

**TUGAS AKHIR – KI141502**

**IMPLEMENTASI PROTOKOL *REVERSE AD-HOC ON DEMAND DISTANCE VECTOR* DENGAN MEMPERTIMBANGKAN FAKTOR KONSUMSI ENERGI PADA TAHAP *ROUTING* DI VANETS**

**CHRISTOPHER ANDREW**

**NRP 05111640000107**

Dosen Pembimbing I

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

Dosen Pembimbing II

Ir. F.X. Arunanto, M.Sc.

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

***(Halaman ini sengaja dikosongkan)***



**TUGAS AKHIR – KI141502**

**IMPLEMENTASI PROTOKOL *REVERSE AD-HOC ON-DEMAND DISTANCE VECTOR* DENGAN MEMPERTIMBANGKAN FAKTOR KONSUMSI ENERGI PADA TAHAP *ROUTING* DI VANETS**

**CHRISTOPHER ANDREW**

**NRP 05111640000107**

**Dosen Pembimbing I**

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

**Dosen Pembimbing II**

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

**Departemen Teknik Informatika**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2020**

***(Halaman ini sengaja dikosongkan)***



**UNDERGRADUATE THESES – KI141502**

**IMPLEMENTATION OF REVERSE AD-HOC   
ON-DEMAND DISTANCE VECTOR PROTOCOL WHILE TAKING ENERGY CONSUMPTION FACTOR INTO CONSIDERATION IN THE ROUTING STAGE OF VANETS**

**CHRISTOPHER ANDREW**

**NRP 05111640000107**

**First Advisor**

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

**Second Advisor**

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

**Department of Informatics**

**Faculty of Intelligent Electrical and Informatics Technology**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2020**

***(Halaman ini sengaja dikosongkan)***

**LEMBAR PENGESAHAN**

**IMPLEMENTASI PROTOKOL *REVERSE AD-HOC   
ON-DEMAND DISTANCE VECTOR* DENGAN MEMPERTIMBANGKAN FAKTOR KONSUMSI ENERGI PADA TAHAP *ROUTING* DI VANETS**

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat

memperoleh gelar Sarjana Komputer

pada

Bidang Studi Arsitektur dan Jaringan Komputer

Program Studi S-1 Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh:

**CHRISTOPHER ANDREW  
NRP: 05111640000107**

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. ..........................  
   (NIP. 19841016 200812 1 002) (Pembimbing 1)
2. Ir. F.X. Arunanto, M.Sc. ..........................  
   (NIP. 19570101 198303 1 004) (Pembimbing 2)

**SURABAYA  
JUNI, 2020**

***(Halaman ini sengaja dikosongkan)***

**IMPLEMENTASI PROTOKOL *REVERSE AD-HOC   
ON-DEMAND DISTANCE VECTOR* DENGAN MEMPERTIMBANGKAN FAKTOR KONSUMSI ENERGI PADA TAHAP ROUTING DI VANETS**

**Nama Mahasiswa : Christopher Andrew**

**NRP : 05111640000107**

**Departemen : Informatika FTIK-ITS**

**Dosen Pembimbing 1 : Dr.Eng. Radityo Anggoro, S.Kom.,**

**M.Sc.**

**Dosen Pembimbing 2 : Ir. F.X. Arunanto, M.Sc.**

Abstrak

Salah satu bentuk jaringan nirkabel adalah Vehicular Ad-hoc Networks (VANETs). VANETs sendiri merupakan pengembangan dari bentuk jaringan nirkabel lainnya, yaitu Mobile Ad-hoc Network (MANET) yang memiliki karakteristik berupa *node-node* dengan mobilitas yang sangat tinggi dan memiliki pola pergerakan yang terbatas. Terdapat banyak *routing* *protocol* yang dapat diimplementasikan pada VANETs, salah satunya adalah Ad-hoc On-demand Distance Vector (AODV).

AODV merupakan salah satu *routing protocol* dalam klarifikasi *reactive routing protocol*, sebuah protokol *routing* yang hanya akan membuat rute informasi ketika *node* sumber memerlukannya. AODV memiliki dua tahap, yaitu *route discovery* dan *route maintenance*. *Route discovery* digunakan untuk meminta dan meneruskan informasi yang nantinya dapat menjadi sebuah rute pengiriman data. Proses pada tahap tersebut meliputi pengiriman *Route Request* (RREQ) dan *Route Reply* (RREP). *Route maintenance* digunakan untuk mendapatkan informasi bila terdapat kesalahan pada rute pengiriman data yang di dalamnya terdapat proses pengiriman *Route Error* (RERR).

Pada Tugas Akhir ini, akan dilakukan modifikasi pada proses pengiriman RREP, yang semula berupa *backtrace* pada satu rute pemgiriman data, menjadi *broadcast* sehingga menghasilkan beberapa rute pengiriman. Modifikasi ini dilakukan karena tingkat mobilitas yang sangat tinggi pada VANETs memungkinkan tidak tersampainya RREP bila hanya bergantung pada satu rute. Hal tersebut mengakibatkan *node* sumber perlu mengirimkan RREQ ulang sehingga meningkatkan *delay* dan menurunkan *packet delivery ratio (PDR)*. Modifikasi tersebut dilakukan untuk meningkatkan probabilitas penerimaan RREP oleh *node* sumber.

Pada Tugas Akhir ini, juga dilakukan modifikasi pada proses pengiriman RREQ, yaitu dengan penambahan filter *energy threshold*. Hal tersebut dilkakukan untuk membatasi ruang lingkup pengiriman RREQ dengan mengeliminasi *node* yang tidak memiliki cukup energi. Modifikasi ini diharapkan akan meningkatkan PDR.

Dari hasil uji coba, **lorem ipsum dolor et amos…**

***Kata kunci: VANETs, AODV, NS2, SUMO, Reverse-AODV, Konsumsi Energi, Energi Residu.***

**IMPLEMENTATION OF REVERSE AD-HOC ON-DEMAND DISTANCE VECTOR PROTOCOL WHILE TAKING ENERGY CONSUMPTION FACTOR INTO CONSIDERATION IN THE ROUTING STAGE OF VANETS**

**Student’s Name : Christopher Andrew**

**Student’s ID : 05111640000107**

**Department : Informatika FTIK-ITS**

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

**Second Advisor : Ir. F.X. Arunanto, M.Sc.**

*Abstract*

*VANETs is one of the wireless networks which derives from its predecessor, MANET. VANETs’ nodes have unique characteristics such as very high mobility and limited movement pattern. There are several routing protocols which can be implemented in VANETs, one of them is AODV.*

*AODV is classified as reactive routing protocolwhich mean that AODV will only create a data transfer route when the source node needed it. AODV have two stages, route discovery stage and route maintenance stage. The route discovery is used to request and forward route informations which consists of RREQ and RREP. In the other hand, route maintenance which consists of RERR, is used to find out any error in the data transfer route.*

