



**BUKU TESIS - KI2501**

## **Studi Kinerja AODV-PNT dengan Penambahan Faktor Jumlah Node Tetangga pada VANET**

**GREZIO ARIFIYAN P**

**NRP. 5115201036**

**DOSEN PEMBIMBING:**

**Prof.Ir.Supeno Djanali, M.Sc, Ph.D**

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

**PROGRAM MAGISTER  
BIDANG KEAHLIAN KOMPUTASI BERBASIS JARINGAN  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017**





BUKU TESIS - KI2501

## **AODV-PNT Performance Study with Added Factor Number of Neighbor Nodes on VANET**

GREZIO ARIFIYAN P

NRP. 5115201036

SUPERVISOR:

Prof.Ir.Supeno Djanali, M.Sc, Ph.D

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

MAGISTER PROGRAMME  
NET CENTRIC COMPUTING  
INFORMATICS ENGINEERING DEPARTMENT  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2017



Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom.)  
di  
Institut Teknologi Sepuluh Nopember Surabaya

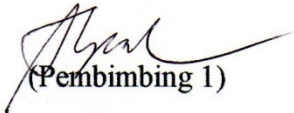
oleh:  
Grezio Arifiyan Primajaya  
Nrp. 5115201036

Dengan judul :  
Studi Kinerja AODV-PNT dengan Penambahan Faktor Jumlah Node  
Tetangga pada VANET

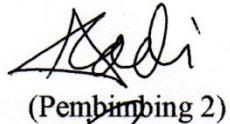
Tanggal Ujian : 13-7-2017  
Periode Wisuda : 2016 Genap

Disetujui oleh:

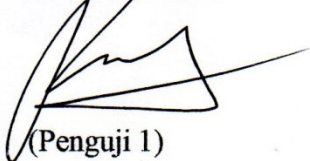
Prof.Ir.Supeno Djanali, M.Sc, Ph.D  
NIP. 194806191973011001

  
(Pembimbing 1)

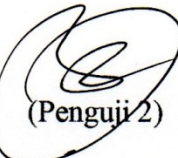
Dr.Eng. Radityo Anggoro, S.Kom, M.Sc  
NIP. 1984101620081210002

  
(Pembimbing 2)

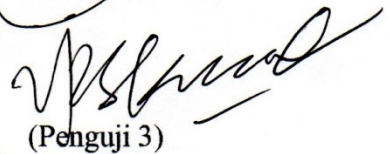
Royyana Muslim I, S.Kom, M.Kom, Ph.D  
NIP. 197708242006041001

  
(Penguji 1)

Tohari Ahmad, S.Kom, MIT, Ph.D  
NIP. 197505252003121002

  
(Penguji 2)

Waskitho Wibisono, S.Kom, M.Eng, Ph.D  
NIP. 19741022200003100

  
(Penguji 3)

  
Dekan FTIF - ITS  
Dr. Agus Zainal Arifin, S. Kom., M. Kom.  
NIP. 197208091995121001

*[Halaman ini sengaja dikosongkan]*

# STUDI KINERJA AODV-PNT DENGAN PENAMBAHAN FAKTOR JUMLAH NODE TETANGGA PADA VANET

Nama Mahasiswa : Grezio Arifiyan P  
NRP : 5115201036  
Pembimbing : Prof.Ir.Supeno Djanali, M.Sc, Ph.D  
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc

## ABSTRAK

VANET (Vehucular Ad hoc Network) merupakan pengembangan dari Mobile Ad hoc Network atau yang sering disebut MANET. Topologi pada VANET dapat berubah dengan cepat karena pergerakan simpul (node) sehingga topologi dapat berubah secara dinamis, selain itu terdapat permasalahan yang cukup menarik untuk diteliti pada VANET yaitu konektivitas antar simpul (node). Ada beberapa *routing* protokol pada MANET yang diterapkan pada VANET salah satunya adalah AODV (*On-deman Distance Protocol*) AODV adalah protokol *reactive* atau *on-deman*, jenis *routing* ini hanya akan membuat route ketika simpul sumber (*source node*) membutuhkannya.

*Routing* protokol AODV dikembangkan lebih lanjut salah satunya adalah menjadi AODV-PNT (*Prediction Node Trend*), dengan menambahkan prediksi kecenderungan simpul dapat meningkatkan performa dalam menentukan rute yang stabil, berdasarkan nilai TWR (*Total Weight of The Route*). Nilai TWR memperhatikan beberapa faktor mobilitas seperti: kecepatan, percepatan, arah dan kualitas tautan (*link quality*) antar kendaraan. Pada penelitian ini, kami mengadopsi formula perhitungan TWR pada AODV-PNT dan juga menambahkan faktor lain yaitu jumlah *node* tetangga dalam menentukan nilai TWR. Dari hasil uji coba, AODV-PNT yang dimodifikasi berhasil meningkatkan nilai *paket delivery ratio* hingga 3,7%. Sedangkan untuk jumlah *routing* paket yang dihasilkan dalam simulasi, protokol AODV-PNT modifikasi menghasilkan *routing* paket 21% lebih banyak dibandingkan dengan protokol AODV-PNT original dan 62,4% lebih sedikit dibandingkan dengan protokol AODV original. Pada penelitian ini protokol modifikasi kami sebut dengan AODV-MNS.

**Kata kunci:** MANET, VANET, AODV, AODV-PNT, TWR, *node* tetangga, AODV-MNS.

*[Halaman ini sengaja dikosongkan]*



## **AODV-PNT PERFORMANCE STUDY WITH ADDED FACTOR NUMBER OF NEIGHBOR NODES ON VANET**

Nama Mahasiswa : Grezio Arifiyan P  
NRP : 5115201036  
Pembimbing : Prof.Ir.Supeno Djanali, M.Sc, Ph.D  
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc

### **ABSTRACT**

VANET (Vehucular Ad hoc Network) is a development of Mobile Ad hoc Network or often called MANET. Topology in VANET can change rapidly because of the movement of node so that topology can change dynamically, besides there is problem interesting enough to be examined at VANET that is connectivity between node. There are several routing protocols on MANET that are applied to VANET one of them is AODV (On-deman Distance Protocol) AODV is a reactive or on-deman protocol, this type of routing will only create a route when the source node needs it.

AODV protocol routing is further developed one of them is to become AODV-PNT (Prediction Node Trend), by adding prediction of node tendency can improve performance in determining stable route, based on TWR value (Total Weight of The Route). TWR values pay attention to several mobility factors such as speed, acceleration, direction and links quality between vehicles. In this study, we adopted the TWR calculation formula on AODV-PNT and also added another factor that is the number of neighboring nodes in determining the TWR value. From the trial results, modified AODV-PNT managed to increase the value of packet delivery ratio up to 3.7%. As for the number of routing packets generated in the simulation, the modified AODV-PNT protocol produced 21% more packet routing than the original AODV-PNT protocol and 62.4% less than the original AODV protocol. In this study the modification protocol we call AODV-MNS.

**Keywords:** MANET, VANET, AODV, AODV-PNT, TWR, Neighbor nodes, AODV-MNS.

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Bismillahirrohmaniروهim.

Alhamdulillahirabil'alamin, segala puji bagi Allah Subhanahu Wata'alla, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tugas Akhir yang berjudul "*Studi Kinerja AODV-PN dengan Penambahan Faktor Jumlah Node Tetangga pada Vanet*" dengan baik dan tepat waktu.

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

1. Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Kedua orang tua penulis, yang telah memberikan dukungan moral, spiritual dan material, semangat, perhatian, selalu setia dan sabar dalam menghadapi curhatan dari penulis, serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis. Tidak lupa kakak penulis Etistika Yuni Wijaya, atas segala dukungannya berupa motivasi dan doa sehingga penulis dapat mengerjakan tesis ini
3. Bapak Dr. Tohari Ahmad, S.Kom., MIT., dan Ibu Henning Titi Ciptaningtyas, S.Kom., M.Kom selaku dosen pembimbing tugas akhir strata satu, yang telah memberikan kepercayaan, dukungan, bimbingan, nasehat, perhatian, serta semua yang telah diberikan kepada penulis.
4. Bapak Prof. Supeno Djanali, M.Sc, Ph.D. Dan pak Dr.Eng Radityo Anggoro, S.Kom, M.Sc., selaku dosen pembimbing pertama dan kedua, atas bimbingan, arahan, serta bantuan ide untuk menyelesaikan Tesis ini.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng, Ph.D, Dr.Eng. selaku ketua program studi pasca sarjana jurusan Teknik Informatika ITS, Ibu Bilqia Amalia., S.Kom., selaku dosen wali penulis, Ibu Dr.Eng. Chastine Fatichah, S.Kom, M.Kom selaku sekretaris prodi S2.
6. Pak Yudi, Pak Sugeng, Mas Jumali dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
7. Teman-teman admin lab NCC 2010 Fajri, Momo, Tasha, Adik-adik 2011 Ade, Faris, Punggi, Risal, Billa, Deasy, Nisa.
8. Kirsna dan Wisnu Bardana sebagai teman yang salalu memotivasi untuk menerapkan hidup yang lebih sehat dan warung sederhana. Mas Daglish Y yang telah membantu dalam materil dan moril, dan sebagai kakak kedua.
9. Seluruh teman Teknik Informatika ITS angkatan 2010 dan teman pasca sarjana 2015, selama bersama kalian sadar atau tidak telah membentuk karakter dan kepribadian penulis.
10. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tugas Akhir ini.

Kesempurnaan tentu sangat jauh tercapai pada Tesis ini, maka penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Juli 2017

Grezio Arifiyan

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

ABSTRAK .....	vii
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xix
DAFTAR LAMPIRAN.....	xxi
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat penelitian.....	2
1.5 Kontribusi .....	3
1.6 Batasan Penelitian.....	3
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA.....	5
2.1 VANET .....	5
2.2 Routing protocol AODV (Ad hoc On-demand Distance Vector).....	6
2.3 AODV-VANET .....	7
2.4 Prediction Node Trend (PNT).....	7
2.5 Routing protokol AODV-PNT.....	8
2.6 Neighbor Node Discovery .....	9
2.7 Path Duration pada VANET .....	10
2.7.1 Least Remaining Distance dan Shortest Path.....	10
2.7.2 Link Residual Life.....	10
2.7.3 Link Distance .....	10
2.7.4 Kepadatan node .....	11

2.7.5 Kecepatan simpul.....	11
2.7.6 Number of Hops .....	11
2.7.7 Average Progress per Hop .....	11
2.8 SUMO.....	12
2.9 NS-2.....	12
BAB 3 METODE PENELITIAN .....	13
3.1 Studi Literatur.....	13
3.2 Desain protokol AODV-MNS.....	15
3.2.1 Route packet dan Proses routing.....	17
3.2.2 Perhitungan TWR .....	19
3.2.3 Perhitungan prediksi <i>future</i> TWR.....	21
3.3 Implementasi .....	24
3.3.1 Implementasi skenario <i>grid</i> .....	25
3.3.2 Implementasi skenario <i>real</i> .....	26
3.3.3 Implementasi protokol.....	28
3.3.4 implementasi perhitungan TWR dan Future TWR.....	31
3.3.5 Implementasi simulasi pada NS-2.35 .....	33
3.4 Uji coba dan Simulasi.....	33
3.5 Analisa Hasil .....	34
3.6 Penyusunan Buku Tesis.....	34
BAB 4 HASIL DAN PEMBAHASAN .....	35
4. 1 Tahapan penelitian.....	35
4. 2 Implementasi sistem .....	35
4. 3 Parameter simulasi.....	36
4. 4 Analisa hasil .....	36
4.4.1 Hasil uji coba skenario <i>grid</i> .....	36
4.4.2 Hasil uji coba modifikasi kecepatan pada skenario <i>grid</i> .....	43
4.4.3 Hasil uji coba densitas tunggu pada skenario <i>grid</i> .....	49

4.4.4 Hasil uji coba skenario <i>real</i> .....	54
4.4.5 Hasil uji coba modifikasi kecepatan pada skenario real .....	60
4.4.6 Hasil uji coba densitas tinggi pada skenario <i>real</i> .....	65
BAB 5 Kesimpulan dan Saran .....	71
5. 1 Kesimpulan .....	71
5. 2 Saran .....	72
DAFTAR PUSTAKA .....	75
LAMPIRAN .....	77
BIODATA PENULIS .....	83

*[Halaman ini sengaja dikosongkan]*



## DAFTAR GAMBAR

Gambar 2.1 Overall framework of VANET (Rehman et al. 2013).....	5
Gambar 2.2 AODV route request (Anggoro et al. 2012).....	6
Gambar 2.3 AODV-PNT route request.....	8
Gambar 3.1 Alur penelitian.....	13
Gambar 3.2 Flowchart proses <i>routing</i> AODV .....	17
Gambar 3.3 Diagram alir proses perhitungan TWR AODV-MNT.....	18
Gambar 3.4 Diagram rancangan simulasi .....	24
Gambar 3.5 Alur pembuatan skenario <i>grid</i> .....	25
Gambar 3.6 Peta skenario <i>grid</i> .....	26
Gambar 3.7 Contoh pergerakan <i>node</i> pada peta <i>grid</i> .....	26
Gambar 3.8 Alur pembuatan skenario <i>real</i> .....	27
Gambar 3.9 Peta skenario real .....	27
Gambar 3.10 Alur modifikasi protokol AODV .....	29
Gambar 3.11 <i>Pseudocode</i> modifikasi pengiriman RREQ.....	30
Gambar 3.12 <i>Pseudocode</i> modifikasi penerimaan paket RREQ .....	31
Gambar 3.13 Perhitungan TWR.....	31
Gambar 3.14 Perhitungan <i>future</i> TWR pada file <i>aodv.cc</i> .....	32

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 3.1 Perbandingan protokol .....	14
Tabel 3.2 Metode yang diusulkan dengan AODV-PNT .....	16
Tabel 3.3 Modifikasi pesan HELLO AODV-MNS .....	19
Tabel 3.4 Pemilihan <i>relay node</i> .....	24
Tabel 4.1 Lingkungan simulasi .....	35
Tabel 4.2 Parameter lingkungan simulasi .....	36
Tabel 4.3 Hasil perhitungan rata-rata PDR(%) skenario grid .....	37
Tabel 4.4 Hasil perhitungan rata-rata <i>end-to-end delay (seconds)</i> skenario real..	38
Tabel 4.5 Hasil perhitungan rata-rata RO ( <i>packets</i> ) skenario <i>grid</i> .....	41
Tabel 4.6 Contoh path duration AODV .....	41
Tabel 4.7 Contoh path duration AODV-MNS .....	41
Tabel 4.8 Hasil perhitungan rata-rata <i>hop count</i> skenario <i>grid</i> .....	42
Tabel 4.9 Contoh pemilihan <i>relay node</i> protokol AODV-MNS.....	43
Tabel 4.10 Contoh pemilihan <i>relay node</i> protokol AODV .....	43
Tabel 4.11 Hasil rata-rata PDR (%) uji coba modifikasi kecepatan skenario grid 44	44
Tabel 4.12 Rata-rata <i>delay (seconds)</i> uji coba modifikasi kecepatan skenario <i>grid</i> .....	46
Tabel 4.13 Rata-rata RO ( <i>packets</i> ) skenario grid dengan variasi kecepatan.....	47
Tabel 4.14 Rata-rata hop skenario <i>grid</i> dengan variasi kecepatan.....	48
Tabel 4.15 Rata-rata PDR (%) uji coba densitas tinggi skenario <i>grid</i> .....	51
Tabel 4.16 Rata-rata <i>delay (seconds)</i> uji coba densitas tinggi skenario <i>grid</i> .....	52
Tabel 4.17 Rata-rata RO ( <i>packets</i> ) uji coba densitas tinggi skenario <i>grid</i> .....	52
Tabel 4.18 Rata-rata <i>hop</i> uji coba densitas tinggi skenario <i>grid</i> .....	54
Tabel 4.19 Hasil rata-rata perhitungan PDR (%) skenario <i>real</i> .....	55
Tabel 4.20 Hasil perhitungan rata-rata <i>end-to-end delay (seconds)</i> skenario <i>real</i> 57	57
Tabel 4.21 Hasil perhitungan rata-rata <i>routing overheads (packets)</i> skenario real .....	57
Tabel 4.22 Hasil perhitungan rata-rata <i>hop count</i> skenario real .....	59
Tabel 4.23 Hasil rata-rata PDR (%) uji coba modifikasi kecepatan skenario <i>real</i> 61	61
Tabel 4.24 Rata-rata <i>delay (seconds)</i> uji coba modifikasi kecepatan skenario <i>real</i> .....	62

Tabel 4.25 Rata-rata RO ( <i>packets</i> ) skenario <i>real</i> dengan variasi kecepatan .....	64
Tabel 4.26 Rata-rata hop skenario <i>real</i> dengan variasi kecepatan .....	64
Tabel 4.27 Rata-rata PDR (%) uji coba densitas tinggi skenario <i>real</i> .....	66
Tabel 4.28 Rata-rata delay (seconds) uji coba densitas tinggi skenario <i>real</i> .....	67
Tabel 4. 29 Rata-rata RO ( <i>packets</i> ) uji coba densitas tinggi skenario <i>real</i> .....	69
Tabel 4.30 Rata-rata hop uji coba densitas tinggi .....	70

## DAFTAR LAMPIRAN

Lampiran A.1 modifikasi paket RREQ pada aodv_packet.h.....	77
Lampiran A.2 modifikasi fungsi sendHello pada file aodv.cc.....	77
Lampiran A.3 penambahan atribut pada file aodv_rtable.h.....	78
Lampiran A.4 penambahan atribut paket HELLO.....	78
Lampiran A.5 penambahan atribut pada Class AODV.....	79
Lampiran A.6 modifikasi fungsi recvHello pada file aodv.cc.....	79
Lampiran A.7 implementasi simulasi NS-2.35.....	80
Lampiran A.8 perhitungan jumlah node tetangga.....	80
Lampiran A.9 perhitungan arah node terhadap node tujuan.....	81

*[Halaman ini sengaja dikosongkan]*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

VANET (*Vehicular ad-hoc Network*) sebuah teknologi baru yang digunakan untuk komunikasi antar kendaraan (*inter-vehicle*) dengan penambahan perangkat. VANET merupakan bagian penting dalam penerapan ITS (*Intelligent Transportation System*), VANET dapat digunakan antara lain untuk mengurangi resiko kecelakaan (Batool & Khan, 2005), mengatur trafik lalu lintas atau dapat membantu kendaraan yang ada di perempatan jalan (Biswas, Tatchikou & Dion 2006).

AODV (*Ad hoc On-Demand Distance Vector*) merupakan *routing* protokol yang akan membuat rute *routing* hanya ketika akan mengirimkan paket sering disebut *on-demand driven routing* protokol. Karakteristik utama VANET adalah kendaraan (*node*) yang bergerak cepat, pergerakan yang cepat tersebut dapat mempengaruhi topologi sehingga perubahan topologi dapat berubah dengan cepat.

Dengan adanya perubahan topologi yang sangat cepat maka perlu pemilihan koneksi transmisi yang lebih baik antar kendaraan (*node*) ketika pembuatan rute (*route discovery*), dengan demikian koneksi dapat berlangsung lama. AODV-PNT menambahkan beberapa faktor dalam pemilihan route untuk memilih route yang terbaik, diharapkan dengan adanya rute yang baik dapat meningkatkan PDR (*packet delivery ratio*), mengurangi kegagalan tautan (*link failure*) dan menurunkan *routing overhead* pada topologi yang dinamis.

Faktor-faktor yang digunakan sebagai perhitungan TWR pada protokol AODV-PNT masih dapat dikembangkan lagi dengan menambahkan faktor lain sehingga dapat meningkatkan kualitas dalam pemilihan *routing route*, beberapa faktor yang dapat ditambahkan antara lain adalah kekuatan sinyal setiap simpul atau kepadatan simpul tetangga (*neighbor node*) dengan penambahan parameter

tersebut diharapkan pemilihan rute *routing* dapat lebih optimal. Metode yang digunakan untuk mendeteksi node tetangga (*neighbor node*) menggunakan pesan HELLO yang dikirim secara periodik untuk mengetahui simpul tetangga (*neighbor node*) pada suatu waktu. Berdasarkan rumus (3.5) untuk perhitungan TWR terdapat *weight factor* sedangkan nilai setiap *weight factor* tidak diketahui. Pada penelitian ini modifikasi protokol AODV-PNT yang telah dimodifikasi dengan nama AODV-PNT

## **1.2 Perumusan Masalah**

Berdasarkan masalah yang diuraikan pada latar belakang, maka rumusan masalah yang akan diselesaikan adalah sebagai berikut:

1. Bagaimana penambahan faktor kestabilan node tetangga dapat memberi perubahan terhadap performa AODV-PNT.
2. Bagaimana performa modifikasi protokol AODV-PNT dibandingkan dengan AODV dan AODV-PNT original berdasarkan packet *delivery ratio* (PDR), *average end-to-end delay*, *routing overhead* (RO) dan *hop count*.

## **1.3 Tujuan Penelitian**

Tujuan yang akan dicapai dalam penelitian ini adalah Meningkatkan performa protokol AODV dengan menambahkan faktor kestabilan mobilitas node tetangga.

## **1.4 Manfaat penelitian**

Manfaat dari penelitian ini adalah Memberikan dampak penambahan faktor jumlah node tetangga pada perhitungan TWR terhadap performa protokol AODV



## **1.5 Kontribusi**

Kontribusi penelitian ini adalah :

1. Meningkatkan performa *routing* protokol yang sudah dikembangkan sebelumnya AODV-PNT dengan penambahan faktor perhitungan simpul tetangga (*neighbor node*) dalam pemilihan rute *routing*.
2. Perbaikan dalam perhitungan nilai TWR dalam pemilihan *next-hop node*.

## **1.6 Batasan Penelitian**

Implementasi VANET pada dunia nyata membutuhkan biaya yang tidak sedikit, maka dalam penelitian ini dibatasi sebagai berikut :

1. *Routing* protokol yang digunakan adalah AODV.
2. Implementasi dan ujicoba menggunakan simulator SUMO dan NS-2.35

*[Halaman ini sengaja di kosongkan]*

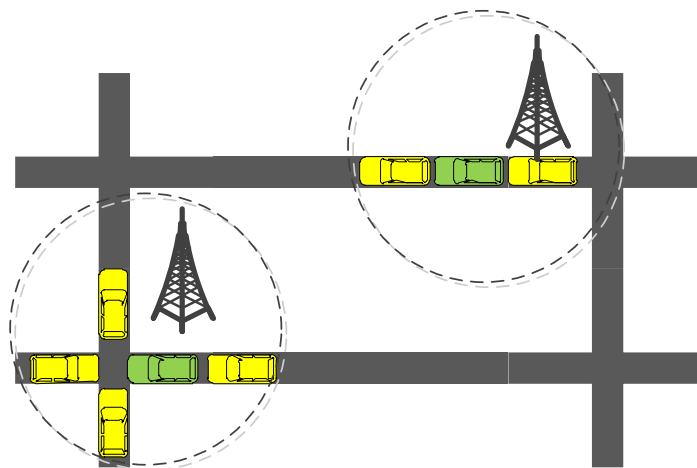
## BAB 2

### DASAR TEORI DAN KAJIAN PUSTAKA

#### 2.1 VANET

VANET (*Vehicular Ad Hoc Network*) adalah pengembangan dari MANET (*Mobile Ad Hoc Network*). VANET berjalan menggunakan jaringan wireless berbasis Ad Hoc yang tambahan pada kendaraan bergerak (mobil, kendaraan umum, dll). VANET menjadi bagian dari ITS (*Intelligent Transportation System*), pada beberapa tahun terakhir banyak penelitian yang dilakukan pada VANET. Secara umum VANET dikembangkan untuk membantu kendaraan-kendaraan untuk berkomunikasi dan memelihara jaringan komunikasi diantara kendaraan tanpa menggunakan bantuan dari central base station atau controller (Rehman, et al., 2013).

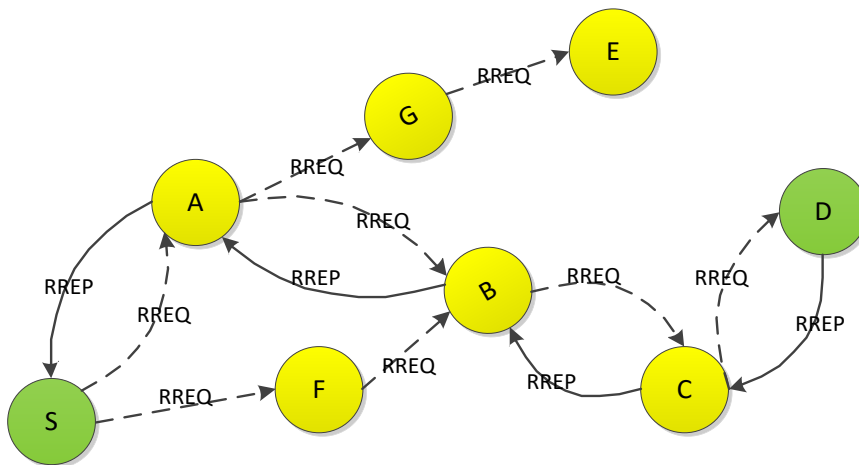
Kendaraan pada VANET dapat berkomunikasi dengan kendaraan lain komunikasi ini dapat disebut dengan komunikasi V2V (*Vehicle to Vehicle*), selain komunikasi V2V kendaraan dapat berkomunikasi dengan infrastruktur seperti RSU (*Road Side Unit*) komunikasi ini disebut dengan V2I (*Vehicle to Infrastructure*) atau komunikasi V2R (*Vehicle to Roadside*), komunikasi V2I dan V2R dapat dilihat pada Gambar 2.1.



Gambar 2.1 Overall framework of VANET (Rehman et al. 2013)

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

AODV adalah sebuah protokol reaktif (C.E. Perkins & E.M. Royer 2002) dimana hanya akan membuat rute ketika ada node yang akan mengirimkan data. Dengan mekanisme on-demand *routing* protokol AODV menghasilkan *routing* over head yang lebih rendah dibandingkan dengan protokol routing yang menggunakan mekanisme proaktif. Ketika simpul pengirim (source node) membutuhkan route ke simpul tujuan (*destination node*), maka simpul pengirim mulai proses pencarian rute dengan menyebar paket Route Request (RREQ) pada jaringan. Jika paket RREQ tidak tahu rute apapun tentang node tujuan dan tidak pernah bertemu, paket RREQ akan menandai reverse path ke pengirim dan menyiarkan ulang RREQ melalui jaringan. Jika simpul pengirim mengetahui rute simpul tujuan, maka *node* akan mengirimkan kembali paket Route Replay (RREP) kepada node pengirim melalui *reverse path* yang telah dibentuk.



