

**ANALISIS KINERJA PROTOKOL ROUTING AODV DAN ADV
PADA VANET BERBASIS IEEE 802.11b**

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun Oleh :
BAYU SUTAWIJAYA
NIM. 0810960037

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
MALANG
2015**

LEMBAR PERSETUJUAN

ANALISIS KINERJA PROTOKOL *ROUTING* AODV DAN ADV PADA VANET BERBASIS IEEE 802.11b

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer



Disusun Oleh :

BAYU SUTAWIJAYA

NIM. 0810960037

Skripsi ini telah disetujui oleh dosen pembimbing

Pada tanggal 9 Juni 2015 :

Dosen Pembimbing I

Dosen Pembimbing II

Kasyful Amron, ST., M.Sc
NIP. 197508032003121003

Sabriansyah R.A., S.T., M.Eng
NIP. 198208092012121004

LEMBAR PENGESAHAN

ANALISIS KINERJA PROTOKOL ROUTING AODV DAN ADV PADA VANET BERBASIS IEEE 802.11b

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun Oleh :

BAYU SUTAWIJAYA

NIM. 0810960037

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 23 Juni 2015

Penguji I

Ir. Heru Nurwasito., M.Kom
NIP. 196504021990021001

Penguji II

Aswin Suharsono, ST.,MT
NIK. 84091906110251

Penguji III

Himawat Aryadita, ST., M.Sc
NIP. 198010182008011003

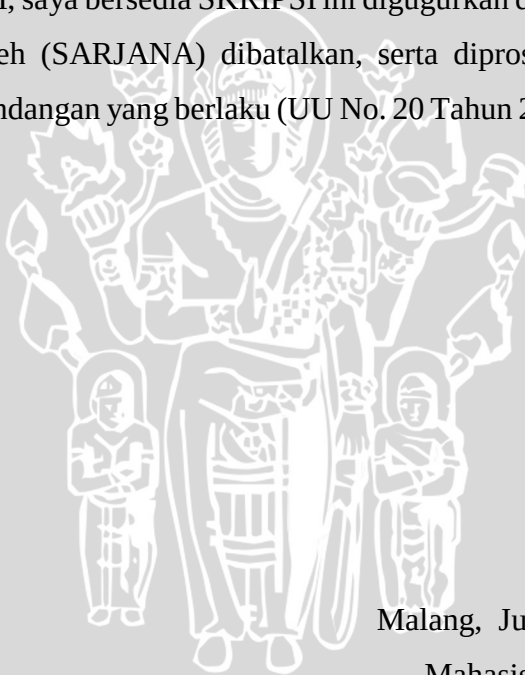
Mengetahui
Ketua Prodi Informatika/Ilmu Komputer

Drs. Marji., M.T
NIP. 196798011992031001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah tulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, Juni 2015

Mahasiswa,

Bayu Sutawijaya

NIM. 0810960037

KATA PENGANTAR

Alhamdulillah rabbil ‘alamin. Puji syukur penulis panjatkan kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul “ANALISA KINERJA PROTOKOL ROUTING AODV DAN ADV PADA VANET BERBASIS IEEE 802.11b”.

Tugas akhir ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya Malang.

Dalam pelaksanaan dan penyusunan tugas akhir ini sudah cukup banyak bantuan yang diberikan berbagai pihak, baik berupa bimbingan dan saran hingga penulisan tugas akhir ini dapat terselesaikan. Oleh karena itu, penulis dalam kesempatan ini mengucapkan terima kasih kepada :

1. Bapak Kasyful Amron, ST., M.Sc dan Bapak Sabriansyah Rizqika Akbar, ST., M.Eng selaku dosen pembimbing 1 dan dosen pembimbing 2 yang telah memberikan pengarahan dan saran dalam penyusunan tugas akhir ini.
2. Keluarga penulis, Bapak Prof. Dr. Abdul Latief Abadi, MS, Ibu Rr. Dewi Anggrahini (Almh), Kakak Bram Hadiwijaya, S.P, M.P, M.S.c dan Kakak Putri Harziani S.E, Ak yang tidak pernah berhenti memberikan dukungan, doa dan kasih sayang kepada penulis..
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku ketua dan sekretasi Program Studi Teknik Informatika Universitas Brawijaya.
5. Bapak Bondan Sapta Prakoso, S.T selaku dosen Penasehat Akademik.
6. Ibu Dany Primanita Kartikasari, S.T yang telah memberikan banyak masukan dan nasehat dalam penyusunan tugas akhir ini.

7. Seluruh Dosen Fakultas Matematika dan Ilmu Pengetahuan Alam dan Program Teknologi Informasi dan Ilmu Komputer yang telah mengajarkan ilmunya kepada penulis.
8. Seluruh staf dan karyawan Program Teknologi Informasi dan Ilmu Komputer yang telah banyak membantu dalam penyusunan skripsi ini.
9. Dr. Agung Pramana Warih Marhendra, M.Si yang telah banyak memberikan masukan dan nasehat kepada penulis.
10. Noer Muhammad Dliyaul Haq, Dwi Prasetyo, Dito Ferrandy Raditya, Cindra Pratama, Zulham Yahya, Echa, Abdul Latief Lintar, Ratna Putri Prati Sasya, Azmi Pratama dan Fahriadi yang telah banyak memberikan bantuan dan pengalaman kepada penulis.
11. Teman-teman warung kopi plonyoter yang telah memberikan pengalaman kepada penulis.
12. Semua teman-teman Ilkom 2008 dan LOF SOBAT MIPA yang senantiasa memberikan semangat dan doa demi terselesaikannya tugas akhir ini.
13. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan selanjutnya. Semoga tugas akhir ini dapat bermanfaat bagi semua pihak, baik penulis maupun pembaca

Malang, Juni 2015

Penulis

ABSTRAK

Bayu Sutawijaya. 2015. Analisis Kinerja Protokol *Routing* AODV dan ADV Pada VANET Berbasis IEEE 802.11b. Skripsi Program Studi Teknik Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Pembimbing : Kasyful Amron, S.T., M.Sc. dan Sabriansyah Rizqika Akbar, S.T., M.Eng.

Jumlah kendaraan yang semakin meningkat memiliki dampak terhadap kemacetan dan tingginya angka kecelakaan lalu lintas. Salah satu solusi dari dampak tersebut adalah dengan cara mengembangkan teknologi dalam *Intelligent Transport System* (ITS) yang memanfaatkan kendaraan yang terintegrasi dengan jaringan yang kemudian disebut dengan *node*. *Node* tersebut nantinya dapat saling berkomunikasi dengan *node* lainnya di dalam jaringan komunikasi. Teknologi ini dikenal dengan *Vehicular Ad hoc Networks* (VANET). Dalam VANET, protokol *routing* merupakan hal yang sangat berpengaruh terhadap kinerja jaringan tersebut. Salah satu protokol *routing* yang dapat digunakan pada jaringan VANET adalah *Ad hoc On-Demand Distance Vector* (AODV) dan *Adaptive Distance Vector* (ADV). Penelitian ini dilakukan sebuah simulasi untuk menganalisis kinerja dari VANET dengan menggunakan program simulator *National Chiao Tung University Network Simulator* (NCTUns). Skenario simulasi yang digunakan adalah dengan membuat sebuah model jaringan VANET berbasis 802.11.b pada lingkungan *urban* dan *highway*. Pertambahan jumlah *node* dan variasi kecepatan *node* menjadi variabel peubah pada simulasi ini. Pada hasil simulasi dapat diketahui bahwa protokol *routing* AODV menunjukkan kinerja yang lebih baik dibandingkan dengan protokol *routing* ADV yang diukur dengan parameter *delay*, *throughput*, *packet loss* dan *Packet delivery ratio* (PDR).

Kata Kunci : VANET, AODV, ADV

ABSTRACT

Bayu Sutawijaya. 2015. Analysis of AODV and ADV Routing Protocol in VANET Based on IEEE 802.11b. Minor Thesis Program Study of Informatics Technology/Computer Science, Information Technology and Computer Science Program, Brawijaya University. Supervisor : Kasyful Amron, S.T., M.Sc. dan Sabriansyah Rizqika Akbar, S.T., M.Eng.

The increasing number of vehicles in the road has a significant impact on traffic congestion and the high number of traffic accidents. One of many solutions for these impact is to develop an Intelligent Transport System (ITS) technology which utilizes vehicles that have been integrated with a network which then called node. These nodes then will be able to communicate with other nodes in a communication network. This technology is known as Vehicular Ad Hoc Networks (VANET). In VANET, routing protocol is a very influential aspect to the performance of the network itself. Some of the routing protocols that can be used in VANET network are Ad hoc On-Demand Distance Vector (AODV) and Adaptive Distance Vector (ADV). This research was done with a simulation that analyzed the performance of VANET using the National Chiao Tung University Network Simulator (NCTUns) simulation program. Simulation scenario used in this research is by building a 802.11.b based VANET network model in urban and highway environment with increasing number of nodes and variation of node speeds as research variables. The simulation results showed that AODV routing protocol showed a better performance in comparison with ADV routing protocol measured by their parameter delay, throughput, packet loss, and Packet Delivery Ratio (PDR).

Keywords : VANET, AODV, ADV

DAFTAR ISI

KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiv
BAB I	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
BAB II	5
2.1 <i>Vehicular Ad hoc Network (VANET)</i>	5
2.1.1 Arsitektur VANET	5
2.1.2 Fitur Pada VANET	6
2.1.3 <i>Inter Vehicle Communication (IVC)</i>	6
2.2 <i>Routing dan Protokol Routing VANET</i>	7
2.2.1 Protokol <i>Routing</i> Proaktif	8
2.2.2 Protokol <i>Routing</i> Reaktif	8
2.2.3 Protokol <i>Routing Hybrid</i>	8
2.3 <i>Ad hoc On-demand Distance Vector (AODV)</i>	8

2.3.1	Tabel <i>Routing</i> AODV	9
2.3.2	<i>Route Discovery</i> AODV	9
2.3.3	Perbaikan <i>Routing</i> AODV	11
2.4	<i>Adaptive Distance Vector</i> (ADV)	12
2.4.1	Variasi Jumlah Pemeliharaan Rute yang aktif	12
2.4.2	Variasi Frekuensi dari <i>Routing Updates</i>	13
2.5	Pengukuran Kinerja	17
2.5.1	<i>End To End Delay</i>	17
2.5.2	<i>Throughput</i>	18
2.5.3	<i>Packet loss</i>	18
2.5.4	<i>Packet Delivery Ratio</i> (PDR)	18
2.6	<i>National Chiao Tung University Network Simulator</i> (NCTUns)	18
2.6.1	Komponen NCTUNS	19
BAB III		22
3.1	Metode Penelitian	22
3.1.1	Studi Literatur	23
3.1.2	Perancangan Sistem	23
3.1.3	Implementasi	24
3.1.4	Pengujian dan Analisa	24
3.1.5	Pengambilan Kesimpulan	25
3.2	Perancangan	26
3.2.1	Perancangan Lingkungan VANET	26
3.2.2	Perancangan Simulasi	27
3.2.3	Perancangan <i>Mobile Node</i>	28
BAB IV		32
4.1	Instalasi NCTUns 6.0	32

4.2	Pengaturan Lingkungan Jaringan VANET	33
4.2.1	Pengaturan Lingkungan <i>Urban</i>	34
4.2.2	Pengaturan Lingkungan <i>Highway</i>	35
4.3	Pengaturan <i>Mobile node</i>	36
4.4	UDP	38
4.5	Pengaturan <i>Protokol Stack</i>	41
BAB V		44
5.1	Implementasi VANET	44
5.2	Pengambilan Data	44
5.3	Hasil Pengolahan Data	47
5.3.1	<i>Delay</i>	47
5.3.2	<i>Throughput</i>	49
5.3.3	<i>Packet Loss</i>	66
5.3.4	<i>Packet Delivery Ratio</i>	68
5.3.4	Uji Statistika	71
5.4	Analisis VANET	72
BAB VI		78
6.1	Kesimpulan	78
6.2	Saran	78
DAFTAR PUSTAKA		79
LAMPIRAN		81

DAFTAR GAMBAR

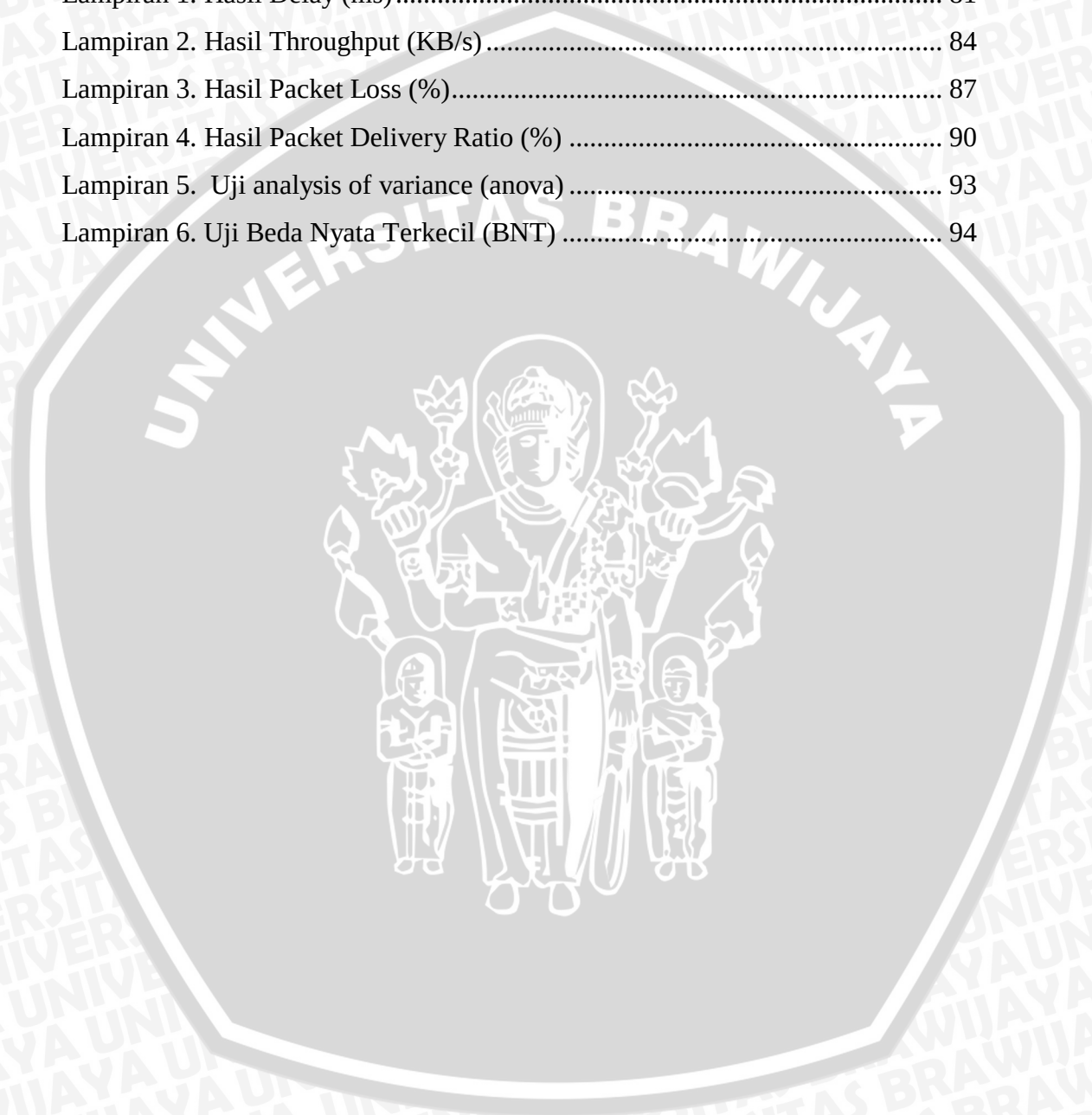
Gambar 2. 1Arsitektur VANET	6
Gambar 2. 2 Komunikasi Vehicle-to-Vehicle (V2V)	7
Gambar 2. 3 Protokol routing ad-hoc.....	7
Gambar 2. 4 paket RREQ.....	10
Gambar 2. 5 paket RREP	11
Gambar 2. 6 paket RERR.....	12
Gambar 2. 7 Entri Routing Update pada ADV	16
Gambar 3. 1 diagram alir metode penelitian.....	22
Gambar 3. 2 diagram alir perancangan sistem	23
Gambar 3. 3 diagram alir pengujian dan analisa sistem.....	24
Gambar 3. 4 lingkungan urban.....	26
Gambar 3. 5 lingkungan highway	27
Gambar 3. 6 diagram alir perancangan <i>mobile node</i>	28
Gambar 3. 7 lingkungan urban 30 node	29
Gambar 3. 8 Lingkungan urban 40 node.....	29
Gambar 3. 9 lingkungan urban 50 node	29
Gambar 3. 10 lingkungan highway 10 node	30
Gambar 3. 11 lingkungan highway 2 0node	30
Gambar 3. 12 lingkungan highway 30 node	30
Gambar 4. 1 pilihan <i>booting</i> NCTUNS 6.0 pada fedora 12.....	32
Gambar 4. 2 tampilan NCTUns 6.0	33
Gambar 4. 3 modul pengaturan ITS road pada urban	34
Gambar 4. 4 modul pengaturan lampu lalu lintas	35
Gambar 4. 5 modul perancangan obstacle	35
Gambar 4. 6 modul pengaturan ITS road pada highway	36
Gambar 4. 7 modul jumlah dan jenis mobile node	37
Gambar 4. 8 modul kecepatan mobile node.....	37
Gambar 4. 9 modul pergerakan node	38

Gambar 4. 10 format pengiriman paket UDP	39
Gambar 4. 11 pengaturan trafik UDP pada source node.....	40
Gambar 4. 12 pengaturan trafik UDP pada node penerima	41
Gambar 4. 13 protocol stack AODV dan ADV	42
Gambar 5.1 file log delay.....	45
Gambar 5. 2file log throughput.....	46
Gambar 5. 3 file log packet loss.....	47
Gambar 5. 4 delay rata-rata urban 8 m/s	48
Gambar 5. 5 delay rata-rata urban 18 m/s	48
Gambar 5. 6 delay rata-rata highway	49
Gambar 5. 7 <i>throughput</i> rata-rata urban 8m/s	50
Gambar 5. 8 throughput urban 30 node 8 m/s.....	50
Gambar 5. 9 <i>throughput</i> urban tertinggi ADV 30 node 8 m/s.....	51
Gambar 5. 10 throughput urban tertinggi AODV 30 node 8 m/s.....	51
Gambar 5. 11 throughput urban 40 node 8 m/s.....	52
Gambar 5. 12 throughput urban tertinggi ADV 40 node 8 m/s	52
Gambar 5. 13 throughput urban tertinggi AODV 40 node 8 m/s.....	53
Gambar 5. 14 <i>throughput</i> urban 50 node 8 m/s	54
Gambar 5. 15 throughput urban tertinggi ADV 50 node 8 m/s	54
Gambar 5. 16 throughput urban tertinggi AODV 50 node 8 m/s.....	55
Gambar 5. 17 throughput rata-rata urban 18m/s	56
Gambar 5. 18 throughput urban 30 node 18 m/s.....	56
Gambar 5. 19 throughput tertinggi ADV 30 node 18 m/s.....	57
Gambar 5. 20 throughput urban tertinggi AODV 30 node 18 m/s.....	57
Gambar 5. 21 <i>throughput</i> urban 40 node 18 m/s.....	58
Gambar 5. 22 <i>throughput</i> urban tertinggi ADV 40 node 18 m/s.....	58
Gambar 5. 23 throughput urban tertinggi AODV 40 node 18 m/s.....	59
Gambar 5. 24 throughput urban 50 node 18 m/s.....	59
Gambar 5. 25 <i>throughput</i> urban tertinggi ADV 50 node 18 m/s.....	60
Gambar 5. 26 <i>throughput</i> urban tertinggi AODV 50 node 18 m/s.....	60
Gambar 5. 27 <i>throughput</i> rata-rata highway	61
Gambar 5. 28 throughput highway 10 node.....	62

Gambar 5. 29 throughput highway tertinggi ADV 10 node.....	62
Gambar 5. 30 throughput highway tertinggi AODV 10 node.....	62
Gambar 5. 31 throughput highway 20 node.....	63
Gambar 5. 32 throughput highway tertinggi ADV 20 node.....	63
Gambar 5. 33 <i>throughput highway</i> tertinggi AODV 20 <i>node</i>	64
Gambar 5. 34 throughput highway 30 node.....	64
Gambar 5. 35t hroughput highway tertinggi ADV 30 node.....	65
Gambar 5. 36 throughput highway tertinggi AODV 30 node.....	65
Gambar 5. 37 packet loss urban 8 m/s	66
Gambar 5. 38 packet loss urban 18 m/s	67
Gambar 5. 39 packet loss highway	68
Gambar 5. 40 packet delivery ratio urban 8 m/s	69
Gambar 5. 41 packet delivery ratiourban 18 m/s	70
Gambar 5. 42 packet delivery ratiohighway	71
Gambar 5. 43 Grafik <i>delay, throughput, packet loss</i> dan PDR AODV pada urban	73
Gambar 5. 44 Grafik <i>delay, throughput, packet loss</i> dan PDR ADV pada urban	73
Gambar 5. 45 Grafik <i>delay, throughput, packet loss</i> dan PDR AODV pada <i>highway</i>	74
Gambar 5. 46 Grafik <i>delay, throughput, packet loss</i> dan PDR ADV pada highway	74
Gambar 5. 47 Perbandingan kinerja AODV dan ADV yang paling baik pada lingkungan urban.....	76
Gambar 5. 48 Perbandingan kinerja AODV dan ADV yang paling baik pada lingkungan <i>highway</i>	77

DAFTAR LAMPIRAN

Lampiran 1. Hasil Delay (ms)	81
Lampiran 2. Hasil Throughput (KB/s)	84
Lampiran 3. Hasil Packet Loss (%).....	87
Lampiran 4. Hasil Packet Delivery Ratio (%)	90
Lampiran 5. Uji analysis of variance (anova)	93
Lampiran 6. Uji Beda Nyata Terkecil (BNT)	94



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini perkembangan dunia jaringan semakin berkembang sangat cepat termasuk teknologi *wireless*. Permintaan komunikasi menggunakan *wireless* mengakibatkan banyaknya penelitian untuk mengembangkan perangkat *wireless* yang baru termasuk dalam *Intelligent Transportation System (ITS)*.