*In this paper, there will be a modification on the AODV behaviour when sending back the RREP in which the original is done by backtrace, thus produce a single data transfer route. The modified behaviour is instead of backtrace, the protocol will broadcast tbe RREP to produce several data transfer routes. This modification is done because of the very high nodes mobility in VANETs may cause the source node cannot receive the RREP if only rely on single data route transfer. This failure of receiving the RREP will increase the end-to-end delay and reduce the PDR.*

*In this final project, there also will be a modification on the process of broadcasting the RREQ. The modification is done by adding an energy filter to a certain threshold to limit the scope of the broadcast itself. By filtering the nodes which doesn’t have enough energy, the probability of dead nodes in the middle of data transfer is eliminated, thus the PDR can be improved.*

*The evaluation shows* ***lorem ipsum dolor et amos…***

***Keyword: VANETs, AODV, NS2, SUMO, Reverse-AODV, Energy Consumption, Residual Energy.***

KATA PENGANTAR

Puji Syukur kepada Allah Yang Maha Esa atas segala berkat dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Implementasi Protokol *Reverse Ad-hoc On-demand Distance Vector* dengan Mempertimbangkan Faktor Konsumsi Energi pada Tahap *Routing* di VANETs”.**

Besar harapan penulis agar apa yang tertulis dalam buku Tugas Akhir ini dapat bermanfaat bagi banyak orang, khususnya untuk perkembangan ilmu pengetahuan saat ini.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini, tentunya sangat banyak bantuan dan dukungan yang penulis terima dari berbagai pihak. Tanpa mengurangi rasa hormat, penulis hendak menyampaikan terima kasih sebesar-besarnya pada:

1. Allah Yang Maha Kuasa atas semua berkat dan rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Augustino Stefanus dan Elizabeth Lysiana selaku papa dan mama penulis atas segala bentuk dukungan, motivasi, dan doa sehingga Tugas Akhir ini terselesaikan.
3. Catharina Angeline selaku adik penulis atas segala dukungan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
4. Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. dan Ir. F.X. Arunanto, M.Sc. selaku dosen pembimbing atas arahan dan bantuannya dalam pengerjaan Tugas Akhir ini.
5. Nurlisa Aulia Setyaningrum yang selalu sabar dalam mendengarkan keluh kesah penulis dan tetap setia dalam memberikan dukungan.
6. Irham M. Fadhil selaku teman seperjuangan penulis atas segala diskusi yang sangat membantu dalam pengerjaan Tugas Akhir ini.
7. Divisi Media dan Informasi ITS IO “Kembang Gula” yang sudah membuat penulis cukup pusing, sedih, dan senang atas tingkah lakunya.
8. Teman-teman TC 16 yang membuat penulis semangat untuk segera lulus dari perkuliahan di ITS.
9. Virus Corona yang mengakibatkan seluruh kegiatan perkuliahan dilakukan di rumah sehingga penulis dapat lebih fokus mengerjakan Tugas Akhir ini.
10. Juga tak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu penulis dalam bentuk apapun dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa masih terdapat kekurangan, kesalaham, maupun kelalaian yang penulis lakukan. Oleh karena itu, saran dan kritik yang membangun sangat dibutuhkan untuk penyempurnaan Tugas Akhir ini.

Surabaya, Juni 2020

Christopher Andrew

DAFTAR ISI

[1.1 Latar Belakang 19](#_Toc42535913)

[1.2 Rumusan Masalah 21](#_Toc42535914)

[1.3 Batasan Permasalahan 21](#_Toc42535915)

[1.4 Tujuan 21](#_Toc42535916)

[1.5 Manfaat 22](#_Toc42535917)

[1.6 Metodolodi 22](#_Toc42535918)

[1.6.1 Penyusunan Proposal Tugas Akhir 22](#_Toc42535919)

[1.6.2 Studi Literatur 22](#_Toc42535920)

[1.6.3 Analisis dan Desain Sistem 23](#_Toc42535921)

[1.6.4 Implementasi Sistem 23](#_Toc42535922)

[1.6.5 Pengujian dan Evaluasi 23](#_Toc42535923)

[1.6.6 Penyusunan Buku 23](#_Toc42535924)

[1.7 Sistematika Penulisan Laporan 24](#_Toc42535925)

[2.1 *Vehicular Ad-hoc Networks* (VANETs) 26](#_Toc42535926)

[2.2 *Ad-hoc On-demand Distance Vector* (AODV) 27](#_Toc42535927)

[2.3 *Network Simulator*-2 (NS-2) 30](#_Toc42535928)

[2.3.1 Instalasi NS-2 30](#_Toc42535929)

[2.3.2 *Trace File* 32](#_Toc42535930)

[2.4 *Open Street Map* (OSM) 33](#_Toc42535931)

[2.5 *Java Open Street Map Editor* (JOSM) 34](#_Toc42535932)

[2.6 *Simulation of Urban Mobility* (SUMO) 34](#_Toc42535933)

[2.7 AWK 36](#_Toc42535934)

[3.1 Deskripsi Umum 37](#_Toc42535935)

[3.2 Perancangan Skenario Mobilitas 38](#_Toc42535936)

[3.2.1 Perancangan Skenario *Grid* 39](#_Toc42535937)

[3.2.2 Perancangan Skenario Riil 40](#_Toc42535938)

[3.3 Perancangan Modifikasi *Routing Protocol* AODV 41](#_Toc42535939)

[3.3.1 Perancangan Penambahan Filter Energi 41](#_Toc42535940)

[3.3.2 Perancangan Modifikasi RREP 41](#_Toc42535941)

[3.4 Perancangan Simulasi pada NS-2 42](#_Toc42535942)

[3.5 Perancangan Metrik Analisis 42](#_Toc42535943)

[3.5.1 Packet Delivery Ratio (PDR) 42](#_Toc42535944)

[3.5.2 *Average End-to-End Delay* 42](#_Toc42535945)

[3.5.3 *Routing Overhead* (RO) 43](#_Toc42535946)

[4.1 Implementasi Skenario Mobilitas 44](#_Toc42535947)

[4.1.1 Skenario *Grid* 44](#_Toc42535948)

[4.1.2 Skenario Riil 44](#_Toc42535949)

[4.2 Implementasi Modifikasi Penambahan Filter Energi pada AODV *Routing Protocol* 44](#_Toc42535950)

DAFTAR ILUSTRASI

[**Gambar 2.0.1** Ilustrasi VANETs 27](#_Toc42536053)

[**Gambar 2.0.1** Ilustrasi proses pencarian rute AODV 28](#_Toc42536054)

[**Gambar 2.2.3.1.1** Potongan perintah untuk melakukan instalasi dependency NS-2 30](#_Toc42536055)