Gambar 2.2 AODV route request (Anggoro et al. 2012)

Pada Gambar 2.2 protokol AODV melakukan proses pembentukan rute, ketika simpul S akan mengirim paket data kepada simpul D, simpul S akan menyebar packet RREQ pada jaringan. Setelah simpul D mendapat paket RREQ yang disebar oleh simpul S, simpul D akan merespon dengan mengirim paket RREP untuk simpul S melalui *reverse path* D-C-B-A-S. Setelah simpul S mendapat packet RREP yang dikirim oleh simpul D, simpul S mengetahui bahwa simpul A merupakan next hop untuk mencapai simpul D. Maka route dari simpul S ke simpul D adalah S-A-B-C-D, rute antara pengirim dan penerima terbentuk

maka route akan disimpan pada tabel routing. Setelah rute dari S – D terbentuk rute membutuhkan pemeliharaan dalam menjaga route karena perubahan topologi pada jaringan yang cukup cepat. Tautan terputus (*Disconnected link*) dapat diketahui dengan cara melihat pesan HELLO yang dikirim secara periodik.

Apabila terjadi perubahan topologi yang mengakibatkan simpul yang dituju tidak dapat dituju dengan menggunakan tabel routing yang ada, maka simpul pengirim akan menyebar paket Route Error (RRER) pada simpul tetangga dan simpul tetangga akan menyebar kembali paket RRER ke simpul tetangga sehingga paket RRER sampai ke simpul tujuan. Simpul yang mendapat paket RRER akan menghapus informasi yang error pada routing table, kemudian simpul sumber (source node) akan melakukan pencarian rute kembali apabila rute tersebut masih dibutuhkan.

### **2.3 AODV-VANET**

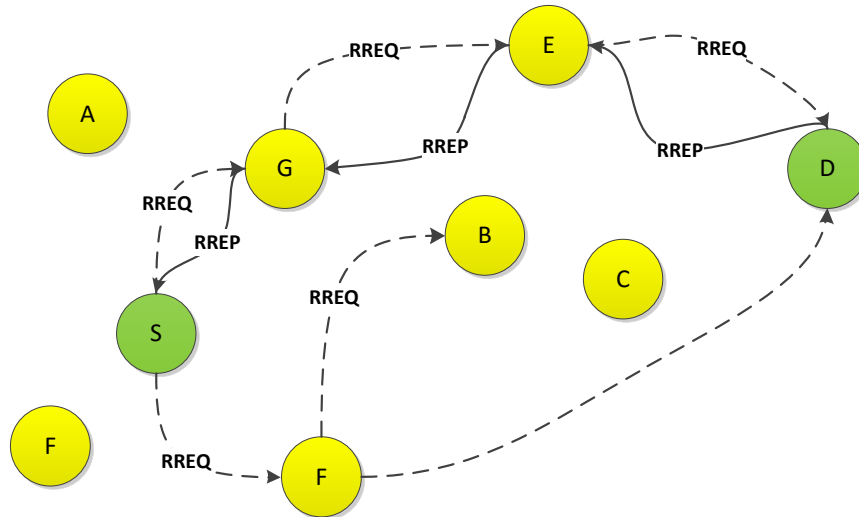
Xi Yu dkk mengusulkan sebuah protokol routing baru bernama AODV-VANET (Yu, Guo & Wong 2011), pada protokol tersebut ditambahkan informasi pergerakan kendaraan ke dalam proses penemuan rute berdasarkan AODV (Ad hoc On-Demand Distance Vector). TWR (Total of Weight of the Route) diperkenalkan untuk memilih rute terbaik bersama-sama dengan estimasi waktu berakhirnya untuk meminimalkan simpul terputus (link breakages). Berdasarkan modifikasi yang dilakukan, protokol AODV-VANET mampu mencapai kinerja routing yang lebih baik dibandingkan dengan AODV tanpa modifikasi.

### **2.4 Prediction Node Trend (PNT)**

PNT (Prediction node trend) (Shen et al. 2014) merupakan sebuah metode prediksi pergerakan node dengan mempertimbangkan beberapa faktor seperti kecepatan dan percepatan sebagai perhitungan TWR, sehingga dalam pemilihan simpul forward dapat memilih simpul yang paling optimal untuk dapat meneruskan paket hingga simpul tujuan (*destination node*).

## 2.5 Routing protokol AODV-PNT

AODV-PNT adalah pengembangan dari routing protokol AODV dan ditambahkan metode PNT. Dikembangkan (Shen et al. 2014) fokus dari AODV-PNT adalah membuat routing protokol yang sesuai dengan *inside a cluster (in-cluster routing)*.



Gambar 2.3 AODV-PNT route request

Tahapan yang dilakukan pada proses route discovery dalam AODV-PNT dimulai dengan source node membuat paket route request (RREQ) dan melakukan *multicast*, sebelum source node mengirim paket RREQ setiap node mengirim paket HELLO. Paket HELLO yang dikirim telah dimodifikasi untuk mendapatkan informasi tentang kecepatan, arah dan link quality dari node tetangga. Setelah informasi didapat kemudian digunakan formula TWR untuk memilih node yang dianggap stabil berdasarkan kriteria pada Tabel 3.4, node yang terpilih kemudian dimasukkan dalam list. Ketika source node mengirimkan paket RREQ paket hanya dikirim pada node yang ada pada list, kemudian neighbor node yang menerima paket RREQ akan memeriksa pada routing table masing-masing. Jika destination sequence number yang terdapat pada paket RREQ sama atau lebih kecil dari yang ada pada routing table dan rute menuju *node* tujuan belum ditemukan, maka paket tersebut tidak akan dilanjutkan (*drop*). Jika *destination sequence number* pada RREQ lebih besar dibandingkan yang terdapat pada routing table, maka entry pada routing table akan diperbarui dan paket tersebut

akan dikirim secara *multicast* kepada neighbor node yang telah dipilih kemudian sekaligus membuat reverse path menuju node pengirim sedangkan neighbor node yang tidak terpilih akan langsung melakukan drop paket. Ketika RREQ telah sampai pada destination node, destination node akan mengirim paket RREP dan membentuk reverse path. Path yang digunakan sesuai dengan path RREQ yang pertama kali sampai pada destination node.

Perbedaan antara AODV dan AODV-PNT adalah penambahan metode PNT dan perhitungan TWR (Total of Weight of the Route) dan future TWR untuk memilih node yang stabil, contoh mekanisme pembentukan path protokol AODV-PNT dapat dilihat pada Gambar 2.3.

## **2.6 Neighbor Node Discovery**

Untuk mengetahui simpul tetangga protokol reaktif mengirim pesan broadcast HELLO secara periodik, simpul (node) yang menerima pesan HELLO akan merespon paket HELLO dengan paket *Acknowledgment*. Salah satu protokol reaktif adalah AODV (C.E. Perkins & E.M. Royer 2002) routing reaktif rentan dengan routing overhead yang disebabkan oleh *flooding* dari RREQ, RREP, RERR dan broadcasting dari pesan HELLO, pesan acknowledgement.

Penelitian untuk meningkatkan performa dari routing protokol AODV terus dilakukan. Pengembangan dilakukan oleh (Mostajeran, Noor & Keshavarz 2013) dan memperkenalkan algoritma I-AODV (Intelligent-AODV) untuk melakukan neighbor node discovery, REQ dan REP header diperiksa untuk membandingkan informasi header dengan tabel routing. Hal ini membantu mengidentifikasi rute baru yang lebih pendek ke tujuan. Dengan demikian, I-AODV bisa menemukan simpul tetangga saat mengirim dan menerima REQ dan REP. Implementasi ini memungkinkan protokol untuk menemukan simpul tetangga dengan cepat dan memanfaatkan informasi simpul tetangga dalam proses penemuan rute.

## 2.7 Path Duration pada VANET

(Raw, Toor & Singh 2012) melakukan survei literatur dan melakukan analisa tentang faktor-faktor yang dapat mempengaruhi durasi path pada VANET, dari analisa tersebut didapatkan:

### 2.7.1 Least Remaining Distance dan Shortest Path

Least Remaining Distance forwarding memilih simpul yang paling dekat dengan simpul tujuan dan memiliki jarak minimum yang tersisa ke simpul tujuan. Shortest Path, memilih jarak terpendek ke tujuan dan memiliki hop yang sedikit.

### 2.7.2 Link Residual Life

Link residual life adalah waktu dimana untuk tautan langsung (direct link) antara dua simpul aktif dan bagian dari rute. Tautan terjauh dari kedua simpul (source dan next-hop) dalam jangkauan transmisi satu sama lain (De et al. 2006). dapat didefinisikan dengan:

$$t = \frac{d}{v_r} \quad (2.1)$$

Dimana :

$d$  = adalah jarak maksimum antara simpul hop-selanjutnya dengan range transmisi maksimum dari simpul asal (source node).

$v_r$  = adalah kecepatan relatif dari simpul asal (source node) dan simpul hop-selanjutnya (next-hop node).

### 2.7.3 Link Distance

Link distance adalah jarak antara 2 simpul, dimana tautan yang tersedia dapat didefinisikan sebagai jarak tautan. Jarak tautan (Link distance) tergantung pada protokol yang digunakan dalam VANET. Jarak tautan (Link distance) akan bertambah jika simpul yang dipilih adalah simpul border (border node) dari *transmission range* sebagai simpul hop-selanjutnya (next-hop node).



#### **2.7.4 Kepadatan node**

Kepadatan simpul adalah jumlah kendaraan per unit area pada range transmisi. Kepadatan simpul akan sangat berpengaruh pada durasi link jika kendaraan yang ada pada *transmission range* jarang (renggang) selain itu dalam pembentukan link menuju simpul tujuan menjadi sangat sulit. Tetapi jika node density cukup padat dan simpul yang dipilih sebagai hop-selanjutnya adalah simpul border maka dapat terjadi Edge Effect. Edge effect adalah ketika simpul yang dipilih sebagai hop-selanjutnya keluar dari range transmisi simpul asal dan terjadi path failure.

#### **2.7.5 Kecepatan simpul**

Arah dari pergerakan simpul dan kecepatan adalah hal yang krusial yang harus dalam perhitungan durasi path antar 2 simpul. Faktor kecepatan simpul tidak harus menyebabkan tautan atau path terputus pada VANETs, durasi tautan (link) juga tergantung pada kecepatan relatif dari simpul karena dapat meningkatkan jarak tautan, perhitungan kecepatan relatif dapat menggunakan persamaan (2.1)

#### **2.7.6 Number of Hops**

Number of hop dapat didefinisikan sebagai jumlah simpul yang ada diantara simpul sumber (source node) dan simpul tujuan (destination node). Jumlah simpul tergantung pada parameter sebelumnya (Least Remaining Distance and Shortest Path, Link Residual Life, Link Distance, Node Density, Velocity of Nodes), setiap parameter mempengaruhi jumlah hop antara simpul sumber (source node) dan simpul tujuan (destination node). Number of Hop seharusnya serendah mungkin untuk mengurangi kemungkinan tautan (link) mudah terputus.

#### **2.7.7 Average Progress per Hop**

Average progress per hop adalah jarak rata-rata antar hop, dimana jarak rata-rata per hop dapat mempengaruhi jumlah hop.

## 2.8 SUMO

SUMO adalah sebuah paket simulasi lalu lintas bebas (open source), pada paket SUMO terdapat paket simulasi dan paket pendukung lainnya yang dapat digunakan secara bebas (open source). SUMO dapat digunakan untuk membantu memodelkan manajemen lalu lintas, kendaraan, dan rute kendaraan. Pada simulasi SUMO kendaraan dapat dimodelkan dengan detail dan eksplisit yang memungkinkan setiap kendaraan bergerak berdasarkan rute sendiri-sendiri.

Implementasi paket simulasi SUMO dimulai pada tahun 2001 dan dapat digunakan secara bebas (open source) pada tahun 2002. SUMO dikembangkan untuk mendukung riset pada bidang lalu-lintas dengan kemampuan untuk implementasi dan mengevaluasi algoritma. (Krajzewicz et al. 2012) melakukan publikasi terbaru tentang pengembangan dan aplikasi SUMO

## 2.9 NS-2

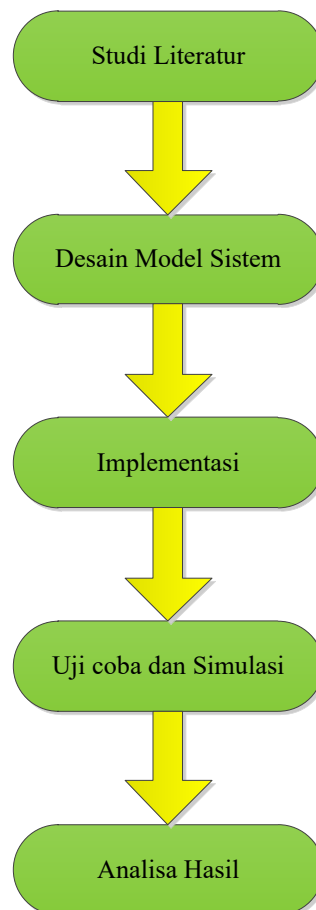
NS adalah simulator diskrit yang dapat digunakan untuk pada penelitian jaringan (isi.edu, 2011). NS dapat digunakan untuk simulasi TCP, routing, dan protokol *multicast* melalui jaringan kabel dan nirkabel (lokal dan satelit). Simulator NS belum terdapat versi yang stabil dengan kata lain NS masih dalam pengembangan, tetapi simulator NS telah banyak digunakan dalam penelitian jaringan bergerak.

NS mulai sebagai varian dari simulator jaringan nyata pada tahun 1989 dan telah berkembang secara substansial selama beberapa tahun terakhir. Pada tahun 1995 pembangunan NS didukung oleh DARPA melalui proyek Vint di LBL, Xerox PARC, UCB, dan USC / ISI. Saat ini pembangunan NS didukung melalui DARPA dengan SAMAN.

## **BAB 3**

### **METODE PENELITIAN**

Metode penelitian ini melalui beberapa tahap meliputi (1) Studi Literatur, (2) Desain Model Sistem, (3) Pembuatan Perangkat Lunak, (4) Uji Coba, dan (5) Analisa Hasil. Alur tahapan-tahapan tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur penelitian

#### **3.1 Studi Literatur**

Studi literatur merupakan ujung tombak berhasil tidaknya sebuah penelitian, Studi literatur dilakukan untuk menggali informasi dan menganalisa perkembangan metodologi yang berkaitan dengan penelitian ini. Beberapa referensi yang dibutuhkan berkaitan dengan penelitian ini adalah :

Tabel 3.1 Perbandingan protokol

Protokol Routing	Kelebihan	Kekurangan
AODV (2003)	<ul style="list-style-type: none"> <li>- Pemakaian jaringan yang rendah dengan cara mengurangi penggunaan <i>control message</i>.</li> <li>- Penambahan <i>sequence number</i> untuk menghindari dari infinite loop.</li> <li>- Memiliki <i>routing over head</i> yang relatif kecil.</li> </ul>	<ul style="list-style-type: none"> <li>- Jika dibandingkan dengan AODV PNT dan AODV-VANET, protokol AODV tidak mempertimbangkan faktor karakter kendaraan sebagai perhitungan pemilihan route.</li> <li>- Memiliki performa yang kurang memuaskan jika kerapatan node rendah tetapi performa akan meningkat jika kerapatan node cukup tinggi (Ali, et al., 2016).</li> </ul>
AODV-VANET (2011)	<ul style="list-style-type: none"> <li>- Memperkenalkan konsep Total Weight of the Route (TWR) untuk pemilihan rute berdasarkan faktor kecepatan, percepatan, arah, dan <i>link quality</i>.</li> <li>- Selain memperkenalkan TWR, AODV-VANET menambahkan perhitungan untuk <i>Expiration Time</i> dari rute yang dipilih.</li> </ul>	<ul style="list-style-type: none"> <li>- Tidak ada perhitungan kestabilan node pada waktu yang akan datang.</li> </ul>
AODV-PNT (2014)	<ul style="list-style-type: none"> <li>- Menggunakan konsep yang sama dengan AODV-VANET.</li> <li>- Menambahkan perhitungan dan membandingkan dengan TWR dengan nilai <i>threshold</i> untuk menentukan kestabilan node</li> </ul>	<ul style="list-style-type: none"> <li>- Tidak ada pertimbangan faktor node tetangga sebagai pertimbangan menentukan nilai TWR.</li> <li>- Nilai bobot setiap faktor konstan untuk setiap skenario yang berbeda-beda.</li> </ul>

1. ITS (*Intelligent Transportation System*) dan VANET (*Vehicular ad-hoc Network*).

2. *Routing* protokol AODV, AODV-AVNET dan AODV-PNT.
3. *Routing* protokol AODV-PNT
4. *Neighbor Discovery Method*

Setelah studi literatur dilakukan perbandingan antar routing protkol AODV, AODV-VANET, dan AODV-PNT. Perbandingan bertujuan untuk mengetahui kelebihan dan kekurangan setiap protokol, dari Tabel 3.1 dapat digunakan sebagai refrensi untuk mengembangkan protokol AODV-PNT. Dari studi literatur yang telah dilakukan maka diperoleh informasi yang berkaitan dengan penelitian yang akan dilakukan. Berdasarkan Tabel 3.1 penelitian ini memfokuskan dalam mengembangkan protokol AODV-PNT dalam perbaikan perhitungan TWR, kami menyebut protokol hasil pengembangan tersebut dengan nama AODV-MNS.

### **3.2 Desain protokol AODV-MNS**

Dari uji coba yang dilakukan oleh (Shen et al. 2014) menunjukkan performa AODV-PNT lebih baik, berdasarkan metrik *packet delivery ratio*, *average end-to-end delay*, dan *routing overhead* (Shen, et al., 2014) dibandingkan dengan protokol AODV original. Meskipun demikian protokol AODV-PNT tidaklah sempurna, protokol tersebut masih dapat dikembangkan lebih lanjut dengan faktor-faktor yang dapat mempengaruhi *path duration* (Raw, Toor & Singh 2012). Pada penelitian ini penulis mengusulkan penambahan dua faktor pada protokol AODV-PNT sebagai modifikasi dalam perhitungan TWR, faktor yang ingin ditambahkan dapat dilihat pada Tabel 3.2 dan akan dijelaskan pada subbab berikutnya.

*Node density* dapat mempengaruhi durasi *path* jika *next hop* yang dipilih adalah *node* yang berada pada area *border node* dan dianggap sebagai *node* yang tidak stabil, ketika *border node* memiliki kecepatan dan percepatan lebih besar dari simpul sumber (*source node*) maka simpul akan berada diluar jangkauan transmisi dan terjadi kegagalan jalur (*path*). Pada jaringan yang padat *node* saling berdekatan bisa menguntungkan bagi protokol AODV karena meningkatkan

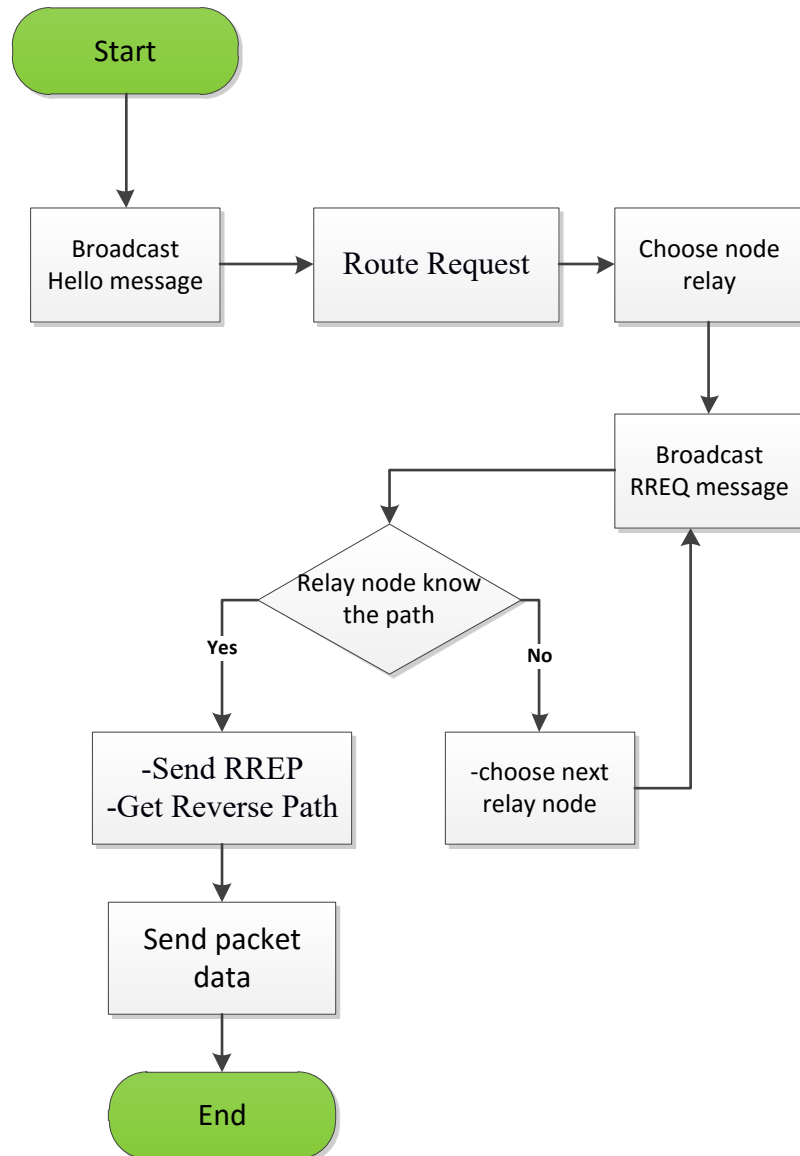
peluang paket terkirim, tetapi tidak selamanya dengan jaringan yang padat performa protokol AODV stabil, bisa jadi PDR yang dihasilkan akan menjadi turun karena banyaknya paket yang tersebar pada jaringan. Protokol AODV-PNT mengusulkan untuk hanya memilih *node* yang stabil dan tidak mengirim pesan kepada semua *node* tetangga, sehingga menjaga RO jaringan untuk tetap rendah. Dengan penambahan faktor tetangga diharapkan dapat meningkatkan performa AODV-PNT dalam pemilihan *relay node* yang stabil.

Tabel 3.2 Metode yang diusulkan dengan AODV-PNT

	AODV-PNT	Penambahan faktor yang diusulkan
Faktor penentu nilai TWR	<ul style="list-style-type: none"> <li>- Kualitas tautan (<i>Link quality</i>) antara simpul hop-selanjutnya (<i>next-hop node</i>) dengan simpul sumber (<i>source node</i>).</li> <li>- Kecepatan,</li> <li>- Percepatan.</li> <li>- Arah.</li> </ul>	<ul style="list-style-type: none"> <li>- Jarak antar <i>node</i></li> <li>- <i>Neighbor node</i> (jumlah simpul tetangga).</li> </ul>
Koefisien bobot masing-masing faktor.	<ul style="list-style-type: none"> <li>- Konstan</li> </ul>	<ul style="list-style-type: none"> <li>- Konstan.</li> </ul>

Selain faktor tetangga pada penelitian ini, penulis mengusulkan penambahan faktor jarak dalam pemilihan *relay node*. Jarak yang dibandingkan adalah jarak *node* tetangga dengan *node* tujuan. Alasan kenapa faktor jarak ditambahkan, karena rute yang dipilih adalah *node* yang memiliki jarak mendekati *node* tujuan, baik secara kecepatan, percepatan, maupun arah. Dengan penambahan faktor yang dilakukan pada perhitungan TWR, protokol AODV-PNT

modifikasi kami sebut AODV-MNS. Untuk perhitungan TWR dengan penambahan faktor baru dijelaskan pada subbab berikutnya.



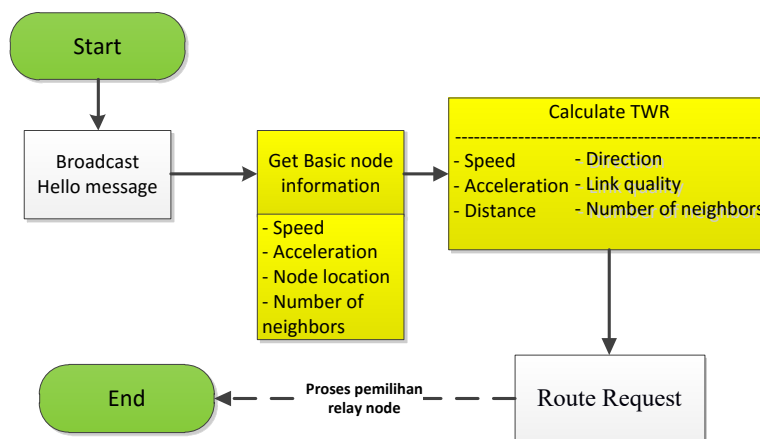
Gambar 3.2 Flowchart proses *routing* AODV

### 3.2.1 Route packet dan Proses routing

Mekanisme route paket AODV-MNS secara umum sama dengan protokol AODV pada Gambar 2.2. Dalam AODV, *node* akan menyebar pesan HELLO kepada *node* sekitar untuk mendapatkan informasi tentang *node* tetangga, pesan HELLO juga digunakan untuk pemilihan jalur *routing*. Mekanisme proses *routing* pada protokol AODV dapat dilihat pada Gambar 3.2. Pada

protokol AODV-PNT pesan HELLO dimodifikasi untuk mendapatkan informasi *node* tetangga dimana informasi tersebut digunakan untuk perhitungan TWR dan *future* TWR, setelah nilai TWR didapat node akan memilih *relay node* sesuai dengan tabel keputusan. Ketika ada paket RREQ yang dikirim oleh *source node*, *source node* hanya akan mengirim paket RREQ kepada *relay node* yang telah dipilih sesuai dengan perhitungan TWR. Jika salah satu *relay node* tahu jalur menuju *destination node*, *node* tersebut akan mengirim pesan RREP kepada *source node* beserta jalur kembalinya (*reverse-path*). Jika *relay node* tidak tahu jalur ke destinasi, *node* tersebut akan menyebarkan paket kembali dengan cara *multicast* kepada *relay node* yang telah dipilih *node* tersebut. Proses ini akan diulangi sampai sebuah *node* menemukan jalur menuju *destination node*.

