lingkungan yang padat dengan jumlah 200 node, menghasilkan penurunan nilai *end-to-end delay* sebesar 879,7462, atau turun menjadi sekitar 82,12% dimana routing protocol AODV yang telah dimodifikasi juga terbukti unggul berdasarkan hal tersebut.

Dapat dilihat rata-rata penurunan yang terjadi untuk nilai *end-to-end delay* adalah 58,78%. Jika ketiga lingkungan tersebut dibandingkan dapat dilihat bahwa dengan jumlah *node* yang lebih

sedikit atau pada lingkungan dengan *node* yang jarang, menghasilkan nilai *end-to-end delay* yang lebih bagus daripada di lingkungan dengan jumlah *node* yang sedang maupun yang padat baik untuk *routing protocol* AODV asli maupun *routing protocol* AODV yang telah dimodifikasi. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan nilai *end-to-end delay* yang lebih bagus dibandingkan AODV asli dengan penurunan nilai E2E yang cukup signifikan, walaupun demikian nilai *end-to-end delay* jika semakin besar maka akan menyebabkan tidak sampainya paket pada node atau mengalami delay pengiriman paket, sehingga kinerja terbaik AODV akan lebih bagus jika nilai *end-to-end delay* yang dihasilkan semakin turun, untuk lebih jelasnya dapat dilihat pada **Gambar 5.6**.




**Gambar 5.6** Grafik End-to-End Delay Skenario *Real*

### 5.2.3 Hasil Uji Coba Fungsionalitas

Simulasi yang telah dijalankan oleh NS-2 hingga proses analisis menggunakan skrip AWK menghasilkan keluaran berupa nilai metrik analisis, namun untuk membuktikan bahwa yang dipilih adalah rute terbaik pada modifikasi aodv maka hasilnya dapat dibuktikan lewat keluaran file python yang menampilkan kestabilan rute, file python tersebut dinamai trace.py yang dapat dilihat pada **lampiran H.1**. Hasil uji coba yang akan dituliskan pada buku ini hanya berupa potongan baris kode yang mewakilkan 50 node, 100 node, 150 node dan 200 node pada skenario *real*.

```
78 ['_48_', '_17_', '_33_', 'DROP-CBK', 'DROP-RET']
79 ['_48_', '_17_', '_33_', '_9_', 'DROP-COL', '_23_', '_4_', '_26_', '_49_']
80 ['_48_', '_17_', '_33_', '_9_', '_23_', '_4_', '_26_', '_49_']
81 ['_48_', '_17_', '_33_', 'DROP-CBK', 'DROP-RET']
82 ['_48_', '_17_', '_8_', '_18_', '_30_', '_28_', '_1_', '_49_']
83 ['_48_', '_17_', '_8_', '_18_', '_30_', '_28_', 'DROP-RET', '_0_', '_3_', '_49_']
```

**Kode Sumber 5.2.3** Potongan Hasil Trace.py 50 *Node* AODV Asli

```
78 ['_48_', '_33_', '_9_', '_44_', '_26_', '_49_']
79 ['_48_', '_33_', '_9_', '_44_', '_26_', '_49_']
80 ['_48_', '_33_', '_9_', '_44_', '_26_', '_49_']
81 ['_48_', '_33_', '_9_', '_44_', '_26_', '_49_']
82 ['_48_', '_33_', '_9_', '_44_', '_26_', '_49_']
83 ['_48_', '_33_', '_9_', '_44_', '_26_', '_49_']
```

**Kode Sumber 5.2.4** Potongan Hasil Trace.py 50 *Node* AODV Modifikasi

```
170 ['_98_', '_20_', '_27_', '_7_', '_17_', '_99_']
171 ['_98_', '_20_', '_27_', 'DROP-CBK', 'DROP-RET']
172 ['_98_', '_20_', '_27_', 'DROP-COL', '_2_', '_7_', '_17_', '_99_']
173 ['_98_', '_20_', '_27_', '_71_', '_60_', '_46_', '_99_']
174 ['_98_', '_20_', '_27_', '_71_', '_60_', '_46_', '_99_']
175 ['_98_', '_20_', '_27_', '_71_', '_60_', '_46_', '_99_']
```

**Kode Sumber 5.2.5** Potongan Hasil Trace.py 100 *Node* AODV Asli

```
170 ['_98_', '_57_', '_33_', '_27_', '_78_', '_34_', '_14_', '_99_']
171 ['_98_', '_57_', '_33_', '_27_', '_78_', '_34_', '_14_', '_99_']
172 ['_98_', '_57_', '_33_', '_27_', '_78_', '_34_', '_14_', '_99_']
173 ['_98_', '_57_', '_33_', '_27_', '_78_', '_34_', '_14_', '_99_']
174 ['_98_', '_57_', '_33_', '_27_', '_78_', '_34_', '_14_', '_99_']
175 ['_98_', '_57_', '_33_', '_27_', '_78_', '_34_', '_14_', '_99_']
```