[**Gambar 2.2.3.1.2** Potongan baris kode pada file ls.h yang perlu diubah 31](#_Toc42536056)

[**Gambar 2.2.3.1.3** Potongan baris kode pada file Makefile.in yang perlu diubah 31](#_Toc42536057)

DAFTAR TABEL

[**Tabel 2.1** Detil penjelasan trace file 32](#_Toc42536079)

BAB I  
PENDAHULUAN

## Latar Belakang

Perkembangan teknologi informasi dan komunikasi pada saat ini memungkinkan riset terhadap desain berbagai macam jaringan dan implementasinya bertumbuh dengan pesat. Salah satu jaringan yang mendapat perhatian cukup besar adalah *Vehicular Ad-hoc Networks* (VANETs) yang merupakan perkembangan dari *Mobile Ad-hoc Network* (MANET). Karakteristik yang membedakan VANETs dengan MANET adalah tingkat mobilitas *node* yang sangat tinggi sehingga menyebabkan perubahan topologi dalam jaringan. Tingkat mobilitas yang sangat tinggi pada setiap *node* pada jaringan juga meningkatkan kemungkinan putusnya rute pengiriman data karena node tersebut bergerak keluar dari jangkauan sinyal transmisi.

Dalam pengiriman paket data dalam jaringan VANETs, diperlukannya *routing protocol* untuk mengatur jalannya paket -paket data tersebut. Protokol tersebut dapat dikelompokkan menjadi dua, yaitu *proactive routing* dan *reactive routing*. *Proactive* *routing protocol* yang bekerja dengan cara membuat table *routing* dan akan meng-*update* tabel tersebut setiap beberapa waktu tertentu tanpa memerhitungkan *bandwidth*, ukuran.dan beban jaringan contohnya adalah *Destination Sequence Distance Vector (DSDV).* Sebaliknya, *reactive routing protocol* akan membuat tabel *routing* ketika ada permintaan untuk pengiriman data atau terjadinya perubahan rute pada jaringan, contohnya adalah *Ad-hoc On-demand Distance Vector (AODV).*

AODV memiliki dua tahapan, yaitu *route discovery* (pencarian rute) dan *route maintenance* (pemeliharaan rute). Pemilihan rute pengiriman data yang stabil pada proses pencarian rute pada AODV sangat diperlukan untuk memastikan bahwa data telah terkirim dengan sempurna. Salah satu cara yang dapat dilakukan adalah dengan mengurangi dan mengeliminasi faktor-faktor yang menyebabkan terputusnya rute dan memilih rute yang memiliki kemungkinan kecil untuk terputus.

AODV terdiri dari proses pengiriman *Route Request (RREQ)* dan *Route Reply (RREP).* Saat mengirimkan RREQ, *node* sumber akan melakukan *broadcast* RREQ tersebut ke *node-node* tetangganya. Hal tersebut berlangsung terus-menerus hingga RREQ tersebut diterima oleh *node* tujuan. Kemudian, pada saat proses pengiriman RREP berlangsung, *node* tujuan tadi akan mengirim RREP tersebut hanya pada satu *node* yang merupakan pengirim RREQ pada tahap sebelumnya. Secara singkat, proses pengiriman RREP merupakan proses *backtrace* dari node tujuan kembali ke *node* sumber. *Route maintenance* berfungsi untuk melakukan deteksi bila terdapat *error* pada rute pengiriman data. *Node* akan mengirimkan *Route Error* *(RERR)* bila *node* tersebut mendeteksi adanya *error* pada rute, misalnya koneksi terputus.

Pada Tugas Akhir ini diusulkan suatu mekanisme *routing discovery* yang telah dimodifikasi untuk meningkatkan tingkat stabilitas rute pengiriman data pada AODV dengan melakukan penambahan filter energi pada saat pengiriman RREQ untuk mengeliminasi *node* yang tidak memiliki energi dan mengubah proses pengiriman RREP menjadi *broadcast* untuk meningkatkan probabilitas *node* sumber dapat menerima RREP.

Hasil akhir yang diharapkan adalah mengetahui perbandingan kinerja antara AODV (AODV asli), AODV dengan penambahan filter energi (E-AODV), AODV dengan pengubahan perilaku pengiriman RREP (R-AODV), dan AODV modifikasi gabungan (RE-AODV) diukur berdasarkan performansi *Packet Delivery Ratio (PDR), Routing Overhead,* dan *End-to-End Delay.*

## Rumusan Masalah

Tugas Akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana penambahan filter energi pada proses *broadcast* saat pengiriman RREQ mempengaruhi performa AODV dalam lingkungan yang dinamis?
2. Bagaimana dampak pengubahan metode pengiriman RREP menjadi *broadcast* dalam lingkungan yang dinamis secara keseluruhan?

## Batasan Permasalahan

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

1. Jaringan yang digunakan adalah jaringan *Vehicular Ad-hoc Networks* (VANETs).
2. *Routing protocol* yang digunakan adalah AODV.
3. Simulasi pengujian jaringan menggunakan *Network Simulator 2* (NS-2).
4. Pembuatan scenario uji coba menggunakan *Simulation of Urban Mobility* (SUMO)

## Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Menurunkan jumlah paket yang hilang pada jaringan
2. Menurunkan *end-to-end delay* pada saat pengiriman paket data dalam jaringan
3. Menurunkan jumlah *control packet* yang di-*broadcast* dalam jaringan.

## Manfaat

Manfaat yang diperoleh dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Memberikan informasi mengenai dampak dari penambahan filter energi pada proses *broadcast* saat pengiriman RREQ dan pengubahan metode pengiriman RREP dari *backtrace* menjadi *broadcast* terhadap kinerja AODV pada VANETs.
2. Membantu perembangan AODV dengan menjadi referensi untuk riset pada masa depan.

## Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan beberapa metodologi, yaitu:

## Penyusunan Proposal Tugas Akhir

Tahapan awal dari pembuatan Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir tersebut berisi pendahuluan, deskripsi, gagasan, dan rancangan metode-metode yang akan dibuat dalam Tugas Akhir ini. Pendahuluan tersebut terdiri dari hal yang menjadi latar belakang diajukannya Tugas Akhir, rumusan masalah yang diangkat, batasan masalah, tujuan, serta manfaat dari pembuatan Tugas Akhir ini. Selain itu, dijabarkan pula beberapa tinjauan pustaka sebagai referensi pendukung dalam pengerjaan Tugas Akhir. Terdapat pula sub-bab jadwal kegiatan yang menggambarkan rencana pengerjaan Tugas Akhir.