Secara keseluruhan mekanisme yang perhitungan TWR protokol AODV-MNS sama dengan protokol AODV-PNT, dimulai dengan mencari informasi node tetangga dengan cara mengirim pesan HELLO yang telah dimodifikasi seperti pada Tabel 3.3, kemudian perhitungan TWR dan *future* TWR untuk memilih *relay node*. Pada AODV-MNS parameter yang digunakan dalam perhitungan TWR adalah kecepatan, percepatan, jarak, arah, *link quality* dan jumlah node tetangga. Mekanisme poses pengiriman RREQ AODV-MNS sama dengan AODV-PNT. Proses penanganan RREP sama dengan AODV original. Proses perhitungan TWR AODV-MNS dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram alir proses perhitungan TWR AODV-MNT



Tabel 3.3 Modifikasi pesan HELLO AODV-MNS

Type	Source IP Address (source id)	Kecepatan (Speed)	Percepatan (Acceleration)	Node position (X, Y)	Number of neighbour
------	-------------------------------	-------------------	---------------------------	----------------------	---------------------

### 3.2.2 Perhitungan TWR

#### 3.2.2.1 Kecepatan dan percepatan kendaraan

Kecepatan dan percepatan kendaraan perlu diperhitungkan karena apa bila perbedaan kecepatan antara simpul sumber (*source node*) dan simpul tujuan (*destination node*) terlalu besar dapat mengakibatkan tautan (*link*) akan mudah terputus, selain itu percepatan kendaraan juga diperhitungkan untuk memilih simpul relay (*relay node*). Sehingga dalam pemilihan kendaraan sebagai simpul relay (*relay node*) dengan mengutamakan kendaraan yang memiliki kecepatan dan percepatan yang relatif sama dengan *source node*. Informasi kecepatan node didapat dari pesan HELLO, sedangkan untuk informasi percepatan kendaraan dihitung menggunakan formula (3.6).

#### 3.2.2.2 Arah kendaraan

Kendaraan yang memiliki arah yang sama akan membuat durasi komunikasi antar kendaraan dapat berlangsung cukup lama. Arah kendaraan menjadi prioritas penting setelah kecepatan dan percepatan kendaraan dalam perhitungan TWR. Pada proses menentukan arah kendaraan menggunakan koordinat simulator NS-2.35, kemudian menghitung sudut antara 2 arah kendaraan.

$$\beta = \text{atan2}(x, y) \quad (3.1)$$

Dimana:

$$x = \cos \theta b * \sin \Delta L$$

$$y = \cos \theta a * \sin \theta b - \sin \theta a * \cos \theta b * \cos \Delta L$$

$\beta$  adalah bearing

$\theta$  adalah latitude

$L$  adalah longitude

Simpul (*node*) tujuan tidak selalu dalam keadaan bergerak, simpul (*node*) tujuan sangat mungkin dalam kondisi berhenti atau tidak bergerak. Pada

kondisi tersebut faktor arah tidak memiliki bobot (*weight factor*) maka ditambahkan faktor jarak dalam perhitungan TWR untuk membantu dalam pemilihan relay simpul (*node*) yang stabil, untuk menghitung jarak antar 2 (dua) simpul (*node*) digunakan formula (3.2):

$$dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.2)$$

Dimana:

$(x_1, y_1)$  adalah koordinat simpul (*node*) 1

$(x_2, y_2)$  adalah koordinat simpul (*node*) 2

### 3.2.2.3 Kualitas tautan (*link quality*) antar kendaraan

Selain kecepatan, percepatan dan arah kendaraan yang perlu diperhatikan adalah link quality antar kendaraan, digunakan formula:

$$Q = \frac{1}{S_{ij}} \quad (3.3)$$

Dimana Q adalah kualitas tautan (*Link Quality*), untuk menghitung nilai digunakan formula berikut:

$$S_{ij} = 1 - \frac{\min(\sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}, r)}{r} \quad (3.4)$$

Dimana:

$r$  adalah jarak komunikasi maximum antara *nod*) yang berdekatan

$i_x, i_y$  adalah koordinat dari simpul (*node*)  $i$

$j_x, j_y$  adalah koordinat dari simpul (*node*)  $y$

### 3.2.2.4 Tetangga (*neighbor node*)

Berdasarkan studi literatur yang dilakukan Vikas Toor dkk (Raw, et al., 2012), kerapatan simpul (*node density*) dapat mempengaruhi durasi path jika simpul hop-selanjutnya yang dipilih adalah simpul yang berada pada area *border node*, ketika *border node* memiliki kecepatan dan percepatan lebih besar dari simpul sumber (*source node*) maka simpul akan berada diluar jangkauan transmisi dan terjadi kegagalan jalur (*path*).

$$TWR = f_s \times |S_n - S_d| + f_a \times |A_n - A_d| + f_d \times |\theta_n - \theta_d| \quad (3.5) \\ + + f_{dist} \times |Dist_n - Dist_d| + f_q \times Q + f_n \times N$$

Dimana:

$S_n$  adalah next-hop node kecepatan (*speed*)

$A_n$  adalah next-hop node percepatan (*acceleration*)

$\theta_n$  adalah next-hop node arah (*direction*)

$S_d$  adalah destination node kecepatan (*speed*)

$A_d$  adalah destination node percepatan (*acceleration*)

$\theta_d$  adalah arah *destination node*

$Q$  adalah kualitas tautan (*link quality*) antara *source* dan *next-hop node*

$N$  adalah simpul (*node*) tetangga

$f_s$  adalah *weight factor* kecepatan

$f_a$  adalah *weight factor* percepatan

$f_d$  adalah *weight factor* arah

$f_q$  adalah *weight factor* kualitas tautan (*link quality*)

$f_n$  adalah *weight factor* simpul tetangga (*neighbor node*)

Berdasarkan persamaan (3.5), dapat diketahui TWR tergantung dari perbedaan kecepatan, percepatan, arah, kualitas tautan (*link quality*) dan simpul tetangga (*neighbor node*) atau kerapatan simpul (*node density*). Simpul hop-selanjutnya (*next-hop node*) yang terbaik jika simpul tersebut memiliki nilai TWR yang kecil dengan nilai kecepatan, percepatan, dan arah relatif sama.

### 3.2.3 Perhitungan prediksi *future* TWR

Dengan karakteristik topologi VANET yang dinamis maka next-hop yang terpilih memiliki peluang keluar dari *transmission range* pada saat pengiriman paket data berikutnya, maka dilakukan perhitungan *future* TWR dari sebuah simpul (*node*) (3-5 detik yang akan datang), dengan

memperhatikan nilai *future* TWR diharapkan rute jalur yang dipilih memiliki durasi waktu koneksi yang cukup lama.

### 3.2.3.1 Prediksi kecepatan dan percepatan kendaraan

Karena selisih waktu kecepatan dan percepatan dengan waktu prediksi kecepatan dan percepatan cukup kecil (3-5 detik), maka nilai prediksi dari kecepatan dan percepatan menggunakan formula (3.6) dan (3.7)

- Perhitungan percepatan

$$a = \frac{\Delta v}{\Delta t} \quad (3.6)$$

Dimana:

$\Delta v$  adalah selisih kecepatan sekarang dan kecepatan sebelumnya.

$\Delta t$  adalah selisih waktu.

- Perhitungan prediksi kecepatan

$$v' = v + a \times t \quad (3.7)$$

Dimana:

$v$  adalah kecepatan sekarang

$a$  adalah akselerasi

$t$  adalah waktu (detik)

### 3.2.3.2 Prediksi arah pergerakan kendaraan

Pada kehidupan nyata simpul (*node*) dapat berbelok, berhenti, dan melakukan pergerakan negative (mundur) maka untuk perhitungan prediksi arah kendaraan menggunakan GPS, kemudian dihitung menggunakan formula (3.8) dan (3.9) sebagai perhitungan posisi yang akan datang.

$$x' = x + v_0 t + \frac{1}{2} a t^2 \quad (3.8)$$

$$y' = y + v_0 t + \frac{1}{2} a t^2 \quad (3.9)$$

dimana :

$x, y$  adalah oordinat sekarang

$v_0$  adalah kecepatan sekarang

$a$  adalah akselerasi

$t$  adalah waktu (detik)

### 3.2.3.3 Prediksi kualitas tautan (*link quality*) antar kendaraan

Setelah mendapatkan nilai prediksi dari kecepatan, percepatan dan arah pergerakan kendaraan, maka dapat dihitung jarak antar kendaraan dan arah kendaraan sehingga didapatkan koordinat berikutnya berdasarkan koordinat saat ini. Formula untuk menghitung prediksi kualitas link (*link quality*) antar kendaraan dapat menggunakan formula (3.3)

### 3.2.3.4 Prediksi tetangga (*neighbor node*)

Karena selisih waktu sekarang dan waktu yang akan diprediksi cukup singkat, maka untuk prediksi jumlah simpul tetangga (*neighbor node*) menggunakan jumlah simpul tetangga (*neighbor node*) saat ini dari *routing table*.

Kriteria *node* yang akan dipilih sebagai simpul relay adalah:

1. TWR saat ini: Jika nilai TWR semakin kecil maka semakin baik, TWR yang paling kecil adalah “*Optimal*”, sedangkan suboptimal berarti “*Good*”.
2. State : didefinisikan adalah *stability threshold* adalah bernilai absolut dari selisih TWR saat ini dengan *future TWR*.
  - Jika  $< W \text{ threshold}$ , maka simpul dianggap sebagai simpul “*stable*”.
  - Jika  $> W \text{ threshold}$ , maka simpul dianggap sebagai simpul “*instable*”.
3. *Future TWR*
  - Jika *future TWR*  $<$  TWR saat ini, maka disebut “*Better*”
  - Jika *future TWR*  $>$  TWR saat ini, maka tidak lebih baik dari TWR saat ini.
4. Judgement:
  - “*Relay node*” berarti *path feasible* (memungkinkan).
  - “*Abandon*” berarti simpul tidak perlu dipilih.

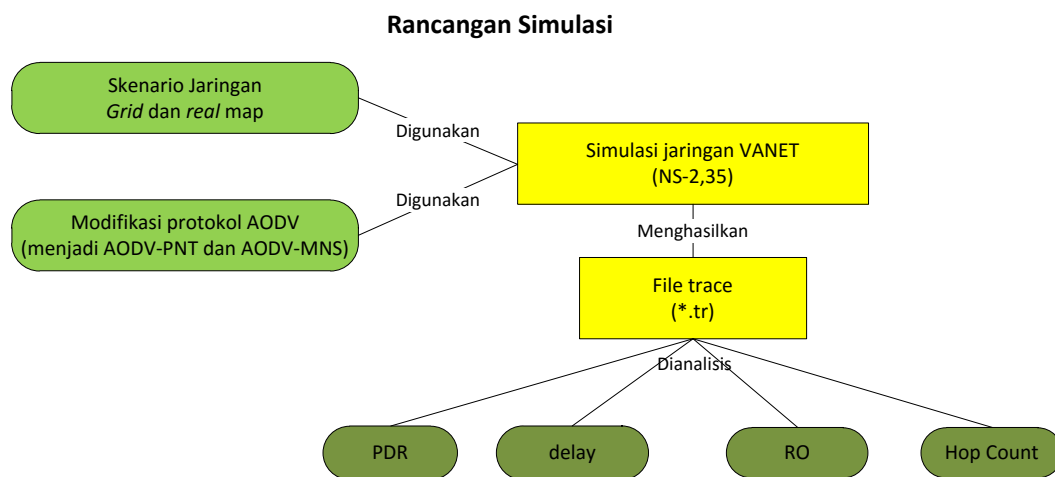
Dalam pemilihan *relay node* mempertimbangkan informasi simpul tetangga dan nilai TWR yang optimal, kemudian memilih simpul yang *stable*, dan yang terakhir melihat *future* TWR apakah lebih baik dari TWR saat ini. Detail dalam pemilihan *relay node* dapat dilihat pada Tabel 3.4.

Tabel 3.4 Pemilihan *relay node*

Current TWR	State	Future TWR	Judgement
Optimal	Instable	Better	Relay node
Optimal	Stable	\	Relay node
Suboptimal	Instable	Better	Relay node
Supoptimal	Stable	\	Relay node
Other cases			Abandon

### 3.3 Implementasi

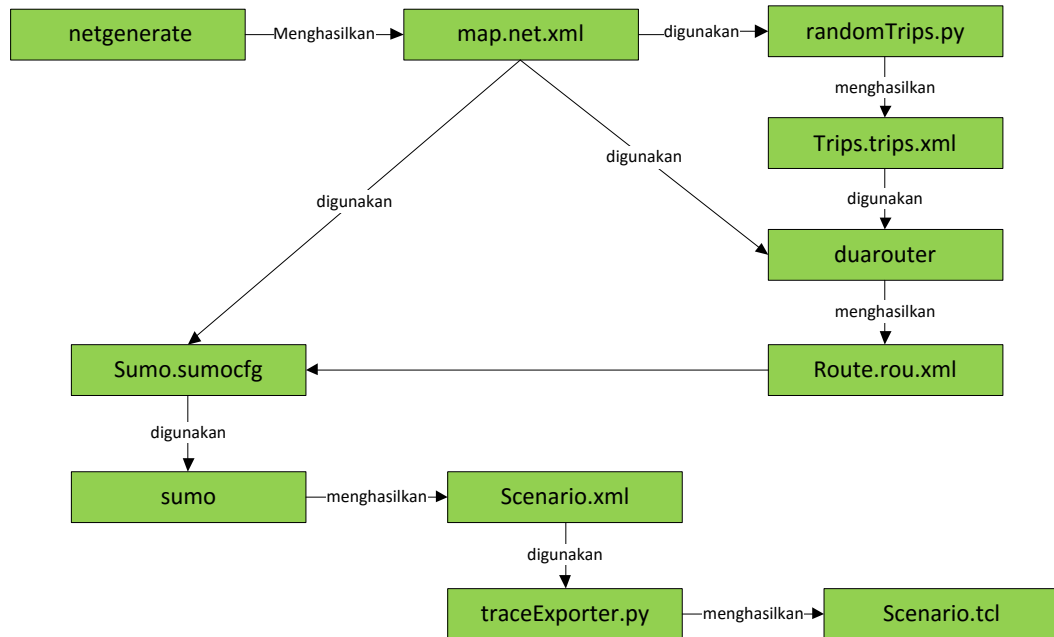
Pada tahap ini, dilakukan implementasi terhadap desain model sistem yang telah dirancang sebelumnya. Implementasi dilakukan menggunakan perangkat lunak simulator NS-2.35 dengan skenario (implementasi) pada peta *grid* dan peta *real*, untuk implementasi pada setiap skenario dijelaskan pada subbab berikutnya dan untuk diagram simulasi dapat dilihat pada Gambar 3.4



Gambar 3.4 Diagram rancangan simulasi

### 3.3.1 Implementasi skenario *grid*

#### Alur pembuatan peta *grid*

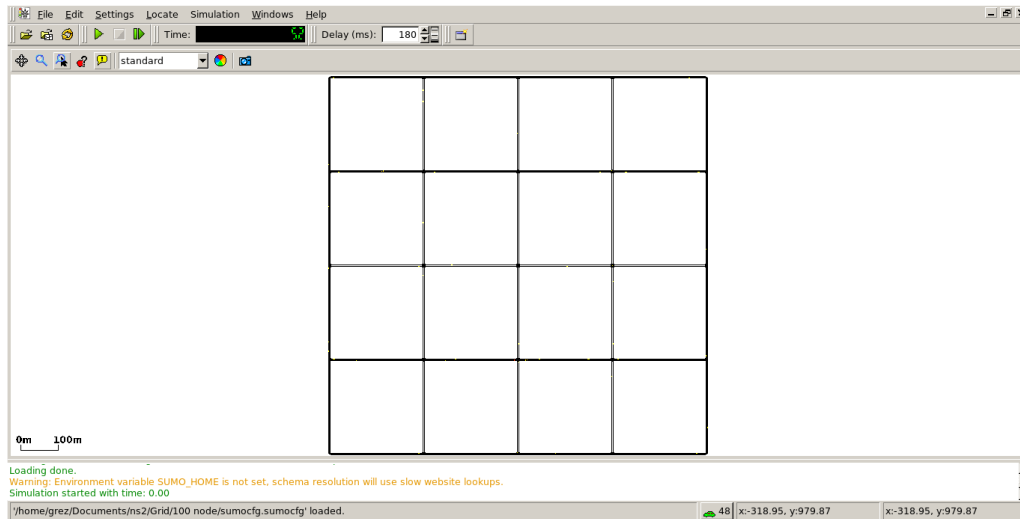


Gambar 3.5 Alur pembuatan skenario *grid*

Skenario *grid* dibuat dengan menggunakan bantuan perangkat lunak SUMO. Skenario yang dibuat adalah dengan luas area 800 m x 800 m dengan panjang setiap ruas jalan 200 m dan jumlah titik persimpangan sebanyak 6 X 6. Kecepatan rata-rata kendaraan 10, 15, 20 dan 25 m/s, untuk menjalankan perintah `netgenerate` dapat dilihat pada *lampiran*. Langkah-langkah pembuatan peta *grid* dengan spesifikasi diatas, digunakan perintah seperti pada lampiran A.10, pada lampiran tersebut terdapat perintah untuk pembuatan peta *grid* dari awal hingga siap gunakan pada simulator NS-2.35, tahapan dalam pembuatan skenario *grid* dapat dilihat pada Gambar 3.5.

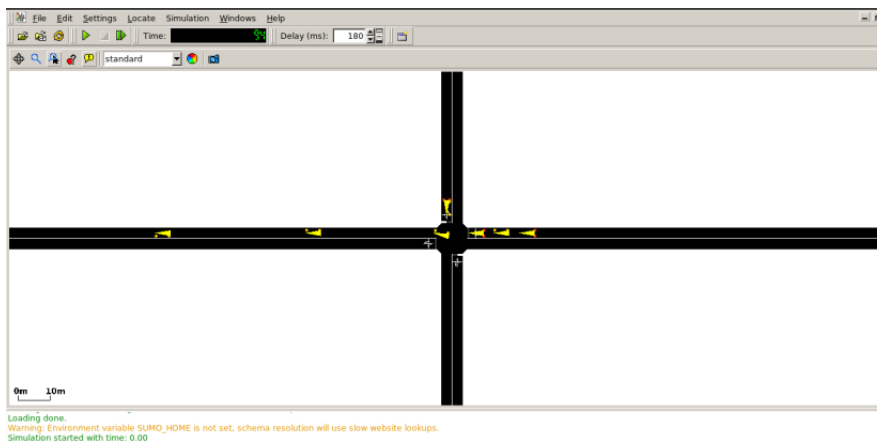
Tahap awal adalah membuat peta, setelah peta terbentuk tampak pada Gambar 3.6, proses selanjutnya adalah membuat titik awal dan titik tujuan kendaraan secara acak dengan menggunakan modul `randomTrips.py`. Tahapan selanjutnya adalah membuat rute yang akan dilalui kendaraan berdasarkan peta yang telah dibuat sebelumnya, kemudian membuat *file* `.sumocfg`, *file* ini digunakan untuk mendefinisikan lokasi *file* `.net.xml` dan `.trips.xml` agar dapat dijalankan menggunakan antarmuka simulator SUMO dan mendefinisikan durasi

simulasi. `.sumocfg` diletakkan satu direktori yang sama dengan `.net.xml` dan `.trips.xml`. Isi file `.sumocfg` dapat dilihat pada lampiran, agar dapat digunakan di NS-2, keluaran dari perintah `sumo` harus dikonversi ke format yang dapat dipahami oleh NS-2 melalui perintah `traceExporter.py`.



Gambar 3.6 Peta skenario *grid*

Setelah semua perintah dijalankan, visualisasi pergerakan kendaraan dapat dilihat pada Gambar 3.7 menggunakan antar muka perangkat lunak SUMO.



Gambar 3.7 Contoh pergerakan *node* pada peta *grid*

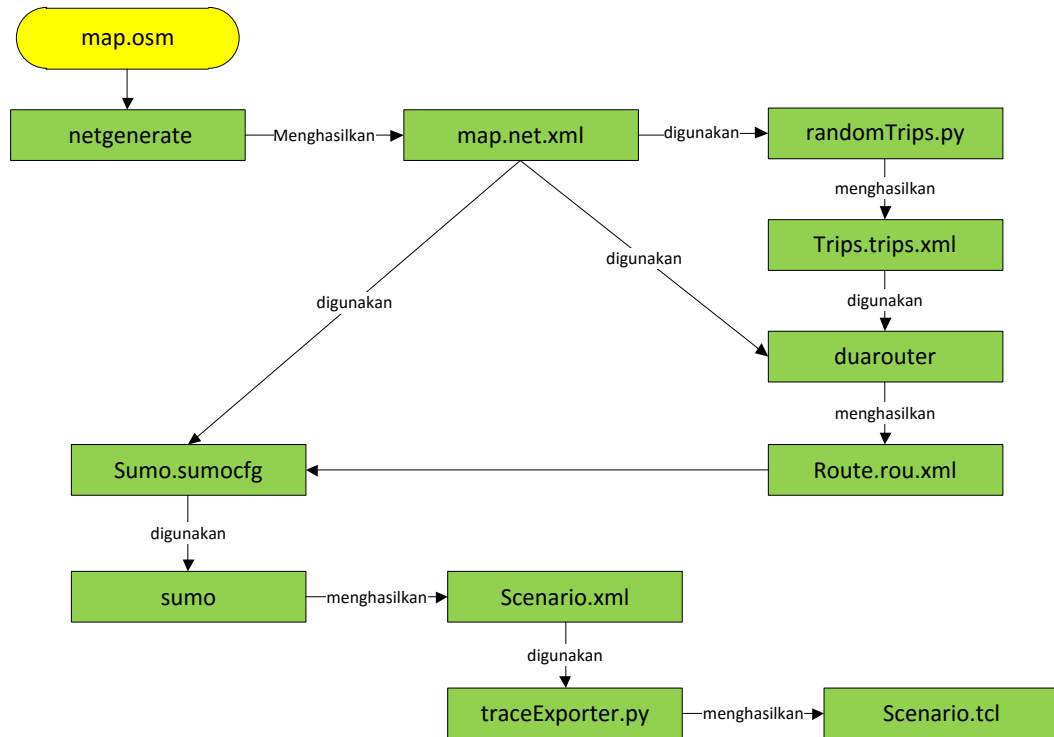
### 3.3.2 Implementasi skenario *real*

Alur pembuatan peta untuk skenario *real* hampir sama dengan pembuatan peta skenario *grid*, alur pembuatan skenario *real* dapat dilihat pada Gambar 3.8. Skenario *real* menggunakan peta yang diambil dari OpenStreetMap. Luas area yang digunakan kurang lebih sama seperti pada skenario *grid* 600 x 700



dengan variasi kecepatan kendaraan 10, 15, 20 dan 25 m/s. Peta diambil dengan cara seleksi wilayah Kota Malang kemudian diekspor kedalam *file* .osm melalui browser.

### Alur pembuatan peta real



Gambar 3.8 Alur pembuatan skenario *real*



Gambar 3.9 Peta skenario *real*

Peta yang telah diekspor kemudian peta dapat disunting dengan perangkat lunak JOSM. Setelah penyuntingan selesai peta kemudian dikonversi menjadi *file* dengan format *.net.xml* dengan perintah *netconvert --osm-files*

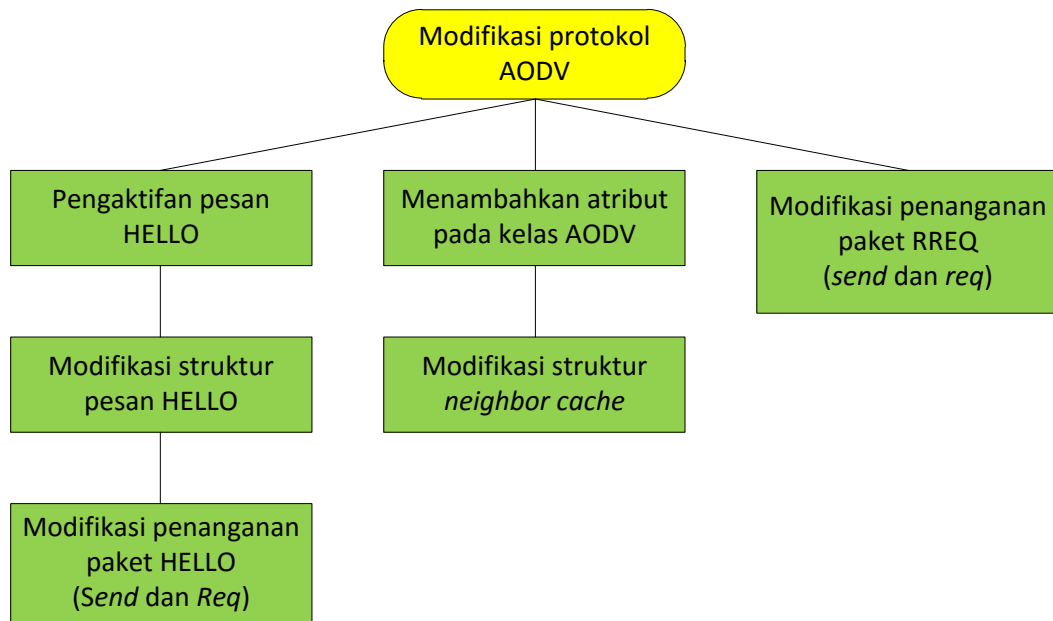
*map.osm -o map.net.xml*, hasil peta yang telah dikonversi dapat dilihat pada Gambar 3.9. Setelah peta terbentuk, langkah selanjutnya sama seperti tahapan dalam membuat skenario *grid*, membuat titik awal dan titik tujuan kendaraan hingga konversi *file* yang dapat dijalankan dengan NS-2.

### 3.3.3 Implementasi protokol

Implementasi protokol dengan cara memodifikasi protokol AODV, modifikasi yang dilakukan pada protokol AODV pada simulator NS-2 dapat dilihat pada gambar Gambar 3.10. Pada simulator NS-2.35 pesan HELLO tidak aktif, untuk mengaktifkan pesan HELLO perlu menonaktifkan atau menghilangkan baris kode *define AODV\_LINK\_LAYER\_DETECTION* dan *define AODV\_USE\_LL\_METRIC* pada skrip *aodv.h*. Untuk implementasi AODV-MNS pada simulator NS-2, perlu penambahan *field* kecepatan (*speed*), akselerasi (*acceleration*) dan koordinat *x* dan *y* pada paket HELLO. Dilakukan penambahan atribut *rp\_speed*, *rp\_accel*, *rp\_x*, dan *rp\_y* pada *struct hdr\_aodv\_reply* pada skrip *aodv\_packet.h*, dapat dilihat pada Lampiran A.4.