**Kode Sumber 5.2.6** Potongan Hasil Trace.py 100 *Node* AODV Modifikasi

```
126 ['_148_', '_28_', '_122_', '_109_', '_62_', '_78_', 'DROP-RET', 'DROP-TTL']
127 ['_148_', '_28_', '_122_', '_109_', '_62_', '_78_', 'DROP-TTL']
128 ['_148_', '_28_', '_122_', '_109_', '_62_', '_78_', 'DROP-TTL']
129 ['_148_', '_28_', '_122_', '_109_', '_62_', '_78_', 'DROP-TTL']
130 ['_148_', '_28_', '_122_', '_109_', '_62_', 'DROP-RET', '_89_', '_51_', '_11_', '_26_', '_149_']
131 ['_148_', '_28_', '_122_', '_109_', 'DROP-CBK', 'DROP-RET']
```

**Kode Sumber 5.2.7** Potongan Hasil Trace.py 150 *Node* AODV Asli

```
126 ['_148_', '_28_', '_122_', '_126_', '_78_', '_26_','_149_']
127 ['_148_', '_28_', '_122_', '_126_', '_78_', '_26_','_149_']
128 ['_148_', '_28_', '_122_', '_126_', '_78_', '_26_','_149_']
129 ['_148_', '_28_', '_122_', '_126_', '_78_', '_26_"_149_']
130 ['_148_', '_28_', '_122_', '_126_', '_78_', '_26_','_149_']
131 ['_148_', '_28_', '_122_', '_126_', '_78_', '_26_','_149_']
```

**Kode Sumber 5.2.8** Potongan Hasil Trace.py 150 *Node* AODV Modifikasi

```
74 ['_198_', '_158_', '_133_', '_40_', '_99_', '_80_', '_199_']
75 ['_198_', '_158_', '_133_', '_40_', '_99_', '_80_', '_199_']
76 ['_198_', '_158_', '_133_', '_40_', '_99_', '_80_', '_199_']
77 ['_198_', '_158_', '_133_', '_40_', '_99_', '_80_', '_199_']
78 ['_198_', '_158_', '_133_', '_40_', 'DROP-CBK', 'DROP-RET']
79 ['_198_', '_62_', 'DROP-COL', 'DROP-COL', '_175_', '_85_', '_111_', '_108_', '_199_']
```

**Kode Sumber 5.2.9** Potongan Hasil Trace.py 200 *Node* AODV Asli

```
74 ['_198_', '_83_', '_107_', '_45_', '_47_', '_143_', '_199_']
75 ['_198_', '_83_', '_107_', '_45_', '_47_', '_143_', '_199_']
76 ['_198_', '_83_', '_107_', '_45_', '_47_', '_143_', '_199_']
77 ['_198_', '_83_', '_107_', '_45_', '_47_', '_143_', '_199_']
78 ['_198_', '_83_', '_107_', '_45_', '_47_', '_143_', '_199_']
79 ['_198_', '_83_', '_107_', '_45_', '_47_', '_143_', '_199_']
```

**Kode Sumber 5.2.10** Potongan Hasil Trace.py 200 *Node* AODV Modifikasi

Potongan hasil trace.py diatas menggambarkan kestabilan rute yang dicapai pada aodv modifikasi dibandingkan aodv asli pada jumlah node sebesar 50 node, 100 node, 150 node dan 200 node, selain itu untuk membuktikan kestabilan rute dapat juga melihat dari total *drop* paket yang dihasilkan, yang dapat dilihat pada **Tabel 5.9**.

**Tabel 5.9 Hasil Uji Fungsionalitas**

| **Simulasi** | | **Total drop paket** |
|---|---|---|
| **50 node** | **Asli** | 48 paket |
| | **Modifikasi** | 28 paket |
| | | |
| **100 node** | **Asli** | 60 paket |
| | **Modifikasi** | 21paket |
| | | |
| **150 node** | **Asli** | 34 paket |
| | **Modifikasi** | 9 paket |
| | | |
| **200 node** | **Asli** | 68 paket |
| | **Modifikasi** | 10 paket |

*(Halaman ini sengaja dikosongkan*

# BAB VI
# KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang diperoleh dari tugas akhir yang telah dikerjakan dan saran terkait pengembangan dari tugas akhir ini yang dapat dilakukan pada masa yang akan datang.

## 6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil uji coba dan evaluasi pada tugas akhir ini adalah sebagai berikut:

1. Semakin besarnya total residu energi yang terpilih pada seleksi rute *routing protocol* AODV semakin kecil Average End-to-End Delay yang dihasilkan.

2. Dampak Implementasi algoritma kondisi *Resenergy* terhadap performa protokol AODV pada skenario *grid* adalah rata – rata kenaikan *Packet Delivery Ratio* (PDR) sebesar 8,30%, rata–rata penurunan *Routing Overhead (RO)* sebesar 21,78%, rata–rata penurunan *End-to-end Delay* (E2E) sebesar 58,88%.

3. Dampak Implementasi algoritma kondisi *Resenergy* terhadap performa protokol AODV pada skenario *real* adalah rata–rata kenaikan *Packet Delivery Ratio* (PDR) sebesar 14,45%, rata–rata penurunan *Routing Overhead (RO)* sebesar 39,19%, rata–rata penurunan *End-to-end Delay* (E2E) sebesar 58,78%.