Teknologi jaringan *ad hoc* banyak dikembangkan di dalam *ITS* untuk mengatasi permasalahan-permasalahan yang sering terjadi dalam lalu lintas. Salah satunya adalah kecelakaan dan kemacetan di jalan raya. Salah satu solusi permasalahan tersebut adalah memanfaatkan kendaraan yang terintegrasi dengan jaringan yang kemudian disebut dengan *node*. *Node* tersebut nantinya dapat berkomunikasi dengan *node* lainnya atau dengan infrastruktur yang ada. Setiap *node* akan saling bertukar informasi situasi lalu lintas dengan *node* lainnya di dalam jaringan komunikasi sehingga informasi situasi lalu lintas dapat tersebar secara menyeluruh. Teknologi ini dikenal dengan *Vehicular ad Hoc Networks (VANET)*.

Vehicular Ad hoc Network (VANET) adalah sebuah teknologi yang memadukan kemampuan komunikasi *wireless* kendaraan menjadi sebuah jaringan. VANET ditandai dengan mobilitas tinggi dan konektivitas yang rendah (Mogi & Wibisono, 2012). Dengan tingginya mobilitas pada *node* maka proses *routing* dituntut sangat cepat sehingga proses pengiriman data dari *source node* ke *source destination* dapat terjadi lebih cepat dan efisien.

Routing adalah proses pencarian rute untuk pengiriman data dari *source* ke *destination*. Terdapat 3 kategori dalam protokol *routing* yaitu protokol *routing* reaktif, proaktif dan *hybrid*. Protokol *routing* reaktif bekerja dengan cara membentuk tabel *routing* jika terdapat permintaan pembuatan rute *link* baru atau terdapat perubahan *link*. Protokol *routing* proaktif bekerja dengan cara membentuk tabel *routing* dan *update* setiap waktu jika terjadi perubahan *link*. Sedangkan protokol *routing hybrid* merupakan gabungan antara protokol *routing* reaktif dan proaktif (hadi, et al., 2011).

Salah satu protokol *routing* yang dapat digunakan pada VANET adalah *Ad hoc on Demand Distance Vector* (AODV) dan *Adaptive Distance Vector* (ADV). *Ad hoc on Demand Distance Vector* (AODV) adalah salah satu contoh protokol *routing* reaktif. Ciri utama AODV adalah menjaga *timer-based state* pada setiap *node* sesuai dengan penggunaan tabel *routing*. Tabel *routing* akan kadaluarsa jika jarang digunakan (hadi, et al., 2011). Sedangkan *Adaptive Distance Vector* (ADV) adalah salah satu contoh protokol *routing hybrid*. ADV merupakan protokol *routing distance vector* yang menggabungkan beberapa karakteristik dari *on-demand* dengan memberikan variasi dari frekuensi dan ukuran dari *routing update* dalam merespon beban jaringan dan kondisi mobilitas jaringan (Boppana & Konduru, 2001).

Penelitian ini bertujuan untuk membandingkan kinerja dari dua protokol *routing* AODV dan ADV berdasarkan parameter *delay*, *throughput*, *packet loss*, dan *packet delivery ratio*. Berdasarkan latar belakang tersebut maka judul pada tugas akhir ini yaitu “ANALISA KINERJA PROTOKOL ROUTING AODV DAN ADV PADA VANET BERBASIS IEEE 802.11b”

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijelaskan pada latar belakang, maka rumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana pengaruh kecepatan *node* terhadap kinerja AODV dan ADV pada VANET.
2. Bagaimana pengaruh jumlah *node* terhadap kinerja AODV dan ADV pada VANET.
3. Bagaimana analisa kinerja AODV dan ADV pada VANET berdasarkan parameter *delay*, *throughput*, *packet loss*, dan *packet delivery ratio*.

1.3 Batasan Masalah

Untuk menjaga agar tidak melebar nya masalah dalam penelitian ini, maka penulis membatasi permasalahan yang dibahas. Batasan masalah dalam penelitian ini yaitu:

1. Sistem disimulasikan dengan aplikasi simulator National Chiao Tung University Network Simulator (NCTUNS) 6.0.
2. Jaringan disimulasikan dengan menggunakan standard 802.11b.
3. Komunikasi yang digunakan adalah *vehicle to vehicle*.
4. Penelitian ini mengabaikan aspek keamanan jaringan.

1.4 Tujuan

Tujuan penelitian ini adalah :

1. Mengetahui pengaruh kecepatan *node* terhadap kinerja AODV dan ADV pada VANET.
2. Mengetahui pengaruh jumlah *node* terhadap kinerja AODV dan ADV pada VANET.
3. Mengetahui dan menganalisa kinerja AODV dan ADV pada VANET berdasarkan parameter *delay*, *throughput*, *packet loss*, dan *packet delivery ratio*.

1.5 Manfaat

Penelitian ini diharapkan memberikan kontribusi bagi ilmu pengetahuan teknologi khususnya pada bidang VANET yang akan diterapkan dengan menggunakan protokol *routing* AODV dan ADV.

1.6 Sistematika Penulisan

Sistematika dalam penelitian yang dilakukan terdiri dari beberapa bab yaitu sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan skripsi.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi teori-teori yang mendukung dalam perancangan VANET

BAB III METODE PENELITIAN DAN PERANCANGAN

Bab ini berisi metode penelitian dan perancangan yang akan dilakukan dalam implementasi VANET

BAB IV IMPLEMENTASI

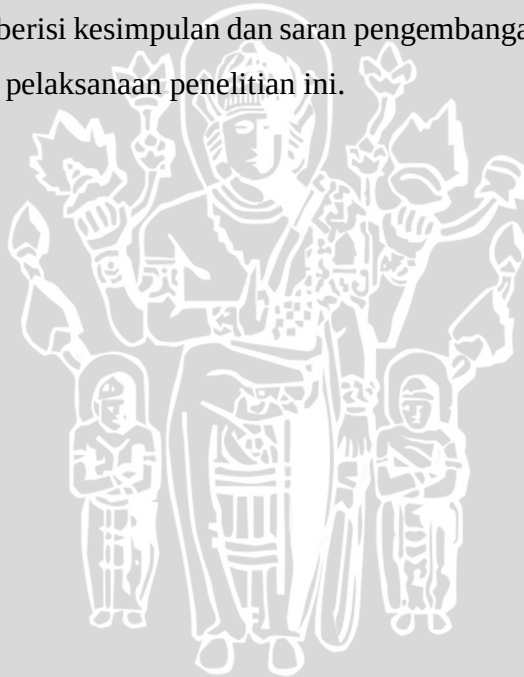
Bab ini berisi proses implementasi dan perancangan yang telah dibuat.

BAB V PENGUJIAN DAN ANALISIS

Bab ini berisi pelaksana pengujian VANET yang telah diimplementasikan dan diperoleh hasil pengujian kemudian dilakukan analisa.

BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran pengembangan dari keseluruhan tahapan pelaksanaan penelitian ini.



BAB II

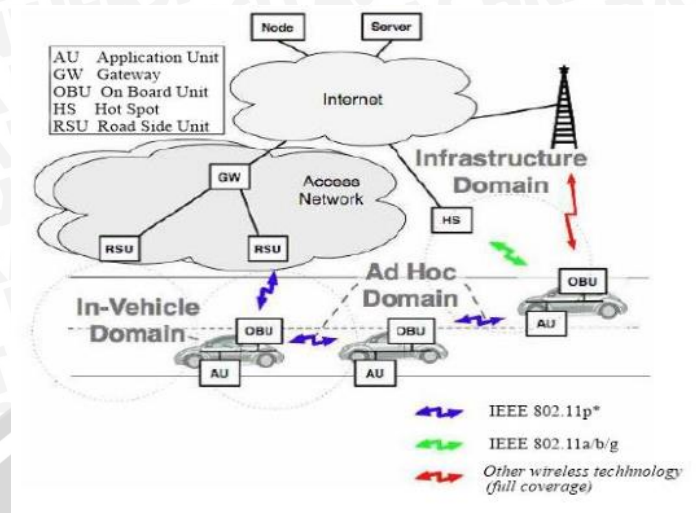
TINJAUAN PUSTAKA

2.1 *Vehicular Ad hoc Network* (VANET)

Vehicular Ad hoc Network (VANET) merupakan turunan dari *Mobile Ad hoc Network* (MANET) yang membangun komunikasi *wireless*, meliputi komunikasi *Inter Vehicle Communication* (IVC), *Vehicle to Roadside* (V2R), atau *Roadside to Roadside* (R2R). Teknologi dalam VANET mengintegrasikan *Wireless Local Area Network* (WLAN), seluler, atau jaringan *ad hoc* untuk mencapai konektivitas berkelanjutan. VANET ini kelak akan sangat berperan pada perkembangan teknologi *Intelligent Transport System* (ITS) dalam menyediakan aplikasi keamanan seperti kemacetan lalu lintas, kontrol kecepatan, kecelakaan sisi jalan, bagian bebas untuk kondisi darurat dan hambatan pada umumnya. Selain aplikasi keamanan, VANET juga menyediakan aplikasi kenyamanan bagi pengguna jalan. Sebagai contoh informasi cuaca, *mobile-commerce*, akses internet, dan aplikasi multimedia lainnya (Faikah, et al.).

2.1.1 *Arsitektur VANET*

Sebuah arsitektur VANET terdiri dari domain-domain berbeda dan komponen-komponen individual. Terdapat tiga domain berbeda yakni kendaraan (*vehicle*), *ad hoc*, dan infrastruktur, termasuk di dalamnya komponen individual (*application unit*, *on board unit*, dan *road side unit*) seperti tertera pada gambar 2.1 (Faikah, et al.).



Gambar 2. 1 Arsitektur VANET

Sumber : (Faikah, et al.)

2.1.2 Fitur Pada VANET

Ketika sistem jaringan pada kendaraan mulai diimplementasikan, banyak isu yang harus diselesaikan, mulai dari pengembangan aplikasi hingga isu ekonomi. Adapun karakteristik unik pada jaringan VANET adalah sebagai berikut:

1. Perubahan topologi jaringan yang cepat dan seringnya fragmentasi, sehingga diameter jaringan yang efektif kecil
2. *Variable*, skala sangat dinamis dan padatnya jaringan
3. Memungkinkan pengendara untuk menyesuaikan perilakunya terhadap data yang diterima di jaringan, sehingga menimbulkan perubahan topologi (Faikah, et al.).

2.1.3 Inter Vehicle Communication (IVC)

Meng-install infrastruktur yang permanen di jalan raya seperti *access points*, *base stations* dapat menelan banyak biaya. Maka komunikasi antar kendaraan (IVC) akan dibutuhkan untuk meningkatkan efektivitas cakupan jaringan kendaraan. Komunikasi IVC murni merupakan jaringan *ad hoc* seperti pada gambar 2.2 (Faikah, et al.).



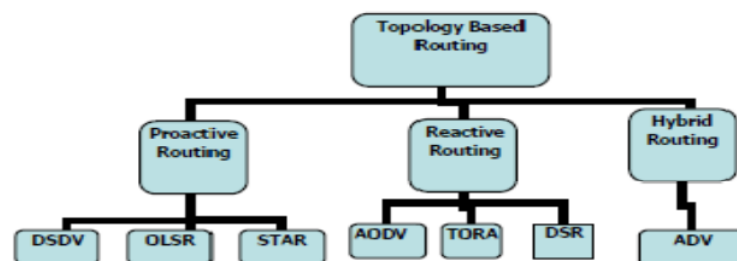
Gambar 2. 2 Komunikasi *Vehicle-to-Vehicle* (V2V)

Sumber: (Zeadally, et al., 2010)

Komunikasi jenis ini banyak digunakan untuk aplikasi-aplikasi keamanan seperti peringatan keselamatan, informasi lalu lintas, peringatan penghalang jalan, peringatan tabrakan persimpangan, dan lain sebagainya. Pada komunikasi ini, masing-masing kendaraan telah dilengkapi dengan GPS (*Global Positioning System*), sensor, alat jaringan, peta digital yang berisi informasi segmen jalan dan alat *computing*. Kendaraan akan mendeteksi sendiri pesan lalu lintasnya dan berkomunikasi dengan kendaraan tetangga dengan mem-*broadcast beacon* atau pesan HELLO secara *periodic* (Faikah, et al.).

2.2 *Routing* dan Protokol *Routing* VANET

Protokol *routing* pada jaringan *ad-hoc* mengatur jalur, pertukaran informasi dan mengambil keputusan jalur *runtime*. terdapat tiga tipe protokol *routing* seperti digambarkan pada gambar 2.3, yaitu protokol *routing* proaktif, reaktif dan *hybrid* (Arijit, et al., 2014).



Gambar 2. 3 Protokol *routing* *ad-hoc*

Sumber: (Arijit, et al., 2014)

2.2.1 Protokol Routing Proaktif

Protokol *routing* proaktif adalah protokol *routing* yang berdasarkan algoritma *shortest path*. Protokol *routing* ini memelihara dan meng-*update* informasi *routing* kepada semua *node* dalam jaringan yang diberikan setiap saat bahkan jika jalur tidak sedang digunakan. Jika beberapa jalur tidak pernah digunakan tetapi *update* mengenai jalur tersebut tetap di *broadcast* di antara *node*. *update* rute secara periodik dilakukan tanpa memperhatikan beban jaringan dan kendala *bandwidth*. Contoh dari tipe protokol ini adalah DSDV dan OLSR (Arijit, et al., 2014).

2.2.2 Protokol Routing Reaktif

Pada protokol *routing* reaktif atau *on demand* dirancang untuk mengatasi masalah *overhead* yang diciptakan oleh protokol *routing* proaktif dengan cara memperbaiki hanya rute yang aktif. Protokol ini menerapkan penentuan rute berdasarkan permintaan dan memperbaiki hanya rute yang saat ini digunakan, sehingga mengurangi beban pada jaringan ketika hanya sebagian rute yang tersedia yang sedang digunakan. Contoh dari tipe protokol *routing* ini adalah AODV dan DSR (Saha, et al., 2013).

2.2.3 Protokol Routing Hybrid

Protokol *routing hybrid* menggabungkan karakteristik dari kedua protokol *routing proactive* dan *reactive* untuk membuat *routing* lebih *scalable* dan efisien. Contoh dari tipe protokol *routing* ini adalah ADV (Dalal, et al., 2012).

2.3 Ad hoc On-demand Distance Vector (AODV)

AODV adalah *on demand routing*, dimana algoritma ini akan membangun rute antara *node* hanya apabila diinginkan oleh *source node*. AODV memelihara rute tersebut sepanjang masih dibutuhkan oleh *source node*. AODV menggunakan *sequence number* untuk memastikan bahwa rute yang dihasilkan adalah *loop-free* dan memiliki informasi *routing* yang paling *update*.

Ciri utama dari AODV adalah menjaga *timer-based state* pada setiap *node* sesuai dengan penggunaan *table routing*. Tabel *routing* akan kadaluarsa jika jarang digunakan (hadi, et al., 2011).

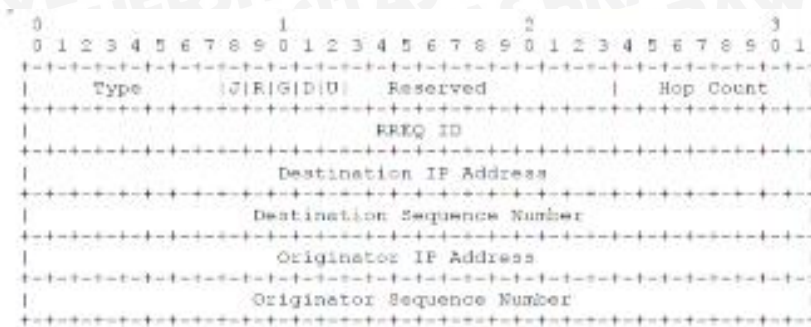
2.3.1 Tabel Routing AODV

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

1. *Destination IP Address*: berisi alamat IP dari *destination node* yang digunakan untuk menentukan rute
2. *Destination Sequence Number*: *destination sequence number* terkait untuk menentukan rute
3. *Next Hop*: *hop* berikutnya, bisa berupa tujuan atau *node* tengah. *Field* ini dirancang untuk meneruskan paket ke tujuan
4. *Hop Count*: jumlah *hop* dari alamat IP *source* sampai ke alamat IP *destination*
5. *Lifetime*: waktu dalam mili detik yang digunakan untuk *node* menerima RREP
6. *Routing Flags*: status sebuah rute. Up (valid), down (tidak valid) atau sedang diperbaiki (hadi, et al., 2011).

2.3.2 Route Discovery AODV

AODV menciptakan suatu rute dengan menggunakan *route request* (RREQ) dan *route reply* (RREP). Ketika *source node* menginginkan suatu rute menuju *destination node* tetapi belum mempunyai rute yang benar, maka *source node* akan menginisialisasi *route discovery process* untuk menemukan rute ke *destination node*. *Source node* akan mem-broadcast paket RREQ menuju *neighbour node*. Paket RREQ berisi *field* yang tertera pada gambar 2.4 di bawah ini.



Gambar 2. 4 paket RREQ

Sumber : (Arijit, et al., 2014)

Nilai *broadcast ID* akan bertambah satu setiap suatu *source node* mengirimkan RREQ yang baru dan digunakan sebagai identifikasi sebuah paket RREQ. Jika *node* yang menerima RREQ memiliki informasi rute menuju *destination node*, maka *node* tersebut akan mengirim paket RREP kembali menuju *source node*. Tetapi jika tidak mengetahui maka *node* tersebut akan mem-*broadcast* ulang RREQ ke *neighbour node* setelah menambahkan nilai *hop counter*. *Node* yang menerima RREQ dengan nilai *source address* dan *broadcast ID* yang sama dengan RREQ yang diterima sebelumnya akan membuang RREQ tersebut. *Source sequence number* yang digunakan oleh suatu *node* untuk memelihara informasi yang *valid* mengenai *reverse path* (jalur balik) menuju ke *source node*. Pada saat RREQ mengalir menuju *destination node* yang diinginkan, maka akan diciptakan *reverse path* menuju ke *node*. Setiap *node* akan membaca RREQ dan mengidentifikasi alamat dari *neighbour node* yang mengirim RREQ tersebut. Ketika *destination node* atau *node* yang memiliki informasi rute menuju *destination* menerima RREQ maka *node* tersebut akan membandingkan nilai *destination sequence number* yang ada di RREQ. Jika nilai *destination sequence number* yang ada di RREQ lebih besar dari nilai yang dimiliki oleh *node* maka paket RREQ tersebut akan di-*broadcast* kembali ke *neighbour node*, sebaliknya jika nilai *destination sequence number* yang ada di *node* lebih besar atau sama dengan nilai yang ada di RREQ maka *node* tersebut akan mengirim *route reply* (RREP) menuju *source node* dengan menggunakan *reverse path* yang telah dibentuk oleh RREQ. Paket RREP berisi *field* yang tertera pada gambar 2.5 di bawah ini.

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
+	+	+	+
	Type	R A	Reserved
+	+	+	+
	Destination IP address		
+	+	+	+
	Destination Sequence Number		
+	+	+	+
	Originator IP address		
+	+	+	+
	Lifetime		
+	+	+	+

Gambar 2. 5 paket RREP

Sumber : (Arijit, et al., 2014)

Intermediate node yang menerima RREP akan meng-update informasi *timeout* (masa aktif rute) jalur yang telah diciptakan. Informasi rute *source* ke *destination* akan dihapus apabila waktu *timeout*-nya habis (hadi, et al., 2011).

2.3.3 Perbaikan Routing AODV

Di dalam AODV setiap *node* bertanggung jawab untuk memelihara informasi rute yang telah disimpan di dalam *routing* tabel. Pada saat pengiriman data apabila terjadi perubahan topologi yang mengakibatkan suatu *node* tidak dapat dituju dengan menggunakan informasi rute yang ada di *routing* tabel, maka suatu *node* akan *route error packet* (RRER) ke *neighbour node* dan *neighbour node* akan mengirim kembali RERR demikian seterusnya hingga menuju *source node*. setiap *node* yang memperoleh RRER ini akan menghapus informasi yang mengalami *error* di dalam *routing* tabel. Kemudian *source node* akan melakukan *route discovery process* kembali apabila rute tersebut masih diperlukan. Paket RERR berisi *field* seperti yang tertera pada gambar 2.6 (S S, et al.).

[illegible]

Gambar 2. 6 paket RERR

Sumber : (Arijit, et al., 2014)

2.4 Adaptive Distance Vector (ADV)

Adaptive Distance Vector (ADV) berjalan dengan dasar dari algoritma *distance vector* yang menggunakan *sequence number* untuk menghindari *long-lived loops*. ADV menggunakan *routing updates* untuk mempelajari dan memelihara rute seperti algoritma *distance vector* yang lainnya. ada dua cara dalam ADV untuk mengurangi *routing overhead*.

1. memberikan variasi jumlah pemeliharaan rute yang aktif
2. memberikan variasi frekuensi dari *routing updates* (Boppa & Konduru, 2001).

2.4.1 Variasi Jumlah Pemeliharaan Rute yang aktif

Untuk mengurangi dari ukuran *routing updates*, ADV melakukan *advertise* dan memelihara rute hanya pada *receiver* aktif. Tidak seperti pada protokol *distance vector* sebelumnya dimana *advertise* dan pemeliharaan rute dilakukan ke semua *node* dalam jaringan. *Node* dapat dikatakan sebagai *receiver* aktif jika *receiver* tersebut merupakan *receiver* dari koneksi yang sekarang sedang aktif. Entri ke dalam tabel *routing* ditandai dengan spesial *flag* yang disebut dengan *receiver flag* untuk mengindikasikan jika *destination* adalah *receiver* aktif.