Penambahan atribut pada kelas AODV menambahkan atribut *lastUpdateTime*, *lastSpeed*, dan *lastAccel* pada skrip *aodv.h* seperti pada Lampiran A.4. Atribut *lastAccel* digunakan untuk menyimpan hasil perhitungan akselerasi. *position\_update\_time\_* didapat dari objek *MobileNode*, atribut tersebut berfungsi untuk mencatat pada detik berapa suatu *node* berubah posisi sehingga atribut tersebut dapat dimasukkan untuk perhitungan akselerasi.

Kelas AODV menyimpan *list* dari *neighbor node* pada kelas *neighbor cache*. Informasi *neighbor cache* akan diperbarui setiap kali *node* menerima paket HELLO, tetapi *list* tersebut tidak memiliki informasi tentang kecepatan, akselerasi, dan koordinat *neighbor node*. Sehingga perlu ditambahkan atribut *double nb\_speed*, *nb\_accel*, *double nb\_x* dan *double nb\_y* seperti pada Lampiran A.5.



Gambar 3.10 Alur modifikasi protokol AODV

Dalam NS-2 proses pengiriman paket HELLO terdiri atas pembuatan paket, pengisian atribut paket, dan penjadwalan event pengiriman. Pada *routing* protokol AODV-MNS, paket HELLO digunakan untuk mengirim informasi kecepatan, akselerasi, dan koordinat dari *node* pengirim. Dengan struktur paket HELLO yang sudah dimodifikasi, pengisian nilai dari atribut-atribut paket HELLO yang baru dapat dilakukan di fungsi *sendHello* pada skrip *aodv.cc* seperti pada Lampiran A.3, sedangkan informasi kecepatan dan koordinat dari *node* pengirim pesan HELLO didapat dari kelas *MobileNode*. Ketika *node* menerima paket HELLO *node* akan mengecek *neighbor cache*. Jika alamat pengirim paket HELLO tersebut tidak ada dalam *list*, maka alamat tersebut akan ditambahkan kedalam *list* dan menyimpan informasi kecepatan, akselerasi, dan koordinatnya. Jika alamat pengirim paket HELLO sudah ada dalam *list*, maka informasi kecepatan, akselerasi, koordinat akan diperbarui. Modifikasi fungsi pesan HELLO dapat dilihat pada Lampiran A.2.

Modifikasi proses pengiriman paket RREQ (*Route Request*) memodifikasi fungsi *sendRequest()* pada skrip *aodv.cc* seperti pada Lampiran A.3. Modifikasi perlu dilakukan karena protokol AODV mengirim pesan RREQ dengan cara *broadcast*, sedangkan AODV-MNS dengan cara *multicast*. Pemilihan

*relay node* berdasarkan perhitungan TWR dan *future TWR*, *node* yang terpilih dijadikan sebagai *relay node* akan disimpan.

---

```
Input: rreq
.
.
    calculate TWR of next-hop nodes;
    calculate future TWR of next-hop nodes;
    classify TWR, stability, and future TWR;
    list relay node set;
.
.
rq new RREQ packet;
fill up RREQ fields;
rqlist_length list:size();
rqrelay_node_set list;
broadcast rq;
```

---

Gambar 3.11 *Pseudocode* modifikasi pengiriman RREQ

Agar perilaku *boradcast* paket RREQ dapat berjalan mirip dengan *multicast* RREQ, maka perlu modifikasi pada struktur paket RREQ menyimpan array dan panjang array dari *relay node set*. Perubahan dilakukan dengan menambahkan array *rq\_eligible\_nodes* untuk menyimpan *relay node set* dan *nodes\_list\_len* untuk menyimpan panjang array seperti pada Lampiran A.1. Setelah modifikasi dilakukan, ketika paket RREQ di *broadcast* pada *node* tetangga, paket akan dicek untuk mengetahui *node* tersebut ada dalam *relay node set*, apabila *node* tersebut ada dalam set maka paket RREQ akan diteruskan atau di *broadcast* ulang, jika *node* tersebut tidak ada dalam set maka paket akan di *drop*, *Pseudocode* modifikasi pengiriman paket RREQ dapat dilihat pada Gambar 3.11.

Proses penerimaan RREQ juga perlu dimodifikasi agar *broadcast* paket RREQ bisa memiliki perilaku yang mirip dengan *multicast* seperti pengiriman paket RREQ, *Pseudocode* modifikasi fungsi *recvRequest* pada skrip *aodv.cc* dapat dilihat pada Lampiran A.2. Agar penerimaan paket RREQ yang di-*broadcast* dapat melakukan (seperti) *multicast*, maka perlu ditambahkan kondisi dimana *node* yang tidak seharusnya menerima paket RREQ (*relay node set*) harus melakukan *drop* paket. Apabila *node* tersebut ada dalam *set* penerima paket RREQ dan *node* bukan merupakan *node* tujuan, maka dilakukan proses perhitungan TWR hingga pembuatan *relay node set* yang baru didapat dan paket

akan tersebut, *Pseudocode* modifikasi penerimaan paket RREQ dapat dilihat pada Gambar 3.12 .

---

```

Input: rreq
.
.
index address of this node;
if index is not in rreqrelay_node_set then
    drop packet;
else
    if rreqdst is not index then
        calculate TWR of next-hop nodes;
        calculate future TWR of next-hop nodes;
        classify TWR, stability, and future TWR;
        new_list new relay node set;
        rreqrelay_node_set new_list;
    end
end
.
.

```

---

Gambar 3.12 *Pseudocode* modifikasi penerimaan paket RREQ

### 3.3.4 implementasi perhitungan TWR dan Future TWR

---

```

// TWR Calculation
// Calculate distance between next-hop and dst
double nb_distance;
nb_distance = sqrt(pow((nb->nb_x - xDst), 2) +
pow((nb->nb_y - yDst), 2));
// radius between this node and neighbor
// minimum radius -> min(√(i x - j x )^2 +
(i y - j y )^2 ; r)
double radius = std::min(sqrt(pow((nb->nb_x - posX), 2) +
pow((nb->nb_y - posY), 2)), (double) maxTxRange);
double quality = 1.0 / (1.0 - (radius / ((double)
maxTxRange + 1.0)));
double modSpeed = fSpeed * nb->nb_speed;
double modAccel = fAccel * nb->nb_accel;
double modDistance = fDistance * nb_distance;
double modDirection = fDirection * abs
(sudut_vertikal - nb_sudut_vertikal);
double modQuality = fQuality * quality;
double modNeighbor = fNeighbor * nb_node();
// TWR = f s × |S n - S d | + f a × |A n - A d | + f d ×
|θ n - θ d | + f q × Q
double TWR = modSpeed + modAccel + modDirection
+ modDistance + modQuality + modNeighbor;

```

---

Gambar 3.13 Perhitungan TWR

---

```

// Calculate future TWR for next [timeModifier] seconds
// Future speed v' = v + a * t
double nb_speedFuture = nb->nb_speed + (nb->nb_accel *
    timeModifier);
// Future position will be used to calc future direction
// Formula: x' = x + v0 t + 0.5at^2
// Future neighbor position
double nb_xFuture = nb->nb_x + (nb->nb_speed * timeModifier)
    + (0.5 * nb->nb_accel * timeModifier * timeModifier);
double nb_yFuture = nb->nb_y + (nb->nb_speed * timeModifier)
    + (0.5 * nb->nb_accel * timeModifier * timeModifier);
// Future this_node position
double iXFuture = posX + (iSpeed * timeModifier)
    + (0.5 * iAccel * timeModifier * timeModifier);
double iYFuture = posY + (iSpeed * timeModifier)
    + (0.5 * iAccel * timeModifier * timeModifier);
// Calculate future distance between next-hop and dst
double futureDistance = sqrt(pow((nb_xFuture - xDst), 2) +
    pow((nb_yFuture - yDst), 2));
// Future radius between this node and neighbor: both using future
values
double futureRadius = std::min(sqrt(pow((nb_xFuture - iXFuture),
    + pow((nb_yFuture - iYFuture), 2))), (double) maxTxRange);
double futureQuality = 1.0 / (1.0 - (futureRadius /
    ((double) maxTxRange + 1.0)));
// Calculate future TWR
modSpeed      = fSpeed * nb_speedFuture;
modAccel      = fAccel * nb->nb_accel;
modDistance   = fDistance * futureDistance;
modDirection  = fDirection *
    abs(sudut_vertikal - nb_sudut_vertikal);
modQuality    = fQuality * futureQuality;
modNeighbor   = fNeighbor * nb_node();
double futureTWR = modSpeed + modAccel + modDistance +
    modDirection + modQuality + modNeighbor;

```

---

Gambar 3.14 Perhitungan *future* TWR pada file aodv.cc

Tahap perhitungan TWR dan pengolahan *relay node* dilakukan oleh *node* yang akan mengirimkan atau meneruskan paket RREQ. Sebelum melakukan perhitungan TWR, nilai *threshold* W, rasio setiap faktor pengali ditentukan dengan cara ujicoba. Implementasi kode untuk perhitungan TWR dapat dilihat pada Gambar 3.13. Khusus untuk *node* tujuan, kecepatan dan akselerasi bernilai nol karena *node* tujuan diam (*stationary node*) koordinat *node* tujuan dituliskan secara *hard code* pada kode sumber.

Proses perhitungan jumlah *node* tetangga menggunakan fungsi *neighbor cache* yang sudah ada, implementasi perhitungan jumlah *node* tetangga dapat dilihat pada Lampiran A.8. Perhitungan TWR dimulai dengan menghitung jarak

antara *nexthop node* dengan *node* tujuan. Kemudian menghitung kualitas hubungan antara *node* pengirim dan *next-hop node*. Selanjutnya kecepatan, akselerasi, arah dari *next-hop node* masing-masing dikurangkan dengan kecepatan, akselerasi dan arah dari *node* tujuan. Kemudian kecepatan, akselerasi, jarak, arah, link quality dan jumlah *node* tetangga dikalikan dengan faktor pengali dan dijumlahkan sehingga menghasilkan nilai TWR, implementasi perhitungan arah kendaraan terhadap *node* tujuan dapat dilihat pada Lampiran A.9.

Setelah melakukan perhitungan TWR, dilakukan perhitungan TWR yang akan datang (*future TWR*) dengan menggunakan informasi yang sudah didapatkan sebelumnya dan prediksi nilai dengan menggunakan beberapa rumus fisika gerak lurus. Implementasi perhitungan *future TWR* dapat dilihat pada Gambar 3.14.

### 3.3.5 Implementasi simulasi pada NS-2.35

Dibutuhkan sebuah *file* OTcl yang berisi deskripsi dari lingkungan simulasi untuk simulasi VANET pada perangkat lunak NS-2.35. *File* tersebut berisikan pengaturan untuk setiap *node* dan beberapa event yang perlu diatur agar berjalan, contoh potongan skrip dapat dilihat pada Lampiran A.7.

## 3.4 Uji coba dan Simulasi

Uji coba performa dari protokol maka dilakukan:

1. Pengujian berdasarkan beberapa metrik analisis antara lain *packet delivery ratio*, *routing over head*, *average delivery delay*, dan *hop count* dengan variasi kecepatan kendaraan.
  - PDR (*Packet delivery ratio*) adalah perbandingan antara jumlah paket yang dikirim dan diterima, formula (3.10) adalah formula untuk mencari nilai PDR.

$$PDR = \frac{Packet_{received}}{Packet_{send}} \quad (3.10)$$

- *Routing overhead* adalah jumlah paket *routing* yang ditransmisikan. Paket *routing* yang ditransmisikan terdiri dari RREQ, RREP, dan RERR. Untuk menghitung *routing overhead* dapat menggunakan formula (3.11)

$$RO = RREQ_{send} + RREP_{send} + RERR_{send} \quad (3.11)$$

- *Average end-to-end delay* adalah nilai rata-rata waktu yang dibutuhkan oleh paket untuk sampai dari source sampai ke tujuan. Rumus *average end-to-end delay* dapat dilihat pada rumus (3.12).

$$Delay = \frac{\sum_{i=0}^n t_{received}[i] - t_{sent}[i]}{pktCounter} \quad (3.12)$$

- *Hop count* adalah jumlah link point-to-point dalam jalur transmisi.
2. Uji coba untuk mendapatkan nilai setiap *weight factor* yang optimal dilakukan dengan cara merubah nilai kombinasi setiap *weight factor* dan diuji pada beberapa skenario yang berbeda.

### 3.5 Analisa Hasil

Pada tahap ini dilakukan analisis dari hasil uji coba dari beberapa metrik *packet delivery ratio*, *routing over head*, *average delivery delay*, dan *hop count*. Diharapkan dari analisis uji coba ini akan diperoleh hasil yang sesuai dengan tujuan penelitian.

### 3.6 Penyusunan Buku Tesis

Penyusunan buku tesis dilakukan sebagai dokumentasi terhadap serangkaian penelitian yang dikerjakan agar dapat dijadikan sebagai bahan pembelajaran, referensi maupun sebuah perbaikan penelitian di masa depan yang berhubungan dengan perbaikan mekanisme *update* informasi.



## BAB 4

### HASIL DAN PEMBAHASAN

Pada bab 4 ini dijelaskan tahapan penelitian mengenai mencari nilai rasio dari setiap faktor, tahapan pengujian yang dilakukan dan analisis terhadap hasil uji coba yang dilakukan.

#### 4.1 Tahapan penelitian

Pada tahapan ini dilakukan pencarian rasio setiap faktor dengan cara ujicoba berdasarkan nilai rekomendasi pada penelitian sebelumnya ( $f_s = 15$ ,  $f_a = 10$ ,  $f_d = 20$ ,  $f_q = 50$  dan  $W = 100$ ), berdasarkan rekomendasi tersebut dilakukan pencarian nilai yang terbaik setelah penambahan faktor jarak ( $f_{ds}$ ) dan faktor tetangga ( $f_n$ ) dengan menggunakan skenario *grid*. Untuk tahap awal pencarian dilakukan perbandingan *routing* metrik RO, *delay*, dan *paket delivery ratio* dengan *routing* protokol AODV dengan jumlah *node* 50 dan 100 *node*. Dari hasil uji coba didapatkan nilai faktor  $f_s = 14$ ,  $f_a = 10$ ,  $f_d = 1$ ,  $f_{ds} = 19$ ,  $f_q = 32$ ,  $f_n = 19$ , nilai faktor yang didapat digunakan sebagai acuan nilai faktor pada penelitian ini.

#### 4.2 Implementasi sistem

Spesifikasi perangkat keras yang digunakan untuk penelitian ini dapat dilihat pada Tabel 4.1 Lingkungan simulasi.

Tabel 4.1 Lingkungan simulasi

Komponen	Spesifikasi
CPU	CPU Intel® Core™ i5-430M @ 2.10GHz × 4
Sistem Operasi	Linux Ubuntu 14.04 LTS 64-bit
Memori	6 GB PC3-12800 DDR3
Penyimpanan	200 GB

Sedangkan perangkat lunak yang digunakan dalam penelitian ini adalah:

1. SUMO versi 0.25.0 untuk pembuatan skenario mobilitas VANET.
2. JOSM versi 10301 untuk penyuntingan peta OpenStreetMap.

3. NS 2.35 untuk simulasi skenario VANET

#### 4.3 Parameter simulasi

Parameter lingkungan simulasi pada perangkat simulator NS-2 dapat dilihat pada Tabel 4.2 Parameter lingkungan simulasi

Tabel 4.2 Parameter lingkungan simulasi

Parameter	Spesifikasi
Network Simulator	NS-2.35
<i>Routing</i> protokol	AODV dan AODV yang telah dimodifikasi
Area simulasi	Grid = 800 m x 800 m Real = 600 m x 700 m
Jumlah kendaraan	10, 30, 50, 75, dan 100 200, 300, 400, 500 dan 600
Radius transmisi	250 m
Kecepatan maksimal kendaraan ( <i>node</i> )	10, 15, 20, 25 m/s
Agen	CBR (Constant Bit Rate)
<i>Node</i> pengirim dan <i>node</i> tujuan	Tetap (tidak bergerak)
Ukuran paket	512 Bytes
Packet rate	2 kB/s
Packet interval	1 paket/s
Protokol MAC	IEEE 802.11p
Model propagasi	Two-ray ground
Parameter faktor	$f_s = 14, f_a = 10, f_d = 1, f_{ds} = 19, f_q = 32, f_n = 19$ dan $W = 100$

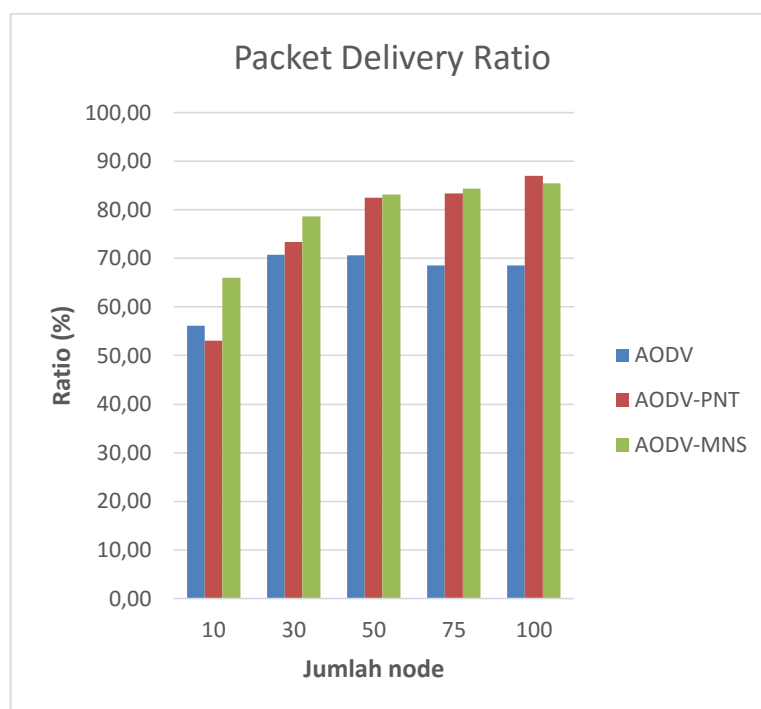
#### 4.4 Analisa hasil

Analisa hasil penelitian ini mencakup pembahasan tentang *routing* metrik yang dibandingkan antara lain RO (*routing overhead*), *delay*, PDR (*packet deliver ratio*) dan *hop count*.

##### 4.4.1 Hasil uji coba skenario grid

Uji coba skenario *grid* dilakukan sebanyak 10 kali dengan skenario mobilitas *random* pada peta *grid* dengan luas area 800m x 800m dan jumlah *node* sebanyak 10, 30, 50, 75 dan 100 dengan kecepatan maksimum dari setiap *node* adalah 15 m/s.

#### 4.4.1.1 Packet delivery ratio (PDR)



Gambar 4.1 Grafik PDR skenario grid

Gambar 4.1 merupakan grafik perbandingan PDR antara protokol AODV, AODV-PNT dengan protokol AODV-MNS. Hasil perhitungan rata-rata PDR dari skenario grid dengan penambahan jumlah *node* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil perhitungan rata-rata PDR(%) skenario grid

Jumlah Node	Jumlah rute AODV			PDR AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	8	7	5	56,1	53,044	66,076	13,03
30	11	4	3	70,75	73,35	78,6	5,25
50	14	8	7	70,65	82,45	83,15	0,70
75	12	6	5	68,5	83,35	84,35	1,00
100	19	7	7	68,5	87	85,5	-1,50

Berdasarkan Gambar 4.1 dapat dilihat bahwa PDR protokol AODV-MNS memiliki performa yang lebih baik dibandingkan dengan protokol AODV-PNT meskipun peningkatan tidak signifikan. Pada jumlah *node* 10, 30, 50 dan 75 PDR protokol AODV-MNS unggul sekitar 0,7%-13% lebih baik dibandingkan dengan protokol AODV-PNT, jika dilihat jumlah pergantian rute protokol

AODV-MNS memiliki perubahan rute yang sedikit dibandingkan dengan protokol yang lain, dengan perubahan rute yang sedikit menandakan relay node semakin stabil, dengan rute yang stabil otomatis PDR akan naik. Ketika jumlah *node* 100 PDR AODV-MNS lebih kecil dibandingkan dengan AODV-PNT, dengan selisih 1,5% penurunan PDR dapat disebabkan karena pemilihan *path* yang kurang tepat karena perhitungan TWR hanya menghitung node tetangga, ketika node tetangga yang dipilih optimal belum tentu *next hop* juga optimal sehingga mengakibatkan PDR menurun. Selain masalah perhitungan TWR, penurunan PDR dapat disebabkan karena banyaknya perubahan rute yang terjadi, dapat dilihat perubahan rute AODV-PNT dan AODV-MNS sama tetapi durasi bertahanya rute AODV-MNS lebih kecil dibandingkan dengan AODV-PNT, tabel durasi bertahanya rute dapat dilihat pada Tabel 4.5. Selisih PDR paling tinggi antara AODV-MNS dan AODV-PNT ketika *node* 10 sebanyak 13,03%. Berdasarkan Tabel 4.3, PDR protokol AODV-PNT lebih kecil dengan protokol AODV original ketika jumlah *node* 10, tetapi seiring penambahan jumlah *node* protokol AODV-PNT dan AODV-MNS selalu unggul dibandingkan dengan protokol AODV original.

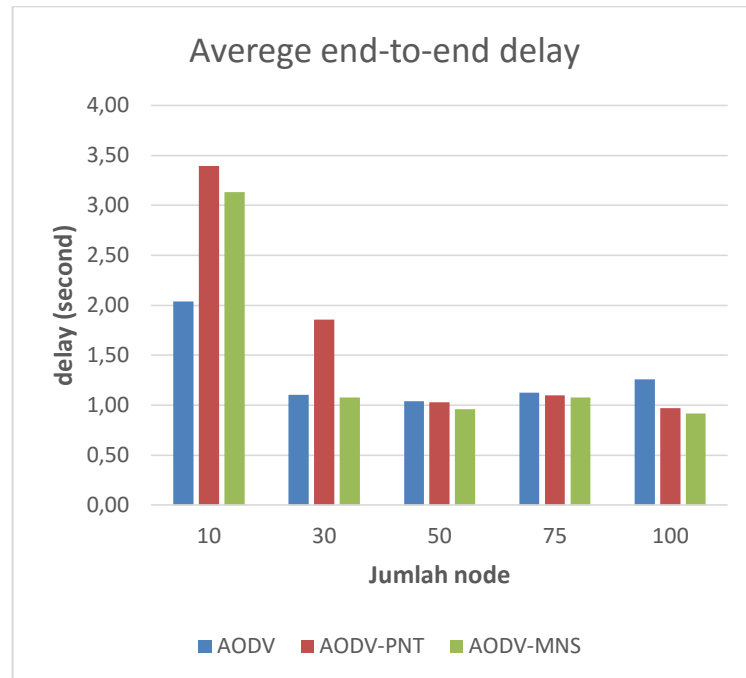
#### 4.4.1.2 Average end-to-end delay

Tabel 4.4 Hasil perhitungan rata-rata *end-to-end delay* (seconds) skenario real

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih PNT dan MNS
10	2,04	3,39	3,13	0,26
30	1,10	1,86	1,08	0,78
50	1,04	1,03	0,96	0,07
75	1,13	1,10	1,08	0,02
100	1,26	0,97	0,92	0,05

*Average end-to-end delay* adalah perhitungan yang digunakan untuk mengetahui nilai rata-rata waktu yang dibutuhkan oleh paket untuk sampai dari *source* sampai ke tujuan. Perbandingan perhitungan *average end-to-end delay* protokol AODV, AODV-PNT dan AODV-MNS dapat dilihat pada Tabel 4.4. Pada jumlah *node* 10 protokol AODV-PNT dan AODV-MNS memiliki *delay* yang tinggi dibandingkan dengan AODV original, 3,39 detik untuk AODV-PNT dan 3,13 detik *delay* yang dihasilkan AODV-MNS. Pada kepadatan node

yang rendah AODV-PNT dan AODV-MNS memerlukan waktu yang cukup lama dalam melakukan *route discovery*, *delay* yang dihasilkan AODV-MNT lebih kecil sekitar 0,25 detik. Pada jumlah pada jumlah *node* 30, 50, 75 dan 100 *delay* yang dihasilkan oleh protokol AODV-MNT lebih kecil dibandingkan dengan protokol AODV original maupun protokol AODV-PNT.



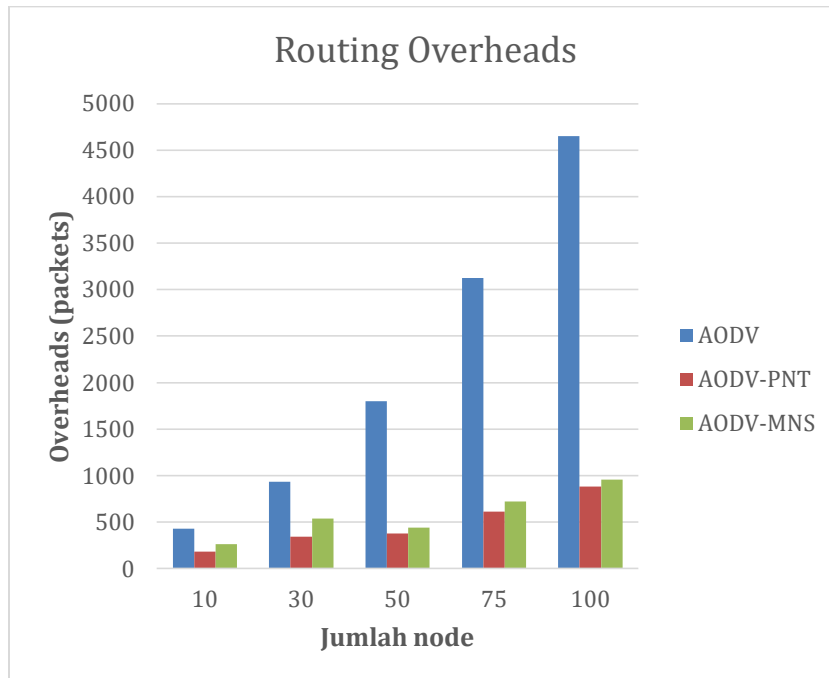
Gambar 4.2 Grafik rata-rata *end-to-end delay* skenario grid

Rata-rata selisih delay antara AODV-PNT dan AODV-MNS adalah 0,23 detik, delay yang dihasilkan AODV-PNT dan AODV-MNS semakin kecil seiring dengan bertambahnya jumlah node, dengan delay yang rendah dan RO yang rendah dapat meningkatkan PDR terbukti PDR yang dihasilkan protokol AODV-PNT dan AODV-MNS meningkat seiring dengan bertambahnya jumlah *node*. Grafik perbandingan rata-rata *end-to-end delay* pada skenario grid dapat dilihat pada Gambar 4.2.