## Studi Literatur

Pada tahap ini, dipelajari sejumlah referensi yang diperlukan dalam melakukan untuk mencapai tujuan dari Tugas Akhir, yaitu topik bahasan yang meliputi VANETs, AODV, NS2, *Open Sreet Map* (OSM), *Java Open Street Map* (JOSM), SUMO, dan AWK

## Analisis dan Desain Sistem

Pada tahap ini dilakukan analisis dari hasil percobaan modifikasi AODV yang telah dibuat. Data yang menjadi fokus analisis berasal dari perhitungan *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay* pada pengiriman paket antar *node*. Hal ini bertujuan untuk merumuskan solusi yang tepat untuk konfigurasi AODV yang kemudian dilakukan simulasi pada VANETs dengan bantuan SUMO.

## Implementasi Sistem

Implementasi merupakan tahap untuk membangun metode-metode yang telah diajukan dalam Proposal Tugas Akhir. Pada tahap ini, dilakukan implementasi menggunakan NS-2 sebagai platform *simulator*, Bahasa pemrograman C/C++, dan SUMO sebagai *tools* untuk membantu uji coba dan implementasi desain sistem yang telah dirancang

## Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian menggunakan SUMO sebagai generator *traffic* untuk membuat simulasi keadaan topologi riil untuk dilakukan pengujian. Hasil dari SUMO tersebut akan dilankan pada NS-2 dan akan menghasilkan sebuah *trace file* (.tr). PDR, RO, dan End-to-End Delay akan dihitung dari *trace file* tersebut. Penghitungan ini dilakukan untuk menguji performa AODV yang telah termodifikasi.

## Penyusunan Buku

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

## Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. **Bab I. Pendahuluan**

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

1. **Bab II. Tinjauan Pustaka**

Bab ini berisi kajian teori atau penjelasan dari metode, algoritma, library, dan tools yang digunakan dalam penyusunan Tugas Akhir ini. Bab ini berisi tentang penjelasan singkat mengenai VANETs, AODV, NS-2, OSM, JOSM, SUMO, dan AWK.

1. **Bab III. Perancangan**

Bab ini berisi pembahasan mengenai perancangan skenario mobilitas grid dan real, perancangan simulasi pada NS-2, perancangan modifikasi AODV, serta perancangan metrik analisis (PDR, RO, dan *End-to-End Delay*).

1. **Bab IV. Implementasi**

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari proses modifikasi protokol AODV, pembuatan simulasi pada NS-2, SUMO, dan perhitungan metrik analisis.

1. **Bab V. Uji Coba dan Evaluasi**

Bab ini berisikan hasil uji coba dan evaluasi dari implementasi yang telah dilakukan untuk menyelesaikan masalah yang dibahas pada Tugas Akhir.

1. **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 solusi ke depannya.

1. **Daftar Pustaka**

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

1. **Lampiran**

Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode sumber program secara keseluruhan.

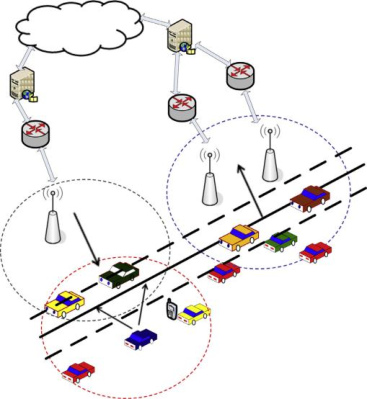
BAB II  
TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai landasan-landasan teori atau penjelasan dari metode dan *tools* yang digunakan dalam Tugas Akhir.

## 2.1 *Vehicular Ad-hoc Networks* (VANETs)

VANETs merupakan pengembangan dari jaringan nirkabel lainnya, yaitu *Mobile Ad-hoc Network* (MANET). VANETs dalam pengembangannya difokuskan pada kendaraan (*vehicle*) yang mendukung teknologi nirkabel dalam bentuk kemampuan untuk mengirim data dan saling berkomunikasi baik antar kendaraan (*vehicle to vehicle* / V2V), antara kendaraan dengan infrastruktur (*vehicle to infrastructure* / V2I), dan antara kendaraan dengan manusia (*vehicle to person* / V2P). VANETs adalah sebuah teknologi baru yang memadukan kemampuan komunikasi secara nirkabel pada kendaraan menjadi sebuah jaringan yang dapat berkomunikasi secara lansung tanpa perlunya mediasi infrastruktur serta memiliki karakteristik mobilitas yang sangat tinggi pada tiap *node* di dalamnya dan memiliki pola pergerakan yang terbatas. *Node* dalam jaringan VANETs dapat dianggap sebagai *router* yang bebas bergerak dan dapat berubah-ubah antara menjadi *client* atau *router*.

*Routing* *protocol* pada VANETs dapat dikategorikan menjadi dua model, yaitu protokol *proactive routing* yang melakukan pemeliharaan tabel *routing* secara berkala dalam kurun waktu tertentu dan protokol *reactive routing* yang hanya akan membuat tabel *routing* jika diperlukan. Pergerakan *node* dalam VANETs juga dapat berubah-ubah setiap saat namun terbatas pada rute lalu lintas yang dapat ditentukan dari koordinat peta. Hal tersebut mengakibatkan setiap *node* akan memperbarui informasi dalam tabelnya sesuai informasi dari *node* lain secara kontinyu. Perubahan pergerakan *node* pada VANETs menjadi salah satu faktor yang mendukung keberhasilan proses pengiriman paket data sehingga diperlukannya informasi jarak antar *node*, kecepatan, dan *delay* transmisi dari tiap *node.* Ilustrasi VANETs dapat dilihat pada Gambar 2.1



V2I

V2V

V2I

Roadside Infrastructure Unit

Cloud

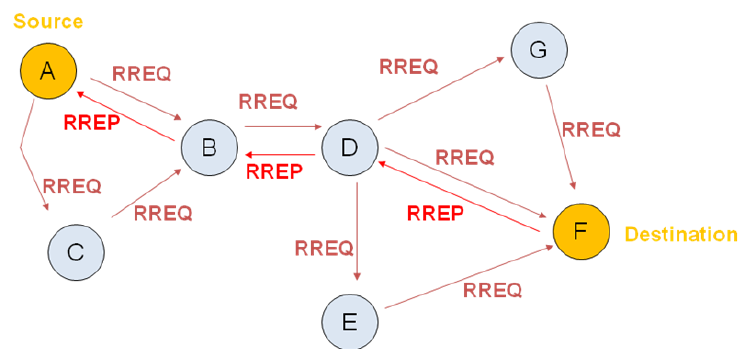
**Gambar 2.0.1** Ilustrasi VANETs

Dalam Tugas Akhir ini, penulis akan melakukan implementasi *routing protocol* AODV yang telah termodifikasi dan menguji performansi protokol tersebut pad alingkungan VANETs.