Koneksi ini dimulai dengan *source* mem-broadcast (*flood*) *network-wide* dengan *advertise* paket kontrol *init-connection* dimana *node destination* merupakan *receiver* aktif. Semua *node* akan menghidupkan *receiver flag* yang sesuai di dalam

tabel *routing* dan memulai *advertise* rute ke *receiver* dalam *update* selanjutnya. *Destination node* yang menjadi target menerima respon paket *init-connection* jika sudah tidak menjadi *receiver* aktif lagi dengan mem-broadcast *network-wide* dengan paket *receiver-alert*. Dengan kedua *broadcast network-wide* tersebut maka semua *node* akan mengetahui tentang *receiver* aktif dan rute dengan cepat.

Pada saat koneksi ditutup, *source* akan mem-broadcast *network-wide* dengan paket kontrol *end-connection* yang mengindikasikan bahwa koneksi telah dihentikan. Jika *destination node* tidak memiliki koneksi lainnya yang aktif, maka *destination* akan mem-broadcast paket kontrol *non-receiver-alert* yang mengindikasikan bahwa *destination* tersebut berhenti menjadi *receiver* aktif mulai dari sekarang. Kemudian *node* mematikan *receiver flag* yang sesuai dalam tabel *routing* dan rute menuju *node* tersebut tidak akan di *advertise* pada saat *routing update* selanjutnya. *Connection-initiation* dan *connection-termination* digunakan sekali pada setiap koneksi. Koneksi tersebut akan membantu dalam memvariasi koneksi aktif yang terbuka. Pembentukan ulang dari paket kontrol dicegah dengan penambahan dan entri *sequence number* dari *node* asal.

Walaupun jika paket *init-connection* dan *receiver-alert* menghilang, *source* akan *advertise receiver* dengan *receiver flag set* (metrik mungkin akan di set *infinity*) dalam semua *update* selanjutnya. Metode *advertising receiver* aktif ini tentu saja lebih lama daripada proses *connection initiation*. Tidak seperti proses pendeteksian rute di dalam protokol *on-demand*, proses *connection-initiation* dalam ADV terutama ditujukan untuk *advertise destination* sebagai *receiver* aktif, meskipun efek sampingnya adalah rute menuju *destination* juga diketahui semua *node* awalnya. Setelah itu rute menuju *receiver* diperbaiki menggunakan *routing update* (Boppana & Konduru, 2001).

2.4.2 Variasi Frekuensi dari Routing Updates

Pada proses ini akan dijelaskan bagaimana frekuensi dari *routing updates* dapat berubah dengan beban dan mobilitas jaringan (Boppana & Konduru, 2001).

2.4.2.1 Forwarding Node

Node disebut sebagai *forwarding node* untuk entri tabel *routing* tertentu dalam *node* tersebut, jika sebelumnya meneruskan setiap paket data untuk *destination* yang sesuai. Pada ADV variabel akan diperbaiki yang disebut dengan paket *handled*, yang bertambah satu pada paket *forwarding* dan dibagi dua setiap kali *update* yang berisikan entri yang di transmitkan. Nilai *non-zero* untuk variabel ini mengindikasikan jika *node* tersebut adalah *forwarding node* untuk entri *routing destination* (Boppa & Konduru, 2001).

2.4.2.2 Trigger Meter

Node seharusnya melakukan *update trigger* jika dalam tiga kondisi sebagai berikut:

1. jika ada beberapa paket data *buffered* karena kekurangan rute
2. jika satu atau lebih *neighbour* membuat permintaan untuk rute baru
3. digunakan untuk *forwarding node* yang bertujuan untuk *advertise* setiap rute fresh yang valid/tidak valid menuju *destination* sehingga rute tetap fresh.

Namun, alih-alih *trigger* melakukan *update* dengan segera setelah mengalami salah satu kondisi di atas, jika *node* menunggu sampai terlihat membutuhkan *update trigger*, *routing overhead* dapat dikurangi secara besar. Oleh karena itu ADV mengukur dari akibat semua peristiwa yang membutuhkan *trigger update* dan menambahkan variabel yang spesial yang disebut dengan *trigger meter* setiap kali peristiwa semacam itu terjadi.

Trigger meter dihubungkan dengan konstanta TRGMETER_FULL, TRGMETER_MED dan TRGMETER_LOW, yang diberikan berdasarkan urutan nilai mereka. *Trigger meter* bertambah berdasarkan level yang tepat tergantung pada prioritas dengan *node* yang mana yang harus di *update trigger* saat melihat kondisi yang memaksa harus *update*. Setiap waktu *trigger meter* dimodifikasi untuk mengecek apakah *trigger meter* melebihi nilai TRGMETER_FULL. Jika iya maka *full update* segera dijadwalkan. Jika *trigger meter* melintasi nilai *threshold* tetapi kurang dari TRGMETER_FULL, maka sebagian *update* segera dijadwalkan. Selain dari itu, tidak ada *update* yang dijadwalkan (Boppa & Konduru, 2001).

2.4.2.3 Trigger Threshold

Trigger threshold digunakan untuk menentukan kapan sebagian *update* perlu dilakukan *trigger*. *Trigger meter* akan di-reset ke nol setelah penjadwalan *update*. *Trigger threshold* akan berubah secara dinamis berdasarkan *history* dari nilai *trigger meter* pada saat melakukan beberapa *update* sebelumnya (Boppana & Konduru, 2001).

2.4.2.4 Mobilitas Node dan Kecepatan Jaringan

Mobilitas jaringan ditentukan oleh pengamatan jumlah perubahan *neighbour* oleh *node* dalam 1-hop *neighbourhood* dalam periode dari jumlah tetap dari *full update*. Jumlah *node* yang keluar dari *range* 1-hop dapat ditentukan dengan jumlah *link* yang rusak sedangkan yang masuk ke dalam *range* dapat ditentukan ketika *update* diterima dari *neighbour* dengan metrik lebih besar dari sebelumnya. Jika jumlah perubahan *neighbour* melebihi dari jumlah yang telah ditetapkan, maka *node* tersebut dikategorikan sebagai jaringan HIGH_SPEED atau sebaliknya dikategorikan sebagai LOW_SPEED (Boppana & Konduru, 2001).

2.4.2.5 Buffer Threshold

Pada saat paket yang baru *buffer* karena kekurangan rute, maka cek dilakukan. Jika jumlah paket yang sudah *buffer* melebihi jumlah yang ditentukan maka disebut dengan *buffer threshold*. Jika melebihi, maka *trigger meter* akan bertambah oleh TRGMETER_MED untuk secara bertahap memaksa untuk *update* (Boppana & Konduru, 2001).

2.4.2.6 Pengiriman Routing Updates

Struktur dari entri *routing update* digambarkan pada gambar 2.7.

Destination IP address (32 bits)			
Next hop IP address (32)			
Sequence number (16)	Metric (8)	ls_receiver (1)	Expected_response (2)
Unused (5)			

Gambar 2. 7 Entri *Routing Update* pada ADV

Sumber : (Boppana & Konduru, 2001)

Dalam *full update*, *node* berisi semua entri dari *receiver aktif* (*node* dengan *receiver flag set*), bahkan jika tidak dibutuhkan *advertise* untuk setiap entri tersebut. *Sequence number* yang digunakan mirip dengan DSDV kecuali pada setiap *update* (full dan sebagian) dengan hasil *sequence number* baru yang lebih tinggi.

Dengan setiap entri *routing update*, *node* mengirimkan nilai respon yang diharapkan dari ZERO (bit sequence 00), LOW (01), MEDIUM (10) atau HIGH (11). Nilai respon yang diharapkan ditentukan menggunakan aturan berikut :

1. Respon yang diharapkan dari HIGH diberikan ketika ada paket menunggu untuk rute tersebut dalam *node buffer* tanpa memperhatikan kecepatan jaringan
2. Dalam jaringan HIGH_SPEED, respon yang diharapkan dari MEDIUM diberikan jika *node* adalah termasuk *forwarding node* untuk *destination* yang diberikan dalam entri *routing*
3. Dalam jaringan LOW SPEED, respon yang diharapkan dari LOW diberikan jika *node* tersebut adalah *forwarding node* pada setiap *neighbour* untuk entri *routing destination*
4. Jika tidak ada kriteria di atas maka respon yang diharapkan di set ke ZERO (Boppana & Konduru, 2001).

2.4.2.7 Proses Penerimaan *Update*

Proses *update* juga memuat salinan *receiver flag* dari *ls_receiver* dari *update* yang diterima (dengan *sequence number* yang lebih tinggi) untuk mengidentifikasi dan menyebarkan *receiver* aktif, bahkan jika paket kontrol

receiver-alert dan *non-receiver-alert* hilang. Karena *forwarding node* perlu menjaga rute tetap fresh, maka *forwarding node* mencoba untuk *trigger update* saat melihat setiap valid/tidak valid rute untuk *receiver* aktif dengan *sequence number* yang lebih tinggi. Hal ini dicapai dengan penambahan *trigger meter* oleh TRGMETER MED.

Setelah setiap entri di dalam *routing update* diproses, *trigger meter* akan bertambah oleh TRGMETER HIGH, TRGMETER MED atau TRGMETER LOW untuk masing-masing respon yang diharapkan untuk HIGH, MEDIUM atau LOW. *Trigger meter* tetap tidak berubah untuk respon yang diharapkan untuk ZERO. Selain pada kasus di atas, *trigger meter* bertambah oleh TRGMETERFULL pada saat *receiver* aktif menemukan bahwa entri *routing* tidak valid di dalam penerimaan *update* dari *neighbour*.

Setelah semua proses *update* entri, nilai akumulasi dari *trigger meter* di cek untuk melihat jika melebihi TRGMETER FULL, dalam kasus ini *full update* melakukan penjadwalan untuk transmisi. Jika tidak maka *trigger meter* di cek jika melintasi *trigger threshold*. Dalam kasus ini *trigger update* melakukan penjadwalan untuk transmisi (Boppana & Konduru, 2001).

2.5 Pengukuran Kinerja

Pengukuran kinerja yang berbeda digunakan untuk mengecek kinerja dari protokol *routing*. Pengukuran kinerja digunakan untuk mengecek efektivitas dari protokol *routing* VANET. Pengukuran kinerja *routing* protokol yang digunakan antara lain adalah *end to end delay*, *throughput*, *packet loss* dan *packet delivery ratio* (Arijit, et al., 2014).

2.5.1 End To End Delay

End to end delay adalah waktu yang diperlukan oleh suatu paket data yang berasal dari *source node* hingga mencapai *destination node*. *End to end delay* secara tidak langsung berhubungan dengan kecepatan transfer data suatu jaringan (Saha, et al., 2013).

2.5.2 *Throughput*

Throughput adalah rata-rata jumlah dari data paket yang sukses terkirim pada komunikasi jaringan atau *node* jaringan. Dengan kata lain *throughput* di definisikan sebagai nilai total dari paket yang diterima pada destination dibandingkan dengan total dari paket yang ditransmisikan. *Throughput* dihitung dengan satuan bytes/detik atau paket data per detik. (Saha, et al., 2013).

2.5.3 *Packet loss*

Packet loss adalah jumlah paket yang rusak atau hilang pada saat transmisi dari source ke destination. Beberapa penyebab *packet loss* adalah :

1. Kongesti yang disebabkan oleh antrian yang telah melewati batas jumlah paket yang dapat ditampung di dalam buffer
2. Kerusakan data (*data corruption*) yang bisa diakibatkan salah pemrosesan pada saat enkapsulasi, *collision* pada jaringan, atau kerusakan pada media transmisi
3. Kerusakan eksternal ini biasanya berupa gangguan catu daya (*power loss*) dimana sumber daya utama turun secara drastik atau bahkan mati dan sumber daya cadangan gagal menggantikan sumber daya utama (Muslim, 2007).

2.5.4 *Packet Delivery Ratio (PDR)*

Packet delivery ratio adalah rasio antara banyaknya paket yang diterima oleh tujuan dengan banyaknya paket yang dikirim oleh sumber (Dhamayanti & Hendrantoro, 2013).

2.6 *National Chiao Tung University Network Simulator (NCTUns)*

NCTUns merupakan program simulator jaringan yang terintegrasi dengan kemampuan simulasi lalu lintas seperti merancang peta dan mengendalikan mobilitas kendaraan. NCTUns memasukkan *Graphic User Interface* (GUI) untuk membantu dalam proses mendesain peta. Program simulator jaringan NCTUns

memiliki banyak fitur yang berguna seperti menggunakan *protocol stack* linux TCP/IP untuk menghasilkan simulasi yang realistis (Rios, et al., 2011).

2.6.1 Komponen NCTUNS

NCTUNS memiliki delapan komponen yaitu:

1. *Graphical User Interface* (GUI)
2. *Simulation Engine* (SE)
3. Kumpulan Modul-modul Protokol
4. *Dispatcher*
5. *Coordinator*
6. *Real-life user-level application programs*
7. *User-level daemons*
8. *Kernel patches* (Wan, et al., 2010).

2.6.1.1 *Graphical User Interface* (GUI)

GUI merupakan lingkungan terintegrasi yang dapat memudahkan user untuk menjalankan simulasi. GUI terdiri dari empat komponen, yaitu “*Topology Editor*”, “*Node Editor*”, “*Performance Monitor*” dan “*Packet Animation Player*”. Keempat komponen tersebut memungkinkan user untuk mengubah topologi jaringan, mengkonfigurasi modul protokol dari *node* jaringan, mengatur parameter-parameter untuk modul protokol, menentukan *traffic* jaringan, menampilkan kurva kinerja, memutar kembali animasi dari transfer paket data, dan sebagainya (Wan, et al., 2010).

2.6.1.2 *Simulation Engine* (SE)

Simulation Engine (SE) adalah inti dari NCTUns. Ini merupakan *user-level program* yang menyediakan layanan simulasi yang dasar dan penting bagi modul protokol, seperti *event scheduling*, *timer management*, *packet manipulation*, dan sebagainya (Wan, et al., 2010).

2.6.1.3 Kumpulan Modul-Modul Protokol

NCTUns menyediakan *platform* berbasis modul. Sebuah modul berhubungan dengan satu lapisan pada *protocol stack*. Misalnya modul ARP mengimplementasikan protokol ARP. Pengguna program dapat memasukkan modul baru ke dalam *protocol stack* yang telah ada, menghapus modul dari *protocol stack*, atau mengganti sebuah modul dengan modul sendiri. Melalui operasi-operasi tersebut, pengguna dapat mengendalikan serta mengubah perilaku dari sebuah perangkat jaringan (Wan, et al., 2010).

2.6.1.4 Dispatcher

Komponen ini dapat mengatur serta menggunakan beberapa server simulasi secara simultan untuk menaikkan *packet delivery ratio* simulasi keseluruhan. Ketika *user* memasukkan tugas simulasi kepada *dispatcher*, *dispatcher* akan memilih *simulation engine* yang tersedia untuk menjalankan tugas tersebut (Wan, et al., 2010).

2.6.1.5 Coordinator

Komponen ini menghubungkan GUI dengan *dispatcher*. *Coordinator* bertanggungjawab akan eksekusi proses dari *simulation engine* untuk menjalankan simulasi, pelaporan status dari *simulation machine* kepada *dispatcher*, serta komunikasi dengan GUI serta *dispatcher* (Wan, et al., 2010).

2.6.1.6 Real-Life User-Level Application Programs

Setiap program aplikasi dapat dijalankan untuk menghasilkan *traffic* jaringan yang realistis, konfigurasi jaringan, memantau *traffic* jaringan, dan sebagainya (Wan, et al., 2010).

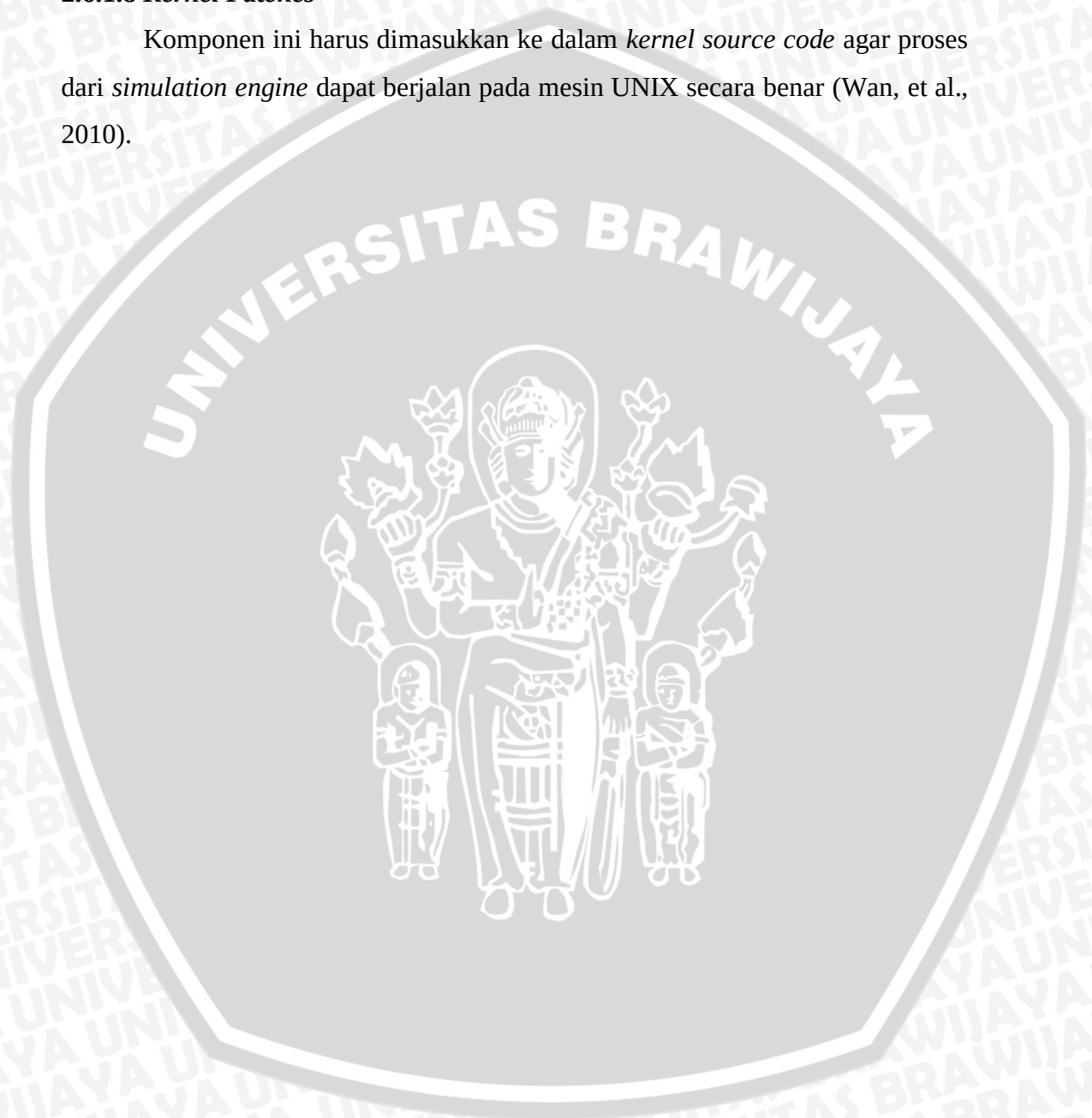
2.6.1.7 User-Level Daemons

NCTUns menyediakan *background process* yang dapat berjalan selama seluruh kasus simulasi. Dengan menjalankan *daemon*, seperti RIP atau OSPF

routing daemons, masukan *routing* yang dibutuhkan oleh simulasi jaringan dapat dihasilkan secara otomatis (Wan, et al., 2010).

2.6.1.8 Kernel Patches

Komponen ini harus dimasukkan ke dalam *kernel source code* agar proses dari *simulation engine* dapat berjalan pada mesin UNIX secara benar (Wan, et al., 2010).



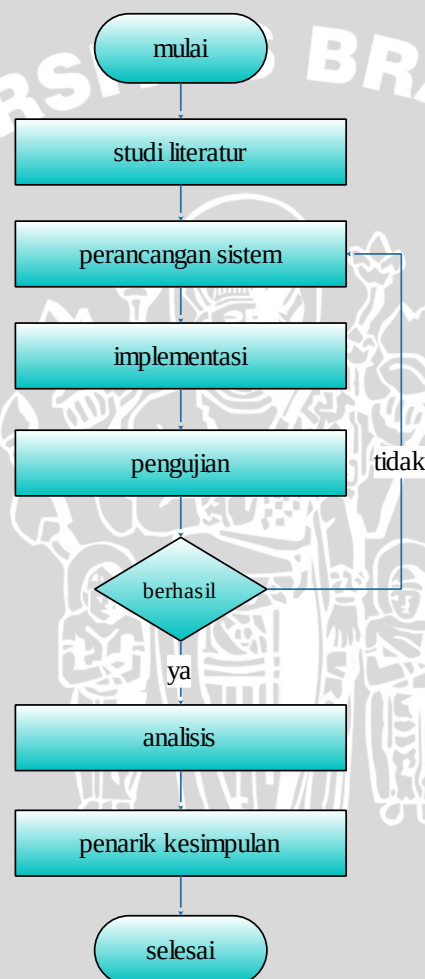
BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Bagian metodologi menjelaskan tahapan yang digunakan dalam penelitian.

Tahapan penelitian digambarkan dalam bentuk diagram alir seperti pada gambar 3.1.