## 6.2 Saran

Saran yang diberikan dari hasil uji coba dan evaluasi pada tugas akhir ini adalah sebagai berikut:

1. Untuk menghasilkan hasil uji coba yang lebih baik, dapat dilakukan penentuan posisi dan pergerakan *node* yang lebih teratur atau stabil.

2. Ide untuk implementasi algoritma kondisi *Resenergy* sudah bagus, kedepannya bisa melakukan implementasi lain dengan menambahkan aspek baru seperti nilai minimum energi yang harus dimiliki *node* , penyimpanan rute pada *routing table*, dan lain sebagainya.

*(Halaman ini sengaja dikosongkan)*

# BAB VII DAFTAR PUSTAKA


[1] J. J. Mulcahy, Huang Shihong and I. Mahgoub, "Autonomic Computing and VANET," in *ResearchGate*, Florida, 2015.

[2] P. Paranavithana and A. Jayakody, "Compromising AODV for Better Performance : Improve Energy Efficiency in AODV," in *National Conference on Technology and Management (NCTM)*, Malabe, Sri Lanka, 2017.

[3] A. Virgono, "SIMULASI DAN ANALISIS PERBANDINGAN PERFORMANSI ROUTING PROTOCOL AODV & DSR," in *e-Proceeding of Engineering*, Jakarta, 2016.

[4] R. G. Engoulou, M. Bellaiche, S. Pierre and A. Quintero, "VANET Security Surveys," *Computer Communication,* vol. 44, p. 2, 2014.

[5] N. Singh, "Network Simulator NS2-2.35," in *International Journal of Advanced Research in Computer Science and Software Engineering*, Jaipur, 2012.

[6] A. A. V, K. B. W and W. P. J, "The AWK Programming Language," in *Addison-Wesley*, Boston, 1988.

[7] M. Haklay and P. Weber, "openstreetMap:User-Generated street Maps," in *PERVASIVE computing*, London, 2008.

[8] E. Borgia, "Effects of Unstable Links on AODV Performance In Real Testbades," in *Hindawi Publishing Corporation*, Pisa, 2007.

*Halaman ini sengaja dikosongkan)*

# LAMPIRAN

## A.1 Kode Fungsi AODV::recvReply ()

```
void AODV::recvReply(Packet *p) {

  struct hdr_ip *ih = HDR_IP(p);
  struct hdr_aodv_reply *rp =
  HDR_AODV_REPLY(p);
  aodv_rt_entry *rt;
  char suppress_reply = 0;
  double delay = 0.0;

  iNode =    (MobileNode *)
  (Node::get_node_by_address (index) );
  ((MobileNode *)iNode)-
  >getLoc(&xpos,&ypos,&zpos);
  iEnergy =  iNode->energy_model()-
  >energy();

  if(ih->daddr() == index
  {
    if(rp->rp_resenergy < iEnergy)
      Packet::free(p);
    else
      iEnergy = rp->rp_resenergy;
  }
  else
    rp->rp_resenergy+=iEnergy;

  rt = rtable.rt_lookup(rp->rp_dst);
  if(rt == 0) rt =
  rtable.rt_add(rp>rp_dst);
```

```
  if ( (rt->rt_seqno < rp-
>rp_dst_seqno)||
      ( (rt->rt_seqno == rp->rp_dst_seqno)
      && (rt->rt_hops > rp->rp_hop_count))
  {
      rt_update(rt, rp->rp_dst_seqno, rp-
      >rp_hop_count, rp->rp_src,
      CURRENT_TIME + rp->rp_lifetime);
      rt->rt_req_cnt = 0;
      rt->rt_req_timeout = 0.0;
      rt->rt_req_last_ttl = rp
      >rp_hop_count;

      if (ih->daddr() == index) {
      rt>rt_disc_latency[(unsigned
      char)rt->hist_indx] = (CURRENT_TIME
      - rp->rp_timestamp)
      / (double) rp->rp_hop_count;
       rt->hist_indx = (rt->hist_indx + 1)
      % MAX_HISTORY;
    }
  Packet *buf_pkt;
  while((buf_pkt = rqueue.deque(rt
  >rt_dst))){
      if(rt->rt_hops != INFINITY2) {
        assert (rt->rt_flags == RTF_UP);
        // Delay them a little to help
         ARP. Otherwise ARP
        // may drop packets. -SRD 5/23/99
        forward(rt, buf_pkt, delay);
        delay += ARP_DELAY;
      }
    }
  }
```

```
  else suppress_reply = 1;

    if(ih->daddr() == index ||
    suppress_reply) Packet::free(p);
    else {
    aodv_rt_entry *rt0
    =rtable.rt_lookup(ih->daddr());

    if(rt0 && (rt0->rt_hops != INFINITY2))
    {
      assert (rt0->rt_flags == RTF_UP);
      rp->rp_hop_count += 1;
      rp->rp_src = index;
      forward(rt0, p, NO_DELAY);

      rt->pc_insert(rt0->rt_nexthop);
    }
    else {
      drop(p, DROP_RTR_NO_ROUTE);
    }
  }
}
```