#### 4.4.1.3 Routing overheads

*Routing overhead* adalah jumlah paket *routing* yang ditransmisikan, paket *routing* yang ditransmisikan terdiri dari RREQ, RREP, dan RERR. Berdasarkan Tabel 4.5 *routing overhead* pada AODV-MNS lebih tinggi dibandingkan dengan protokol AODV-PNT original rata-rata peningkatan RO

kurang lebih adalah 106,06 paket, selisih RO paling tinggi ketika jumlah node 30 dan 75. RO yang dihasilkan AODV-MNS lebih besar dibandingkan AODV-PNT dapat disebabkan protokol AODV-MNS memilih banyak *node* yang dianggap sebagai *node* stabil sehingga paket yang ditransmisikan bertambah sehingga RO juga meningkat.



Gambar 4.3 Hasil perhitungan rata-rata RO skenario *grid*

Jika protokol AODV-PNT dan AODV-MNS dibandingkan dengan protokol AODV, RO yang dihasilkan jauh lebih rendah karena AODV-PNT dan AODV-MNS hanya memilih *next-hop node* yang stabil dengan perhitungan TWR, *future TWR* dan melakukan *multicast RREQ* terhadap kumpulan *next-hop node (relay node set)* berdasarkan nilai TWR sedangkan AODV original melakukan *broadcast RREQ* sehingga lebih banyak jumlah paket *routing* yang beredar di jaringan. Perbandingan jumlah *routing overheads* yang dihasilkan setiap protokol dapat dilihat pada Gambar 4.3.

Salah satu penyebab meningkatnya RO pada jaringan adalah *link failure* yang terlalu sering sehingga node pengirim akan mengirimkan paket RERR pada jaringan untuk mendapatkan path kembali, contoh *path duration* pada ujicoba skenario *real* dengan jumlah 50 node yang terbentuk dapat dilihat pada

Tabel 4.6 dan Tabel 4.7. AODV-MNS memiliki jumlah *link failure* yang lebih sedikit dibandingkan dengan AODV original. Berdasarkan Tabel 4.6 dan Tabel 4.7 jumlah paket yang diterima oleh AODV-MNS lebih banyak dibandingkan dengan AODV original, dengan RO yang dihasilkan menjadi semakin rendah dan diharapkan PDR dapat meningkat.

Tabel 4.5 Hasil perhitungan rata-rata RO (*packets*) skenario *grid*

Jumlah Node	Path Duration AODV			RO AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	23,73	26,52	41,50	431,70	184,20	265,60	44,19
30	17,36	48,74	51,99	935,20	343,90	541,10	57,34
50	14,94	22,37	25,42	1798,10	377,60	441,70	16,98
75	16,84	28,65	41,98	3124,40	610,50	724,40	18,66
100	10,83	31,57	29,99	4649,00	880,70	954,40	8,37

Berdasarkan Tabel 4.5 perbandingan durasi bertahannya rute, protokol AODV-MNS memiliki rata-rata durasi paling lama, disusul AODV-PNT, dan terakhir AODV. Dari perbandingan jumlah pergantian rute (Tabel 4.3) dan durasi rute (Tabel 4.5), dapat disimpulkan bahwa rute yang dipilih oleh metode yang diusulkan lebih stabil daripada AODV-PNT dan AODV karena memiliki durasi rute yang lama.

Tabel 4.6 Contoh path duration AODV

Path No	Durasi (detik)	Jumlah Paket Terkirim
1	9,96	5
2	33,96	28
3	20,45	16
4	9,98	5
5	12,96	8
6	6,96	2
7	19,45	15
8	18,94	14
9	13,96	9
10	14,97	10
11	17,92	13
12	9,96	5
13	19,45	9
<b>Total paket</b>		139

Tabel 4.7 Contoh path duration AODV-MNS

Path No	Durasi (detik)	Jumlah Paket Terkirim
1	18,98	14
2	34,48	29
3	21,47	16
4	67,99	62
5	30,99	25
6	25,00	26
<b>Total paket</b>		172

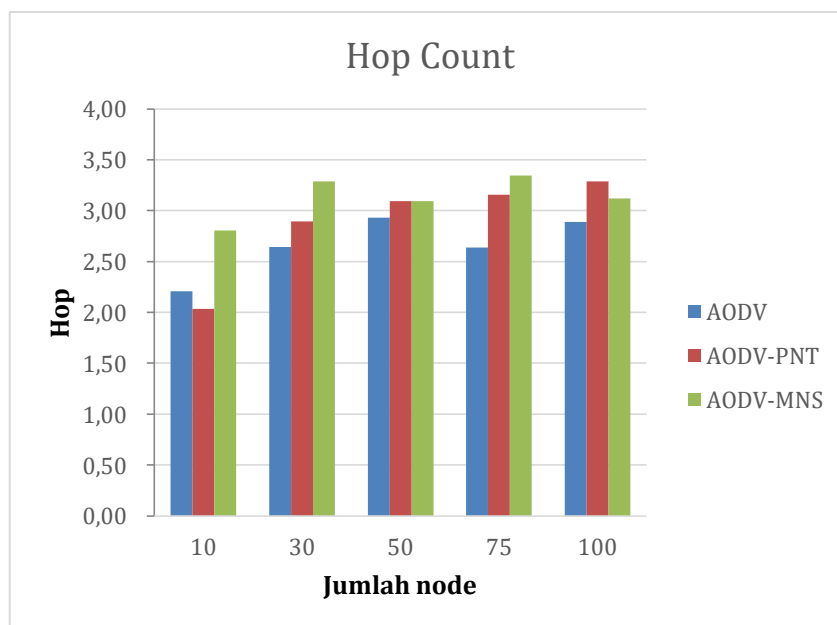
#### 4.4.1.4 Hop count

Tabel 4.8 Hasil perhitungan rata-rata *hop count* skenario *grid*

Jumlah node	AODV	AODV-PNT	AODV-MNS	Selisih PNT dan MNS
10	2,21	2,04	2,81	0,76
30	2,64	2,89	3,29	0,39
50	2,93	3,09	3,09	-0,0020
75	2,64	3,15	3,34	0,18
100	2,89	3,29	3,12	-0,16

*Hop count* adalah jumlah *link point-to-point* dalam jalur transmisi, hasil rata-rata *hop count* pada setiap skenario dapat dilihat pada Tabel 4.8. Berdasarkan nilai rata-rata hop protokol AODV-MNS modifikasi memiliki jumlah hop yang cukup banyak dibandingkan dengan protokol AODV-PNT, dengan rata-rata selisih kurang lebih 0,23 *hop* lebih panjang, pada jumlah *node* 50 dan 100 jumlah *hop* yang dihasilkan protokol AODV-MNS 0,002-0,16 hop lebih kecil dibandingkan dengan AODV-PNT, penurunan jumlah hop dapat disebabkan karena *relay node* yang dipilih saat pembentukan *path* sedikit.

Dengan jumlah *hop* yang lebih banyak dibandingkan AODV bukan menjadi masalah bagi protocol AODV-MNS, karena *relay node* yang dipilih oleh protokol AODV-MNS lebih stabil sehingga dapat meningkatkan PDR, grafik perbandingan rata-rata jumlah hop dapat dilihat pada Gambar 4.4



Gambar 4.4 Grafik rata-rata *hop count* skenario *grid*



Tabel 4.9 Contoh pemilihan *relay node* protokol AODV-MNS

Path No	Durasi (detik)	Relay node
1	18,98	51-16-14-47-50
2	34,48	51-25-42-50
3	21,47	51-18-31-6-50
4	67,99	51-29-32-50
5	30,99	51-18-17-49-50
6	25,00	51-16-40-29-50

Contoh perbandingan *path* yang terbentuk pada skenario *grid* dengan jumlah *node* 50 dapat dilihat pada Tabel 4.9 dan Tabel 4.10, 50 dan 51 adalah *source* dan *destination node*. Dapat dilihat meskipun rata-rata *path* yang dibentuk oleh protocol AODV-MNS memiliki *hop* lebih banyak dibandingkan dengan protokol AODV, tetapi frekuensi perubahan *path* AODV-MNS lebih kecil dibandingkan dengan AODV original 13 banding 6, dengan sedikitnya perubahan *path* pada durasi pengiriman paket cbr yang sama maka terbukti bahwa relay node yang dipilih dengan perhitungan TWR lebih stabil dibandingkan dengan AODV original

Tabel 4.10 Contoh pemilihan *relay node* protokol AODV

Path No	Durasi (detik)	Relay node
1	9,96	51-34-1-50
2	33,96	51-5-21-6-42-50
3	20,45	51-16-10-50
4	9,98	51-39-31-50
5	12,96	51-12-23-10-50
6	6,96	51-21-36-10-50
7	19,45	51-17-21-2-50
8	18,94	51-41-32-50
9	13,96	51-13-30-50
10	14,97	51-16-27-50
11	17,92	51-28-33-6-50
12	9,96	51-28-47-35-50
13	19,45	51-34-35-50

#### 4.4.2 Hasil uji coba modifikasi kecepatan pada skenario grid

Uji coba modifikasi kecepatan pada skenario *grid* dilakukan sebanyak 10 kali dengan skenario mobilitas *random* pada peta *grid* dengan luas area 800m x 800m dan jumlah *node* sebanyak 10, 30, 50, 75 dan 100 dengan variasi kecepatan maksimum dari setiap *node* adalah 10,15,20 dan 25 m/s, tetapi hasil yang

ditampilkan hanya variasi kecepatan pada jumlah node 75 untuk hasil uji coba pada variasi jumlah node yang lain dapat dilihat pada lampiran.

#### 4.4.2.1 Packet delivery ratio (PDR)

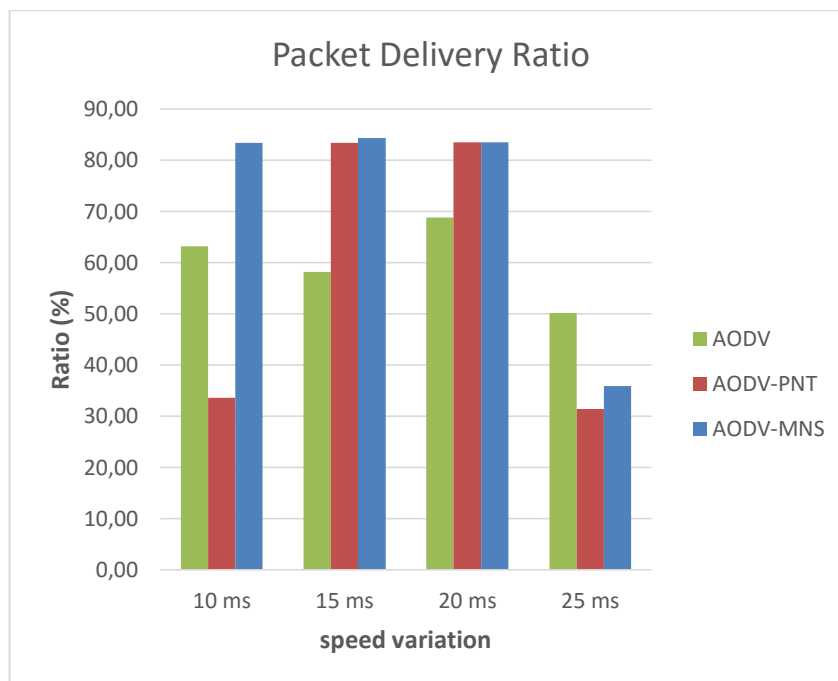
Nilai PDR yang dihasilkan dengan modifikasi kecepatan pada skenario *real* menunjukkan bahwa rata-rata PDR protokol AODV-MNS selalu unggul dibandingkan dengan protokol AODV-PNT biasa, dari 4 variasi kecepatan AODV-MNS selisih paling tinggi ketika kecepatan maksimal 10 m/s dengan selisih 49%, jika dilihat jumlah pergantian rute protokol AODV-MNS pada variasi kecepatan 10 m/s memiliki perubahan rute yang sedikit dibandingkan dengan protokol yang lain, dengan perubahan rute yang sedikit menandakan relay node semakin stabil sehingga dapat menghasilkan PDR yang tinggi. Pada variasi kecepatan 25 m/s AODV-MNS dan AODV-PNT tidak lebih baik dibandingkan dengan protokol AODV original, PDR yang dihasilkan AODV-MNS lebih kecil 14% dibandingkan dengan AODV original, . Menurunnya performa AODV-MNS dan AODV-PNT dapat disebabkan karena perhitungan yang dilakukan dalam pemilihan *relay node*. Berdasarkan Tabel 4.11 performa AODV-MNS dan AODV-PNT dapat berjalan stabil dengan rata-rata PDR diatas 80% ketika pada kecepatan 10-20 m/s.

Tabel 4.11 Hasil rata-rata PDR (%) uji coba modifikasi kecepatan skenario grid

Variasi Kecepatan (m/s)	Jumlah Rute AODV			PDR AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	12	6	4	63,19	33,57	83,42	49,85
15	12	6	5	58,20	83,35	84,35	1,00
20	9	5	6	68,76	83,50	83,45	-0,05
25	15	7	6	50,11	31,39	35,90	4,51

Rata-rata selisih kenaikan PDR pada protokol AODV-MNS dibandingkan dengan protokol AODV adalah 9%, sedangkan rata-rata kenaikan PDR yang dihasilkan AODV-MNS dan AODV-PNT adalah 13%. Grafik perbandingan PDR skenario *real* antara protokol AODV dan AODV-MNS dapat dilihat pada Gambar 4.5. Pada variasi kecepatan 20 m/s PDR yang dihasilkan AODV-MNS lebih kecil dibandingkan dengan AODV-PNT 0,05%,

penurunan PDR dapat disebabkan karena *relay node* yang dipilih bergerak cukup cepat sehingga kemungkinan path terputus semakin tinggi, selain itu jumlah perubahan rute AODV-MNS lebih besar dibandingkan dengan AODV-PNT.

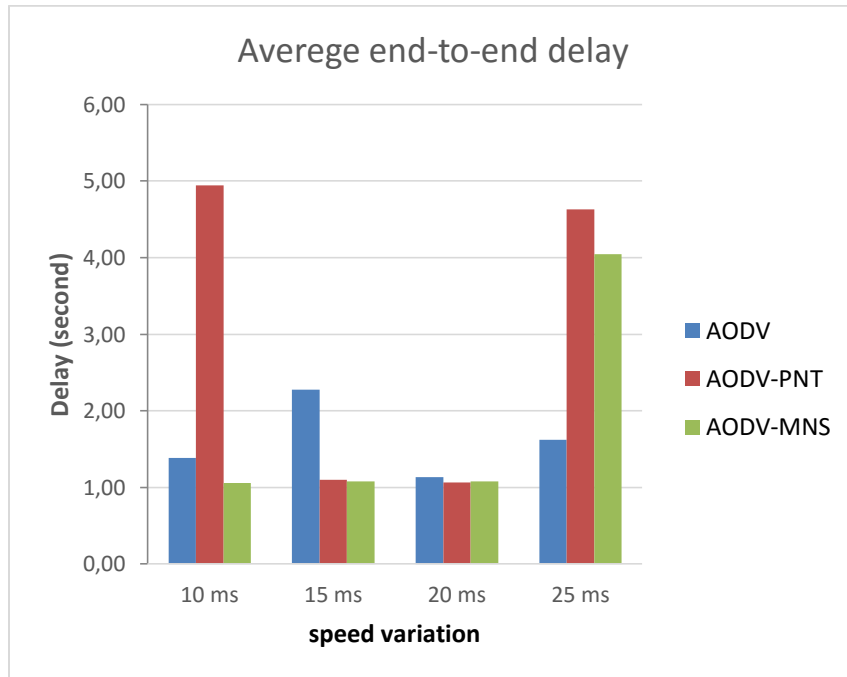


Gambar 4.5 Grafik PDR dengan modifikasi kecepatan pada skenario *real*

#### 4.4.2.2 Average end-to-end delay

Berdasarkan Gambar 4.6 delay yang dihasilkan oleh routing protokol AODV-MNS lebih kecil dibandingkan AODV original pada variasi kecepatan 10, 15, dan 20 m/s, sedangkan *delay* yang dihasilkan oleh protokol AODV-PNT dibandingkan dengan AODV lebih tinggi ketika variasi kecepatan 15 m/s dan 20 m/s dengan *delay* yang besar mengakibatkan penurunan PDR AODV-PNT dapat dilihat pada Gambar 4.5. Dari 4 variasi kecepatan, *delay* yang dihasilkan AODV-MNS hampir semua lebih kecil dibandingkan AODV-PNT, pada kecepatan 20 m/s delay yang dihasilkan AODV-MNS sedikit lebih besar dibandingkan dengan AODV-PNT, meningkatnya *delay* disebabkan karena *relay node* yang dipilih bergerak cukup cepat sehingga kemungkinan path terputus semakin tinggi, apabila *path* yang dipilih sering putus maka kemungkinan yang akan terjadi adalah meningkatnya RO dan delay terbukti

dengan RO yang dihasilkan AODV-MNS pada variasi kecepatan 20 m/s lebih besar dibandingkan dengan AODV-PNT (Tabel 4.13).



Gambar 4.6 Grafik end-to-end delay dengan modifikasi kecepatan pada skenario grid

Tabel 4.12 Rata-rata *delay (seconds)* uji coba modifikasi kecepatan skenario *grid*

Variasi Kecepatan (m/s)	AODV	AODV-PNT	AODV-MNS	Selisih AODV PNT dan MNS
10	1,38	4,94	1,05	3,89
15	2,27	1,09	1,07	0,02
20	1,13	1,06	1,08	-0,02
25	1,62	4,62	4,04	0,58

Penurunan nilai rata-rata *delay* AODV-MNS dibandingkan dengan AODV original pada variasi kecepatan 10, 15 dan 20m/s adalah 0,52 detik. Sedangkan selisih delay yang dihasilkan AODV-MNS dibandingkan dengan AODV-PNT adalah 1,2971 detik. Berdasarkan Gambar 4.6 pada variasi kecepatan 25 m/s delay yang dihasilkan AODV-PNT dan AODV-MNS dibandingkan AODV original lebih tinggi, tingginya delay yang dihasilkan dari kedua routing protokol tersebut mengakibatkan penurunan pada PDR.

#### 4.4.2.3 Routing overheads

Nilai rata-rata RO yang dihasilkan AODV-PNT dan AODV-MNS tidak terpengaruh oleh variasi kecepatan, berdasarkan Gambar 4.7 nilai rata-rata RO yang dihasilkan oleh protokol AODV-PNT dan AODV-MNS lebih kecil dibandingkan dengan protokol AODV original. RO protokol AODV yang dihasilkan paling tinggi ketika variasi kecepatan 20 m/s, nilai rata-rata RO dapat dilihat pada Tabel 4.13.

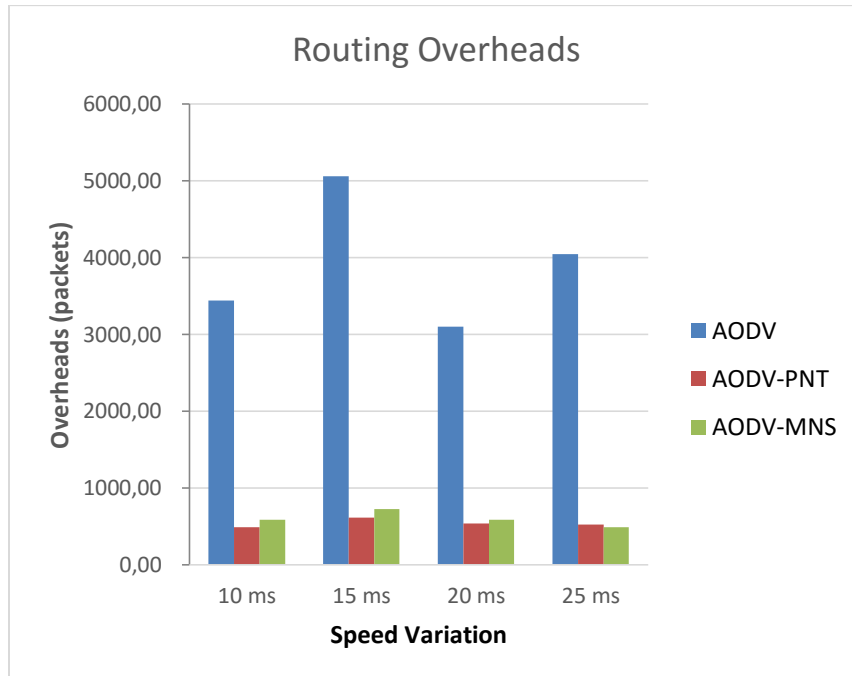
Tabel 4.13 Rata-rata RO (*packets*) skenario grid dengan variasi kecepatan

Variasi Kecepatan (m/s)	Path Duration AODV			RO AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	15,03	12,89	46,21	3439,16	487,97	585,48	97,51
15	14,20	32,28	38,24	5061,00	610,50	72,40	113,90
20	19,78	37,90	32,32	3096,42	536,87	585,48	48,61
25	11,18	12,35	12,48	4041,97	524,61	490,94	-33,67

Jika dilihat secara keseluruhan dengan variasi kecepatan nilai rata-rata, RO yang dihasilkan protokol AODV-PNT dan AODV-MNS lebih kecil dibandingkan dengan RO AODV original dengan rata-rata selisih 3233 paket lebih sedikit, sedangkan RO yang dihasilkan AODV-PNT lebih kecil dibandingkan dengan AODV-MNS. Pada variasi kecepatan 10, 15 dan 20 m/s, selisih RO yang dihasilkan AODV-PNT 86,68 paket lebih sedikit dibandingkan AODV-MNS. Berdasarkan Tabel 4.13 perbandingan durasi bertahannya rute, protokol AODV-MNS memiliki rata-rata durasi paling lama, disusul AODV-PNT, dan terakhir AODV. Dari perbandingan jumlah pergantian rute (Tabel 4.11) dan durasi rute, dapat disimpulkan bahwa rute yang dipilih oleh metode yang diusulkan lebih stabil daripada AODV-PNT dan AODV karena memiliki durasi rute yang lama.

Ketika variasi kecepatan 25 m/s RO yang dihasilkan AODV-PNT lebih besar dibandingkan dengan AODV-MNS dengan selisih rata-rata sekitar 34 paket data, dengan selisih tersebut dapat memberikan dampak pada nilai PDR terbukti pada variasi kecepatan 25 m/s PDR yang dihasilkan oleh protokol AODV-PNT lebih besar dibandingkan dengan AODV-MNS dengan selisih 4%. Berdasarkan Tabel 4.13 perhitungan TWR yang dilakukan AODV-PNT

dan AODV-MNS dapat menurunkan RO pada jaringan, tetapi penurunan RO tidak selalu dapat meningkatkan PDR terbukti pada variasi kecepatan 25 m/s PDR AODV-PNT dan AODV-MNS tidak lebih baik daripada PDR yang dihasilkan AODV original.



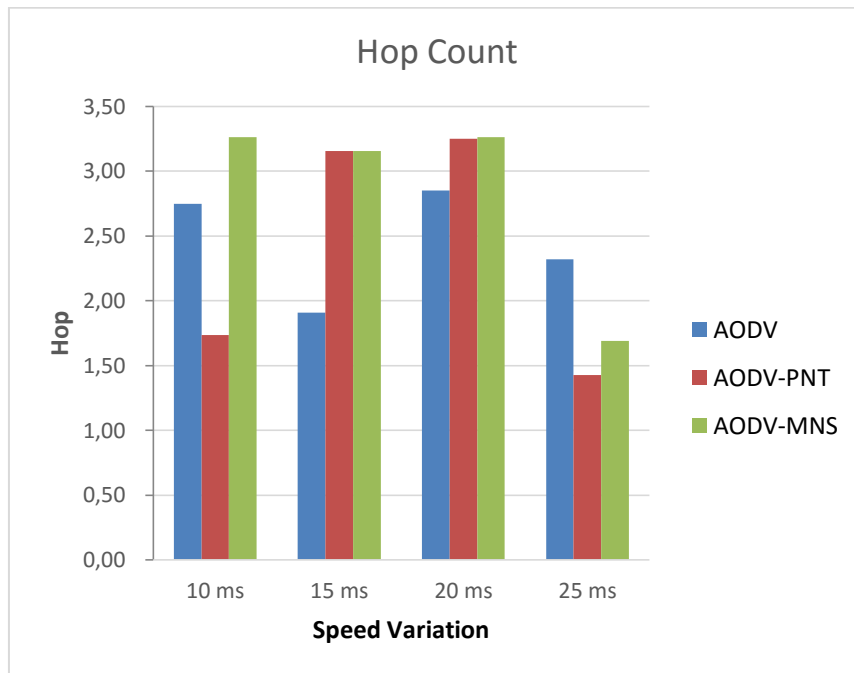
Gambar 4.7 Grafik *routing overheads* skenario real dengan modifikasi kecepatan

#### 4.4.2.4 Hop count

Tabel 4.14 Rata-rata hop skenario *grid* dengan variasi kecepatan

Variasi Kecepatan (m/s)	AODV	AODV-PNT	AODV-MNS	Selisih AODV MNS dan PNT
10	2,75	1,74	3,26	1,53
15	1,91	3,15	3,15	0,00
20	2,85	3,25	3,26	0,01
25	2,32	1,43	1,69	0,26

Jumlah rata-rata *hop count* yang dihasilkan oleh *routing* protokol AODV-MNS lebih besar dibandingkan dengan rata-rata *hop count* yang dihasilkan oleh *routing* protokol AODV-PNT, berdasarkan pada Tabel 4.14 AODV-MNS memiliki rata-rata 2,84 *hop*, sedangkan AODV-PNT memiliki rata-rata 2,39 *hop*.