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

AODV adalah satu *routing protocol* yang termasuk dalam kategori *reactive routing protoco*, sebuah protokol yang hanya akan membuat rute pada saat diperlukan. AODV dikembangkan oleh C.E. Perkins, E.M. Belding-Royer, dan S. Das pada July 2003 di *Nokia Research Center, University of California*.

Karakteristik utama pada AODV adalah menjaga *timer-based* *state* pada setiap *node* untuk sesuai dengan penggunaan tabel *routing.* Tabel *routing* akan dianggap kadaluarsa apabila jarang digunakan.Terdapat dua tahapan dalam AODV, yaitu *route discovery* dan *route maintenance*. *Routing discovery* memiliki dua bentuk pesan yang berupa *Route Request* (RREQ) dan *Route Reply* (RREP), sedangkan *route maintenance* hanya memiliki sebuah pesan, yaitu *Route Error* (RERR).

AODV merupakan sebuah metode routing pesanantar *node* dalam sebuah jaringan nirkabel. AODV memungkinkan setiap *node* untuk melewatkan pesan mereka melalui lingkungannya kepada *node* lain yang tidak terhubung secara langsung. AODV dapat melakukan hal tersebut melalui proses pencarian rute pengiriman data yang bisa dilalui pesan agar sampai ke tujuan yang tidak berulang (*looping*). AODV juga menangani perubahan rute dan membuat rute baru apabila terdapat *error*. Ilustrasi pencarian rute oleh AODV dapat dilihat pa

**Gambar 2.0.1** Ilustrasi proses pencarian rute AODV

Pada setiap *node* pada AODV akan memiliki sebuah tabel *routing* dengan atribut sebagaiberikut:

1. *Destination Address*:alamat dari *node* tujuan pengiriman.
2. *Destination Sequence Number*: urutan *node ID* dari *node* sumber hingga *node* tujuan yang membentuk sebuah rute.
3. *Next Hop*: alamat *node* yang akan meneruskan paket data.
4. *Hop Count*: jumlah *hop* yang telah dilakukan paket data sebelum sampai ke *node* tujuan
5. *Lifetime*: waktu dalam milidetik yang diperlukan untuk *node* menerima RREP.
6. *Routing flags*: status rute. Terdapat tiga tipe status, yaitu *up* (rute valid), *down* (rute tidak valid), dan sedang diperbaiki.

Sebagai contoh proses *route discovery* dalam AODV, ilustrasi pada Gambar 2.2 menggambarkan proses *node* sumber (*node* A) mencari rute untuk *node* tujuan (*node* F). *Node A* akan membuat RREQ dan melakukan *broadcast* kepada semua *node* yang terhubung langsung dengannya (*node* tetangga). Jika *destination sequence number (*DSN*)* pada RREQ lebih besar daripada DSN pada tabel *routing* node penerima, maka atribut DSN pada tabel *routing* akan diperbarui dan paket tersebut akan diteruskan ke *node* tetangga dari *node* penerima tersebut sekaligus membuat *reverse path* kembali menuju sumber. Sebaliknya, bila DSN pada RREQ sama atau lebih kecil daripada DSN pada tabel *routing*, maka paket data tersebut akan di-*drop*. Paket RREQ akan diteruskan hingga diterima oleh *node* F. Kemudian, apabila rute dari *node* A ke *node* F telah terbentuk dalam tabel *routing* dan memiliki *routing flags* berstatus valid, maka *node* F akan mengirimkan RREP kembali menuju *node* A*.*

Dalam Tugas Akhir ini, penulis menggunakan AODV yang akan diimplementasikan pada VANETs dalam beberapa skenario.

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

*Network Simulator* (NS) adalah salah satu *interpreter* berorientasi obyek dan *discrete event-driven* yang dikembangkan oleh *University of California, Berkeley* dan USC ISI sebagai bagian dalam proyek *Virtual Internet Testbed* (VINT). NS yang banyak dikenal sebagai NS-2 (NS versi 2) berkembang menjadi *tools* yang sangat berguna untuk melakukan simulasi jaringan yang melibatkan *Local Area Network* (LAN) dan *Wide Area Network* (WAN). Beberapa tahun terakhir, fungsi NS-2 berkembang dan lebih difokuskan untuk memasukkan jaringan nirkabel (*wireless network*) dan jaringan *ad-hoc*.

Pada Tugas Akhir ini, NS-2 digunakan untuk melakukan simulasi lingkungan VANETs dengan protokol AODV yang telah dimodifikasi. NS-2 akan menghasilkan *trace file* di akhir simulasi yang dapat digunakan untuk mengukur kinerja protokol AODV yang telah termodifikasi tersebut.

## Instalasi NS-2

NS-2 dalam instalasinya memerlukan beberapa *package* yang harus sudah terinstalasi. Instalasi *dependency* yang dibutuhkan dapat dilakukan dengan *command* yang ditunjukkan pada Gambar 2.3

sudo apt-get install build-essential autoconf automake libxmu-dev gcc5

**Gambar 2.2.3.1.1** Potongan perintah untuk melakukan instalasi dependency NS-2

Setelah proses instalasi *dependency* selesai, lakukan ekstraksi pada *package* NS-2 dan ubah baris kode ke-137 pada *file* ls.h di *folder* linkstate menjadi seperti pada Gambar 2.4

Void eraseAll(){**this->**erase(baseMap::begin(), baseMap::end()); }

**Gambar 2.2.3.1.2** Potongan baris kode pada file ls.h yang perlu diubah

Kemudian ubah juga kode pada *file* Makefile.in pada folder ns-2.xx pada Gambar 2.5 (bagian – menunjukkan bagian yang harus dihapus, sedangkan bagian + menunjukkan bagian yang harus ditambah)

…

* CC = @CC@
* CPP = @CXX@
* CC = **gcc-5**
* CPP = **g++-5**
* CCOPT = @V\_CCOPT@
  + CCOPT = @V\_CCOPT@ **-g**
* DEFINE = -DTCP\_DELAY\_BIND\_ALL -DNO\_TK @V\_DEFINE@ @V\_DEFINES@ @DEFS@ -DNS\_DIFFUSION -DSMAC\_NO\_SYNC -DCPP\_NAMESPACE=@CPP\_NAMESPACE@ -DUSE\_SINGLE\_ADDRESS\_SPACE -Drng\_test
  + DEFINE = -DTCP\_DELAY\_BIND\_ALL -DNO\_TK @V\_DEFINE@ @V\_DEFINES@ @DEFS@ -DNS\_DIFFUSION -DSMAC\_NO\_SYNC -DCPP\_NAMESPACE=@CPP\_NAMESPACE@ -DUSE\_SINGLE\_ADDRESS\_SPACE -Drng\_test   
    **-DNDEBUG -DDEBUG**