**B.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)      1500;
set opt(y)      1500;
set val(ifqlen) 1000;
set val(nn)     200;
set val(seed)   1.0;
set val(adhocRouting)   AODV;
set val(stop)   200;
set val(cp)      "cbr200.tcl";
set val(sc)      "scenario.tcl";

set ns_          [new Simulator]
set topo         [new Topography]
set tracefd      [open result-en.tr w]
set namtrace     [open result-en.nam w]

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

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

$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) \
     -energyModel $val(energy_mod) \
     -InitialEnergy $val(energy_init) \
     -txPower $val(tx_power) \
     -rxPower $val(rx_power) \
     -idlePower $val(idle_power) \
     -sleepPower $val(sleep_power) \
     -topoInstance $topo \
     -agentTrace ON \
     -routerTrace ON \
     -macTrace ON \
     -movementTrace ON \

Phy/WirelessPhy set  RXThresh_ 5.57189e-
11 ;
Phy/WirelessPhy set  CSThresh_ 5.57189e-
11 ;

for {set i 0} {$i < $val(nn)} {incr i} {
     set node_($i) [$ns_ node]
     $node_($i) random-motion 0;
}

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

puts "Loading scenario file..."
source $val(sc)
```

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

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i)
reset";
}

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

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

### C.1 Kode Konfigurasi *Traffic*

```
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(198) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(199) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 1000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 0 "$cbr_(0) start"
```

### D.1 Kode Skrip AWK *Packet Delivery 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;
}
```

### E.1 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;
```

### F.1 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;
         }
    }
```

## G.1 Kode *File* Trace.py

```
import collections
import sys
import operator

# Usage: python trace_cbr.py <tracefile.tr>
filename = sys.argv[1]

lines = [line.rstrip('\n') for line in
open(filename)]
links = dict()
drop_cbk = 0
drop_nrte = 0
drop_arp = 0
drop_ttl = 0
drop_col = 0
drop_end = 0
for line in lines:
  line = line.split()
  node = line[2]
  packet_id = line[5]
  layer = line[6]

  if layer == "cbr":packet_id = int(packet_id)
      if packet_id in links:
            if node not in links[packet_id]:
                links[packet_id].append(node)
      else:
            links[packet_id] = [node]
      if line[0] == "D":
            drop_cause = "DROP-" + line[4]
```

```
links[packet_id].append(drop_cause)
      if drop_cause == "DROP-CBK":
            drop_cbk += 1
      elif drop_cause == "DROP-NRTE":
            drop_nrte += 1
      elif drop_cause == "DROP-ARP":
            drop_arp += 1
      elif drop_cause == "DROP-TTL":
            drop_ttl += 1
      elif drop_cause == "DROP-COL":
            drop_col += 1
      elif drop_cause == "DROP-END":
            drop_end += 1

links_sorted =
collections.OrderedDict(sorted(links.items()))
for key, value in links_sorted.iteritems():
      print key, value
print "Drop CBK: ", drop_cbk
print "Drop NRTE: ", drop_nrte
print "Drop IFQ ARP: ", drop_arp
print "Drop TTL: ", drop_ttl
print "Drop END: ", drop_end
print "Total Drop: ", (drop_cbk + drop_nrte +
drop_arp + drop_ttl + drop_end)
```

## H.1 Tabel Hasil Skenario *Grid* 50 *Node*

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.8794 | 29.6914 | 135.1570 | 2032.1846 |
| | **Modifikasi** | 0.8601 | 34.7349 | 936.4500 | 2017.8898 |
| | | | | | |
| **2** | **Asli** | 0.7677 | 29.3421 | 154.7800 | 2035.0958 |
| | **Modifikasi** | 0.8942 | 24.6452 | 437.4110 | 2063.8828 |
| | | | | | |
| **3** | **Asli** | 0.7677 | 31.3026 | 902.6160 | 2029.1668 |
| | **Modifikasi** | 0.8350 | 32.5389 | 492.8200 | 2039.4064 |
| | | | | | |
| **4** | **Asli** | 0.7650 | 17.2810 | 731.0170 | 2049.0925 |
| | **Modifikasi** | 0.7868 | 21.2645 | 520.5270 | 2050.2558 |
| | | | | | |
| **5** | **Asli** | 0.8214 | 22.0124 | 488.8870 | 2048.8724 |
| | **Modifikasi** | 0.8934 | 22.8068 | 543.5390 | 2053.6158 |
| | | | | | |
| **6** | **Asli** | 0.8485 | 22.2857 | 86.7704 | 2061.8005 |
| | **Modifikasi** | 0.9119 | 24.2841 | 604.1900 | 2052.9904 |
| | | | | | |
| **7** | **Asli** | 0.4314 | 37.6591 | 735.1800 | 2240.6788 |
| | **Modifikasi** | 0.4706 | 32.5833 | 458.6770 | 2237.7002 |
| | | | | | |
| **8** | **Asli** | 0.5000 | 37.4257 | 187.2720 | 2093.8480 |
| | **Modifikasi** | 0.8670 | 33.2386 | 132.9490 | 2018.5705 |
| | | | | | |
| **9** | **Asli** | 0.7949 | 27.3290 | 890.7710 | 2072.9689 |
| | **Modifikasi** | 0.7659 | 22.0637 | 282.8020 | 2038.0262 |
| | | | | | |
| **10** | **Asli** | 0.8528 | 25.0595 | 408.5710 | 2082.7958 |
| | **Modifikasi** | 0.9069 | 21.0162 | 595.8410 | 2059.9089 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.7429** | **27.9389** | **472.1021** | **2074.6504** |
| | **Modifikasi** | **0.8192** | **26.9176** | **500.5206** | **2081.6957** |