Gambar 4.8 Grafik rata-rata hop count dengan modifikasi kecepatan pada skenario grid

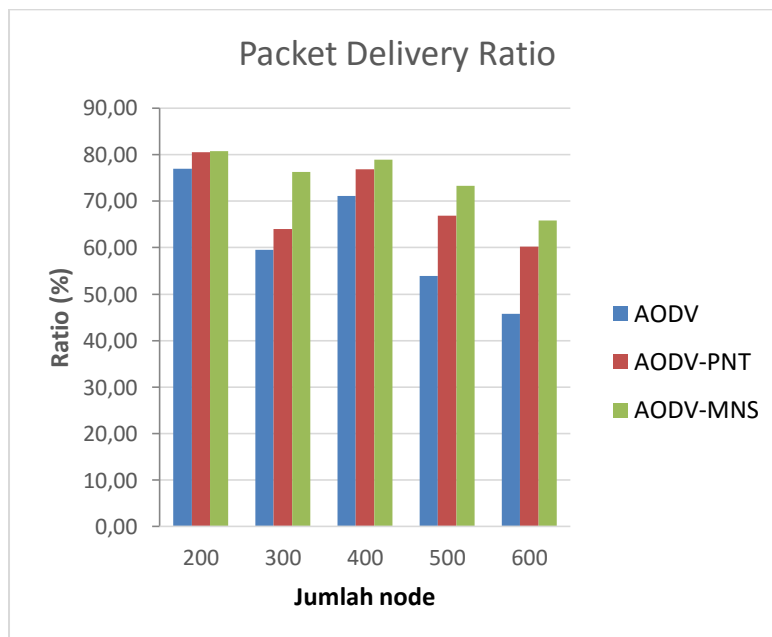
Berdasarkan grafik pada Gambar 4.8, jumlah *hop* yang dihasilkan *routing* protokol AODV-MNS lebih banyak jika dibandingkan dengan AODV-PNT, dengan rata-rata selisih 0,45 *hop* lebih banyak dibandingkan AODV-PNT. Tetapi pada variasi kecepatan 25 m/s jumlah hop AODV-PNT dan AODV-MNS lebih kecil dibandingkan AODV original, penurunan jumlah hop mempengaruhi nilai PDR, protokol AODV-PNT dan AODV-MNS kesulitan dalam memilih *relay node* maka perlu adanya perubahan nilai bobot faktor kecepatan ketika terjadi variasi kecepatan yang tinggi.

#### 4.4.3 Hasil uji coba densitas tinggi pada skenario grid

Uji coba densitas tinggi pada skenario *grid* dilakukan sebanyak 10 kali dengan skenario mobilitas *random* pada peta *grid* dengan luas area 800m x 800m dengan variasi jumlah *node* sebanyak 200, 300, 400, 500 dan 600 kecepatan rata-rata setiap *node* adalah 15 m/s. Uji coba dilakukan untuk mengetahui seberapa baik performa protokol AODV-MNS ketika diimplementasikan pada jaringan yang sangat padat.

#### 4.4.3.1 Packet delivery ratio (PDR)

Nilai PDR yang dihasilkan dengan modifikasi jumlah node pada skenario *grid* menunjukkan bahwa protokol AODV-MNS selalu unggul dibandingkan dengan protokol AODV-PNT, dari 4 variasi jumlah node selisih paling tinggi pada variasi 300 *node* dengan selisih 12,26%. Pada variasi 200 node protokol AODV-MNS dan AODV-PNT memiliki selisih PDR yang sangat kecil sekitar 0,25%, nilai rata-rata PDR yang dihasilkan dapat dilihat pada Tabel 4.15. Dari hasil uji coba variasi jumlah node yang dilakukan, protokol AODV-MNS memiliki rata-rata PDR 75% sedangkan rata-rata PDR protokol AODV adalah 61%.



Gambar 4.9 Grafik PDR dengan modifikasi jumlah *node* pada skenario *grid*

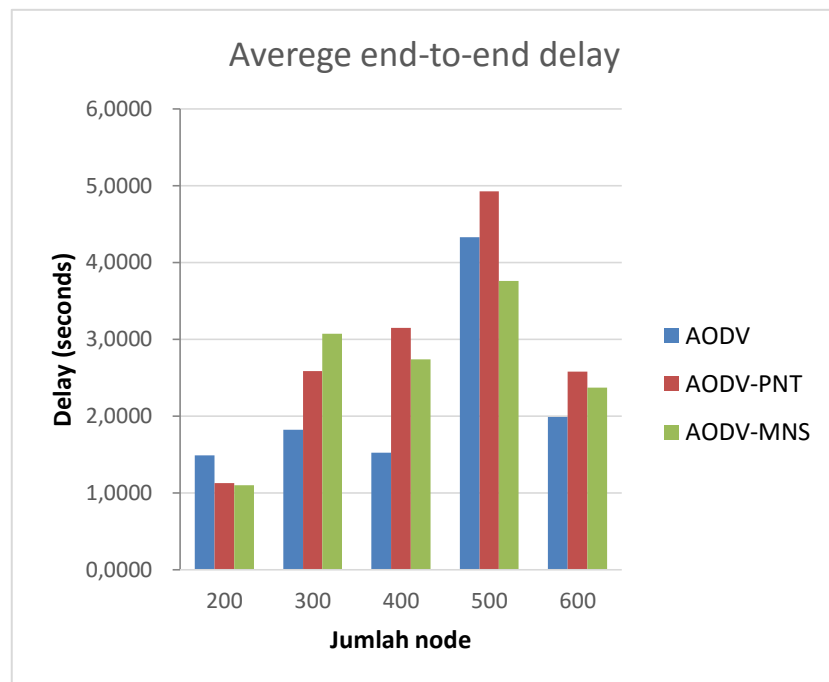
Berdasarkan Gambar 4.9 semua protokol mengalami penurunan nilai rata-rata PDR seiring dengan penambahan jumlah node, penurunan PDR disebabkan karena tingginya RO pada jaringan, tidak dapat dipungkiri bahwa masalah utama pada jaringan yang padat adalah RO, RO yang dihasilkan dalam uji coba ini dapat dilihat pada Gambar 4.11. Protokol AODV-MNS dan AODV PNT selalu unggul dibandingkan dengan AODV original meskipun ada penurunan nilai rata-rata PDR yang dihasilkan. Rata-rata penurunan PDR pada protokol AODV-MNS adalah 3%, AODV-PNT 5% dan AODV original 7%.



Tabel 4.15 Rata-rata PDR (%) uji coba densitas tinggi skenario *grid*

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV MNS dan PNT
200	77,00	80,50	80,75	0,25
300	59,56	63,99	76,25	12,26
400	71,07	76,86	78,93	2,07
500	53,92	66,87	73,27	6,40
600	45,72	60,19	65,90	5,70

#### 4.4.3.2 Average end-to-end delay



Gambar 4.10 Grafik end-to-end delay dengan modifikasi jumlah node pada skenario *grid*

Berdasarkan Gambar 4.10 *delay* yang dihasilkan oleh routing protokol AODV-MNS cenderung meningkat, delay yang paling tinggi ketika jumlah node 500, sedangkan paling rendah adalah ketika jumlah node 200. Berdasarkan grafik tersebut peningkatan delay yang dihasilkan sedikit mempengaruhi performa AODV-MNS, terbukti pada PDR yang dihasilkan cenderung menurun seiring bertambahnya jumlah *node*. Delay yang dihasilkan AODV-MNS tidak selalu lebih kecil dibandingkan protokol AODV original. Berdasarkan Tabel 4. 16 dengan jumlah variasi *node* 300 *delay* yang dihasilkan AODV-MNS lebih besar dibandingkan dengan AODV-PNT, meningkatnya *delay* dapat disebabkan oleh lamanya proses pemilihan relay

*node*, peningkatan *delay* juga dapat disebabkan karena perhitungan TWR dan *future* TWR.

Tabel 4. 16 Rata-rata delay (*seconds*) uji coba densitas tinggi skenario *grid*

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV PNT dan MNS
200	1,49	1,13	1,10	0,03
300	1,82	2,59	3,07	-0,49
400	1,52	3,15	2,74	0,41
500	4,33	4,93	3,76	1,17
600	1,99	2,58	2,37	0,20

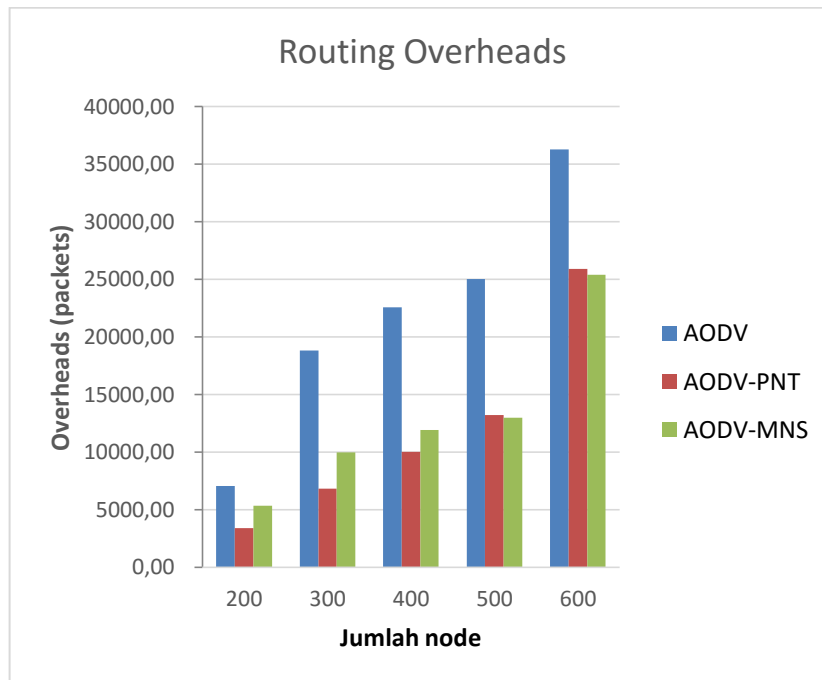
#### 4.4.3.3 Routing overheads

Nilai rata-rata RO yang dihasilkan AODV, AODV-PNT dan AODV-MNS tidak terpengaruh oleh variasi kecepatan, berdasarkan Gambar 4.11 nilai rata-rata RO yang dihasilkan oleh protokol AODV-PNT dan AODV-MNS lebih kecil dibandingkan dengan protokol AODV original. RO protokol AODV-MNS lebih kecil daripada RO AODV-MNS ketika variasi jumlah node 500 dan 600.

Tabel 4.17 Rata-rata RO (*packets*) uji coba densitas tinggi skenario *grid*

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV PNT dan MNS
200	7031,5	3380,0	5322,5	-1942,50
300	18791,4	6814,7	9974,1	-3159,40
400	22583,3	10014,3	11902,4	-1888,10
500	25000,0	13191,3	12992,6	198,70
600	36251,7	25882,5	25377,0	505,50

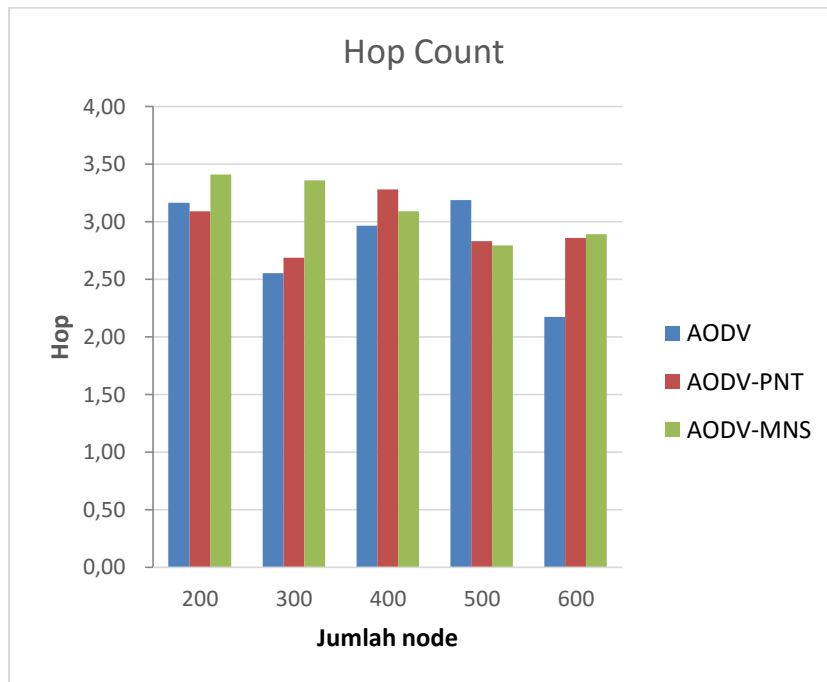
Berdasarkan data diatas perhitungan TWR yang dilakukan AODV-PNT dan AODV-MNS dapat menurunkan RO pada jaringan, tetapi penurunan RO tidak selalu dapat meningkatkan PDR terbukti pada variasi kecepatan 25 m/s PDR AODV-PNT dan AODV-MNS tidak lebih baik daripada PDR yang dihasilkan AODV original. Pada jumlah *node* 200, 300 dan 400 jumlah RO yang dihasilkan AODV-MNS lebih besar dibandingkan dengan AODV-PNT, meningkatnya RO disebabkan AODV-MNS memilih *relay node* lebih banyak dibandingkan AODV-PNT sehingga paket yang disebar juga semakin banyak.



Gambar 4.11 Grafik *routing overheads* modifikasi jumlah node pada skenario grid

#### 4.4.3.4 Hop count

Berdasarkan grafik pada Gambar 4.12, jumlah *hop* yang dihasilkan *routing* protokol AODV-MNS lebih bervariasi jika dibandingkan dengan uji coba variasi node dibawah 100, AODV-MNT memilih *relay node* yang stabil berdasarkan perhitungan TWR, pada uji coba ini pada variasi node 500 dan 600 node hampir tidak dapat bergerak sehingga banyak node yang berhenti. Dalam perhitungan untuk pemilihan *relay node* bergantung pada nilai *weight* faktor, pada penelitian ini *weight* faktor yang paling besar adalah *link quality*, jumlah node tetangga kemudian disusul dengan faktor lain sehingga ketika banyak *node* yang berhenti diantara *source* dan *destination node*, maka node tersebut dianggap stabil kemudian dipilih sebagai *relay node* hal tersebut dapat menyebabkan *hop count* meningkat. Pada variasi jumlah node 600 *hop count* yang dihasilkan AODV-MNS lebih tinggi dibandingkan dengan AODV-PNT maupun AODV original, dengan penambahan *hop count* yang dihasilkan tidak mempengaruhi nilai PDR, terbukti bahwa nilai PDR protokol AODV-MNS masih lebih unggul dibandingkan dengan protokol AODV-PNT. Tabel 4.14 adalah hasil rata-rata *hop* uji coba densitas tinggi.



Gambar 4.12 Grafik rata-rata hop count dengan variasi jumlah node pada skenario grid

Tabel 4. 18 Rata-rata *hop* uji coba densitas tinggi skenario *grid*

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV MNS dan PNT
200	3,17	3,09	3,41	0,32
300	2,55	2,69	3,36	0,67
400	2,96	3,28	3,09	-0,19
500	3,19	2,83	2,79	-0,04
600	2,17	2,86	2,89	0,03

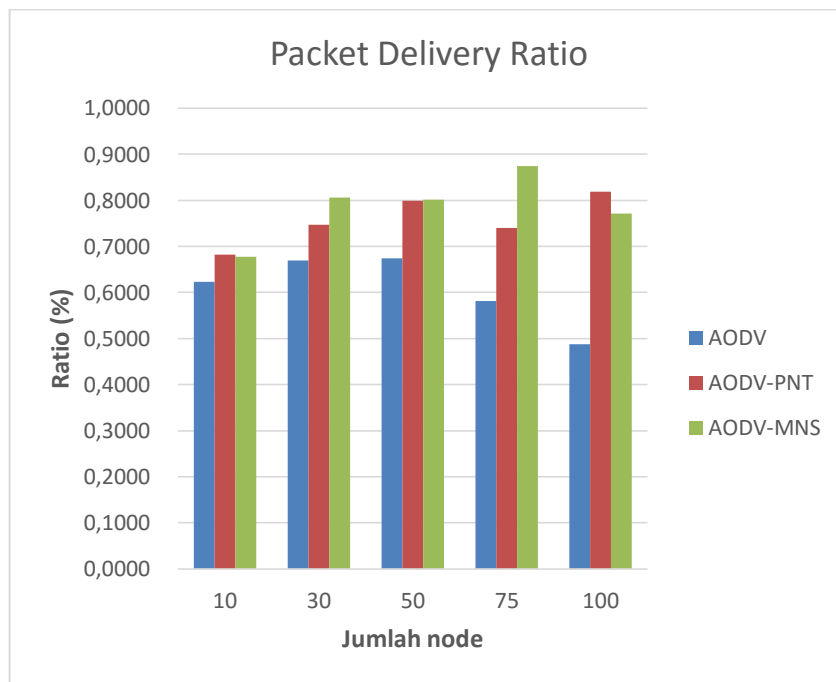
#### 4.4.4 Hasil uji coba skenario *real*

Uji coba skenario *real* sama seperti uji coba skenario *grid* dilakukan sebanyak 10 kali dengan skenario mobilitas *random* pada peta *grid* dengan luas area 600m x 700m dengan jumlah variasi *node* 10, 30, 50, 75 dan 100 dengan kecepatan maksimum dari setiap *node* adalah 15 m/s.

##### 4.4.4.1 Packet delivery ratio (PDR)

Nilai PDR yang dihasilkan pada skenario *real* menunjukkan bahwa protokol AODV-MNS tidak selalu unggul jika dibandingkan dengan protokol AODV-PNT, dari 5 variasi jumlah *node* AODV-MNS unggul pada variasi jumlah *node* 30, 50, 75 selisih paling tinggi ketika jumlah *node* 75 sebesar 13,4%, dapat dilihat pada Tabel 4.19 jumlah pergantian rute protokol AODV-

MNS memiliki perubahan rute yang sedikit dibandingkan dengan protokol yang lain, dengan perubahan rute yang sedikit menandakan relay node semakin stabil, dengan rute yang stabil otomatis PDR akan naik. Berdasarkan Tabel 4.19 nilai PDR AODV paling tinggi ketika jumlah *node* 50, namun setelah penambahan jumlah node PDR AODV mengalami penurunan. Penurunan PDR protokol AODV disebabkan oleh jumlah RO yang tinggi pada jaringan seperti pada Gambar 4.15, dimana terjadi peningkatan RO pada setiap penambahan *node*. Sedangkan protokol AODV-PNT dan AODV-MNS memiliki kecenderungan PDR yang meningkat meskipun ada penambahan jumlah *node* kecuali pada jumlah *node* 10, dengan nilai rata-rata PDR 80%.



Gambar 4.13 Grafik *packet deliver ratio* skenario *real*

Tabel 4.19 Hasil rata-rata perhitungan PDR (%) skenario *real*

Jumlah Node	Jumlah rute AODV			PDR (%) AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	9	7	7	62,3	68,2	67,7	-0,50
30	6	6	4	66,9	74,75	80,55	5,80
50	9	5	3	67,4	79,9	80,15	0,25
75	12	9	6	58,2	74,05	87,45	13,40
100	7	5	3	48,75	81,85	77,15	-4,70

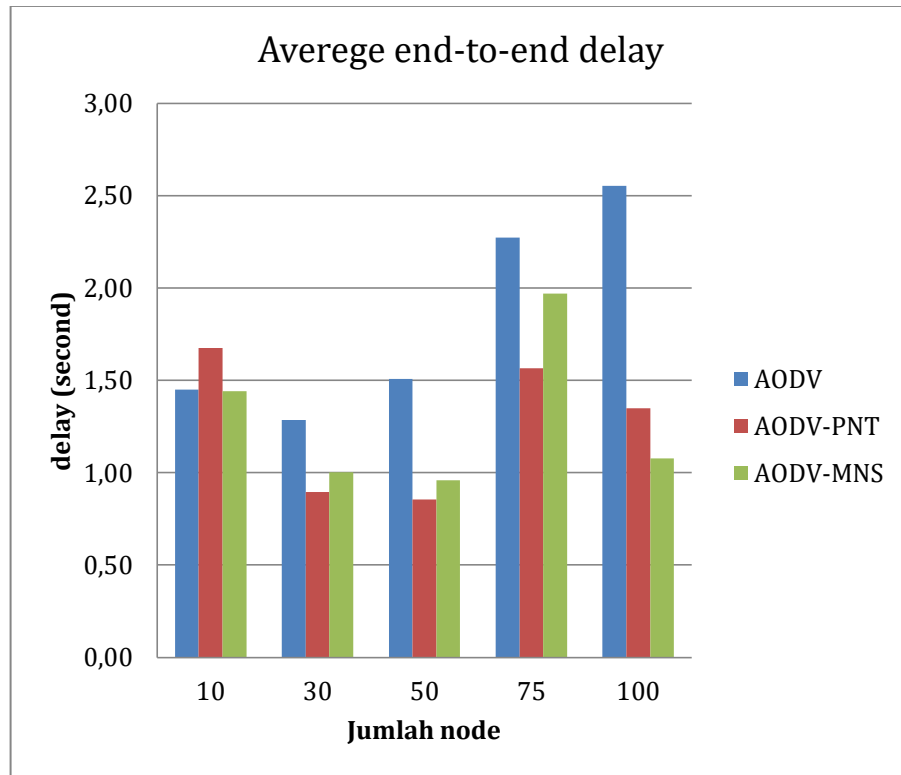
Pada saat jumlah node 10 dan 100 node PDR AODV-MNS lebih kecil dibandingkan PDR AODV-PNT, penurunan PDR AODV-MNS dapat disebabkan pemilihan path yang kurang tepat karena perhitungan TWR, selain itu penurunan PDR dapat disebabkan karena *node* tetangga yang dipilih sebagai *relay node* optimal belum tentu *node* tetangga tersebut memiliki *next hop* optimal, sehingga mengakibatkan penurunan PDR, dapat dilihat perubahan rute AODV-PNT dan AODV-MNS sama tetapi durasi bertahanya rute AODV-MNS lebih kecil dibandingkan dengan AODV-PNT, tabel durasi bertahanya rute dapat dilihat pada Tabel 4.21. Grafik perbandingan PDR skenario real antara protokol AODV dan AODV-MNS dapat dilihat pada Gambar 4.13.

#### 4.4.4.2 Average end-to-end delay

Berdasarkan Tabel 4.20 ketika jumlah *node* 10, *delay* yang dihasilkan oleh *routing* protokol AODV-MNS hampir sama dengan *routing* protokol AODV original dengan selisih 0,0101 detik, tetapi jika dibandingkan dengan AODV-PNT, nilai *delay* yang dihasilkan AODV-PNT lebih tinggi, dapat dilihat pada Gambar 4.14. *Delay* yang cukup tinggi pada jumlah *node* 10 dapat disebabkan oleh lamanya proses pemilihan *relay node*, peningkatan *delay* juga dapat disebabkan karena perhitungan TWR dan *future* TWR karena dengan jumlah 10 *node* saat simulasi cukup sulit untuk memilih *relay node* karena pilihan *node* yang dapat digunakan sebagai *relay node* juga sedikit.

Pada saat jumlah *node* 30 dan 50 *delay* yang dihasilkan AODV-PNT dan AODV-MNS memiliki kecenderungan menurun, selisih *delay* AODV-MNS dan AODV-PNT original adalah 0,1078 dan 0,1056 detik, pada saat jumlah *node* 75 rata-rata *delay* yang dihasilkan oleh protokol AODV-MNS dan AODV-PNT meningkat tajam, tetapi *delay* yang dihasilkan masih dibawah rata-rata *delay* AODV original. Rata-rata *delay* yang tinggi dihasilkan AODV-MNS saat jumlah *node* 75 berbanding lurus dengan rata-rata RO, dimana nilai rata-rata RO AODV-MNS pada saat jumlah *node* 75 adalah paling tinggi. Peningkatan nilai rata-rata *delay* pada AODV-MNS pada saat jumlah *node* 75 kemungkinan dapat disebabkan oleh RO yang tinggi, selisih rata-rata *delay*

antara AODV-PNT dan AODV-MNS adalah 0,0221 detik. Grafik perbandingan rata-rata *delay* yang dihasilkan *routing* protokol AODV, AODV-PNT dan AODV-MNS dapat dilihat pada Gambar 4.14.



Gambar 4.14 Grafik rata-rata *end-to-end delay* skenario *real*

Tabel 4.20 Hasil perhitungan rata-rata *end-to-end delay (seconds)* skenario *real*

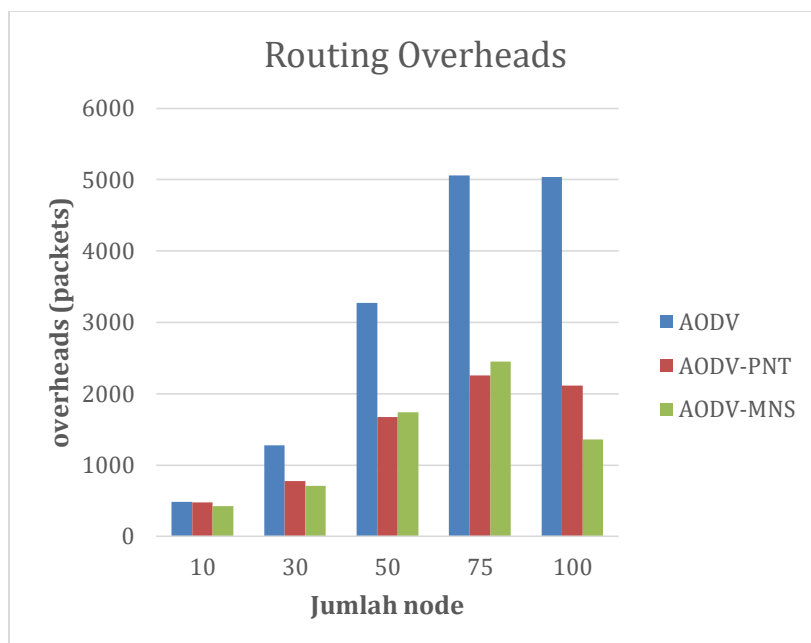
Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih PNT dan MNS
10	1,45	1,67	1,44	-0,23
30	1,28	0,89	1,00	0,10
50	1,50	0,85	0,95	0,10
75	2,27	1,56	1,96	0,40
100	2,55	1,34	1,07	-0,27

#### 4.4.4.3 Routing overheads

Tabel 4.21 Hasil perhitungan rata-rata *routing overheads (packets)* skenario *real*

Jumlah Node	Path Duration AODV			RO AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	11,55	19,49	19,04	484,1	474	428,6	-9,58
30	10,72	26,85	34,37	1275,3	776,8	709,3	-8,69
50	23,20	41,69	60,30	3269,1	1675,9	1741,9	3,94
75	12,48	12,45	29,33	5061,0	2253,7	2448,4	8,64
100	29,28	30,86	69,65	5037,1	2114,8	1357,2	-35,82

Nilai rata-rata RO yang dihasilkan berbanding lurus dengan jumlah *node*, berdasarkan Tabel 4.21 nilai rata-rata RO yang dihasilkan oleh protokol AODV-PNT dan AODV-MNS lebih kecil dibandingkan dengan protokol AODV original. Berdasarkan Tabel 4.21 perbandingan durasi bertahannya rute, protokol AODV-MNS memiliki rata-rata durasi paling lama, disusul AODV-PNT, dan terakhir AODV. Dari perbandingan jumlah pergantian rute (Tabel 4.19) dan durasi rute Tabel 4.5, dapat disimpulkan bahwa rute yang dipilih oleh metode yang diusulkan lebih stabil daripada AODV-PNT dan AODV karena memiliki durasi rute yang lama. Nilai rata-rata RO protokol AODV semakin meningkat seiring penambahan jumlah *node*, begitu juga dengan *routing* protokol AODV-MNS dan AODV-PNT memiliki kecenderungan yang sama, tetapi pada saat jumlah *node* 100, nilai rata-rata RO yang dihasilkan oleh *routing* protokol AODV-MNS dan PNT menurun.