Gambar 3. 1 diagram alir metode penelitian

Sumber : [Perancangan]

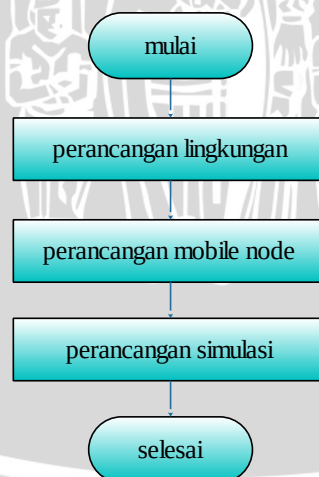
3.1.1 Studi Literatur

Studi literatur dilakukan untuk mempelajari dan memahami konsep yang terkait dengan teori-teori yang mendukung dalam perancangan dan simulasi protokol *routing* AODV dan ADV pada VANET. Sumber bacaan yang digunakan berupa *text book*, *paper*, dan sumber bacaan *softcopy* yang diperoleh dari internet. Teori-teori yang digunakan yaitu :

- VANET
- AODV
- ADV
- Pengukuran kinerja
- NCTUns.

3.1.2 Perancangan Sistem

Simulasi VANET dengan menggunakan protokol *routing* AODV dan ADV dilakukan pada 2 skenario yaitu *highway* dan *urban* yang akan dilakukan dengan menggunakan simulator NCTUns 6.0. Diagram alir perancangan sistem ditunjukkan pada gambar 3.2.



Gambar 3. 2 diagram alir perancangan sistem

Sumber : [Perancangan]

3.1.3 Implementasi

Tahap selanjutnya adalah melakukan implementasi dalam bentuk simulasi sesuai dengan tahap perancangan sistem yang telah ditentukan. Simulasi dilakukan dengan menggunakan program simulator NCTUns 6.0.

3.1.4 Pengujian dan Analisa

Pada tahap ini, simulasi dilakukan dalam beberapa skenario dan hasilnya diolah sehingga dapat dianalisa. Pengujian ini ini di lakukan untuk melihat bagaimana kinerja dari protokol *routing* AODV dan ADV dengan perbedaan kecepatan *node* dan penambahan *node* untuk masing-masing kecepatan. Tujuan utamanya adalah melihat bagaimana kinerja pada masing-masing protokol *routing* pada lingkungan *urban* dan *highway*. Tahapan pengujian dan analisa digambarkan pada diagram alir sebagai berikut.



Gambar 3. 3 diagram alir pengujian dan analisa sistem

Sumber : [Perancangan]

Simulasi ini diujikan berdasarkan 3 skenario yaitu sebagai berikut :

1. Pengujian skenario *urban* dengan kecepatan *node* 8 m/s.

Tahap pengujian ini dilakukan dengan cara mensimulasikan VANET pada lingkungan *urban* dengan kecepatan masing-masing *node* adalah 8 m/s. Untuk setiap simulasi jumlah *node* yang diujikan berbeda. Jumlah *node* yang diujikan untuk setiap simulasi adalah 30, 40, dan 50 *node* dan masing-

masing-masing simulasi diujikan dengan menggunakan protokol *routing* AODV dan ADV.

2. Pengujian skenario *urban* dengan kecepatan *node* 18 m/s.

Tahap pengujian ini dilakukan dengan cara mensimulasikan VANET pada lingkungan *urban* dengan kecepatan masing-masing *node* adalah 18 m/s. Untuk setiap simulasi jumlah *node* yang diujikan berbeda. Jumlah *node* yang diujikan untuk setiap simulasi adalah 30, 40, dan 50 *node* dan masing-masing simulasi diujikan dengan menggunakan protokol *routing* AODV dan ADV.

3. Pengujian skenario *highway*.

Tahap pengujian ini dilakukan dengan cara mensimulasikan VANET pada lingkungan *highway* dengan kecepatan masing-masing *node* yang berbeda. Kecepatan 50% *node* adalah 20 m/s dan 50% *node* yang lain adalah 36 m/s. Untuk setiap simulasi jumlah *node* yang diujikan berbeda. Jumlah *node* yang diujikan untuk setiap simulasi adalah 10, 20, dan 30 *node* dan masing-masing simulasi diujikan dengan menggunakan protokol *routing* AODV dan ADV.

Setelah tahapan implementasi maka data file log diolah dengan menggunakan program microsoft excel kemudian dilakukan analisa dimana hasil analisa akan digunakan untuk pengambilan kesimpulan. Analisa kinerja VANET menggunakan 4 parameter yaitu *delay*, *throughput*, *packet loss*, *packet delivery ratio* untuk setiap simulasi.

3.1.5 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah melakukan analisis data. Kesimpulan merupakan hasil akhir yang diperoleh dan didasarkan pada kesesuaian dengan teori dan praktek yang telah dilakukan. Selanjutnya penulisan saran untuk memperbaiki kesalahan yang terjadi untuk penelitian selanjutnya.

3.2 Perancangan

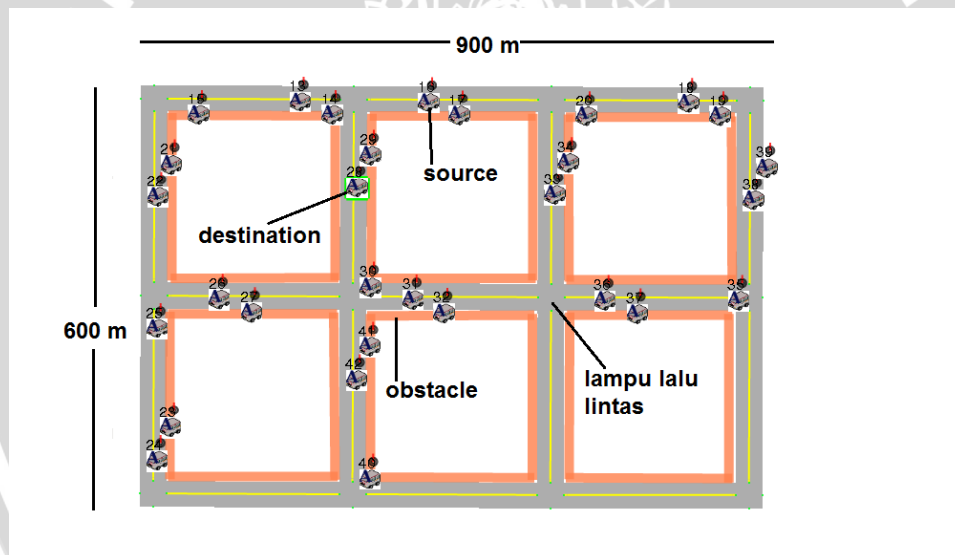
Tahapan perancangan pada simulasi VANET dengan menggunakan protokol *routing* AODV dan ADV adalah sebagai berikut.

3.2.1 Perancangan Lingkungan VANET

Pada penelitian ini terdapat 2 lingkungan yang digunakan yaitu *urban* dan *highway* dimana pada lingkungan *urban* mengikuti kondisi lingkungan perkotaan untuk melihat bagaimana pengaruh proses *routing* jika terhalang oleh *obstacle* dan lingkungan *highway* mengikuti kondisi lingkungan jalan tol.

3.2.1.1 Perancangan Lingkungan Urban

Perancangan pada lingkungan *urban* digambarkan seperti pada gambar 3.4.



Gambar 3. 4 lingkungan urban

Sumber : [perancangan]

a. Perancangan Jalan

Pada Lingkungan *urban* ini menggunakan area simulasi 900 m x 600 m dan untuk jalan pada *urban* menggunakan 2 tipe jalan yaitu *single-lane* dan *crossroad*. *Single-lane* adalah tipe jalan yang terdiri dari 2 jalur yang berlawanan arah. Lebar masing-masing jalan adalah 20 meter. Sedangkan *crossroad* berfungsi untuk menghubungkan masing-masing persimpangan jalan. Pada *crossroad* secara

default telah terpasang 4 lampu lalu lintas. *Traffic light* ini diaktifkan hanya pada pertigaan dan perempatan jalan sehingga menghasilkan *traffic* jaringan yang realistis.

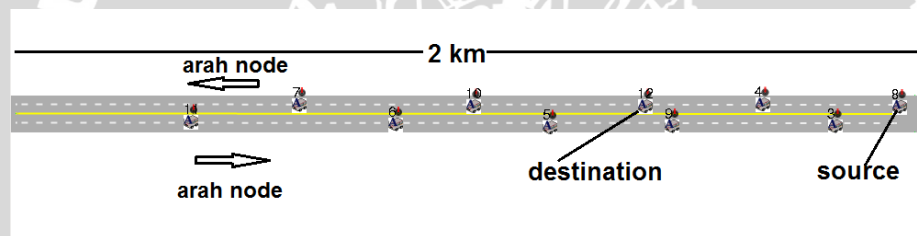
b. Perancangan Radio Obstacle

radio obstacle adalah objek persegi panjang yang dapat menghalangi sinyal *wireless* seluruhnya atau hanya mengurangi kekuatan sinyal dari *wireless*. pada perancangan *urban* ini *obstacle* dibangun untuk menghalangi sinyal *wireless* seluruhnya.

3.2.1.2 Perancangan Lingkungan Highway

Perancangan pada lingkungan *highway* digambarkan seperti pada gambar

3.3.



Gambar 3. 5 lingkungan *highway*

sumber : [Perancangan]

Pada lingkungan *highway* ini menggunakan area simulasi dengan panjang jalan 2 km dan untuk kondisi jalan menggunakan tipe jalan *multi-lane* yang terdiri dari 4 jalur dengan 2 jalur adalah jalur satu arah. Lebar masing-masing jalur adalah 20 meter.

3.2.2 Perancangan Simulasi

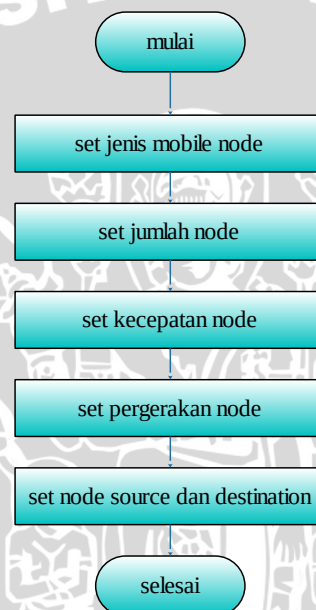
Perancangan simulasi dari jaringan VANET yang disimulasikan adalah sebagai berikut :

1. Setiap simulasi berjalan selama 200 detik

2. Data yang ditransmisikan dalam simulasi adalah paket UDP sebesar 1400 byte. Paket data dikirimkan mulai detik ke-0 hingga detik ke-200 dan masing-masing dilakukan dalam simulasi terpisah
3. Jaringan yang digunakan pada masing-masing *node* adalah IEEE 802.11b.

3.2.3 Perancangan *Mobile Node*

Langkah-langkah dalam perancangan *mobilenode* yang dilakukan dalam simulasi adalah sebagai berikut.

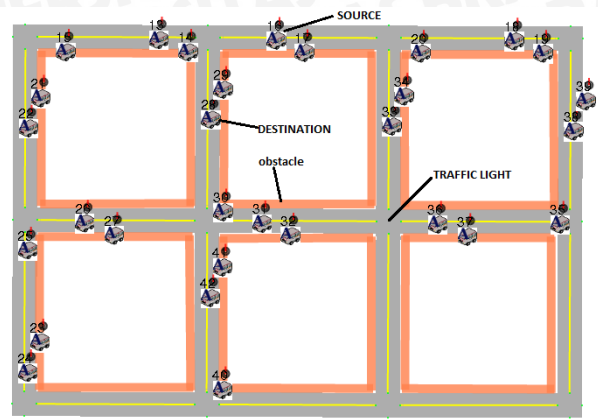


Gambar 3. 6 diagram alir perancangan *mobile node*

sumber : [Perancangan]

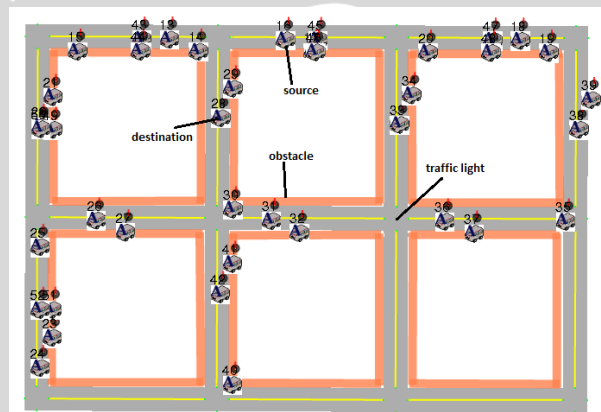
Pada perancangan *mobile node*, jenis *mobile node* yang akan disimulasikan adalah menggunakan ITS car 802.11b dengan peletakan *node* yang disusun secara *random* dan jumlah *node* yang berbeda-beda.

Pada lingkungan *urban* jumlah *node* yang disimulasikan adalah 30, 40, dan 50 *node* seperti pada gambar di bawah ini.



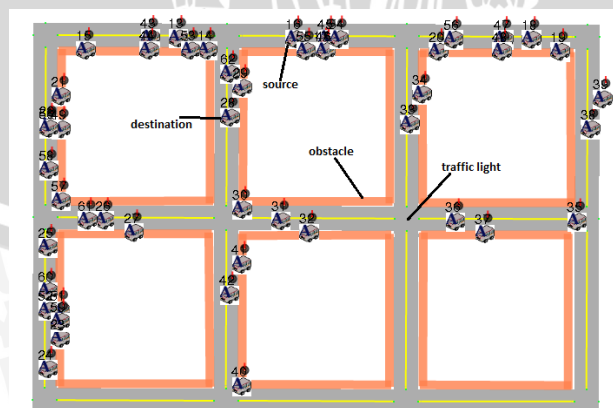
Gambar 3. 7 lingkungan urban 30 node

Sumber : [perancangan]



Gambar 3. 8 Lingkungan urban 40 node

Sumber: [perancangan]

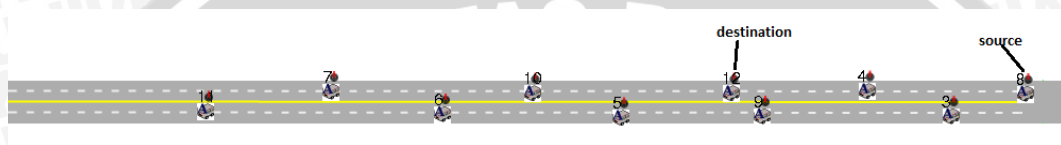


Gambar 3. 9 lingkungan urban 50 node

Sumber : [perancangan]

Pada masing-masing lingkungan tersebut kecepatan masing-masing *node* yang akan disimulasikan adalah 8 m/s dan 18 m/s untuk setiap simulasi. Pergerakan masing-masing *node* disusun secara *random* sehingga *node* bebas bergerak ke arah mana saja. Langkah selanjutnya adalah pemilihan *node* yang akan dijadikan *source* dan *destination*. *Node* yang dijadikan *source* dan *destination* berjumlah satu yaitu *node* 16 yang menjadi *source* dan *node* 28 yang menjadi *destination*.

Pada lingkungan *highway* jumlah *node* yang disimulasikan adalah 10, 20, dan 30 *node* seperti pada gambar di bawah ini.



Gambar 3. 10 lingkungan *highway* 10 *node*

Sumber : [perancangan]



Gambar 3. 11 lingkungan *highway* 20 *node*

Sumber : [perancangan]



Gambar 3. 12 lingkungan *highway* 30 *node*

Sumber : [perancangan]

Pada masing-masing lingkungan tersebut kecepatan masing-masing *node* yang akan disimulasikan berbeda dengan lingkungan *urban* yang pada setiap simulasi kecepatan masing-masing *node* adalah sama. Pada lingkungan *highway* untuk setiap simulasi kecepatan yang digunakan adalah 50% *node* 20 m/s dan 50% *node* sisanya dengan kecepatan 36 m/s. Langkah selanjutnya adalah pemilihan *node* yang akan dijadikan *source* dan *destination*. *Node* yang dijadikan *source* dan *destination*

berjumlah satu yaitu *node* 8 yang menjadi *source* dan *node* 12 yang menjadi *destination*.



BAB IV

IMPLEMENTASI

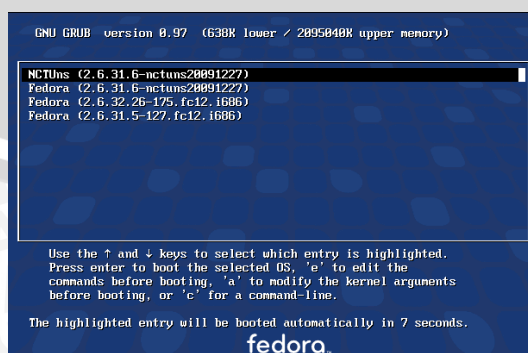
Bab ini menjelaskan tentang implementasi VANET dengan menggunakan protokol *routing* AODV dan ADV pada simulator NCTUns 6.0.

4.1 Instalasi NCTUns 6.0

Dalam penelitian ini, penulis menggunakan program simulator NCTUns 6.0 yang ditanamkan dalam sistem operasi fedora 12. Sistem operasi fedora tersebut ditanam dalam sistem operasi Microsoft windows 8 dengan menggunakan program aplikasi virtual machine VMWare Workstation.

Seluruh proses instalasi simulator NCTUns 6.0 dilakukan dalam aplikasi terminal pada fedora dengan menggunakan status sebagai root user. Instalasi NCTUns 6.0 dilakukan melalui langkah-langkah sebagai berikut :

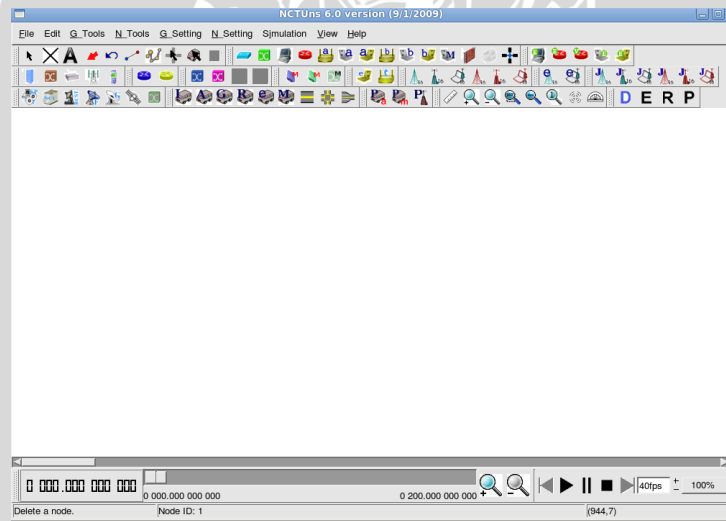
1. Jalankan file program skrip dengan perintah `./install.sh`. selama proses ini program `install.sh` akan memunculkan beberapa kali pertanyaan. Untuk instalasi normal cukup jawab “yes” atau “Y” pada setiap pertanyaan. Seluruh paket program akan terpasang pada direktori `/usr/local/nctuns`. Selanjutnya, seluruh pilihan “`${tempat instalasi}`” merujuk kepada “`/usr/local/nctuns`”
2. *Reboot* sistem untuk menggunakan kernel yang baru terpasang. Ketika *boot loader* linux menampilkan pilihan *kernel*, pilih *kernel* “NCTUNS”



Gambar 4. 1 pilihan *booting* NCTUNS 6.0 pada fedora 12

Sumber : [Perancangan]

3. Setelah masuk ke dalam terminal dan *login* sebagai root, lakukan beberapa pengaturan variabel lingkungan. Masukkan perintah berikut ini pada *.bashrc*
Export NCTUNSHOME=\$ {tempat instalasi}
Export NCTUNS_TOOLS=\${tempat instalasi}/tools
Export NCTUNS_BIN={tempat instalasi}/bin
4. Set semua variabel yang telah dimasukkan dengan perintah
source \${tempat instalasi}/etc/nctuns.bash
5. Non-aktifkan setting firewall yang telah dimasukkan dengan perintah
service iptables stop
6. Jalankan program dispatcher pada \$ {tempat instalasi}/bin.
7. Jalankan program coordinator pada \$ {tempat instalasi}/bin
8. Jalankan program nctunscient pada \${tempat instalasi}/bin.



Gambar 4. 2 tampilan NCTUns 6.0

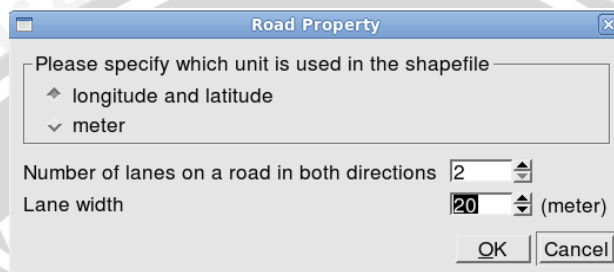
Sumber : [Perancangan]

4.2 Pengaturan Lingkungan Jaringan VANET

Tahapan ini akan membahas tentang pembuatan dan pengaturan lingkungan pada jaringan VANET yaitu *urban* dan *highway*. Proses yang dilakukan yaitu pengaturan jalan, pengaturan *obstacle* dan lampu lalu lintas pada lingkungan *urban*.

4.2.1 Pengaturan Lingkungan Urban.

pembuatan ITS *road* dalam NCTUns dilakukan pada mode topology editor. Dalam pemilihan ITS *road* pada lingkungan *urban*, jenis yang digunakan adalah *single lane road* dimana jalur jalan yang digunakan adalah 2 jalur. Lebar jalur adalah 20 meter. Pembuatan jalan digambarkan pada gambar 4.3.