## I.1 Tabel Hasil Skenario Grid 100 Node

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.7696 | 55.1465 | 1157.1200 | 4060.4462 |
| | **Modifikasi** | 0.7980 | 73.3418 | 178.0380 | 4009.4756 |
| | | | | | |
| **2** | **Asli** | 0.8960 | 50.221 | 210.4380 | 4066.9305 |
| | **Modifikasi** | 0.8291 | 67.5636 | 139.5340 | 4038.6734 |
| | | | | | |
| **3** | **Asli** | 0.8308 | 58.1728 | 639.7180 | 4056.1307 |
| | **Modifikasi** | 0.9077 | 43.0621 | 233.4680 | 4066.9404 |
| | | | | | |
| **4** | **Asli** | 0.8235 | 37.6429 | 1153.4900 | 4045.2562 |
| | **Modifikasi** | 0.8873 | 44.6409 | 190.6350 | 4067.4851 |
| | | | | | |
| **5** | **Asli** | 0.7424 | 62.9252 | 1622.1100 | 4016.9490 |
| | **Modifikasi** | 0.9059 | 44.3388 | 94.2784 | 4096.0709 |
| | | | | | |
| **6** | **Asli** | 0.7512 | 48.6225 | 611.0500 | 4033.3677 |
| | **Modifikasi** | 0.8049 | 35.2727 | 496.7800 | 4068.0731 |
| | | | | | |
| **7** | **Asli** | 0.8227 | 60.7545 | 374.4490 | 4007.2759 |
| | **Modifikasi** | 0.9100 | 35.0769 | 167.0320 | 4116.5590 |
| | | | | | |
| **8** | **Asli** | 0.8010 | 70.8323 | 734.6150 | 4024.3475 |
| | **Modifikasi** | 0.7665 | 51.6556 | 472.4340 | 4050.7278 |
| | | | | | |
| **9** | **Asli** | 0.8049 | 54.5212 | 244.0790 | 4060.4462 |
| | **Modifikasi** | 0.8850 | 46.2938 | 119.5530 | 4009.4756 |
| | | | | | |
| **10** | **Asli** | 0.8041 | 63.5897 | 3106.3000 | 4020.8246 |
| | **Modifikasi** | 0.8776 | 53.1628 | 195.3710 | 4022.4717 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.7920** | **27.9389** | **1080.0051** | **4038.5490** |
| | **Modifikasi** | **0.8541** | **26.9176** | **232.5627** | **4051.6755** |

## J.1 Tabel Hasil Skenario Grid 150 Node

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.8794 | 105.9778 | 385.4850 | 5976.7850 |
| | **Modifikasi** | 0.8601 | 77.4674 | 88.9926 | 5979.7570 |
| | | | | | |
| **2** | **Asli** | 0.7677 | 87.9779 | 185.4450 | 6044.3425 |
| | **Modifikasi** | 0.8942 | 73.0695 | 515.3500 | 6034.8817 |
| | | | | | |
| **3** | **Asli** | 0.7677 | 97.2152 | 854.2360 | 6004.8492 |
| | **Modifikasi** | 0.8350 | 49.0430 | 189.4400 | 6031.7416 |
| | | | | | |
| **4** | **Asli** | 0.7650 | 111.3750 | 360.5600 | 5989.6879 |
| | **Modifikasi** | 0.7868 | 61.7486 | 57.8880 | 6043.4276 |
| | | | | | |
| **5** | **Asli** | 0.8214 | 133.2653 | 1139.6300 | 6036.4748 |
| | **Modifikasi** | 0.8934 | 79.8324 | 98.0770 | 6011.0864 |
| | | | | | |
| **6** | **Asli** | 0.8485 | 97.0506 | 1323.9500 | 6043.7842 |
| | **Modifikasi** | 0.9119 | 99.5867 | 471.1270 | 5985.2475 |
| | | | | | |
| **7** | **Asli** | 0.4314 | 80.1429 | 2221.5000 | 6034.6617 |
| | **Modifikasi** | 0.4706 | 70.4022 | 191.7730 | 6066.7327 |
| | | | | | |
| **8** | **Asli** | 0.5000 | 71.7667 | 637.2590 | 6085.2119 |
| | **Modifikasi** | 0.8670 | 33.2258 | 497.9170 | 6063.4696 |
| | | | | | |
| **9** | **Asli** | 0.7949 | 92.7229 | 929.8250 | 6008.3394 |
| | **Modifikasi** | 0.7659 | 97.1395 | 512.1770 | 6010.3915 |
| | | | | | |
| **10** | **Asli** | 0.8528 | 77.9075 | 1532.0200 | 5981.5973 |
| | **Modifikasi** | 0.9069 | 76.9497 | 271.4040 | 6023.4510 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.8138** | **95.5402** | **956.9910** | **6020.5734** |
| | **Modifikasi** | **0.8868** | **71.8465** | **289.4146** | **6025.0187** |