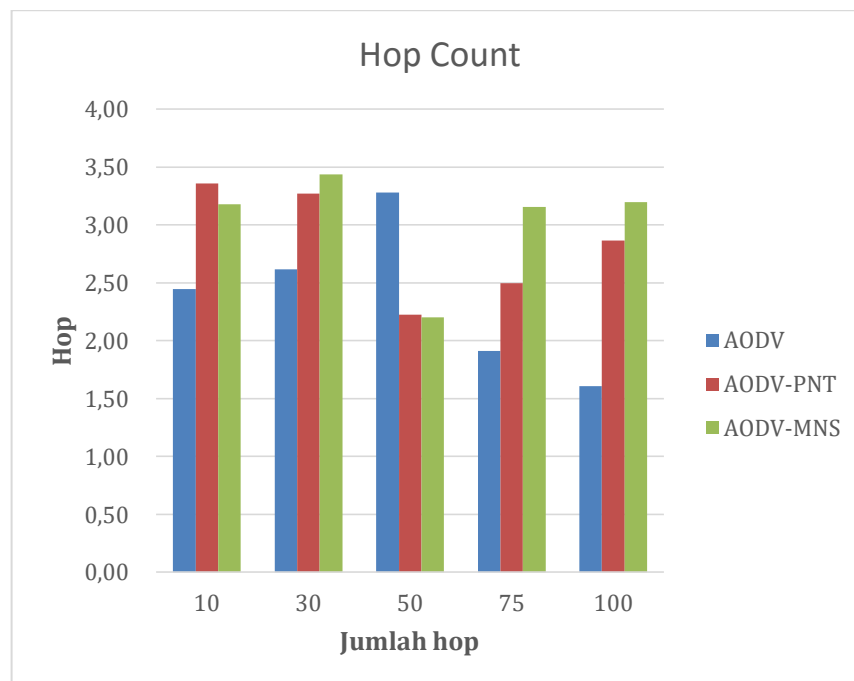
Gambar 4.15 Grafik *routing overheads* skenario *real*

Jika dilihat secara keseluruhan nilai rata-rata RO yang dihasilkan AODV-PNT dan AODV-MNS lebih kecil dibandingkan RO AODV original, RO yang dihasilkan AODV-PNT lebih besar dibandingkan AODV-MNS dengan rata-rata selisih 121,96 paket. RO AODV-PNT lebih kecil dibandingkan RO AODV-MNS ketika jumlah *node* 50 dan 75 dengan selisih 66 dan 194,7 paket.



Selisih tertinggi pada saat jumlah *node* 100 sebanyak 757,6 paket. Dengan rata-rata RO yang lebih kecil dibandingkan AODV-PNT, modifikasi protokol AODV-PNT menjadi AODV-MNS cukup berhasil karena RO yang dihasilkan lebih kecil. Penambahan *weight faktor* dapat meningkatkan seleksi *node*, sehingga *node* yang dipilih sebagai *relay node* lebih stabil.

#### 4.4.4.4 Hop count



Gambar 4. 16 Grafik *hop count* skenario real

Tabel 4.22 Hasil perhitungan rata-rata *hop count* skenario real

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih MNS dan PNT
10	2,45	3,36	3,18	-0,18
30	2,62	3,27	3,44	0,16
50	3,28	2,23	2,20	-0,02
75	1,91	2,49	3,15	0,65
100	1,61	2,87	3,20	0,33

Berdasarkan pada Tabel 4.22, jumlah rata-rata *hop count* yang dihasilkan oleh *routing* protokol AODV-MNS lebih lebih besar dibandingkan dengan rata-rata *hop count* yang dihasilkan oleh *routing* protokol AODV-PNT. AODV-MNS memiliki rata-rata 3,03 *hop count* pada setiap variasi *node*,

sedangkan AODV-PNT memiliki rata-rata 2,84 *hop count* pada setiap variasi *node*. Berdasarkan grafik pada Gambar 4. 16, jumlah *hop* yang dihasilkan *routing* protokol AODV-MNS lebih banyak jika dibandingkan dengan AODV-PNT, dengan rata-rata selisih 0,19 hop. Selisih jumlah hop paling kecil ketika jumlah *node* 30, meskipun dengan jumlah hop yang lebih panjang AODV-MNS memiliki PDR yang lebih tinggi dibandingkan dengan AODV original maka penambahan jumlah hop tidak mempengaruhi PDR.

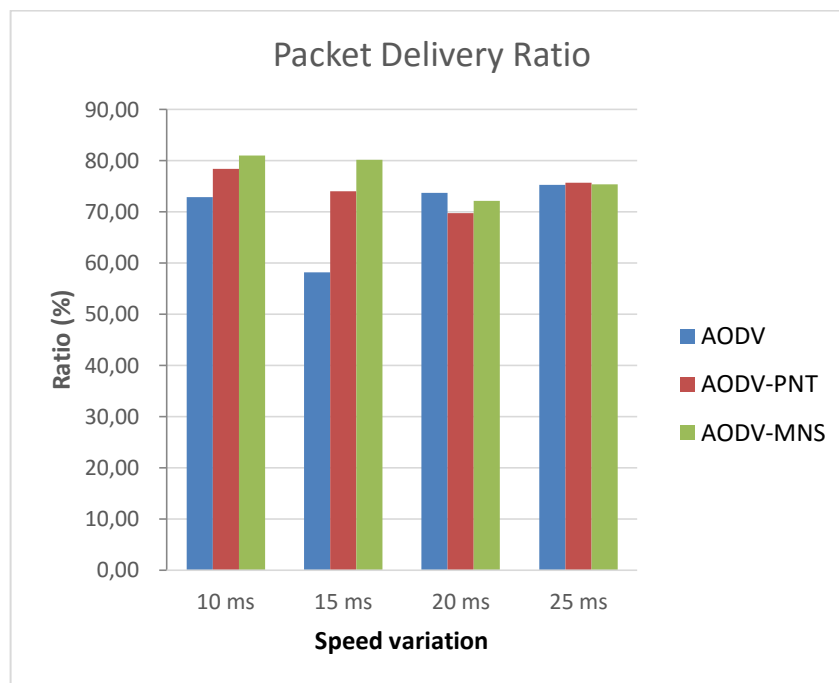
#### **4.4.5 Hasil uji coba modifikasi kecepatan pada skenario real**

Uji coba modifikasi kecepatan pada skenario *real* dilakukan sebanyak 10 kali dengan skenario mobilitas *random* pada peta *real* dengan luas area 600m x 700m dan jumlah *node* sebanyak 10, 30, 50, 75 dan 100 dengan variasi kecepatan maksimum dari setiap *node* adalah 10,15,20 dan 25 m/s, tetapi hasil yang ditampilkan hanya variasi kecepatan pada jumlah node 75 untuk hasil uji coba pada variasi jumlah node yang lain dapat dilihat pada lampiran.

##### **4.4.5.1 Packet delivery ratio (PDR)**

Nilai PDR yang dihasilkan dengan modifikasi kecepatan pada skenario *real* pada Tabel 4.23 menunjukkan bahwa protokol AODV-MNS tidak selalu unggul dibandingkan dengan protokol AODV-PNT biasa, dari 4 variasi kecepatan AODV-MNS memiliki selisih paling tinggi ketika kecepatan maksimal 15 m/s dengan selisih 6%, salah satu faktor yang dapat meningkatkan PDR adalah jumlah pergantian rute, protokol AODV-MNS memiliki perubahan rute yang sedikit dibandingkan dengan protokol yang lain dengan perubahan rute yang sedikit menandakan relay node semakin stabil. Pada variasi kecepatan 25 m/s PDR AODV-MNS dan AODV-PNT lebih kecil dibandingkan dengan PDR AODV original, PDR yang dihasilkan AODV-MNS lebih kecil 0,2 % dibandingkan dengan AODV original, menurunnya performa AODV-MNS dan AODV-PNT dapat disebabkan karena perhitungan saat pemilihan *relay node*. Berdasarkan Gambar 4. 16 performa AODV-MNS dan AODV-PNT dapat berjalan stabil dengan rata-rata PDR diatas 8% ketika pada kecepatan 10-20 m/s. Rata-rata selisih kenaikan PDR pada protokol AODV-MNS dibandingkan dengan protokol

AODV adalah 7%, selisih kenaikan PDR protokol AODV-MNS paling tinggi ketika variasi kecepatan 10 m/s sekitar 8%. Grafik perbandingan PDR skenario *real* antara protokol AODV dan AODV-MNS dapat dilihat pada Gambar 4.17. Pada variasi kecepatan 20 m/s AODV-PNT dan AODV-MNS memiliki PDR lebih kecil dibandingkan dengan AODV original, pada variasi kecepatan 25 m/s rata-rata PDR dari semua protokol hampir sama sekitar 75%.



Gambar 4.17 Grafik PDR dengan modifikasi kecepatan pada skenario *real*

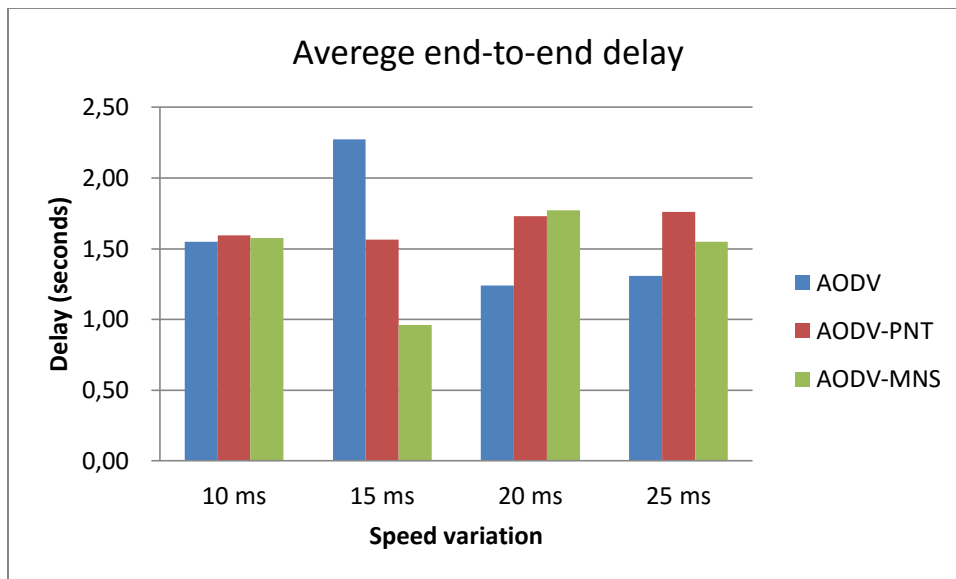
Tabel 4.23 Hasil rata-rata PDR (%) uji coba modifikasi kecepatan skenario *real*

Variasi Kecepatan (m/s)	Jumlah Rute AODV			PDR AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	10	8	8	72,85	78,38	81,02	2,65
15	12	9	6	58,20	74,05	80,15	6,10
20	10	8	8	73,66	69,75	72,16	2,40
25	12	7	8	75,21	75,65	75,41	-0,25

#### 4.4.5.2 Average end-to-end delay

Berdasarkan Gambar 4.18 delay yang dihasilkan oleh routing protokol AODV-MNS lebih kecil dibandingkan AODV original pada variasi kecepatan 10, 15, dan 20 m/s. Sedangkan AODV-PNT dibandingkan dengan AODV

unggul ketika variasi kecepatan 15 dan 20 ms, selain itu delay yang dihasilkan AODV-PNT lebih besar dibandingkan dengan AODV original. Dari 4 variasi kecepatan delay yang dihasilkan AODV-MNS hampir semua variasi kecepatan delay yang dihasilkan lebih kecil dibandingkan AODV-PNT, pada kecepatan 20 m/s delay yang dihasilkan AODV-MNS sedikit lebih besar dibandingkan dengan AODV-PNT. Penurunan nilai rata-rata *delay* AODV-MNS dibandingkan dengan AODV original pada variasi kecepatan 10, 15 dan 20m/s adalah 0,52 detik. Sedangkan selisih delay yang dihasilkan AODV-MNS dibandingkan dengan AODV-PNT adalah 1,2971 detik. Berdasarkan **Error! eference source not found.** pada variasi kecepatan 25 m/s delay yang dihasilkan AODV-PNT dan AODV-MNS dibandingkan AODV original lebih tinggi, tingginya delay yang dihasilkan dari kedua routing protokol tersebut mengakibatkan penurunan pada PDR.



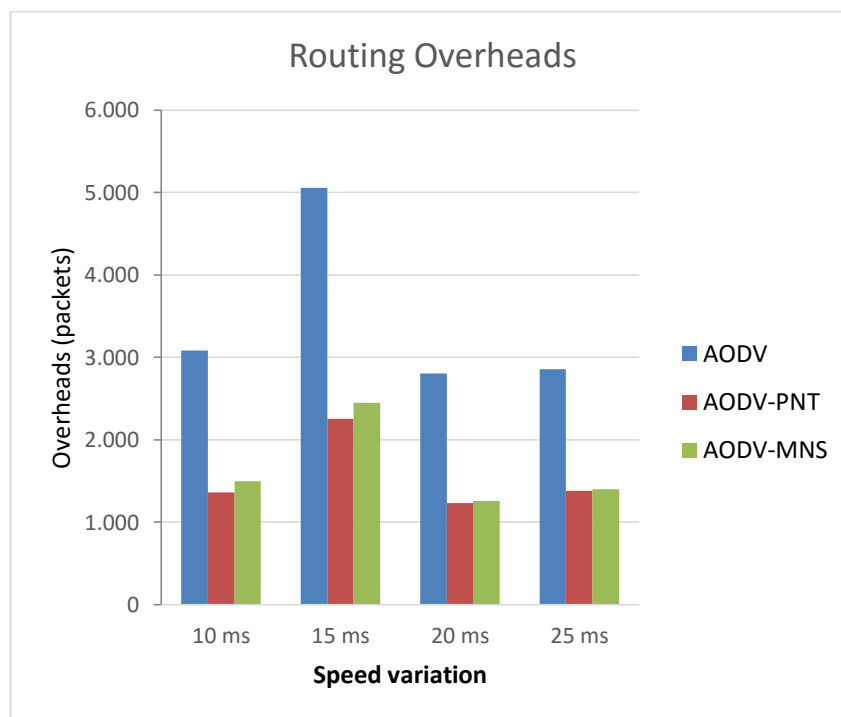
Gambar 4.18 Grafik *end-to-end delay* dengan modifikasi kecepatan pada skenario *grid*

Tabel 4.24 Rata-rata *delay (seconds)* uji coba modifikasi kecepatan skenario *real*

Variasi Kecepatan (m/s)	AODV	AODV-PNT	AODV-MNS	Selisih MNS dan PNT
10	72,85	78,38	81,02	2,65
15	58,20	74,05	80,15	6,10
20	73,66	69,75	72,16	2,40
25	75,21	75,65	75,41	-0,25

### 4.4.5.3 Routing overheads

Nilai Nilai rata-rata RO yang dihasilkan AODV-PNT dan AODV-MNS tidak terpengaruh oleh variasi kecepatan, berdasarkan Gambar 4.19 nilai rata-rata RO yang dihasilkan oleh protokol AODV-PNT dan AODV-MNS lebih kecil dibandingkan dengan protokol AODV original. RO protokol AODV yang dihasilkan paling tinggi ketika variasi kecepatan 15 m/s.



Gambar 4.19 Grafik *routing overheads* skenario *real*

Berdasarkan Tabel 4.25 variasi kecepatan nilai rata-rata RO yang dihasilkan protokol AODV-PNT dan AODV-MNS lebih kecil dibandingkan RO AODV original dengan rata-rata selisih 1440, sedangkan RO yang dihasilkan AODV-PNT lebih kecil dibandingkan dengan AODV-MNS pada variasi kecepatan 10, 15 dan 20 m/s, selisih RO yang dihasilkan AODV-PNT 93,12 paket lebih sedikit. Ketika variasi kecepatan 25 m/s RO yang dihasilkan AODV-PNT lebih besar dibandingkan dengan AODV-MNS dengan selisih rata-rata sekitar 16 paket data, dengan selisih tersebut tidak memberikan dampak apapun terhadap PDR berbeda pada saat uji coba skenario *grid*, t dengan selisih 34 data dapat memberikan dampak pada nilai PDR. Berdasarkan Tabel 4.25 perbandingan durasi bertahannya rute, protokol AODV-MNS

memiliki rata-rata durasi paling lama, disusul AODV-PNT, dan terakhir AODV. Dari perbandingan jumlah pergantian rute (Tabel 4.23) dan durasi rute, dapat disimpulkan bahwa rute yang dipilih oleh metode yang diusulkan lebih stabil daripada AODV-PNT dan AODV karena memiliki durasi rute yang lama.

Tabel 4.25 Rata-rata RO (*packets*) skenario *real* dengan variasi kecepatan

Variasi Kecepatan (m/s)	Path Duration AODV			RO AODV			Selisih MNS - PNT
	Original	PNT	MNS	Original	PNT	MNS	
10	19,07	24,09	24,76	3086,26	1359,26	1495,81	136,55
15	14,20	20,96	31,22	5061,00	2253,70	2448,40	194,70
20	19,23	21,94	22,54	2803,97	1235,48	1260,16	24,68
25	17,03	26,12	23,35	2857,23	1382,90	1399,39	16,48

Berdasarkan data diatas dengan perhitungan TWR yang dilakukan AODV-PNT dan AODV-MNS dapat menurunkan RO pada jaringan, tetapi penurunan RO yang dihasilkan tidak selalu dapat meningkatkan PDR terbukti pada variasi kecepatan 25 m/s PDR AODV-PNT dan AODV-MNS lebih kecil dibandingkan PDR AODV original. Dari hasil uji coba *skenario grid* dan *skenario real* dapat disimpulkan kecepatan dapat mempengaruhi performa pada AODV-PNT dan AODV-MNS, khususnya jika terjadi penambahan kecepatan diatas 20 m/s dari dua skenario uji coba dilakukan AODV original lebih unggul tetapi hipotesis masih perlu pembuktian lebih lanjut dengan uji coba variasi kecepatan dengan uji coba variasi kecepatan diatas 20 m/s.

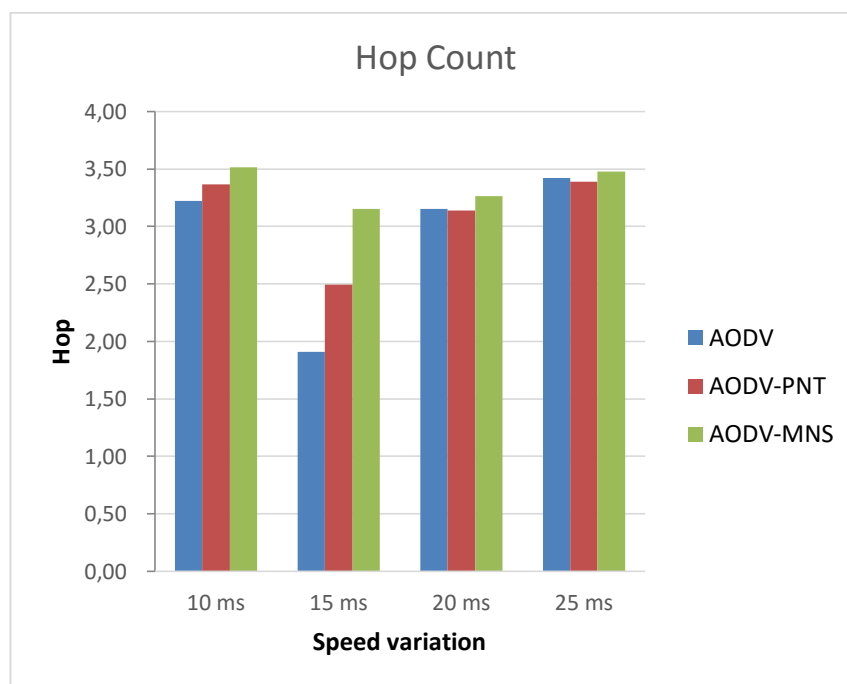
#### 4.4.5.4 Hop count

Tabel 4.26 Rata-rata hop skenario *real* dengan variasi kecepatan

Variasi Kecepatan (m/s)	AODV	AODV-PNT	AODV-MNS	Selisih MNS dan PNT
10	3,22	3,37	3,51	0,15
15	1,91	2,49	3,15	0,66
20	3,15	3,14	3,26	0,12
25	3,42	3,39	3,48	0,09

Jumlah rata-rata *hop* yang dihasilkan oleh *routing* protokol AODV-MNS lebih besar dibandingkan dengan rata-rata *hop* yang dihasilkan oleh *routing* protokol AODV-PNT, berdasarkan pada Tabel 4.26 dan Gambar 4.20. AODV-

MNS memiliki rata-rata 3,35 *hop count* pada setiap variasi node, sedangkan AODV-PNT memiliki rata-rata 3,09 *hop count* pada setiap variasi *node*.



Gambar 4.20 Grafik rata-rata *hop count* dengan modifikasi kecepatan pada skenario *real*

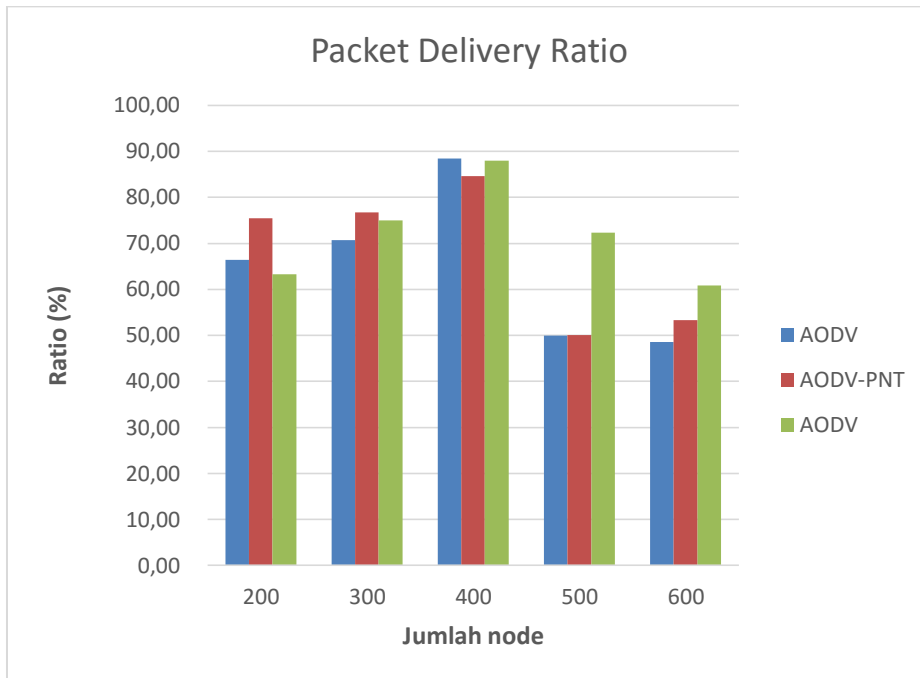
#### 4.4.6 Hasil uji coba densitas tinggi pada skenario *real*

Uji coba densitas tinggi pada skenario *real* dilakukan sebanyak 10 kali dengan skenario mobilitas *random* pada peta *real* dengan luas area 600m x 700m dengan variasi jumlah *node* 200, 300, 400, 500 dan 600 untuk kecepatan rata-rata setiap *node* adalah 15 m/s. Uji coba dilakukan untuk mengetahui seberapa baik performa protokol AODV-MNS ketika dijalankan pada *node* dengan densitas tinggi.

##### 4.4.6.1 Packet delivery ratio (PDR)

Berdasarkan Tabel 4.27, dari 5 variasi uji coba nilai PDR AODV-MNS tidak selalu unggul jika dibandingkan AODV-PNT. Pada variasi jumlah *node* 200 PDR AODV-MNS lebih kecil 12,14% dibandingkan dengan AODV-PNT, sedangkan pada jumlah *node* 300 PDR AODV-MNS lebih kecil 1,77% dibandingkan dengan AODV-PNT. Penurunan rata-rata PDR AODV-MNS dapat disebabkan oleh banyaknya *node* yang dipilih untuk *relay node* sehingga banyak paket RREQ yang disebar pada jaringan, banyaknya paket RREQ yang

disebar dapat meningkatkan kemungkinan *path* yang terbentuk tetapi banyaknya *path* yang terbentuk tidak menjamin PDR tinggi dapat dilihat pada tabel Tabel 4.6 dan Tabel 4.7. Pada jumlah node 400, 500 dan 600 PDR AODV-MNS memiliki PDR 3,7% hingga 22,8% lebih baik dibandingkan AODV-PNT.