Gambar 4. 3 modul pengaturan ITS *road* pada *urban*

Sumber : [Perancangan]

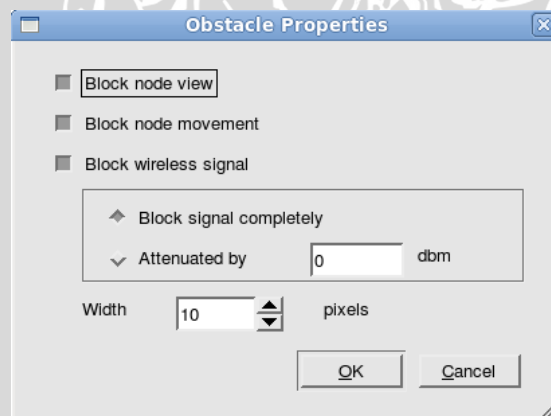
Kemudian sebagai penghubung jalan pada setiap persimpangan jalan menggunakan jenis ITS *road crossroad* dimana pada *crossroad* juga telah terpasang 4 lampu lalu lintas secara otomatis. Lampu lalu lintas tersebut dikendalikan oleh *signal agent*. *Signal agent* berfungsi untuk mengontrol *signal state* dari 4 lampu lintas pada *crossroad*. *Signal state* mempunyai dua komponen yaitu *signal logic* dan *signal information API*. *Signal logic* berfungsi untuk mengatur *signal state* pada saat *signal state* harus berubah. Sedangkan *signal information API* adalah sebutan bagi *signal agent* yang mengupdate database informasi sinyal yang tersimpan pada Simulation Engine (SE). Pengaturan lampu lalu lintas digambarkan pada gambar 4.4.



Gambar 4. 4 modul pengaturan lampu lalu lintas

Sumber : [Perancangan]

Setelah pembuatan jalan dan pengaturan lampu lalu lintas, tahap selanjutnya adalah pembuatan *radio obstacle* yang diletakkan pada masing-masing sisi jalan untuk menghalangi proses *routing*. *Obstacle* di-set dengan menghalangi pandangan dan sinyal *wireless* secara total. Pengaturan *obstacle* digambarkan pada gambar 4.5.

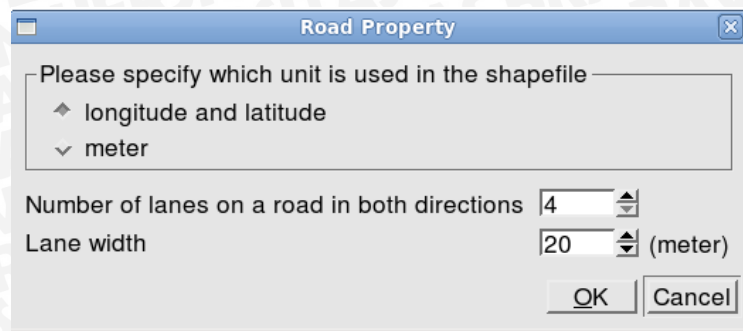


Gambar 4. 5 modul perancangan *obstacle*

Sumber : [Perancangan]

4.2.2 Pengaturan Lingkungan *Highway*.

Dalam pemilihan ITS *road* pada lingkungan *highway*, jenis yang digunakan adalah *multi lane road* dimana jalur jalan yang digunakan adalah 4 jalur. Lebar jalur adalah 20 meter. Pembuatan jalan digambarkan pada gambar 4.6.



Gambar 4. 6 modul pengaturan ITS road pada *highway*

Sumber : [Perancangan]

Pada lingkungan *highwaycrossroad* diletakkan pada masing-masing ujung jalan. Lampu lalu lintas pada *crossroad* ini tidak diaktifkan karena hanya berfungsi sebagai tempat putar balik *node*.

4.3 Pengaturan *Mobile node*

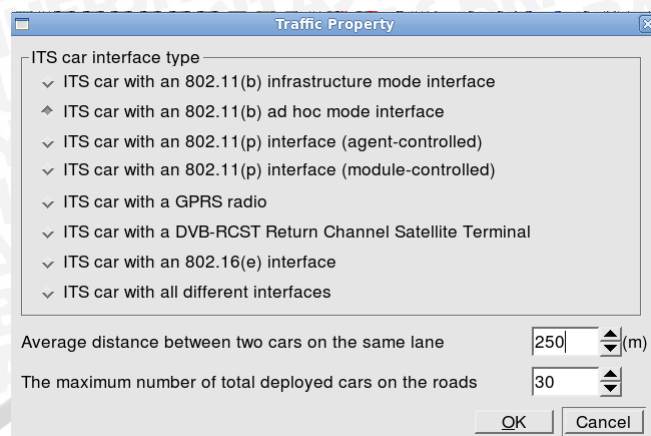
Tahapan ini akan membahas tentang pembuatan dan pengaturan *mobile node* pada jaringan VANET yang nantinya akan diletakkan pada masing-masing lingkungan jaringan yang telah dibuat.. tahapan pada proses ini yaitu :

1. Pengaturan jenis *mobile node*.

Pada penelitian ini, *mobile node* diletakkan secara otomatis oleh simulator sehingga *mobile node* tersebar secara acak pada lingkungan yang telah dibuat. Modul *mobile node* terletak pada *Ntools – ITS network– deploy cars automatically*. Pada modul tersebut terdapat banyak pilihan *interfaceITS car*. Pada penelitian ini menggunakan *ITS car with an 802.11(b) ad hoc mode interface* dimana MAC protocol pada masing-masing *node* menggunakan standard IEEE 802.11b dan jaringan tersebut bersifat *ad-hoc*.

2. Pengaturan jumlah *node*.

Pengaturan jumlah *node* dilakukan pada modul yang sama dengan pengaturan jenis *mobile node*. Jumlah *node* diatur sesuai dengan skenario yang akan diujikan. Pengaturan jumlah dan jenis *mobile node* digambarkan seperti pada gambar 4.7.

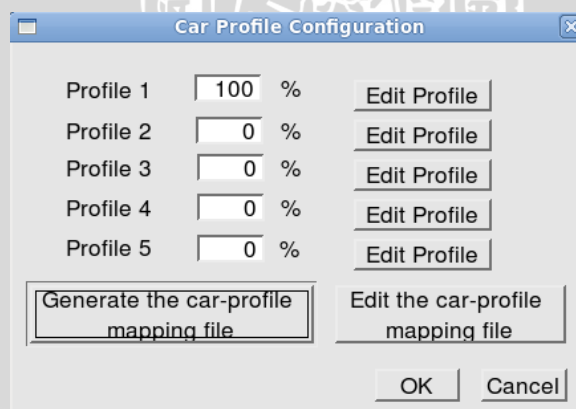


Gambar 4. 7 modul jumlah dan jenis *mobile node*

Sumber : [Perancangan]

3. Pengaturan kecepatan *node*.

Kecepatan *mobile node* pada simulator NCTUns menggunakan satuan meter per second. Pengaturan kecepatan *node* dilakukan pada mode *edit property*. Modul kecepatan *node* terletak pada *Ntools – ITS network – configure car profiles*. Pengaturan modul kecepatan *mobile node* digambarkan seperti pada gambar 4.8.



Gambar 4. 8 modul kecepatan *mobile node*

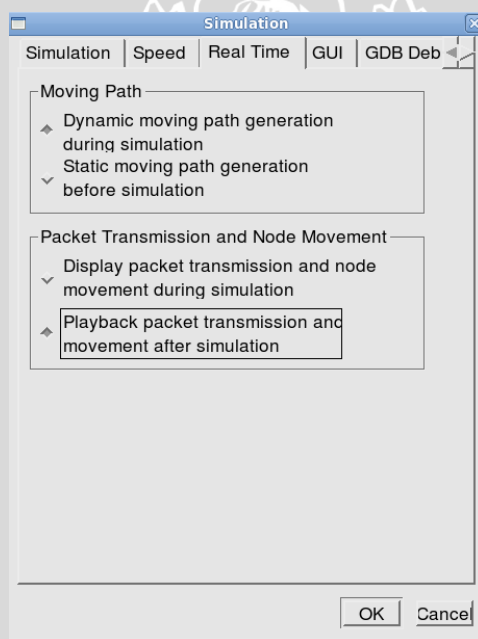
Sumber : [Perancangan]

Masing-masing profile berisi kecepatan *node* yang berbeda-beda. Pada *profile 1* kecepatan maksimal adalah 18 m/s, pada *profile 2* kecepatan maksimal adalah 36

m/s, pada *profile* 3 kecepatan maksimal adalah 20 m/s, pada *profile* 4 kecepatan maksimal adalah 8 m/s, dan pada *profile* 5 kecepatan maksimal adalah 50 m/s. Pada penelitian ini pada lingkungan *urban* menggunakan kecepatan *nodeprofile* 1 dan *profile* 4, sedangkan pada lingkungan *highway* menggunakan kecepatan *nodeprofile* 2 dan *profile* 3.

4. Pengaturan pergerakan *node*

Pada penelitian ini *node* bergerak secara random sehingga *node* bebas bergerak kemana saja. Modul pengaturan pergerakan *node* terletak pada *Gsetting – simulation*. Pada *tab simulation* pada pilihan *moving path* pergerakan *node* yang diaktifkan adalah *dynamic moving path generation during simulation*. Pengaturan modul pergerakan *node* digambarkan seperti pada gambar 4.9.

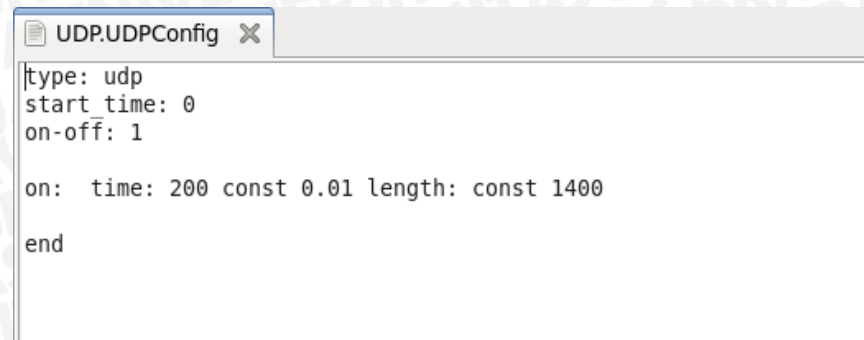


Gambar 4. 9 modul pergerakan *node*

Sumber : [Perancangan]

4.4 UDP

Pada penelitian ini paket data yang dikirimkan adalah paket UDP sebesar 1400 byte. Paket UDP yang dikirimkan di inisialisasi dalam bentuk file yang diberi nama UDP.UDPCfg. isi file UDP.UDPCfg adalah sebagai berikut.



Gambar 4. 10 format pengiriman paket UDP

Sumber : [Perancangan]

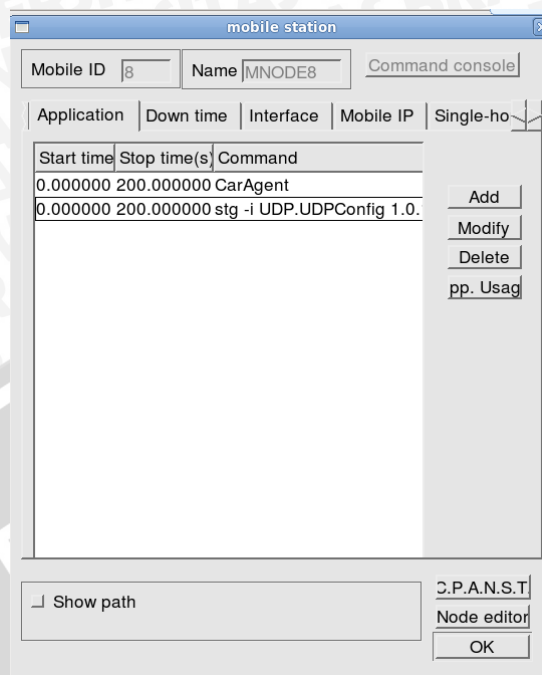
Pada gambar 4.10 dapat dilihat bahwa paket yang dikirimkan adalah paket UDP sebesar 1400 byte. Paket dikirimkan mulai dari detik ke-0 selama 200 detik. Pengiriman dikirimkan secara simultan dengan rentang waktu 0.01 detik kemudian file UDP tersebut dipanggil ke dalam program simulator NCTUNS yang dilakukan dalam mode Editor pada *node* pengirim data. Perintah yang digunakan untuk mengirimkan data adalah sebagai berikut :

```
stg -i UDP.UDPConfig 1.0.1.16
```

Penjelasan mengenai perintah di atas adalah sebagai berikut :

- stg -i menandakan jenis perintah pengiriman data dari file yang akan dipanggil.
- UDP.UDPConfig merupakan file UDP yang dipanggil untuk dikirimkan menuju IP *destination*
- 1.0.1.16 merupakan alamat IP tujuan pengiriman paket data dari *source node*.

Berikut ini adalah gambar dari *source node* setelah dilakukan pengaturan pengiriman data.



Gambar 4. 11 pengaturan trafik UDP pada *source node*

Sumber : [Perancangan]

Setelah melakukan pengaturan pada *source node*, selanjutnya dilakukan pengaturan pada *destination node*. Pengaturan ini dilakukan karena jika *destination node* tidak dilakukan pengaturan trafik, maka walaupun alamat IP dari *destination node* telah ditentukan pada source, *destination node* tersebut tidak akan menerima dan memproses hasil data yang terkirim oleh *source node*.. Selain itu, pengaturan untuk menghasilkan logfile dari parameter jaringan dilakukan pada tahap ini. Dalam tipe paket UDP, NCTUns menyediakan pilihan untuk menghasilkan *logfile* dari parameter *throughput* serta hasil *delay* dan *packet loss* per paket yang dikirim. Perintah yang digunakan untuk pengaturan pada *node* tujuan adalah sebagai berikut:

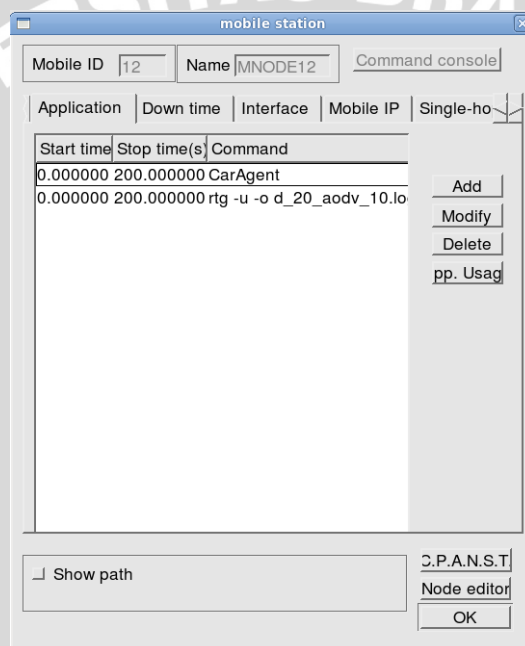
```
rtg -u -o d_20_aodv_10.log -w t_20_aodv_10.log
```

Penjelasan dari perintah di atas adalah sebagai berikut :

- `rtg -u` menandakan jenis paket yang ingin diterima, yaitu paket UDP.

- -o d_20_aodv_10.log merupakan perintah untuk menghasilkan dan menyimpan logfile dari *delay* dan *packet loss* dengan nama d_20_aodv_10.log.
- -w t_20_aodv_10.log merupakan perintah untuk menghasilkan dan menyimpan logfile dari *throughput* pada *node* penerima dengan nama t_20_aodv_10.log.

Berikut ini adalah gambar dari *destination node* setelah dilakukan pengaturan penerimaan data.

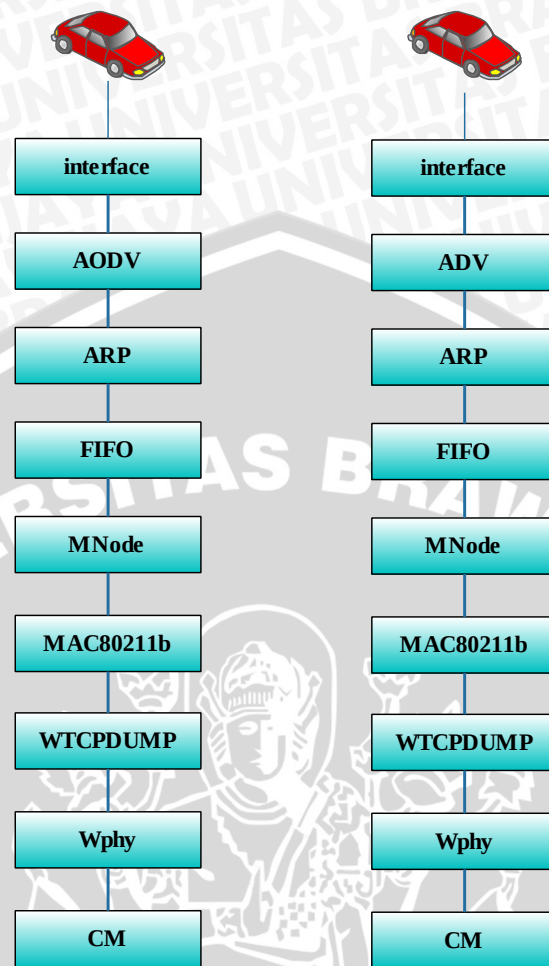


Gambar 4. 12 pengaturan trafik UDP pada *node* penerima

Sumber : [Perancangan]

4.5 Pengaturan *Protokol Stack*

Pada penelitian ini membandingkan 2 protokol *routing* pada masing-masing skenario yang telah dijelaskan di atas. Protokol *routing* yang diujikan adalah protokol *routing* AODV dan ADV. Berikut ini merupakan gambaran *protocol stack* dari setiap *node* dalam jaringan



Gambar 4. 13 *protocol stack AODV dan ADV*

Sumber : [perancangan]

Fungsi dari masing-masing *protocol stack* adalah sebagai berikut :

1. *Interface*. Pada modul *interface* berisi alamat IP dari masing-masing *node*.
2. AODV/ADV. Pada modul AODV/ADV berfungsi untuk mensimulasikan protokol *routing* AODV/ADV.
3. ARP. *Address Resolution Protocol* (ARP) merupakan penerjemah pengalamatan dari alamat IP menjadi alamat MAC.
4. FIFO. *First In First Out* (FIFO) merupakan metode *queuing* dimana paket diperlakukan sama dengan menempatkannya pada sebuah antrian lalu dilayani dengan urutan yang sama ketika paket tersebut memasuki antrian.
5. MNode. Pada modul MNode berfungsi untuk mensimulasikan *node* yang digunakan bersifat *ad-hoc*.

6. MAC80211b. modul ini berfungsi untuk mensimulasikan protokol MAC IEEE 802.11b dan juga menyimpan paket log yang datanya akan diolah untuk dianalisa.
7. WTCPDUMP. Modul ini berfungsi untuk meng-*capture* proses pengiriman dan penerimaan paket pada media *wireless*.
8. Wphy. Modul ini berfungsi untuk mensimulasikan *link bandwidth*, *frequency channel* dan *transmission power*.
9. CM. modul *Channel Mode* (CM) berfungsi untuk mensimulasikan *wireless channel* seperti *fading variance*, *average building height*, *average building distance*, *antenna height*.



BAB V

PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan implementasi, pengambilan data dan analisis kinerja VANET dengan menggunakan protokol *routing* AODV dan ADV yang dilakukan dengan menggunakan simulator NCTUns 6.0.

5.1 Implementasi VANET

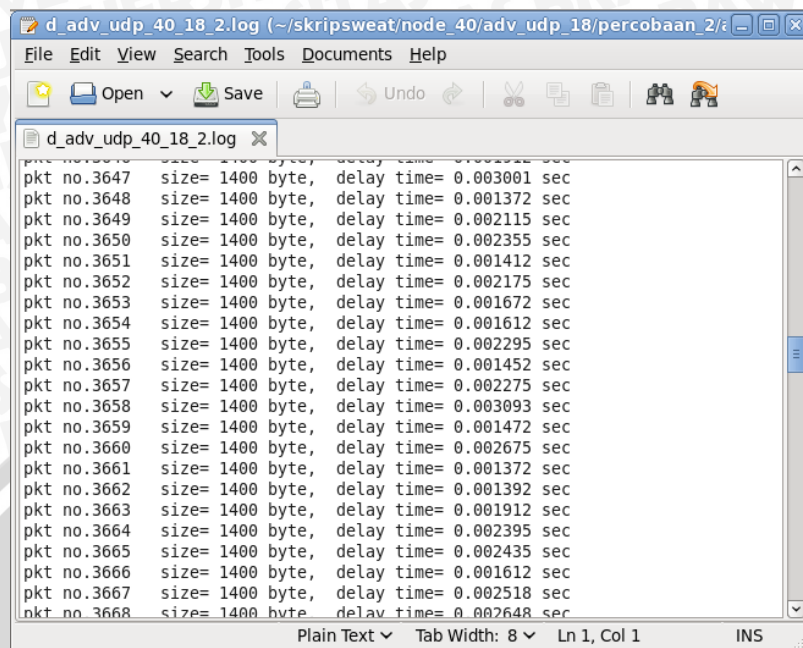
Pada tahapan ini akan dilakukan implementasi dari masing-masing skenario. Implementasi dilakukan berdasarkan tiga skenario yaitu skenario *urban* dengan kecepatan *node* 8 m/s, skenario *urban* dengan kecepatan *node* 18 m/s dan skenario *highway*. implementasi dilakukan sama seperti yang telah dirancang pada bab 3.

5.2 Pengambilan Data

Proses pengambilan data dilakukan dengan mengambil file log yang sudah ditentukan. Pengambilan data dilakukan ketika setelah melakukan proses simulasi. Dari hasil pengambilan data yang dilakukan, dapat diketahui nilai *delay*, *throughput*, *packet loss*, dan *packet delivery ratio*.

1. Delay

Untuk menganalisa nilai *delay* terlebih dahulu mengambil file log *delay* yang berisi kinerja per paket selama simulasi seperti digambarkan pada gambar 5.1.