## K.1 Tabel Hasil Skenario Grid 200 Node

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.8049 | 114.5212 | 577.0330 | 7984.3228 |
| | **Modifikasi** | 0.8775 | 67.7877 | 133.1250 | 8085.4308 |
| | | | | | |
| **2** | **Asli** | 0.7931 | 143.9379 | 1533.6300 | 7987.1952 |
| | **Modifikasi** | 0.8980 | 93.4545 | 493.5040 | 7974.0268 |
| | | | | | |
| **3** | **Asli** | 0.8454 | 92.6400 | 713.7500 | 7978.5756 |
| | **Modifikasi** | 0.9366 | 62.3333 | 74.5448 | 7979.1395 |
| | | | | | |
| **4** | **Asli** | 0.8010 | 142.3515 | 823.9540 | 7978.2929 |
| | **Modifikasi** | 0.8867 | 87.1333 | 316.3080 | 7995.3045 |
| | | | | | |
| **5** | **Asli** | 0.8316 | 136.5644 | 605.8270 | 8024.6202 |
| | **Modifikasi** | 0.8636 | 100.8772 | 138.9710 | 7984.2637 |
| | | | | | |
| **6** | **Asli** | 0.8711 | 140.6272 | 336.5690 | 7970.5290 |
| | **Modifikasi** | 0.9064 | 92.5217 | 530.3360 | 7999.3834 |
| | | | | | |
| **7** | **Asli** | 0.8068 | 108.4192 | 676.2520 | 8012.0810 |
| | **Modifikasi** | 0.9109 | 64.6957 | 82.7271 | 7995.4387 |
| | | | | | |
| **8** | **Asli** | 0.8458 | 113.2118 | 1244.1100 | 7985.9535 |
| | **Modifikasi** | 0.9227 | 86.6480 | 245.4160 | 8072.2001 |
| | | | | | |
| **9** | **Asli** | 0.9208 | 94.0323 | 1141.4200 | 7995.1930 |
| | **Modifikasi** | 0.8827 | 89.5491 | 70.2693 | 8002.7884 |
| | | | | | |
| **10** | **Asli** | 0.9275 | 64.4583 | 264.6760 | 7985.7766 |
| | **Modifikasi** | 0.9000 | 76.8333 | 87.9044 | 8020.6238 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.8448** | **115.0764** | **791.7221** | **7990.2540** |
| | **Modifikasi** | **0.8985** | **82.1834** | **334.6690** | **8010.8600** |

## L.1 Tabel Hasil Skenario *Real* 50 Node

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.8634 | 21.5028 | 554.6800 | 2026.5473 |
| | **Modifikasi** | 0.8492 | 35.7515 | 1723.4100 | 2015.9517 |
| | | | | | |
| **2** | **Asli** | 0.8119 | 28.9756 | 719.2780 | 2003.0714 |
| | **Modifikasi** | 0.8515 | 30.3488 | 632.0810 | *2039.4656* |
| | | | | | |
| **3** | **Asli** | 0.7526 | 24.3973 | 689.6870 | 2055.1183 |
| | **Modifikasi** | 0.8607 | 10.8786 | 667.5380 | 2094.4502 |
| | | | | | |
| **4** | **Asli** | 0.7887 | 26.2353 | 541.4980 | 2062.2077 |
| | **Modifikasi** | 0.9036 | 20.1348 | 523.4010 | 2061.4197 |
| | | | | | |
| **5** | **Asli** | 0.5459 | 32.6903 | 1186.2600 | 2017.1173 |
| | **Modifikasi** | 0.6332 | 23.3333 | 1621.5400 | 2068.7325 |
| | | | | | |
| **6** | **Asli** | 0.8995 | 52.6341 | 2280.0300 | 2060.0418 |
| | **Modifikasi** | 0.8454 | 47.8947 | 1883.6700 | 2051.6243 |
| | | | | | |
| **7** | **Asli** | 0.8454 | 63.6311 | 1277.0100 | 2033.1182 |
| | **Modifikasi** | 0.8267 | 31.5294 | 566.2600 | 2011.4548 |
| | | | | | |
| **8** | **Asli** | 0.8593 | 16.2456 | 528.3890 | 2044.3958 |
| | **Modifikasi** | 0.8798 | 19.1366 | 127.2750 | 2032.7886 |
| | | | | | |
| **9** | **Asli** | 0.7635 | 21.4323 | 1032.3400 | 2060.1477 |
| | **Modifikasi** | 0.8144 | 14.7975 | 1520.5500 | 2047.3329 |
| | | | | | |
| **10** | **Asli** | 0.8390 | 27.5465 | 454.4930 | 2038.3619 |
| | **Modifikasi** | 0.8768 | 19.2809 | 477.6150 | 2071.1100 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.7969** | **31.5291** | **926.3665** | **2040.0127** |
| | **Modifikasi** | **0.8341** | **25.3086** | **974.3340** | **2045.4161** |