…

**Gambar 2.2.3.1.3** Potongan baris kode pada file Makefile.in yang perlu diubah

## *Trace File*

*Trace file* (.tr) merupakan *file* *output* hasil simulasi yang dilakukan oleh NS-2 dan berisikan informasi detail mengenai proses pengiriman paket data selama simulasi berlangsung. *Trace file* digunakan untuk menganalisis performa protokol *routing* yang disimulasikan. *Trace file* sendiri dapat dibaca berdasarkan posisi informasi dalam nomor kolom tertentu karena setiap kolom mewakili informasi yang berbeda-beda. Detil penjelasan *trace file* terdapat pada Tabel 2.1

**Tabel 2.1** Detil penjelasan trace file

|  |  |  |
| --- | --- | --- |
| Nomor Kolom | Penjelasan | Isi |
| 1 | *Event* | s: *sent*  r: *received*  f: *forwarded*  d: *dropped* |
| 2 | *Time* | *Timestamp* terjadinya *event* |
| 3 | *Node ID* | \_x\_: dari 0 hingga banyak *node* dalam topologi |
| 4 | *Layer* | AGT: *application*  RTR: *routing*  LL: *link layer*  IFQ: *packet queue*  MAC: MAC  PHY: *physical* |
| 5 | *Flag* | --- : Tidak ada |
| 6 | *Sequence Number* | Nomor paket |
| 7 | *Packet Type* | AODV: paket *routing* AODV  cbr: berkas paket *Constant Bit Rate* (CBR)  RTS: *Request To Send* yang dihasilkan oleh MAC 802.11  CTS: *Clear* *To* *Send* yang dihasilkan oleh MAC 802.11  ACK: MAC ACK  ARP: Paket LL *Address Resolution Protocol* |
| 8 | Ukuran paket | Ukuran paket pada *layer* saat itu |
| 9 | *MAC Detail* | [a b c d]  a: perkiraan waktu paket  b: alamat penerima  c: alamat penerima  d: IPheader |
| 10 | *Flag* | --- : tidak ada |
| 11 | Detail IP *source*, *destination*, dan *nexthop* | [a:b c:d e f]  a: IP source node  b: port source node  c: IP destination node (jika -1 berarti broadcast)  d: port destination node  e: IP header ttl  f: IP nexthop (jika 0 berarti node 0 atau broadcast) |

## *Open Street Map* (OSM)

OSM adalah sebuah proyek berbasis web untuk membuat peta dunia yang bersifat gratis dan terbuka. OSM dibangun sepenuhnya oleh sukarelawan dengan melakukan survei menggunakan GPS, mendigitalisasi citra satelit dan mengumpulan serta membebaskan data geografis yang tersedia di publik. Melalui *Open Data Commons Open Database License 1.0*, kontributor OSM dapat memiliki, memodifikasi, dan membagikan data peta secara luas tanpa terhalang oleh keterbatasan legal maupun teknis.

Pada Tugas Akhir ini, data yang tersedia dalam OSM digunakan untuk membuat skenario lalu lintas berdasarkan peta daerah di Surabaya. Peta yang diambil lalu digunakan pada simulasi skenario riil pada VANETs.

## *Java Open Street Map Editor* (JOSM)

JOSM Editor adalah aplikasi untuk menyunting data yang didapat pada OSM.

Pada Tugas Akhir ini, aplikasi tersebut digunakan untuk menyunting dan merapikan peta yang didapatkan dari OSM Penyuntingan yang dilakukan meliputi menghilangkan dan menyambungkan ruas jalan yang ada serta menghilangkan gedung pada peta.

## *Simulation of Urban Mobility* (SUMO)

SUMO merupakan paket simulasi lalu lintas yang bersifat gratis dan terbuka yang mulai dikembangkan pada tahun 2001 yang kemudian berkembang menjadi simulasi lalu lintas dengan kelengkapan fitur dan permodelannya termasuk kemampuan jalannya jaringan untuk membaca format yang berbeda

SUMO juga memungkinkan pengguna untuk mendefinisikan kendaraan dengan sifat tertentu seperti ukuran kendaraan, kecepatan maksimum, percepatan, dan perlambatannya. SUMO juga menyediakan pilihan bagi pengguna untuk menentukan rute acak kendaraan. Terdapat pula pilihan model sistem transportasi umum yang memastikan kendaraan dating dan berangkat sesuai dengan jadwal.

SUMO terdiri dari beberapa *tools* yang dapat membantu pembuatan simulasi lalu lintas pada tahapan yang berbeda-beda. Adapun fungsi *tools* yang digunakan dalam pembuatan Tugas Akhir ini adalah sebagai berikut:

* **netgenerate**, merupakan *tool* yang berfungsi untuk membuat peta agar memiliki bentuk tertentu seperti *grid*, *spider*, dan *random*. Sebelum melakukan netgenerate, pengguna dapat menentukan kecepatan maksimum jalannya kendaraan dan membuat *traffic light* pada peta. Hasil dari netgenerate adalah *file* yang berekstensi .net.xml

Pada Tugas Akhir ini, netgenerate digunakan untuk membuat peta scenario *grid*.

* **netconvert,** merupakan program CLI yeng memiliki fungsi utama untuk melakukan konversi dari peta seperti OSM menjadi format *native* SUMO. Pada Tugas Akhir ini, netconvert digunakan untuk melakukan konversi peta dari *Open Street Map.*
* **randomTrips.py,** adalah sebuah *tool* dalam SUMO yang berguna untuk memproduksi rute acak yang akan dilalui oleh kendaraan (*node*) dalam simulasi.
* **duarouter,** adalah *tool* untuk melakukan perhitungan rute berdasarkan definisi atau parameter yang diberikan dan memperbaiki kerusakan pada rute.
* **sumo,** merupakan program yang melakukan simulasi lalu lintas berdasarkan data yang didapatkan baik dari netgenerate (skenario grid) maupun netconvert (skenario riil) dari randomTrips.py

Hasil simulasi dapat di-*export* ke sebuah *file* untuk selanjutnya dilakukan konversi menjadi format lain.

* **sumo-gui,** merupakan *Graphical User Interface* (GUI) untuk melihat simulasi yang dilakukan oleh SUMO secara grafis.
* **traceExporter.py,** merupakan *tool* untuk melakukan konversi *output* dari sumo menjadi format yang dapat digunakan pada simulator lain. Pada Tugas Akhir ini, traceExporter.py digunakan untuk melakukan konversi data menjadi format .txt yang dapat digunakan pada NS-2

Pada Tugas Akhir ini, SUMO digunakan untuk menghasilkan skenario pada VANETs, peta area simulasi, dan pergerakan node sehingga menyerupai keadaan lalu lintas yang sebenarnya. Untuk setiap skenario yang diproduksi oleh SUMO akan dihasilkan pergerakan *node* yang acak sehingga setiap skenario berbeda antara yang satu dengan yang lain.