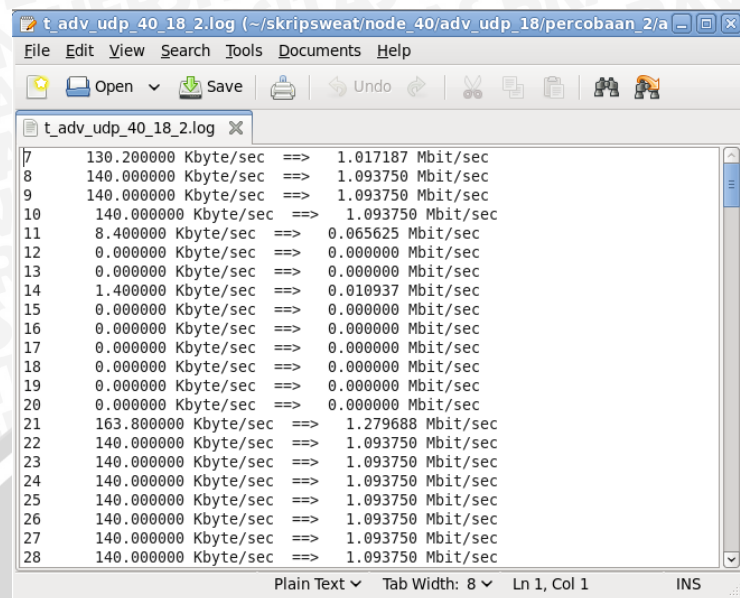
Gambar 5. 1 file log *delay*

Sumber : [Hasil Pengujian]

dari gambar di atas maka didapatkan file log *end to end delay* dari masing-masing paket yaitu waktu yang dibutuhkan suatu paket data dari *node source* menuju *node destination* selama waktu simulasi 200 detik. Kemudian dari file log tersebut diolah ke dalam program microsoft excel dan dicari rata-rata *end to end delay*.

2. *Throughput*

Untuk menganalisa *throughput* terlebih dahulu mengambil file log *throughput* yang berisi *throughput* per detik seperti pada gambar 5.2.



Line	Throughput (Kbyte/sec)	Throughput (Mbit/sec)
7	130.200000	1.017187
8	140.000000	1.093750
9	140.000000	1.093750
10	140.000000	1.093750
11	8.400000	0.065625
12	0.000000	0.000000
13	0.000000	0.000000
14	1.400000	0.010937
15	0.000000	0.000000
16	0.000000	0.000000
17	0.000000	0.000000
18	0.000000	0.000000
19	0.000000	0.000000
20	0.000000	0.000000
21	163.800000	1.279688
22	140.000000	1.093750
23	140.000000	1.093750
24	140.000000	1.093750
25	140.000000	1.093750
26	140.000000	1.093750
27	140.000000	1.093750
28	140.000000	1.093750

Gambar 5. 2file log throughput

Sumber : [Hasil Pengujian]

Dari gambar di atas maka didapatkan *throughput* per detik. Kemudian file log tersebut diolah dengan menggunakan program microsoft excel sehingga didapatkan *throughput* rata-rata per 200 detik.

3. Packet loss

Untuk menganalisa *packet loss* terlebih dahulu mengambil file log yang sama dengan file log *delay* yang berisi jumlah *packet loss* seperti pada gambar 5.3.

```

d_adv_udp_40_18_2.log
File Edit View Search Tools Documents Help
Open Save Undo
d_adv_udp_40_18_2.log
pkt no.997 size= 1400 byte, delay time= 0.001892 sec
pkt no.998 size= 1400 byte, delay time= 0.001792 sec
pkt no.999 size= 1400 byte, delay time= 0.001412 sec
pkt no.1000 size= 1400 byte, delay time= 0.001412 sec
pkt no.1001 size= 1400 byte, delay time= 0.001572 sec
pkt no.1002 size= 1400 byte, delay time= 0.005738 sec
pkt no.1003 size= 1400 byte, delay time= 0.001472 sec
pkt no.1004 size= 1400 byte, delay time= 0.002648 sec
pkt no.1005 size= 1400 byte, delay time= 0.001572 sec
pkt no.1006 size= 1400 byte, delay time= 0.001652 sec
packet no.1007 to no.1069 lost, accumulate 723 pkts lost!!
pkt no.1070 size= 1400 byte, delay time= 2.879177 sec
packet no.1071 to no.2040 lost, accumulate 1693 pkts lost!!
pkt no.2041 size= 1400 byte, delay time= 0.013158 sec
pkt no.1731 out of order
pkt no.1731 size= 1400 byte, delay time= 3.124984 sec
pkt no.2037 out of order
pkt no.2037 size= 1400 byte, delay time= 0.086097 sec
pkt no.2038 out of order
pkt no.2038 size= 1400 byte, delay time= 0.077674 sec
pkt no.2039 out of order
pkt no.2039 size= 1400 byte, delay time= 0.069470 sec
Plain Text Tab Width: 8 Ln 454, Col 38 INS

```

Gambar 5. 3 file log *packet loss*

Sumber : [Hasil Pengujian]

Dari gambar di atas maka didapatkan berapa jumlah total *packet loss* selama simulasi.

4. *Packet delivery ratio*

Untuk menganalisa *packet delivery ratio* menggunakan file log *delay*. Perhitungannya adalah jumlah paket yang dikirimkan dikurangi dengan paket lost.

5.3 Hasil Pengolahan Data

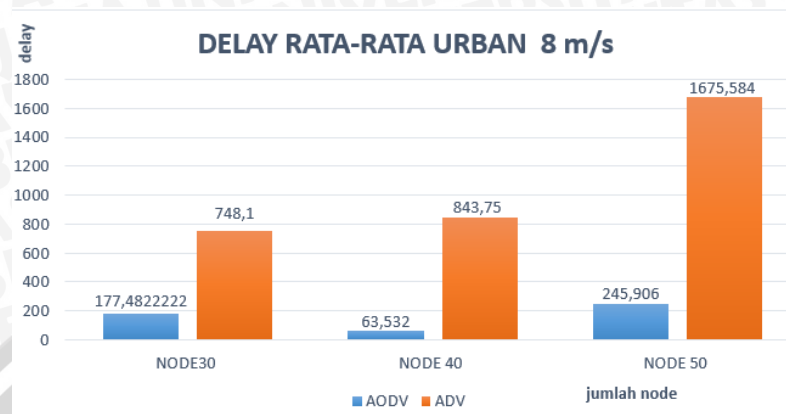
Setelah dilakukan pengambilan data maka dilakukan pengolahan data yang meliputi *delay*, *throughput*, *packet loss*, dan *packet delivery ratio*.

5.3.1 *Delay*

Dari pengolahan data yang dilakukan maka diperoleh nilai *delay* yang dibutuhkan dalam komunikasi VANET dengan menggunakan protokol *routing* AODV dan ADV dalam skenario *urban* dan *highway*. Untuk tabel pengolahan data *packet delivery ratio* dapat dilihat pada lampiran 1.

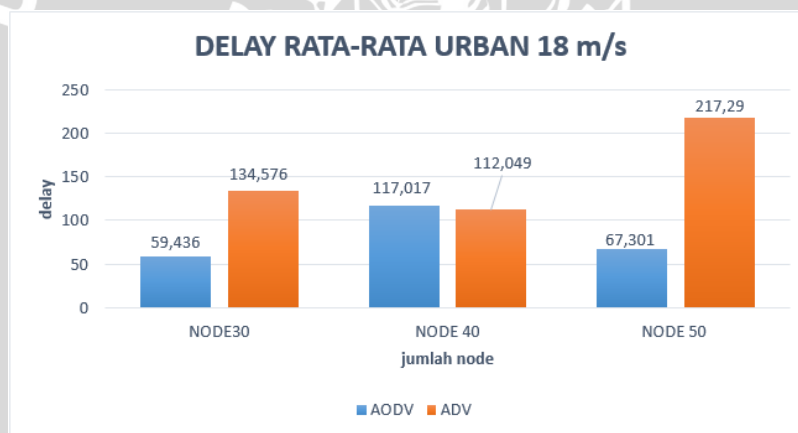
5.3.1.1 Delay Urban

Untuk *delay* pada skenario *urban* dapat dilihat pada gambar 5.4 dan 5.5.



Gambar 5. 4 *delay* rata-rata urban 8 m/s

Sumber : [Hasil Pengujian]



Gambar 5. 5 *delay* rata-rata urban 18 m/s

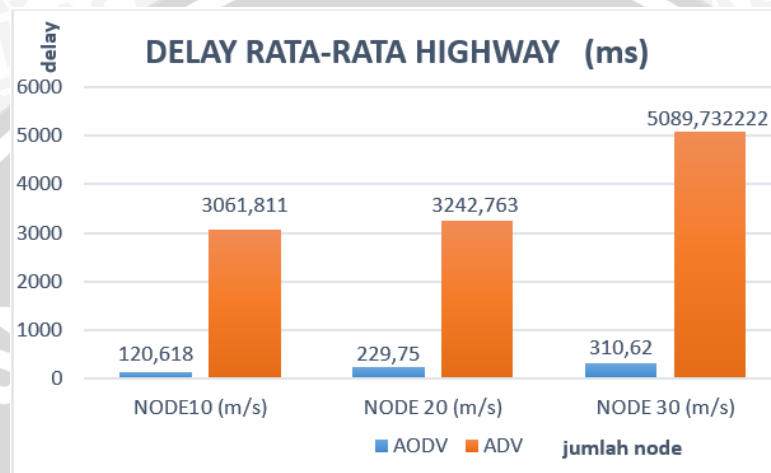
Sumber : [Hasil Pengujian]

Dari gambar 5.4 dan 5.5 di atas didapatkan bahwa pada lingkungan *urban* dengan kecepatan *node* 8 m/s, AODV memiliki *delay* rata-rata yang lebih kecil daripada ADV. AODV memiliki *delay* rata-rata paling kecil pada *node* 40 yaitu 843,75 ms dan ADV memiliki *delay* rata-rata paling kecil pada *node* 30 yaitu 748,1 ms. Sedangkan pada kecepatan *node* 18 m/s didapatkan bahwa AODV juga memiliki *delay* rata-rata yang lebih kecil daripada ADV. AODV memiliki *delay* paling kecil

pada *node* 30 yaitu 59,436 ms dan ADV memiliki *delay* paling kecil pada *node* 40 yaitu 112,049 ms.

5.3.1.2 Delay Highway

Untuk *delay* pada skenario *highway* dapat dilihat pada gambar 5.6.



Gambar 5. 6 *delay* rata-rata *highway*

Sumber : [Hasil Pengujian]

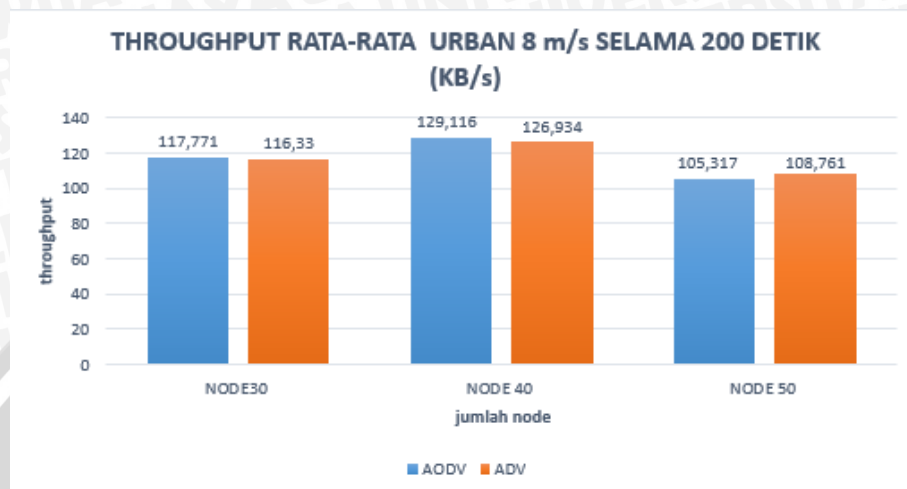
Pada lingkungan *highway*, AODV memiliki *delay* rata-rata yang lebih kecil daripada ADV. Semakin bertambahnya jumlah *node* maka semakin bertambahnya *delay* rata-rata. Pada ADV pada *node* 20 mengalami kenaikan *delay* rata-rata dari 3061,811 ms menjadi 3242,763 ms dan mengalami kenaikan lagi pada *node* 30 menjadi 5089,732 ms. Sama halnya dengan ADV, pada AODV *node* 20 mengalami kenaikan *delay* rata-rata dari 229, 75 ms menjadi 120, 618 ms dan mengalami kenaikan lagi pada *node* 30 menjadi 310,62 ms.

5.3.2 Throughput

Dari pengolahan data yang dilakukan diperoleh nilai *throughput* rata-rata yang dibutuhkan dalam komunikasi VANET dengan menggunakan protokol AODV dan ADV pada skenario *urban* dan *highway*. Untuk tabel pengolahan data *throughput* dapat dilihat pada lampiran 2.

5.3.2.1 Throughput Urban

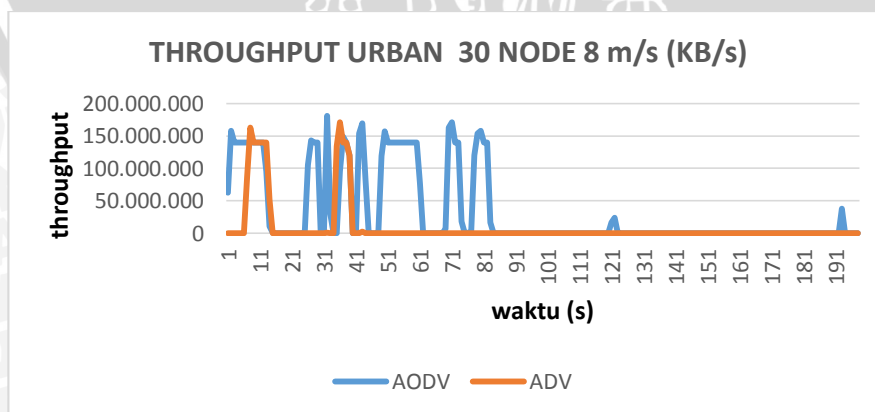
Untuk *throughput* pada skenario urban 8 m/s dapat dilihat pada gambar 5.7.



Gambar 5. 7 *throughput* rata-rata urban 8m/s

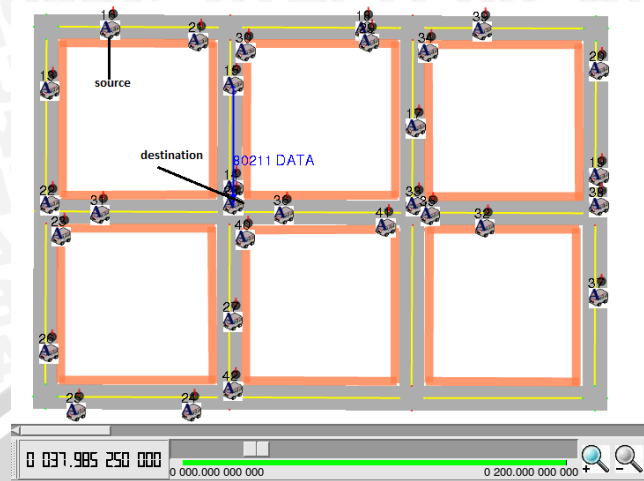
Sumber : [Hasil Pengujian]

Dari gambar 5.7 di atas di dapatkan bahwa pada kecepatan 8 m/s protokol *routing* AODV memiliki *throughput* rata-rata yang hampir sama dengan protokol *routing* ADV. AODV memiliki *throughput* rata-rata paling tinggi pada *node* 40 yaitu 129,116 KB/s sedangkan ADV memiliki *throughput* rata-rata paling tinggi juga pada *node* 40 yaitu 126,934 KB/s.



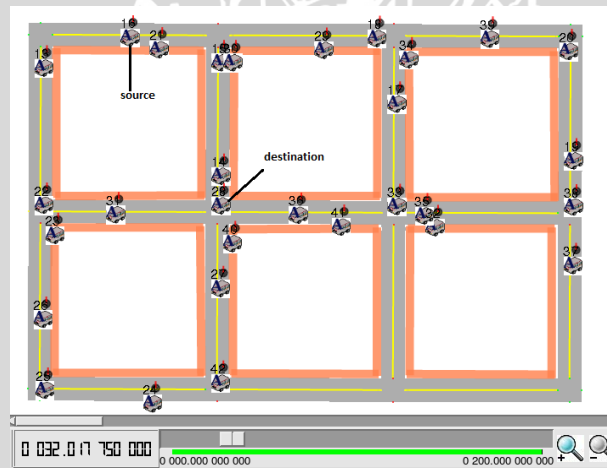
Gambar 5. 8 *throughput* urban 30 node 8 m/s

Sumber : [Hasil Pengujian]



Gambar 5. 9 *throughput urban tertinggi ADV 30 node 8 m/s*

Sumber : [Hasil Pengujian]

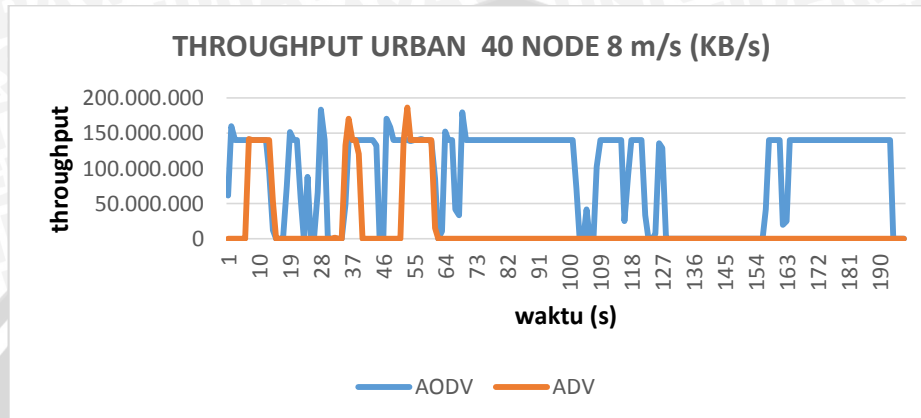


Gambar 5. 10 *throughput urban tertinggi AODV 30 node 8 m/s*

Sumber : [Hasil Pengujian]

Pada gambar 5.8, 5.9 dan 5.10 di atas di dapatkan bahwa pada *node 30 throughput* tertinggi pada AODV didapatkan pada detik ke-32 yaitu 180,6 KB/s. Sedangkan pada ADV *throughput* tertinggi didapatkan pada detik ke 37 yaitu 170,8 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node 16* mengirimkan paket data kepada *node 30*, kemudian *node 30* mengirimkan paket data kepada *node 15*, dan *node 15* mengirimkan paket data kepada *node 28* yang

merupakan *destination node*. Sedangkan *throughput* tertinggi AODV didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 15 dan *node* 15 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*.



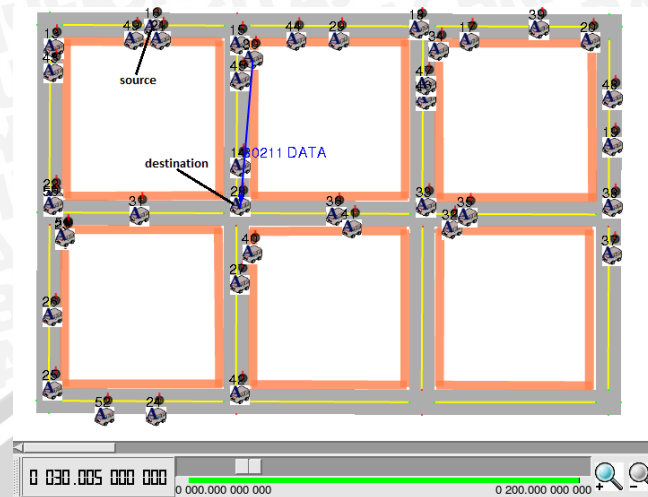
Gambar 5. 11 throughput urban 40 node 8 m/s

Sumber : [Hasil Pengujian]



Gambar 5. 12 throughput urban tertinggi ADV 40 node 8 m/s

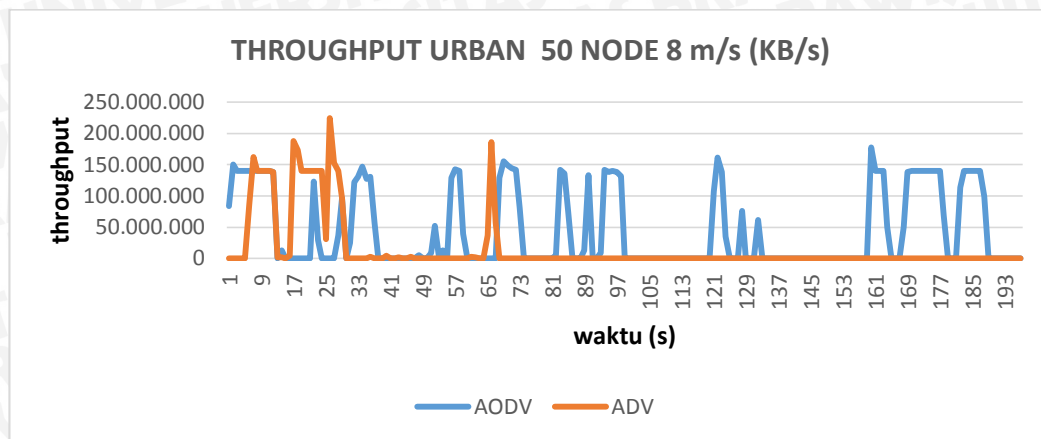
Sumber : [Hasil Pengujian]



Gambar 5. 13 *throughput urban tertinggi AODV 40 node 8 m/s*

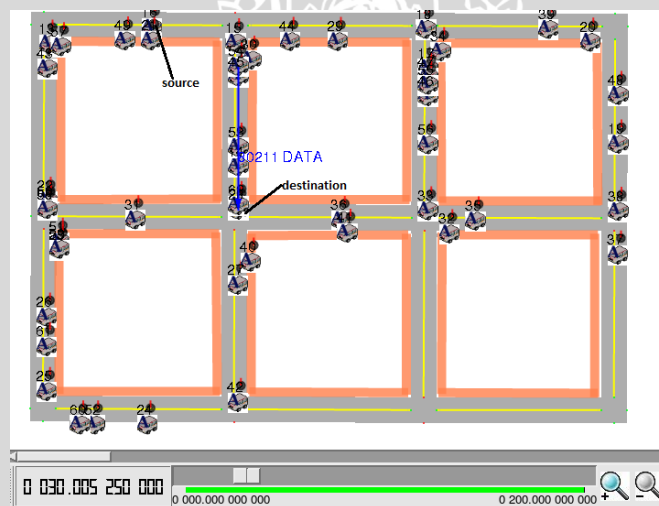
Sumber : [Hasil Pengujian]

Pada gambar 5.11, 5.12 dan 5.13 di atas di dapatkan bahwa pada *node 40 throughput* tertinggi pada AODV didapatkan pada detik ke-55 yaitu 186,2 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-30 yaitu 183,4 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node 16* mengirimkan paket data kepada *node 13*, kemudian *node 13* mengirimkan paket data kepada *node 25*, kemudian *node 25* mengirimkan paket data kepada *node 40*, kemudian *node 40* mengirimkan data kepada *node 28* yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node 16* mengirimkan paket data kepada *node 30*, kemudian *node 30* mengirimkan paket data kepada *node 28* yang merupakan *destination node*.