**M.1 Tabel Hasil Skenario Real 100 Node**

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.7927 | 70.7190 | 926.5600 | 4074.5818 |
| | **Modifikasi** | 0.8990 | 34.7079 | 108.5590 | *4092.9645* |
| | | | | | |
| **2** | **Asli** | 0.8250 | 78.3152 | 1031.3700 | 4016.9180 |
| | **Modifikasi** | 0.8952 | 40.4787 | 58.8460 | *4096.9523* |
| | | | | | |
| **3** | **Asli** | 0.6766 | 71.6618 | 1375.0900 | 3998.0225 |
| | **Modifikasi** | 0.8934 | 45.5455 | 144.5060 | *4084.2907* |
| | | | | | |
| **4** | **Asli** | 0.8660 | 59.6071 | 906.4110 | 4037.3191 |
| | **Modifikasi** | 0.8985 | 36.3277 | 94.5044 | *4098.9617* |
| | | | | | |
| **5** | **Asli** | 0.8431 | 68.3721 | 396.0490 | 4067.9018 |
| | **Modifikasi** | 0.8077 | 46.9524 | 258.9960 | *4092.5759* |
| | | | | | |
| **6** | **Asli** | 0.9064 | 51.5761 | 324.3260 | 4039.5063 |
| | **Modifikasi** | 0.8485 | 60.8929 | 72.2553 | 4090.8408 |
| | | | | | |
| **7** | **Asli** | 0.8010 | 52.7329 | 935.1160 | 4156.1402 |
| | **Modifikasi** | 0.8636 | 55.0760 | 302.6530 | 4226.3921 |
| | | | | | |
| **8** | **Asli** | 0.7970 | 43.2422 | 1394.9800 | 4078.0258 |
| | **Modifikasi** | 0.9067 | 46.7314 | 107.8010 | 4001.2031 |
| | | | | | |
| **9** | **Asli** | 0.8241 | 57.0854 | 702.1990 | 4081.1237 |
| | **Modifikasi** | 0.8636 | 55.1696 | 722.2240 | 3989.1822 |
| | | | | | |
| **10** | **Asli** | 0.9246 | 46.5978 | 666.2910 | 4068.3244 |
| | **Modifikasi** | 0.8911 | 40.2444 | 742.9120 | 4095.2750 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.8224** | **59.2313** | **855.3582** | **4067.5528** |
| | **Modifikasi** | **0.8771** | **45.6356** | **266.2970** | **4086.4651** |

## N.1 Tabel Hasil Skenario Real 150 Node

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.8250 | 91.2848 | 1582.3500 | 6069.0431 |
| | **Modifikasi** | 0.9340 | 55.1413 | 412.8430 | *6288.5326* |
| | | | | | |
| **2** | **Asli** | 0.8515 | 91.8488 | 1067.0000 | 6009.5039 |
| | **Modifikasi** | 0.8333 | 81.1152 | 577.4400 | 6199.1570 |
| | | | | | |
| **3** | **Asli** | 0.8308 | 82.2515 | 959.7070 | 6092.1163 |
| | **Modifikasi** | 0.9655 | 30.9490 | 60.3355 | 6096.1265 |
| | | | | | |
| **4** | **Asli** | 0.5415 | 151.2973 | 1719.6000 | 6000.9664 |
| | **Modifikasi** | 0.9239 | 43.8571 | 151.1200 | 6067.8359 |
| | | | | | |
| **5** | **Asli** | 0.6520 | 105.7744 | 1560.4000 | 6168.7394 |
| | **Modifikasi** | 0.9150 | 57.9235 | 88.0327 | 6154.9288 |
| | | | | | |
| **6** | **Asli** | 0.8829 | 65.3923 | 356.7520 | 6001.2555 |
| | **Modifikasi** | 0.9293 | 50.4891 | 498.1720 | 6010.8813 |
| | | | | | |
| **7** | **Asli** | 0.7236 | 103.2222 | 2172.2000 | 5993.6245 |
| | **Modifikasi** | 0.8643 | 64.5814 | 241.1440 | 6034.4258 |
| | | | | | |
| **8** | **Asli** | 0.8550 | 55.3450 | 1668.6200 | 6040.2735 |
| | **Modifikasi** | 0.9064 | 76.3913 | 331.3820 | 6018.6559 |
| | | | | | |
| **9** | **Asli** | 0.4184 | 151.5610 | 952.9240 | 6000.2039 |
| | **Modifikasi** | 0.9010 | 56.1648 | 162.2520 | 6000.2840 |
| | | | | | |
| **10** | **Asli** | 0.7910 | 84.2138 | 1270.8700 | 6042.0501 |
| | **Modifikasi** | 0.8856 | 80.6966 | 401.5250 | 6001.3029 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.7372** | **98.2191** | **1331.0423** | **6041.7776** |
| | **Modifikasi** | **0.9058** | **59.7309** | **292.4246** | **6087.2131** |