Gambar 4.21 Grafik PDR dengan modifikasi jumlah *node* pada skenario *real*

Tabel 4.27 Rata-rata PDR (%) uji coba densitas tinggi skenario *real*

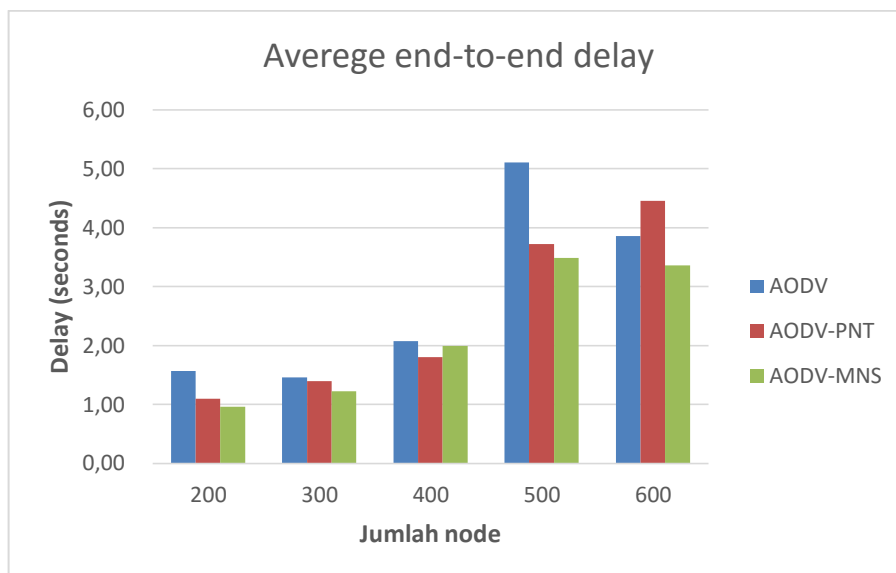
Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV MNS dan PNT
200	66,45	75,45	63,32	-12,14
300	70,68	76,77	75,00	-1,77
400	88,48	84,58	88,05	3,47
500	50,00	50,13	72,31	22,18
600	48,55	53,33	60,85	7,52

Pada Gambar 4.21 menunjukkan rata-rata PDR AODV, AODV-PNT dan AODV-MNS paling tinggi ketika jumlah node 400, pada jumlah node 500 dan 600 rata-rata jumlah PDR yang dihasilkan masing-masing setiap node menurun, tidak seperti hasil uji coba kepadatan tinggi pada skenario grid dimana PDR masing-masing node paling tinggi ketika jumlah node 200.



Berdasarkan hasil uji coba densitas tinggi menggunakan skenario *grid* dan skenario *real* setiap protokol memiliki performa puncak.

#### 4.4.6.2 Average end-to-end delay



Gambar 4.22 Grafik end-to-end delay dengan modifikasi jumlah *node* pada skenario *real*

Tabel 4.28 Rata-rata delay (seconds) uji coba densitas tinggi skenario *real*

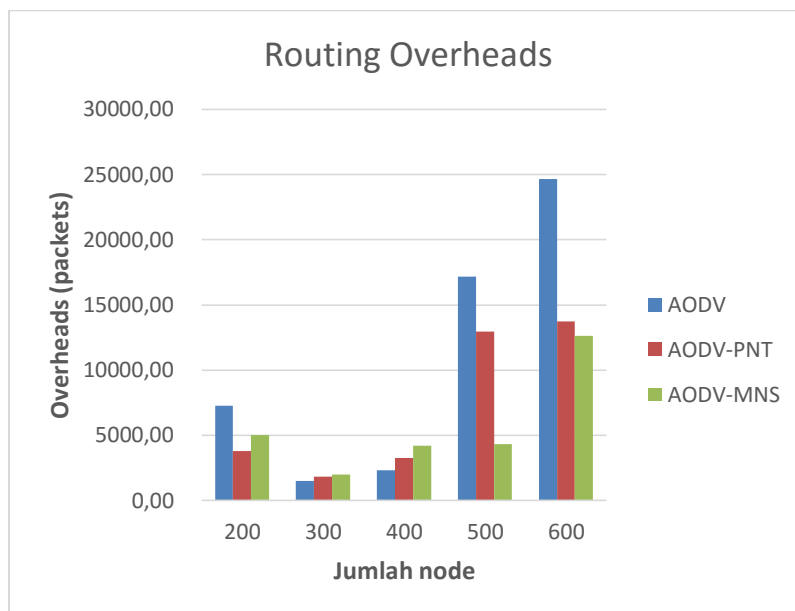
Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV PNT dan MNS
200	1,57	1,10	0,97	0,13
300	1,46	1,40	1,22	0,18
400	2,07	1,81	1,99	-0,19
500	5,11	3,73	3,48	0,24
600	3,86	4,46	3,36	1,09

Berdasarkan Gambar 4.21 delay yang dihasilkan oleh routing protokol AODV-MNS kecenderungan meningkat, delay yang paling tinggi ketika jumlah node 500, sedangkan paling rendah adalah ketika jumlah node 200. Berdasarkan grafik tersebut, peningkatan delay yang dihasilkan sedikit mempengaruhi performa AODV-MNS terbukti pada rata-rata PDR yang dihasilkan pada jumlah node 500 dan 600 memiliki kecenderungan penurunan. Dari uji coba yang dilakukan delay yang dihasilkan AODV-MNS lebih kecil dibandingkan protokol AODV original, ketika jumlah node 200, 300, 500 dan 600 delay yang dihasilkan AODV-MNS lebih kecil dibandingkan dengan

AODV-PNT. Berdasarkan Tabel 4.28, pada jumlah node 400 delay yang dihasilkan AODV-MNS lebih besar dibandingkan dengan AODV-PNT, peningkatan delay dapat disebabkan oleh meningkatnya *delay* dapat disebabkan oleh lamanya proses pemilihan relay *node*, peningkatan *delay* juga dapat disebabkan karena perhitungan TWR dan *future* TWR.

#### 4.4.6.3 Routing overheads

RO AODV-MNS pada jumlah node 200, 500 dan 600 lebih kecil dibandingkan dengan AODV original, pada jumlah node 300 dan 400 jumlah RO AODV-MNS lebih besar dibandingkan AODV original. RO AODV-MNS lebih kecil karena AODV-MNS hanya memilih next-hop node yang stabil dan melakukan multicast RREQ terhadap kumpulan next-hop node (relay node set) sedangkan AODV melakukan broadcast RREQ sehingga lebih banyak jumlah paket routing yang beredar di jaringan.



Gambar 4.23 Grafik *routing overheads* modifikasi jumlah node pada skenario grid

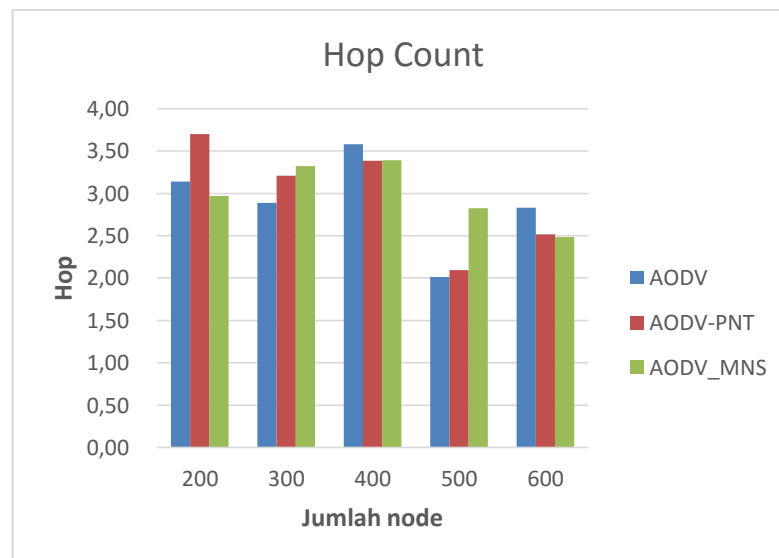
Nilai rata-rata RO yang dihasilkan protokol AODV-MNS pada jumlah node 200, 300 dan 400 lebih besar dibandingkan dengan AODV-PNT, pada jumlah node 500 dan 600 jumlah RO yang dihasilkan AODV-MNS lebih kecil dibandingkan dengan AODV-PNT. Peningkatan RO AODV-MNS pada jumlah *node* 200, 300 dan 400 dapat disebabkan karena pemilihan node. Pada keadaan

lalulintas yang padat *node* hampir tidak bergerak sehingga *node* dianggap sebagai *node* yang stabil karena nilai percepatan dan kecepatan kecil tetapi belum tentu *node* yang dijadikan sebagai *relay node* memiliki *node* tetangga yang stabil. Pada jumlah *node* 500 dan 600 perhitungan TWR yang dilakukan AODV-MNS dapat menurunkan RO pada jaringan, tetapi penurunan RO tidak selalu dapat meningkatkan PDR terbukti pada variasi jumlah *node* 500 dan 600 PDR AODV-PNT memiliki kecenderungan turun.

Tabel 4. 29 Rata-rata RO (*packets*) uji coba densitas tinggi skenario *real*

Jumlah Node	AODV	AODV-PNT	AODV-MNS	Selisih AODV PNT dan MNS
200	7271,82	3801,73	5039,64	-1237,91
300	1483,82	1822,45	1987,00	-164,55
400	2305,73	3273,09	4209,36	-936,27
500	17169,09	12964,36	4345,73	8618,64
600	24676,70	13742,44	12644,11	1098,33

#### 4.4.6.4 Hop count



Gambar 4.24 Grafik rata-rata hop count dengan variasi jumlah *node* pada skenario *real*

Berdasarkan grafik pada Gambar 4.12, jumlah *hop* yang dihasilkan *routing* protokol AODV-MNS lebih bervariasi jika dibandingkan dengan uji coba variasi *node* dibawah 100, AODV-MNT akan memilih *relay node* yang stabil berdasarkan perhitungan TWR. Sedangkan pada uji coba dengan variasi 500 dan 600 *node*, *node* hampir tidak dapat bergerak sehingga faktor kecepatan

dan percepatan tidak berpengaruh terhadap perhitungan, sehingga banyak node yang dianggap stabil karena node diam hal tersebut dapat mengakibatkan RO, *delay* dan *hop count* naik. Pada variasi jumlah node 300 dan 500 *hop count* yang dihasilkan AODV-MNS lebih tinggi dibandingkan dengan AODV-PNT maupun AODV original. Dengan penambahan *hop count* yang dihasilkan tidak mempengaruhi nilai PDR, terbukti bahwa nilai PDR protokol AODV-MNS masih lebih unggul dibandingkan dengan protokol pembanding.

Tabel 4.30 Rata-rata hop uji coba densitas tinggi

<b>Jumlah Node</b>	<b>AODV</b>	<b>AODV-PNT</b>	<b>AODV-MNS</b>	<b>Selisih AODV PNT dan MNS</b>
<b>200</b>	3,14	3,70	2,97	0,73
<b>300</b>	2,89	3,21	3,32	-0,11
<b>400</b>	3,58	3,38	3,39	0,00
<b>500</b>	2,01	2,09	2,83	-0,73
<b>600</b>	2,83	2,52	2,48	0,03

## BAB 5

### Kesimpulan dan Saran

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari serangkaian penelitian yang telah dilakukan dan saran yang dapat diambil untuk pengembangan penelitian

#### 5.1 Kesimpulan

Pengujian dan analisis yang telah dilakukan menghasilkan beberapa kesimpulan penelitian, yaitu:

- 1 Berdasarkan ujicoba yang dilakukan pada skenario *grid* dan real performa AODV-MNS mengalami peningkatan PDR serta penurunan pada RO dan *delay* dibandingkan dengan AODV original.
- 2 Pada skenario *grid* dan real PDR AODV-MNS sedikit lebih unggul dibandingkan dengan AODV-PNT, tetapi RO yang dihasilkan oleh AODV-MNS lebih tinggi dibandingkan dengan AODV-PNT .
- 3 *Packet delivery ratio* (PDR), PDR pada skenario *grid* AODV-MNS meningkat antara 1,20%-20% dibandingkan dengan AODV-PNT sedangkan pada skenario *real* meningkat antara 0,3%-18%. Jika dibandingkan dengan AODV original, PDR AODV-MNS meningkat antara 17%-24% pada skenario *grid*, 8,6%-58,2% pada skenario *real*.
- 4 *Average end-tp-end delay*, *delay* AODV-MNS dibandingkan dengan *delay* AODV cenderung menurun seiring penambahan jumlah *node* sedangkan AODV cenderung meningkat seiring bertambahnya jumlah *node*, kecuali pada jumlah node 10 *delay* yang dihasilkan kurang lebih hampir sama dengan *delay* AODV. AODV-MNS dibandingkan dengan AODV-PNT, pada skenario *grid* *delay* AODV-MNS lebih kecil 2,41%-41% sedangkan pada skenario *real* *delay* AODV-MNS lebih besar sekitar 12,3%-25,7%.
- 5 *Routing overhead* (RO) pada AODV-MNS lebih sedikit dibandingkan dengan AODV original, dikarenakan AODV-MNS hanya mengirim pesan kepada relay node yang telah dipilih sesuai perhitungan TWR node sedangkan AODV melakukan *broadcast* kepada setiap node tetangga. Rata-rata RO yang

dihasilkan oleh AODV-MNS pada skenario *grid* adalah 585,4 dan 1337,08 untuk skenario *real*, sedangkan AODV original 2187,68 untuk skenario *grid* dan 3025,32 skenario *real*. Jumlah RO AODV-MNS pada skenario *grid* lebih besar dibandingkan dengan RO AODV-PNT dengan selisih RO 106,06, tetapi pada skenario *real* RO AODV-MNT lebih kecil dibandingkan dengan AODV-PNT dengan selisih RO 121,96.

- 6 *Hop Count*, rata-rata *hop count* AODV-MNS lebih besar dibandingkan dengan AODV-PNT 0,23 pada skenario *grid* dan 0,19 pada skenario *real*. Jumlah rata-rata hop kon AODV-MNS juga lebih besar dibandingkan dengan AODV original dengan selisih rata-rata *hop count* pada skenario *grid* adalah 0,46 sedangkan pada skenario *real* selisih rata-rata *hop count* adalah 0,66.
- 7 Variasi kecepatan dapat mempengaruhi nilai PDR pada setiap routing protokol, dari uji coba skenario *grid* dan *real* routing protokol AODV-MNS dan AODV-PNT berjalan dengan baik pada variasi kecepatan 10-20 m/s, sedangkan AODV original memiliki rata-rata PDR lebih baik ketika variasi kecepatan 25 m/s.
- 8 Uji coba densitas tinggi dengan jumlah hingga 600 node, menunjukkan PDR yang dihasilkan meningkat secara fluktuatif dengan PDR tertinggi pada 400 *node* dan RO yang dihasilkan semakin tinggi pada jumlah 500 dan 600 *node*. PDR routing protokol AODV-PNT dan AODV-MNS tetap unggul dibandingkan dengan AODV original, RO yang dihasilkan AODV-PNT dan AODV-MNS lebih kecil digandingkan dengan AODV original.

## 5.2 Saran

Saran yang dapat diberikan dari hasil uji coba dan evaluasi adalah sebagai berikut:

1. Studi lebih lanjut mengenai pengaruh dari koefisien *weight* faktor dan *threshold* W yang digunakan pada rumus TWR, dengan penelitian tersebut diharapkan *weight* faktor dapat dibuat adaptif menyesuaikan kondisi lingkungan.

2. Perlu adanya studi lebih lanjut terhadap setiap nilai yang dihasilkan oleh faktor sebelum dikalikan dengan weight faktor, karena nilai yang tidak relevan karena selisih nilai setiap faktor terlalu besar

*[Halaman ini sengaja dikosongkan]*



## DAFTAR PUSTAKA

- Ali, TE, Dulaimi, LAKA & Majeed, YE 20016, 'Review and Performance Comparison of VANET Protocols: AODV, DSR, OLSR, DYMO, DSDV & ZRP', *Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*, *Al-Sadeq International*.
- Anggoro, R, Kitasuka, T, Nakamura, R & Aritsugi, M 2012, 'Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments', *2012 Third International Conference on Networking and Computing*, pp. 259-263.
- Batool, F & Khan, SA 2005, 'Traffic estimation and real time predication using Ad Hoc Network', *Proceedings of IEEE International Conference on Emerging Technologies 2005*, Islamabad, Pakistan.
- Biswas, S, Tatchikou, R & Dion, F 2006, 'Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety', *IEEE Communications Magazine*, pp. 74 - 82.
- C.E. Perkins & E.M. Royer 2002, 'Ad-hoc on-demand distance vector routing', *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*.
- De, S, Caruso, AM, Chaira, T & Chessa, S 2006, 'Bounds on hop distance in greedy routing approach in wireless ad hoc networks', *International Journal of Wireless and Mobile Computing*, vol 1, pp. 131-140.
- Krajzewicz, D, Erdmann, J, Behrisch, M & Laura, B 2012, 'Recent Development and Applications of SUMO – Simulation of Urban MObility', *International Journal on Advances in Systems and Measurements*, vol 5.
- Mostajeran, E, Noor, RM & Keshavarz, H 2013, 'A Novel Improved Neighbor Discovery Method for an Intelligent-AODV in Mobile Ad hoc Networks', *International Conference of Information and Communication Technology (ICoICT)*.

- Raw, RS, Toor, V & Singh, AN 2012, 'Path Duration Analysis In Vehicular Ad-hoc Network', *International Journal on AdHoc Networking Systems (IJANS)*, vol 2.
- Rehman, S-U, Khan, MA, Zia, T & Zheng, L 2013, 'Vehicular ad-Hoc networks (VANETs)—An overview and challenges', *EURASIP Journal on Wireless Communications and Networking* 3, pp. 29-38.
- S. Corson, JM 1992, *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, RFC editor.
- Shen, X, Wu, Y, Xu, Z & Lind, X 2014, 'AODV-PNT: An improved version of AODV routing protocol with predicting node trend in VANET', *Advanced Infocomm Technology (ICAIT), 2014 IEEE 7th International Conference on*.
- Toh, CK 2002, *Ad Hoc Mobile Wireless Networks*, Prentice Hall.
- Wang Kun, YZLSCZMG 2015, 'NPR: A Node Activity-based Probabilistic Routing Algorithm in Delay Tolerant-Mobile Sensor Networks', *Communications (ICC), 2015 IEEE International Conference on*.
- Yu, X, Guo, H & Wong, W-C 2011, 'A Reliable Routing Protocol for VANET Communications', *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*.

## LAMPIRAN

### Lampiran A.1 modifikasi paket RREQ pada aodv\_packet.h

```
struct hdr_aodv_request {
    u_int8_t      rq_type;      // Packet Type
    u_int8_t      reserved[2];
    u_int8_t      rq_hop_count; // Hop Count
    u_int32_t     rq_bcast_id;  // Broadcast ID

    nsaddr_t      rq_dst;       // Destination IP Addr
    u_int32_t     rq_dst_seqno; // Destination Seq Num
    nsaddr_t      rq_src;       // Source IP Address
    u_int32_t     rq_src_seqno; // Source Sequence Num

    double        rq_timestamp; // when REQUEST sent;
    // used to compute route discovery latency
    // AODV-PNT addition
    nsaddr_t      *rq_eligible_nodes = NULL;
    u_int32_t     nodes_list_len;
```

### Lampiran A.2 modifikasi fungsi sendHello pada file aodv.cc

```
1670 void AODV::sendHello() {
1671     Packet *p = Packet::alloc();
1672     struct hdr_cmn *ch = HDR_CMN(p);
1673     struct hdr_ip *ih = HDR_IP(p);
1674     struct hdr_aodv_reply *rh = HDR_AODV_REPLY(p);
1675     // AODV-PNT
1676
1677     MobileNode *iNode;
1678     iNode = (MobileNode *)
1679             (Node::get_node_by_address(index));
1679     double iSpeed = ((MobileNode *) iNode)->speed();
1680     double now = ((MobileNode *) iNode)->getUpdateTime();
1681     rh->rp_x = iNode->X(); // x coordinate
1682     rh->rp_y = iNode->Y(); // y coordinate
1683
1684     if (now - lastUpdateTime == 0) { // if not updated,
1685         just in case
1686         rh->rp_accel = lastAccel;
1687     } else {
1688         // acceleration = delta speed / delta time
1689         rh->rp_accel = (iSpeed - lastSpeed) / (now -
1690             lastUpdateTime);
1691         lastAccel = rh->rp_accel;
1692         lastSpeed = iSpeed;
1693     }
1694     rh->rp_speed = iSpeed; // speed
1695     lastUpdateTime = now; // Update its latest update
1696     time
1697     // Fill other HELLO message attributes then send
```

### Lampiran A.3 penambahan atribut pada file aodv\_rtable.h

```
46 class AODV_Neighbor {
47     friend class AODV;
48     friend class aodv_rt_entry;
49 public:
50     AODV_Neighbor(u_int32_t a) { nb_addr = a; }
51
52 protected:
53     LIST_ENTRY(AODV_Neighbor) nb_link;
54     nsaddr_t nb_addr;
55
56     double nb_expire; // ALLOWED_HELLO_LOSS *
57     HELLO_INTERVAL
58     // AODV-PNT variable to store speed, accel, pos
59     attributes
60     double nb_speed;
61     double nb_accel;
62     double nb_x;
63     double nb_y
64     double nb_destx;
65     double nb_desty;
66 };
```

### Lampiran A.4 penambahan atribut paket HELLO

```
116 struct hdr_aodv_reply {
117     u_int8_t rp_type; // Packet Type
118     u_int8_t reserved[2];
119     u_int8_t rp_hop_count; // Hop Count
120     nsaddr_t rp_dst; // Destination IP Address
121     u_int32_t rp_dst_seqno; // Destination Sequence Number
122     nsaddr_t rp_src; // Source IP Address
123     double rp_lifetime; // Lifetime
124     double rp_timestamp; // when corresponding REQ sent;
125     // used to compute route discovery latency
126
127     // penambahan field untuk AODV-PNT hello message
128     double rp_speed;
129     double rp_accel;
130     double rp_x;
131     double rp_y
132     double rp_destx;
133     double rp_desty;
```

## Lampiran A.5 penambahan atribut pada Class AODV

```
282 //AODV Contractor Attributes
283 nsaddr_t index; // IP Address of this node
284 u_int32_t seqno; // Sequence Number
285 int bid; // Broadcast ID
286 // AODV-PNT
287 double lastUpdateTime; // store last update time
288 double lastAccel; // store last acceleration
289 double lastSpeed; // store last speed
```

## Lampiran A.6 modifikasi fungsi recvHello pada file aodv.cc

```
Void AODV::recvHello(Packet *p) {
//struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
AODV_Neighbor *nb;

nb = nb_lookup(rp->rp_dst);
if(nb == 0) {
nb_insert(rp->rp_dst);
// Get this neighbor once again
nb = nb_lookup(rp->rp_dst);
// Extract speed and accel info from hello message
nb->nb_speed = rp->rp_speed;
nb->nb_accel = rp->rp_accel;
nb->nb_x = rp->rp_x;
nb->nb_y = rp->rp_y;
nb->nb_destx = rp->rp_destx;
nb->nb_desty = rp->rp_desty;
}
else {
// Extract speed and accel info from hello message
nb->nb_speed = rp->rp_speed;
nb->nb_accel = rp->rp_accel;
nb->nb_x = rp->rp_x;
nb->nb_y = rp->rp_y;
nb->nb_destx = rp->rp_destx;
nb->nb_desty = rp->rp_desty;

nb->nb_expire = CURRENT_TIME +
(1.5 * ALLOWED_HELLO_LOSS * HELLO_INTERVAL);
}
Packet::free(p);
}
```

## Lampiran A.7 implementasi simulasi NS-2.35

```
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqLen) 50 ;# max packet in ifq
set val(rp) AODV ;# routing protocol
set val(nn) 10 ;# number of node
set val(cbrsize) 512 ;# 512 Bytes
set val(cbrRate) 2KB
set val(cbrInterval) 1 ;# 1 packet per second
set val(stop) 360
set val(mobilityfile) "mobility1.tcl"
set val(activityfile) "activity.tcl"
.
.
set channel_ [new $val(chan)]
$ns_ node-config \
    -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqLen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel $channel_ \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF \
    -topoInstance $topo
.
.
```

## Lampiran A.8 perhitungan jumlah node tetangga

```
// calculated number neighbour node
int AODV::nb_node() {
    int jumlahNode = 0;
    AODV_Neighbor *cc = nbhead.lh_first;
    for (; cc; cc=cc->nb_link.le_next) {
        jumlahNode++;
    }
    return jumlahNode;
}
```

## Lampiran A.9 perhitungan arah node terhadap node tujuan

```
//Source-Destination direction
double delta_x;
double delta_y;
double tgtheta;
double theta;
double sudut_vertikal;

delta_x = DST_X - SRC_X;
if (delta_x < 0) delta_x*=1;
delta_y = DST_Y - SRC_Y;
if (delta_y < 0) delta_y*=1;

tgtheta = delta_y / delta_x;
theta = atan(tgtheta) * 180 / 3.14159265;
if (delta_x >0 && delta_y > 0){
    sudut_vertikal = 90 - theta; //masukkan
}
.
.
else if (delta_x < 0 && delta_y == 0){
    sudut_vertikal = 270;
}
.
.
//Neighbor node direction
double destX = nb->nb_destx;
double destY = nb->nb_desty;
double nb_delta_x;
double nb_delta_y;
double nb_tgtheta;
double nb_theta;
double nb_sudut_vertikal;
if (destX == 0 && destY == 0)
{
    destX = nb->nb_x;
    destY = nb->nb_y;
}
nb_delta_x = nb->nb_x - destX;
nb_delta_y = nb->nb_y - destY;
if(nb_delta_x !=0)
    nb_tgtheta = nb_delta_y / nb_delta_x;
else nb_tgtheta = 0;
    nb_theta = atan(nb_tgtheta) * 180 / 3.14159265;
if (nb_delta_x > 0 && nb_delta_y > 0){
    nb_sudut_vertikal = 90 - nb_theta;
}
.
.
else if (nb_delta_x < 0 && nb_delta_y == 0){
    nb_sudut_vertikal = 270;
}
}
```

*[Halaman ini sengaja dikosongkan]*



## BIODATA PENULIS



**Grezio Arifiyan Primajaya**, bisa dipanggil Grezio, lahir di kota Malang, 26 Agustus 1992. Penulis adalah anak ke-2 dari 2 bersaudara dibesarkan di kab. Malang, Jawa Timur. Penulis menempuh pendidikan SDN Gedogwetan III (1996 - 2004), SMP 02 Turen (2004 - 2007), SMA Negeri Kepanjen (2007 - 2010). Pada tahun 2010, penulis mengikuti SNMPTN dan diterima strata satu jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya angkatan 2010 yang terdaftar dengan NRP 5110100196. Tahun 2015 penulis menempuh strata dua di Jurusan Teknik Informatika terdaftar dengan NRP 5115201036, penulis mengambil bidang minat Net Centric Computing (NCC). Selama menempuh kuliah, penulis juga aktif sebagai anggota departemen dalam negeri di Himpunan Mahasiswa Teknik Computer (HMTC) C-1A. Penulis juga aktif sebagai administrator lab NCC selama masa perkuliahannya. Pada beberapa acara kampus, penulis juga beberapa kali aktif menjadi panitia, baik sebagai anggota maupun koordinator. Penulis dapat dihubungi melalui alamat e-mail [grez.arifiyan@gmail.com](mailto:grez.arifiyan@gmail.com)