## AWK

AWK adalah bahasa pemrograman yang digunakan untuk melakukan *text processing* dan ekstraksi data. AWK dalam penggunaannya merupakan sebuah program filter untuk teks, serupa dengan perintah grep pada terminal linux. AWK dapat digunakan untuk mencari bentuk/model dalam sebuah berkas teks dan mengubahnya ke dalam bentuk teks yang lain. AWK juga dapat melakukan proses aritmatika. Oleh karena itu, AWK dinilai sebagai *tool* yang cocok untuk *jobs* dan pelengkap *filter* standar.

Pada Tugas Akhir ini, AWK digunakan untuk membuat *script* yang berfungsi untuk menghitung *Packet Delivery Ratio (PDR), Routing Overhead (RO), dan End-to-End Delay* dari *trace file NS2.*

## GDB (GNU *Debugger*)

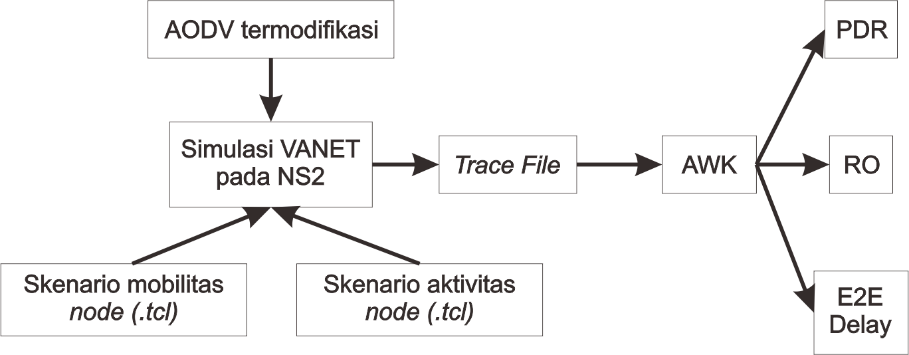
GDB adalahaplikasi yang dapat digunakan untuk melakukan *debugging* yang dapat diaplikasikan pada banyak bahasa pemrogramman, salah satunya C++ yang menjadi dasar NS-2. GDB dikembangkan oleh Richard Stallman pada tahun 1986 yang kini termasuk aplikasi perangkat lunak gratis dalam lisensi *GNU General Public License* (GPL).

BAB III  
PERANCANGAN SISTEM

Perancangan merupakan bagian penting dalam proses pembuatan suatu sistem secara teknis sehingga pada bab ini, secara khusus menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir. Berawal dari deskripsi umum sistem hingga perencangan skenario, alur, dan implementasinya.

## Deskripsi Umum

Pada Tugas Akhir ini akan diimplementasikan *routing protocol* AODV dengan memodifikasi pada bagian proses *route discovery* dan dilakukan pengujian melalui simulasi pada program NS-2. Diagram dari rancangan simulasi dari AODV asli dan AODV telah termodifikasi dapat dilihat pada Gambar 3.1

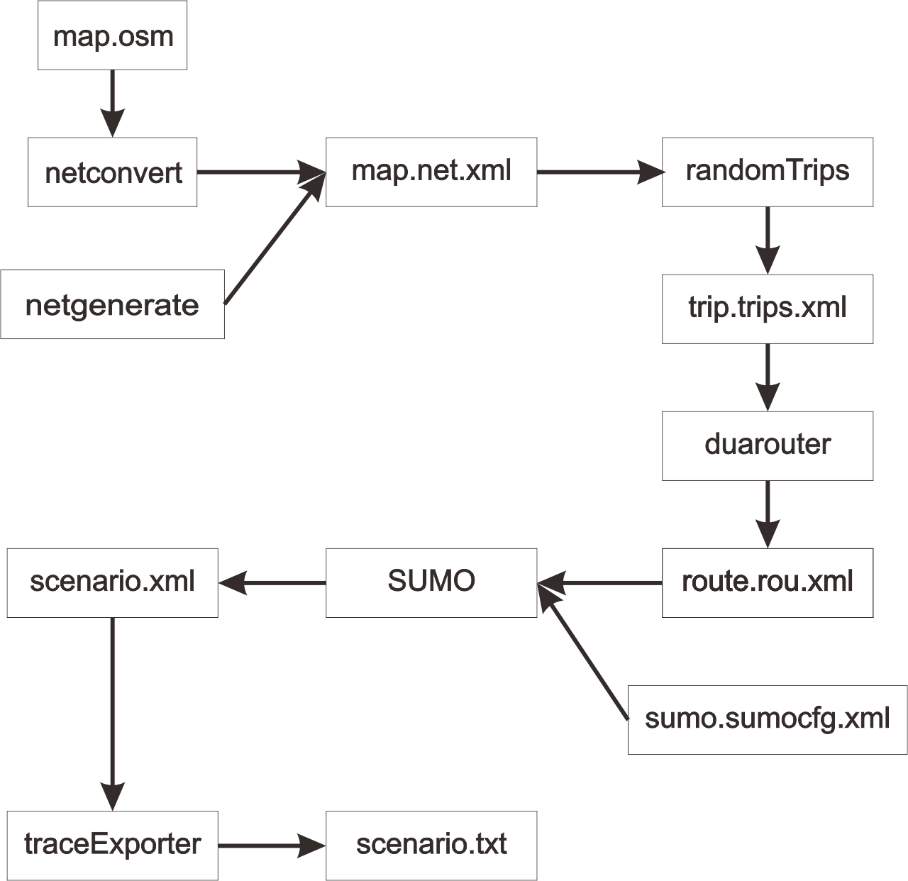


**Gambar 3.1** Diagram alur perancangan simulasi AODV termodifikasi

Modifikasi akan diawali dengan penambahan atribut *threshold* energi pada paket RREQ. Kemudian dilakukan penambahan filter energi dengan *threshold* tertentu pada saat *node* menerima pesan *broadcast* RREQ. Setiap *node* yang menerima pesan RREQ akan melakukan komparasi antara nilai pada atribut *threshold* energi pada pesan RREQ dengan energi residual *node* itu sendiri. Bila *node* tersebut tidak memiliki cukup energi, maka paket data tersebut akan di-*drop.* Lorem ipsum dolor et amos….