## O.1 Tabel Hasil Skenaro Real 200 Node

| Simulasi | | PDR | RO | E2E | Resenergy |
|---|---|---|---|---|---|
| **1** | **Asli** | 0.7688 | 115.2288 | 885.5150 | 7984.3228 |
| | **Modifikasi** | 0.9314 | 68.8421 | 83.1056 | 8085.4308 |
| | | | | | |
| **2** | **Asli** | 0.9450 | 64.0423 | 169.4470 | 7987.1952 |
| | **Modifikasi** | 0.9650 | 35.6684 | 44.9426 | 7974.0268 |
| | | | | | |
| **3** | **Asli** | 0.8621 | 85.9200 | 677.7370 | 7978.5756 |
| | **Modifikasi** | 0.9154 | 40.1739 | 39.0695 | 7979.1395 |
| | | | | | |
| **4** | **Asli** | 0.6904 | 116.2206 | 1446.3600 | 7978.2929 |
| | **Modifikasi** | 0.9122 | 76.0642 | 92.5141 | 7995.3045 |
| | | | | | |
| **5** | **Asli** | 0.8750 | 27.8857 | 313.8810 | 8024.6202 |
| | **Modifikasi** | 0.9653 | 31.4872 | 58.9735 | 7984.2637 |
| | | | | | |
| **6** | **Asli** | 0.6683 | 116.0146 | 1772.1400 | 7970.5290 |
| | **Modifikasi** | 0.9559 | 67.7538 | 221.4250 | 7999.3834 |
| | | | | | |
| **7** | **Asli** | 0.7917 | 132.7895 | 1881.4700 | 8012.0810 |
| | **Modifikasi** | 0.9750 | 26.4923 | 51.5511 | 7995.4387 |
| | | | | | |
| **8** | **Asli** | 0.3881 | 212.3846 | 1874.3400 | 7985.9535 |
| | **Modifikasi** | 0.9113 | 49.0595 | 67.7852 | 8072.2001 |
| | | | | | |
| **9** | **Asli** | 0.8060 | 74.7778 | 741.3390 | 7995.1930 |
| | **Modifikasi** | 0.9752 | 49.6954 | 189.0160 | 8002.7884 |
| | | | | | |
| **10** | **Asli** | 0.7340 | 144.4564 | 951.3060 | 7985.7766 |
| | **Modifikasi** | 0.9104 | 59.9126 | 47.4758 | 8020.6238 |

| | | | | | |
|---|---|---|---|---|---|
| **rata-rata** | **Asli** | **0.7529** | **108.9720** | **1071.3535** | **7990.2540** |
| | **Modifikasi** | **0.9417** | **50.5149** | **191.6073** | **8010.8600** |

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS

Dicky Kaisar Utomo lahir di Jakarta pada tanggal 26 April 1997. Penulis menempuh pendidikan formal di TK Al Hasanah Depok (2001-2002), TK An Nissa Depok (2002-2003), SD N Mekarjaya 29 Depok (2003-2006), SD N 1 Pengkol, Jepara (2006-2009), SMP N 1 Jepara RSBI (2009-2012), SMA N 1 Jepara RSBI (2012-2015), dan Informatika ITS Surabaya (2015-2019). Bidang studi yang diambil oleh penulis saat berkuliah di Departemen Informatika ITS adalah Arsitektur Jaringan Komputer (AJK). Penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer-Informatika pada Divisi Hubungan Luar Kabinet Optimasi (2016-2017) sebagai Staff, dan organisasi Himpunan Mahasiswa Teknik Computer-Informatika pada Divisi Hubungan Luar Kabinet Inspirasi sebagai Staff Ahli (2017-2018). Penulis juga aktif dalam kegiatan kepanitian seperti SCHEMATICS Divisi Danus pada Sub-Divisi Sponsorship sebagai Staff (2016-2017), SCHEMATICS Divisi Danus pada Sub-Divisi Sponsorship sebagai Staff Ahli (2017-2018), FTIF FESTIVAL 2017 sebagai Divisi Danus pada Sub-Divisi Sponsorship sebagai Staff Ahli, dan ITS EXPO 2016 pada Divisi Wahana Teknologi. Penulis pernah menjalani kerja praktik di PT Pertamina EP Kantor Pusat Jakarta periode Juli 2018, Selama berkuliah, penulis mengikuti ekstrakurikuler ITS Debate Club (IDS) sebagai anggota (2015-2017). Penulis dapat dihubungi melalui nomor *handphone* +629620827761 dan +627881442994 atau di email kaisardicky@gmail.com.

*(Halaman ini sengaja dikosongkan)*