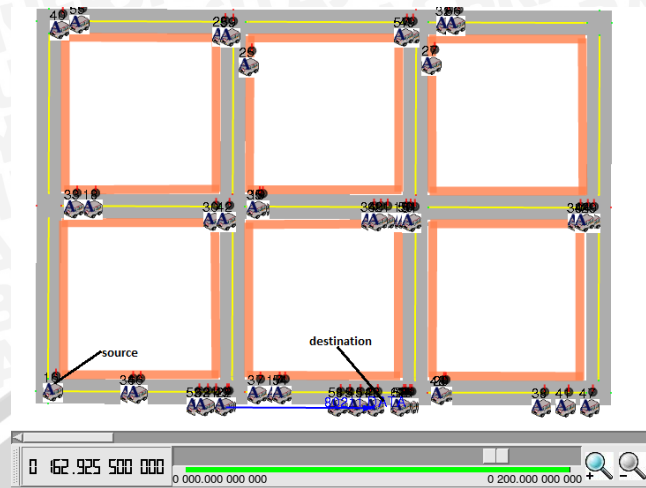
Gambar 5. 14 throughput urban 50 node 8 m/s

Sumber : [Hasil Pengujian]



Gambar 5. 15 throughput urban tertinggi ADV 50 node 8 m/s

Sumber : [Hasil Pengujian]

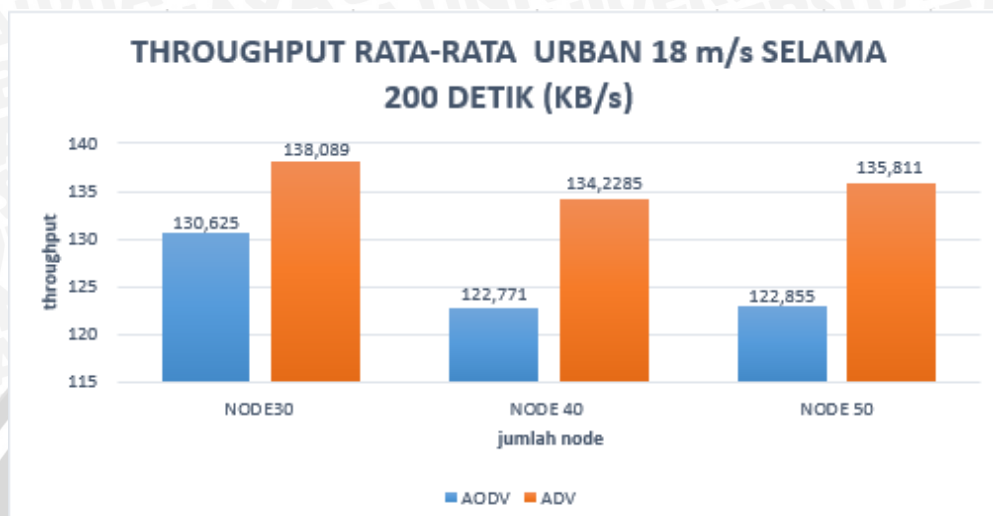


Gambar 5. 16 *throughput urban tertinggi AODV 50 node 8 m/s*

Sumber : [Hasil Pengujian]

Pada gambar 5.14, 5.15 dan 5.16 di atas di dapatkan bahwa pada *node* 50 ADV memiliki *throughput* maksimal yang lebih tinggi daripada AODV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-30 yaitu 224 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-162 yaitu 177.8 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 15, kemudian *node* 15 mengirimkan paket data kepada *node* 54, kemudian *node* 54 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 46, kemudian *node* 46 mengirimkan paket data kepada *node* 22, kemudian *node* 22 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*.

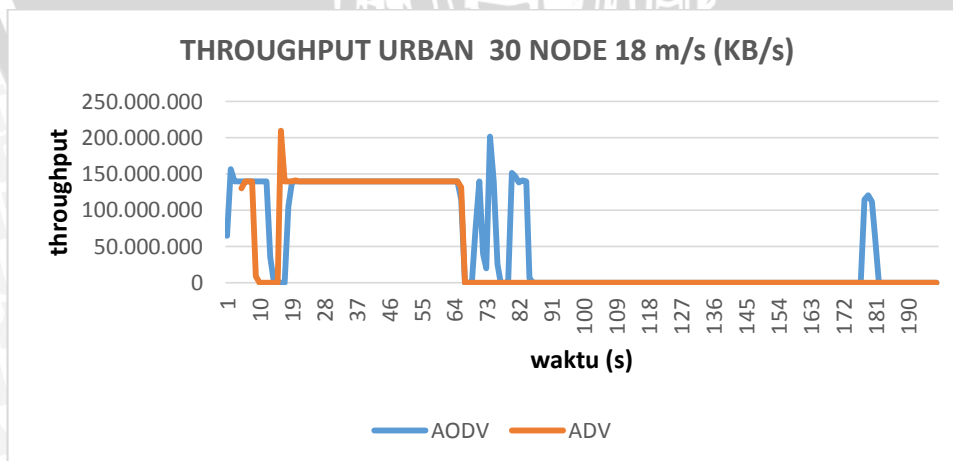
Untuk *throughput* pada skenario urban 18 m/s dapat dilihat pada gambar 5.17.



Gambar 5. 17 *throughput* rata-rata urban 18m/s

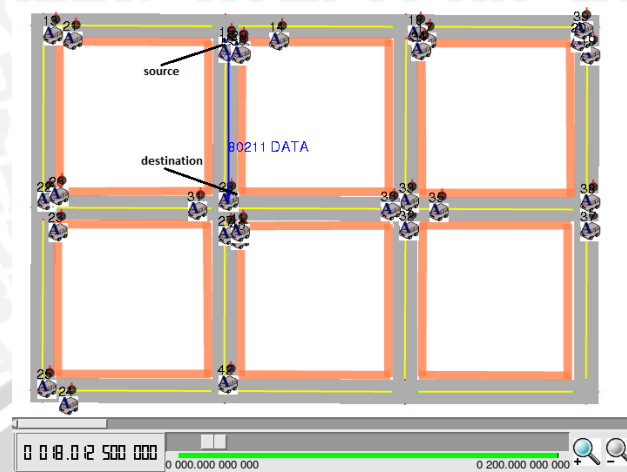
Sumber : [Hasil Pengujian]

Dari gambar 5.17 di atas di dapatkan bahwa pada kecepatan 18 m/s AODV memiliki *throughput* rata-rata paling tinggi pada node 30 yaitu 130,625 KB/s. Sedangkan pada ADV memiliki *throughput* rata-rata paling tinggi juga pada node 30 yaitu 138,089 KB/s.



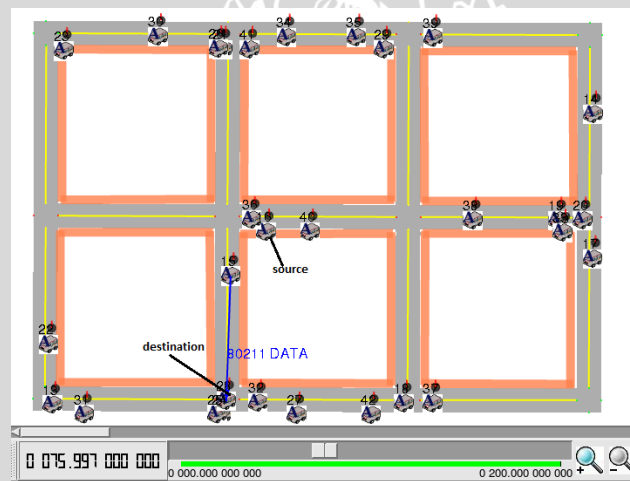
Gambar 5. 18 *throughput* urban 30 node 18 m/s

Sumber : [Hasil Pengujian]



Gambar 5. 19 *throughput* tertinggi ADV 30 node 18 m/s

Sumber : [Hasil Pengujian]

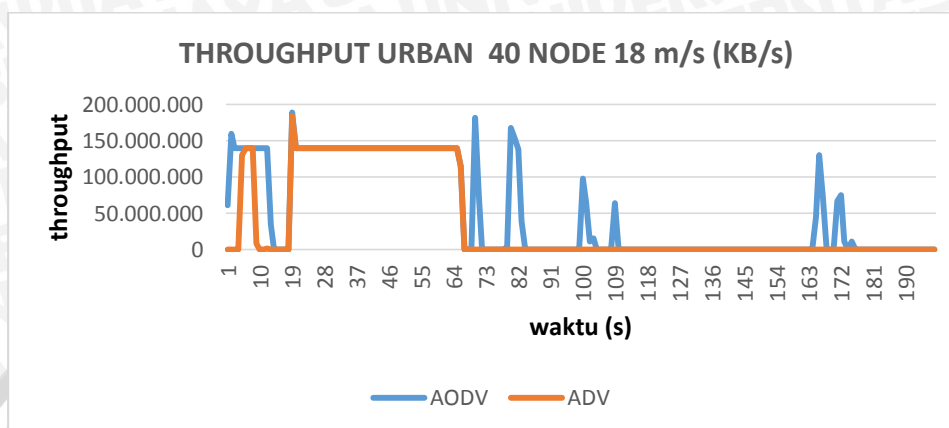


Gambar 5. 20 *throughput* urban tertinggi AODV 30 node 18 m/s

Sumber : [Hasil Pengujian]

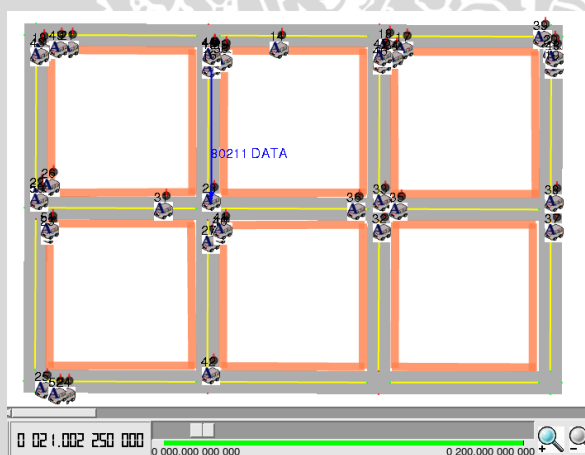
Pada gambar 5.18, 5.19 dan 5.20 di atas di dapatkan bahwa pada *node* 30 ADV memiliki *throughput* maksimal yang lebih tinggi daripada AODV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-18 yaitu 210 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-76 yaitu 201,6 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 36, kemudian *node* 36 mengirimkan paket data kepada *node* 15,

kemudian *node* 15 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*.



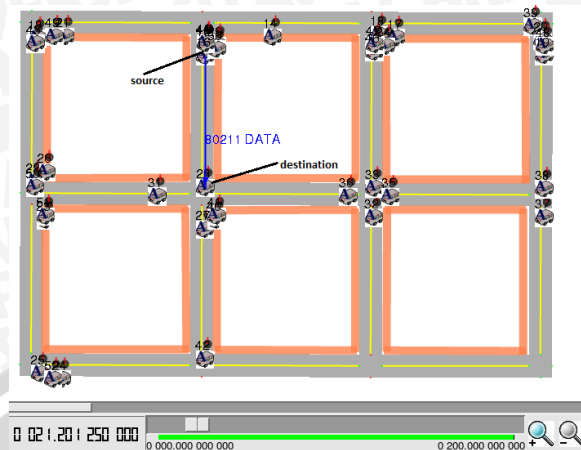
Gambar 5. 21 throughput urban 40 node 18 m/s

Sumber : [Hasil Pengujian]



Gambar 5. 22 throughput urban tertinggi ADV 40 node 18 m/s

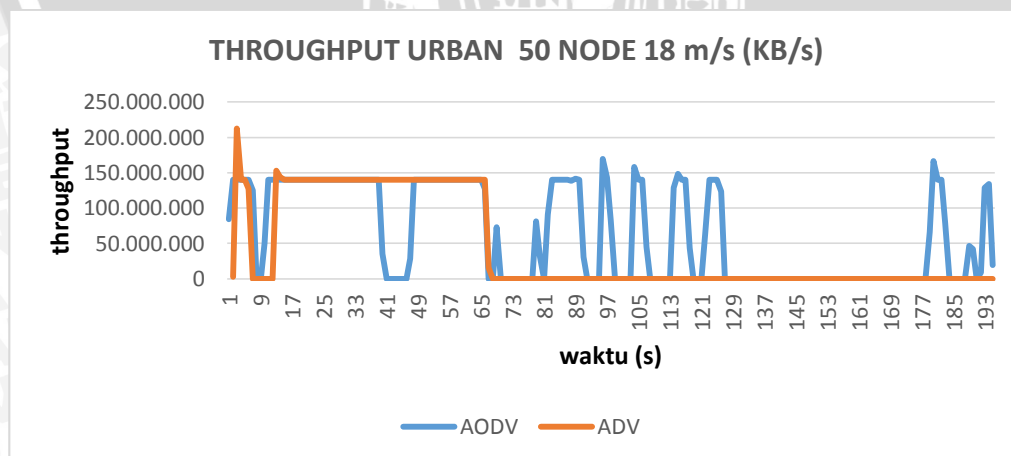
Sumber : [Hasil Pengujian]



Gambar 5. 23 *throughput urban tertinggi AODV 40 node 18 m/s*

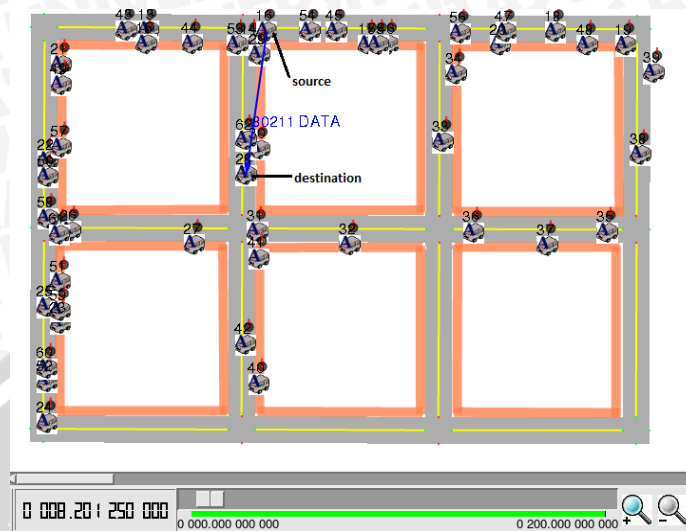
Sumber : [Hasil Pengujian]

Pada gambar 5.21, 5.22 dan 5.23 di atas di dapatkan bahwa pada *node 40* AODV memiliki *throughput* maksimal yang lebih tinggi daripada ADV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-21 yaitu 183,8 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-21 yaitu 189 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node 16* mengirimkan paket data kepada *node 28* yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node 16* mengirimkan paket data kepada *node 28* yang merupakan *destination node*.



Gambar 5. 24 *throughput urban 50 node 18 m/s*

Sumber : [Hasil Pengujian]



Gambar 5. 25 *throughput urban tertinggi ADV 50 node 18 m/s*

Sumber : [Hasil Pengujian]



Gambar 5. 26 *throughput urban tertinggi AODV 50 node 18 m/s*

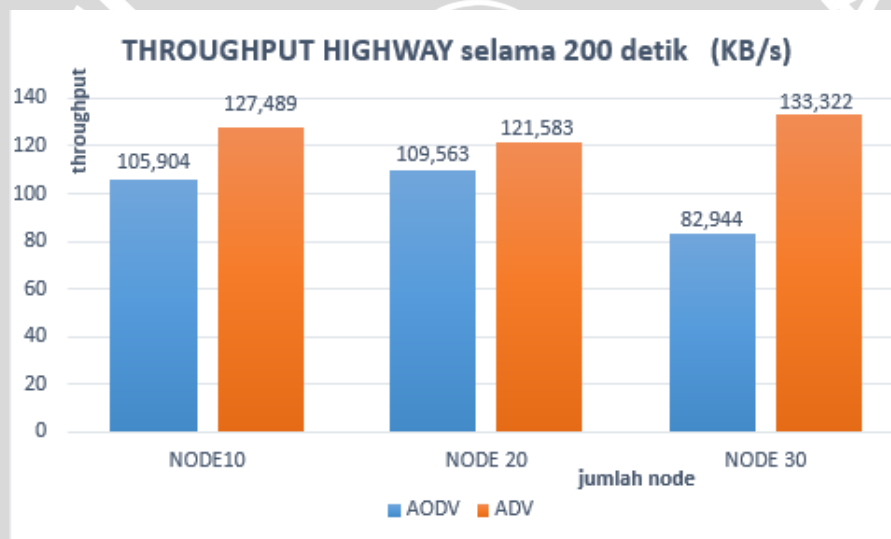
Sumber : [Hasil Pengujian]

Pada gambar 5.24, 5.25 dan 5.26 di atas di dapatkan bahwa pada *node 50* ADV memiliki *throughput* maksimal yang lebih tinggi daripada AODV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-8 yaitu 212,8 KB/s. Sedangkan pada

AODV *throughput* tertinggi didapatkan pada detik ke-101 yaitu 169,4 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node* 16 mengirimkan paket data kepada *node* 55, kemudian *node* 55 mengirimkan paket data kepada *node* 15, kemudian *node* 15 mengirimkan paket data kepada *node* 43, kemudian *node* 43 mengirimkan paket data kepada *node* 28 yang merupakan *destination node*.

5.3.2.2 Throughput Highway

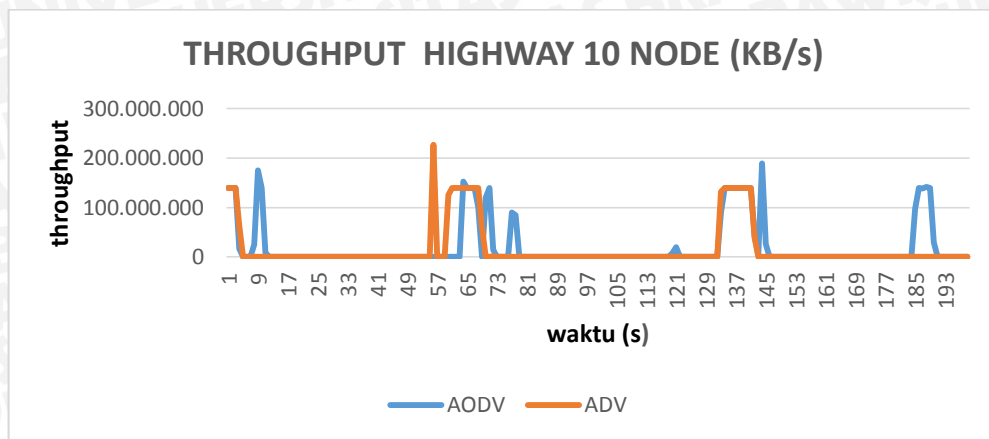
Untuk *throughput* pada skenario *highway* dapat dilihat pada gambar 5.27.



Gambar 5. 27 *throughput* rata-rata *highway*

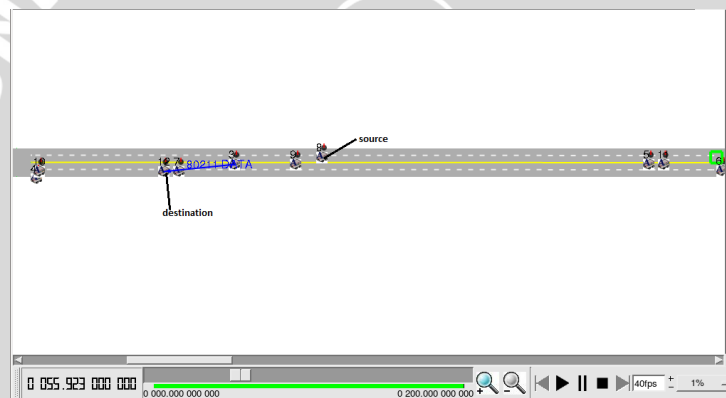
Sumber : [Hasil Pengujian]

Dari gambar 5.27 di atas di dapatkan bahwa pada lingkungan *highway*, AODV memiliki *throughput* rata-rata paling tinggi pada *node* 20 yaitu 109,563 KB/s. Sedangkan pada ADV memiliki *throughput* rata-rata paling tinggi pada *node* 30 yaitu 133,322 KB/s.