## Perancangan Skenario Mobilitas

Tahap ini dimulai dengan membuat area untuk simulasi, pergerakan *node*, serta implementasi dari pergerakan *node*. Dalam Tugas Akhir ini, terdapat dua macam area simulasi, yaitu peta *grid* dan peta riil. Peta *grid* merupakan contoh jalur lalu lintas yang telah disimplifikasi menjadi berpetak-petak yang digunakan sebagai simulasi awal VANETs karena lebih sederhana dan stabil. Peta ini dibangun dengan menentukan panjang dan jumlah petak yang diinginkan menggunakan SUMO. Sedangkan peta riil adalah peta asli/nyata yang digunakan sebagai area simulasi yang didapatkan melalui aplikasi *OpenS treet Map*. Pada Tugas Akhir ini, peta riil yang digunakan merupakan salah satu area di kota Surabaya, Indonesia. Alur perancangan baik peta *grid* maupun peta riil dapat dilihat pada Gambar 3.2



**Gambar 3.2** Diagram alur perancangan skenario

## Perancangan Skenario *Grid*

Perancangan skenario mobilitas *grid* diawali dengan merancang luas area peta *grid* yang dibutuhkan. Luas area tersebut bisa didapatkan dengan cara menentukan terlebih dahulu jumlah titik persimpangan yang diinginkan, sehingga dari jumlah persimpangan tersebut dapat diketahui berapa banyak peta yang dibutuhkan. Dengan mengetahui jumlah petak yang dibutuhkan, dapat ditentukan panjang tiap petak. Pada simulasi, luas area yang dibutuhkan yaitu X m X m dengan X titik persimpangan sehingga menghasilkan peta *grid* berpetak X dan panjang tiap petak adalah Xm.

Peta *grid* yang telah ditentukan datilnya tersebut kemudian diproduksi dengan menggunakan salah satu *tools* SUMO, yaitu netgenerate. Selain jumlah titik persimpangan dan panjang tiap grid, dibutuhkan juga pengaturan kecepatan kendaraan. *File* *output* yang dihasilkan oleh netgenerate tersebut akan memiliki ekstensi .net.xml yang kemudian digunakan untuk membuat serangkaian informasi pergerakan *node* secara acak menggunakan *tools* SUMO lainnya, yaitu randomTrips dan duarouter.

Skenario mobilitas *grid* dihasilkan dengan melakukan penggabungan file peta *grid* dan *file* pergerakan *node* menjadi *file* bertipe XML. *File* tersebut kemudian akan dikonversikan menjadi file TXT agar dapat diterapkan pada NS-2. Konversi tersebut dilakukan menggunakan *tool* SUMO traceExporter.

## Perancangan Skenario Riil

Perancangan skenario mobilitas riil diawali dengan memilih daerah peta yang akan digunakan sebagai area simulasi. Pada Tugas Akhir ini, *Open Street Map* digunakan untuk menghasilkan peta dari daerah yang telah ditentukan. Setelah melakukan pemilihan daerah peta yang diinginkan, dilakukan pengunduhan dengan memanfaatkan fitur *­built-in* *Open Street Map*, yaitu fitur *export* menjadi file dengan ekstensi .osm

Setelah mendapatkan peta daerah simulasi, peta tersebut dikonversi menjadi file yang berekstensi .net.xml menggunakan salah satu *tool* SUMO, yaitu netconvert. Tahapan berikutnya membuat pergerakan *node* menggunakan randomTrips dan duarouter.

Layaknya pada peta *grid*, skenario mobilitas riil dihasilkan dengan melakukan penggabungan *file* peta riil dan *file* pergerakan node menjadi *file* dengan ekstensi .xml yang kemudian dikonversi menjadi .txt dengan menggunakan traceExporter.

## Perancangan Modifikasi *Routing Protocol* AODV

Lorem ipsum dolor et amos…

## Perancangan Penambahan Filter Energi

Penambahan filter energi dilakukan dengan menambah atribut pada struktur paket RREQ. Atribut tambahan tersebut merupakan *threshold* energi minimum yang diperlukan untuk menyelesaikan proses pengiriman paket data. Kemudian pada saat *node* menerima paket RREQ, *node* tersebut akan melakukan komparasi antara nilai energi residu *node* penerima pesan RREQ dengan nilai *threshold* energi pada paket RREQ yang diterima. Apabila sama atau lebih besar, paket akan dilanjutkan seperti bias a, namun bila lebih kecil, paket RREQ tersebut akan di drop. Nilai threshold didapatkan melalui uji coba secara langsung dengan nilai tertentu. *Pseudocode* untuk melakukan filtrasi energi dapat dilihat pada Gambar 3.3

if self.residual\_energy < RREQ.energy\_threshold  
then

packet\_drop (RREQ)

endif

**Gambar 3.3.1** Pseudocode filter energi

## Perancangan Modifikasi RREP

Lorem ipsum dolor et amos

## Perancangan Simulasi pada NS-2

Simulasi VANETs pada NS-2 dilakukan dengan menggabungkan *file* skenario yang telah diproduksi menggunakan SUMO dan *script file* dengan ekstensi .tcl yang berisikan konfigurasi lingkungan simulasi.

Lorem ipsum dolor et amos

## Perancangan Metrik Analisis

Berikut ini merupakan parameter – parameter yang akan dianalisis pada Tugas Akhir ini untuk dapat membandingkan performa dari *routing protocol* AODV yang asli dengan AODV yang telah dimodifikasi:

## Packet Delivery Ratio (PDR)

PDR merupakan perbandingan jumlah paket data yang dikirim dengan jumlah paket data yang diterima. Nilai PDR yang tinggi memiliki arti proses pengiriman paket memiliki tingkat kesuksesan yang tinggi. Rumus untuk menghitung PDR dapat dilihat pada persamaan 3.1

(3.1)

Keterangan:

PDR : Packet Delivery Ratio

*received* : jumlah paket data yang diterima

*sent* : jumlah paket data yang dikirim

## *Average End-to-End Delay*

*Average End-to-End Delay* didapatkan dari rata-rata delay antara waktu paket data terkirim dan waktu paket data tersebut diterima dalam satuan milidetik. Delay seluruh paket selama simulasi dijumlahkan kemudian dibagi dengan jumlah paket yang berhasil diterima berdasarkan persamaan 3.2

(3.2)

Keterangan:

*E2E* : *End-to-End Delay*

*CBRRecvTime* : waktu *node* asal mengirimkan paket

*CBRSentTime* : waktu *node* tujuan menerima paket

*recvnum* : jumlah paket yang berhasil diterima

## *Routing Overhead* (RO)

RO merupakan jumlah paket kontrol *routing* yang ditransmisikan dalam setiap pengiriman paket data ke *node* tujuan selama simulasi terjadi. RO didapatkan dengan menjumlahkan semua paket control *routing* yang ada, baik itu RREQ, RREP, maupun RERR melalui persamaan 3.3

(3.3)

BAB IV  
IMPLEMENTASI

Pada bab ini akan dibahas mengenai proses implementasi dari perancangan sistem yang telah dibuat pada bab sebelumnya. Proses implementasi sistem tersebut berupa kode sumber untuk membangun program

## Implementasi Skenario Mobilitas

## Skenario *Grid*

## Skenario Riil

## Implementasi Modifikasi Penambahan Filter Energi pada AODV *Routing Protocol*