Gambar 5. 28 throughput highway 10 node

Sumber : [Hasil Pengujian]



Gambar 5. 29 throughput highway tertinggi ADV 10 node

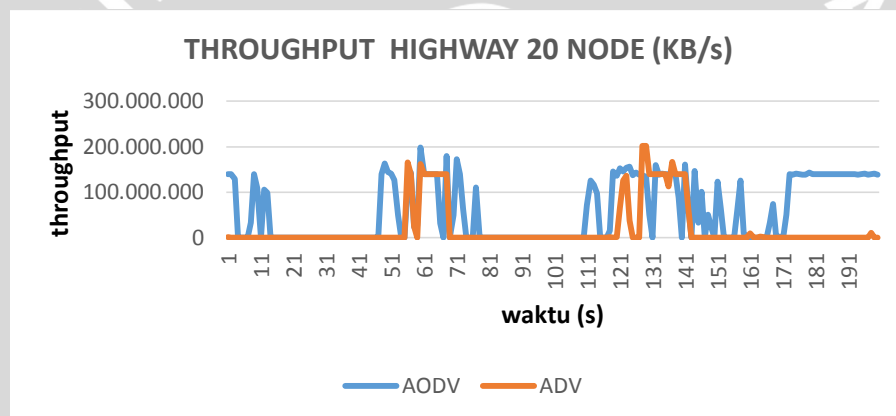
Sumber : [Hasil Pengujian]



Gambar 5. 30 throughput highway tertinggi AODV 10 node

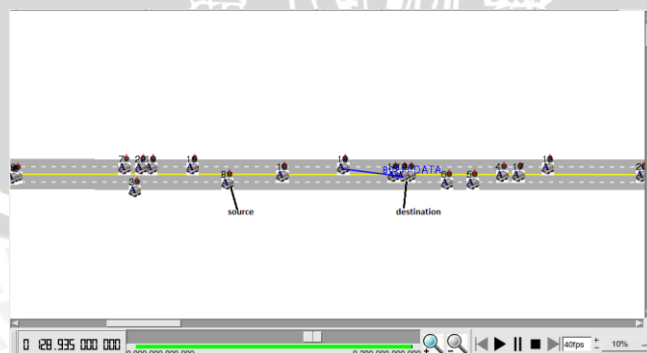
Sumber : [Hasil Pengujian]

Pada gambar 5.28, 5.29 dan 5.30 di atas di dapatkan bahwa pada *node* 10 ADV memiliki *throughput* maksimal yang lebih tinggi daripada AODV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-56 yaitu 226,8 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-144 yaitu 189,0 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node* 8 mengirimkan paket data kepada *node* 3, kemudian *node* 3 mengirimkan paket data ke *node* 12 yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node* 8 mengirimkan paket data kepada *node* 4, kemudian *node* 4 mengirimkan paket data kepada *node* 12 yang merupakan *destination node*.



Gambar 5. 31 *throughput highway 20 node*

Sumber : [Hasil Pengujian]



Gambar 5. 32 *throughput highway tertinggi ADV 20 node*

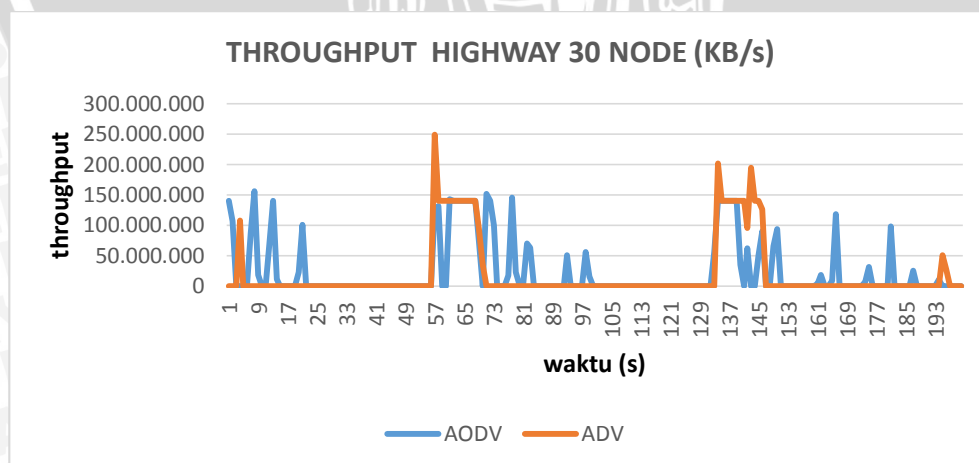
Sumber : [Hasil Pengujian]



Gambar 5. 33 *throughput highway tertinggi AODV 20 node*

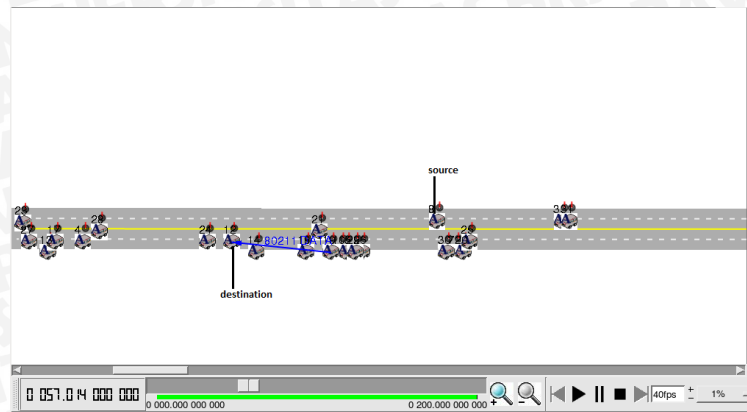
Sumber : [Hasil Pengujian]

Pada gambar 5.31, 5.32 dan 5.33 di atas di dapatkan bahwa pada *node 20* ADV memiliki *throughput* maksimal yang lebih tinggi daripada AODV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-129 yaitu 201,6 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-60 yaitu 198,8 KB/s. *throughput* tertinggi ADV didapatkan pada saat *source node 8* mengirimkan paket data kepada *node 10*, kemudian *node 10* mengirimkan paket data ke *node 18*, kemudian *node 18* mengirimkan paket data kepada *node 12* yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node 8* mengirimkan paket data kepada *node 18*, kemudian *node 18* mengirimkan paket data kepada *node 12* yang merupakan *destination node*.



Gambar 5. 34 *throughput highway 30 node*

Sumber : [Hasil Pengujian]



Gambar 5. 35 *throughput highway tertinggi ADV 30 node*

Sumber : [Hasil Pengujian]



Gambar 5. 36 *throughput highway tertinggi AODV 30 node*

Sumber : [Hasil Pengujian]

Pada gambar 5.34, 5.35 dan 5.36 di atas di dapatkan bahwa pada *node 50* ADV memiliki *throughput* maksimal yang lebih tinggi daripada AODV. Pada ADV *throughput* tertinggi didapatkan pada detik ke-57 yaitu 249,2 KB/s. Sedangkan pada AODV *throughput* tertinggi didapatkan pada detik ke-8 yaitu 155,4 KB/s. *Throughput* tertinggi ADV didapatkan pada saat *source node 8* mengirimkan paket data kepada *node 19*, kemudian *node 19* mengirimkan paket data kepada *node 12* yang merupakan *destination node*. Sedangkan pada AODV *throughput* tertinggi didapatkan pada saat *source node 8* mengirimkan paket data kepada *node 31*, kemudian *node 31* mengirimkan paket data kepada *node 17*, kemudian *node 17* mengirimkan paket data kepada *node 9*, kemudian *node 9* mengirimkan paket data

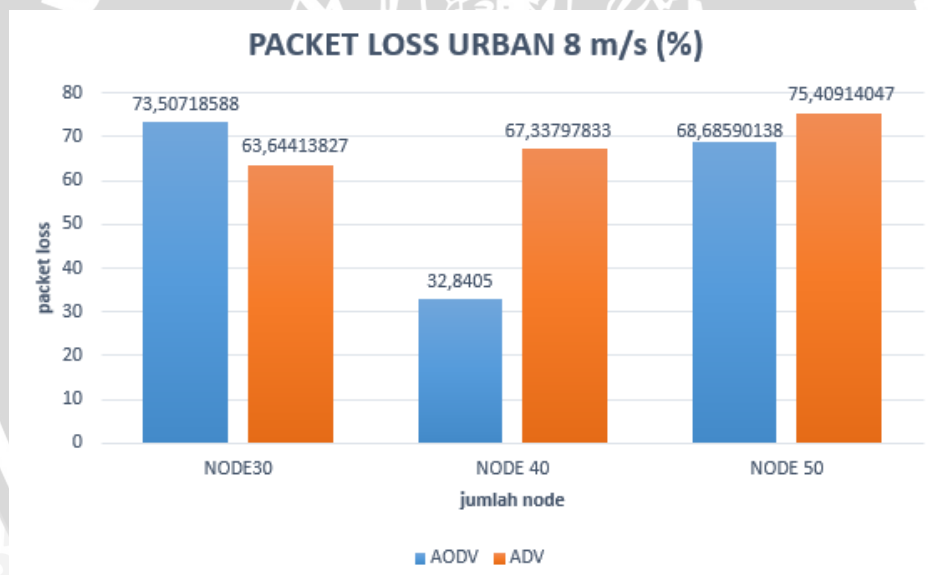
kepada *node* 32, kemudian *node* 32 mengirimkan paket data kepada *node* 12 yang merupakan *destination*.

5.3.3 Packet Loss

Dari pengolahan data yang dilakukan diperoleh nilai *packet loss* yang dibutuhkan dalam komunikasi VANET dengan menggunakan protokol *routing* AODV dan ADV dalam skenario *urban* dan *highway*. Untuk tabel pengolahan data *packet loss* dapat dilihat pada lampiran 3.

5.3.3.1 Packet loss Urban

Untuk *packet loss* pada skenario *urban* 8 m/s dapat dilihat pada gambar 5.37.

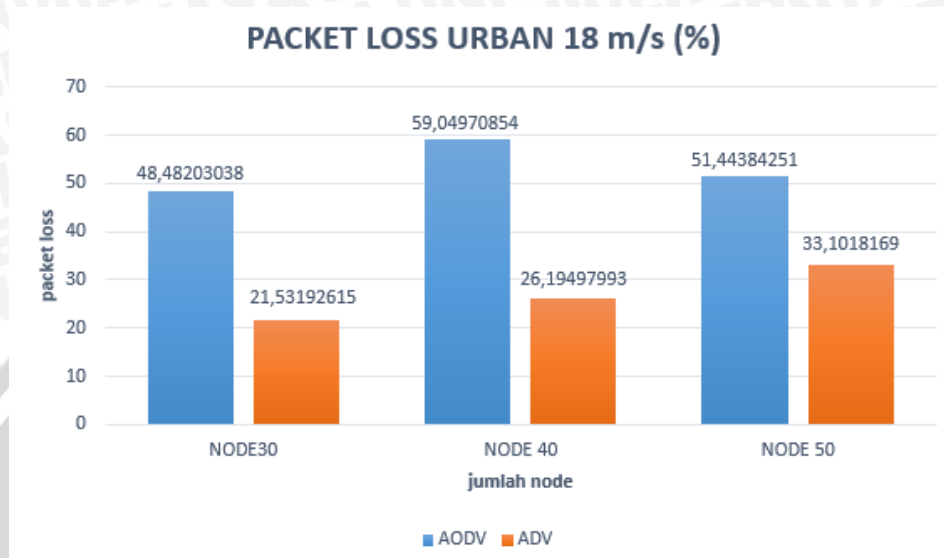


Gambar 5. 37 packet loss urban 8 m/s

Sumber : [Hasil Pengujian]

pada gambar 5.37 dapat dilihat bahwa pada lingkungan *urban* dengan kecepatan *node* 8 m/s, AODV memiliki *packet loss* yang paling kecil pada *node* 40 yaitu 32,8405%. Sedangkan pada ADV memiliki *packet loss* yang paling kecil pada *node* 30 yaitu 63,64413827%.

Untuk *packet loss* pada skenario *urban* 18 m/s dapat dilihat pada gambar 5.38.



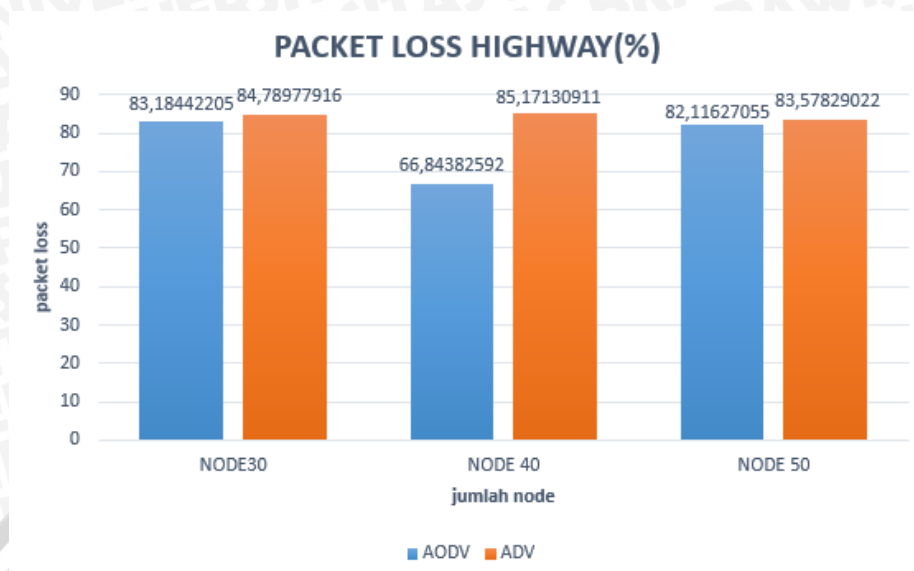
Gambar 5. 38 *packet loss urban* 18 m/s

Sumber : [Hasil Pengujian]

pada gambar 5.38 dapat dilihat bahwa pada lingkungan *urban* dengan kecepatan *node* 18 m/s, AODV memiliki *packet loss* paling kecil pada *node* 30 yaitu 48,482-3038%. Sedangkan pada ADV memiliki *packet loss* paling kecil pada *node* 50 yaitu 33,1018169%.

5.3.3.2 Packet loss Highway

Untuk *packet loss* pada skenario *highway* dapat dilihat pada gambar di 5.39.



Gambar 5. 39 *packet loss highway*

Sumber : [Hasil Pengujian]

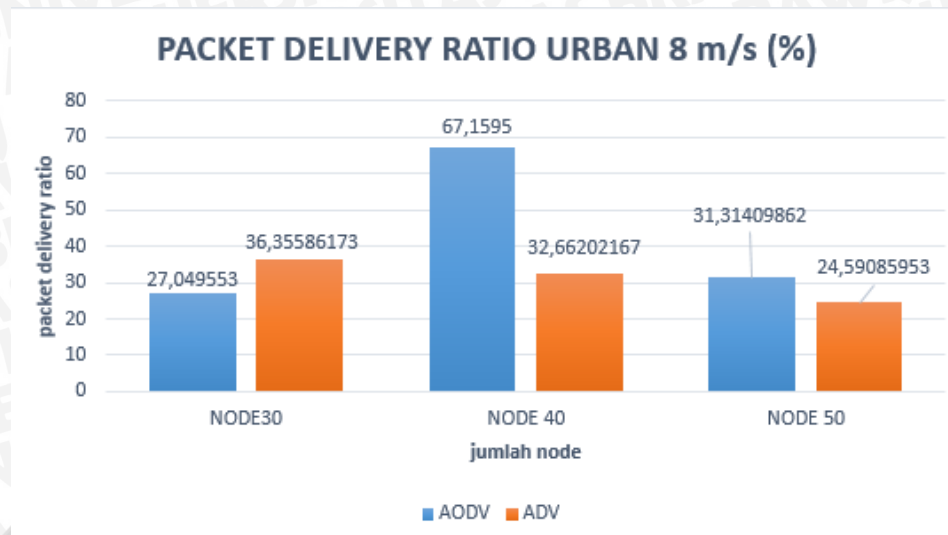
pada gambar 5.39 dapat dilihat bahwa pada lingkungan *highway*, AODV memiliki *packet loss* paling kecil pada *node* 40 yaitu 66,84382592%. Sedangkan pada ADV memiliki *packet loss* paling kecil pada *node* 50 yaitu 83,5789022%.

5.3.4 *Packet Delivery Ratio*

Dari pengolahan data yang dilakukan diperoleh nilai *packet delivery ratio* yang dibutuhkan dalam komunikasi VANET dengan menggunakan protokol AODV dan ADV dalam skenario *urban* dan *highway*. Untuk tabel pengolahan data *packet delivery ratio* dapat dilihat pada lampiran 4.

5.3.4.1 *Packet Delivery Ratio Urban*

Untuk *packet delivery ratio* pada skenario *urban* 8 m/s dapat dilihat pada gambar 5.40.

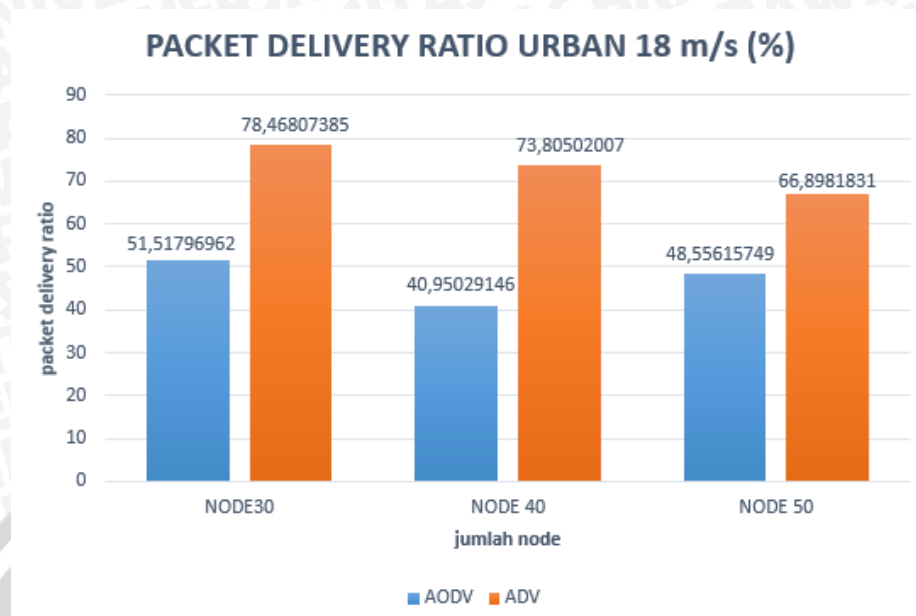


Gambar 5. 40 packet delivery ratio urban 8 m/s

Sumber : [Hasil Pengujian]

pada gambar 5.40 dapat dilihat bahwa pada lingkungan *urban* 8 m/s, AODV memiliki *packet delivery ratio* paling tinggi pada *packet delivery ratio* 40 yaitu 67,1595%. Sedangkan pada ADV memiliki *packet delivery ratio* paling tinggi pada *node* 30 yaitu 36,35586173.

Untuk *packet delivery ratio* pada skenario *urban* 18 m/s dapat dilihat pada gambar 5.41.



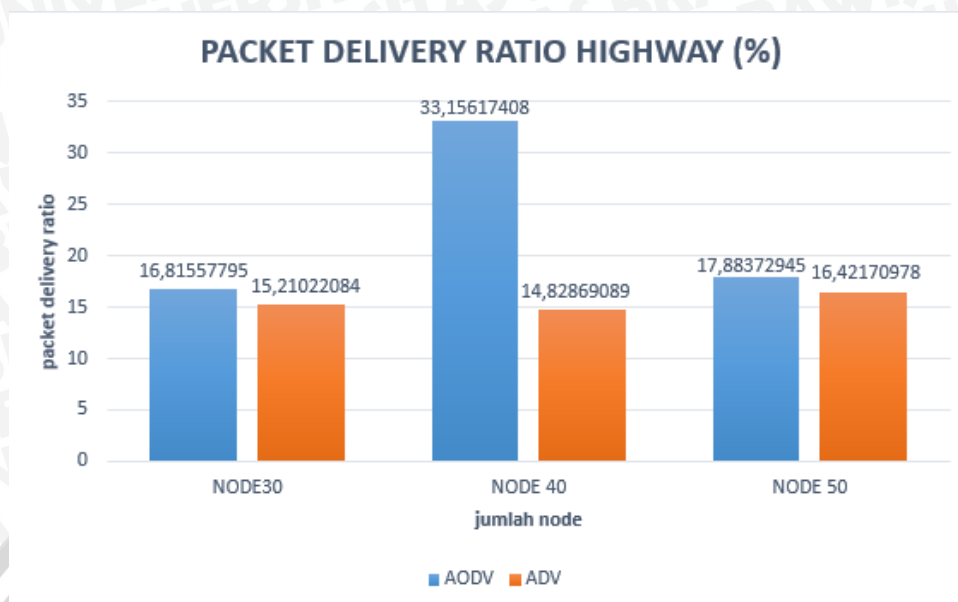
Gambar 5. 41 *packet delivery ratio*urban 18 m/s

Sumber : [Hasil Pengujian]

pada gambar 5.41 dapat dilihat bahwa pada lingkungan *urban* 18 m/s , AODV memiliki *packet delivery ratio* paling tinggi pada *node* 30 yaitu 51,51796962%. Sedangkan pada ADV memiliki *packet delivery ratio* paling tinggi pada *node* 50 yaitu 66,8981831%.

5.3.4.2 Packet Delivery Ratio Highway

Untuk *packet delivery ratio* pada skenario *highway* dapat dilihat pada gambar 5.42.



Gambar 5. 42 *packet delivery ratio highway*

Sumber : [Hasil Pengujian]

pada gambar 5.42 dapat dilihat bahwa pada lingkungan *highway*, AODV memiliki *packet delivery ratio* paling tinggi pada *node* 40 yaitu 33,15617408%. Sedangkan pada ADV memiliki *packet delivery ratio* pada *node* 50 yaitu 16,42170978%.

5.3.4 Uji Statistika

Setelah dilakukan pengolahan data untuk mendapatkan rata-rata dari parameter *delay*, *throughput*, *packet loss* dan *packet delivery ratio*, kemudian dilakukan dua uji statistika yaitu uji *analysis of variance* (ANOVA) dan uji Beda Nyata Terkecil (BNT).

5.3.4.1 Uji *Analysis of Variance* (ANOVA)

Setelah hasil rata-rata didapatkan kemudian data hasil masing-masing percobaan di uji dengan uji anova dengan taraf kepercayaan 5% untuk melihat apakah kecepatan dan jumlah *node* memberikan pengaruh nyata terhadap parameter *delay*, *throughput*, *packet loss* dan *packet delivery ratio*. Hasil dari uji anova digambarkan seperti pada lampiran 5.

Dari hasil uji anova didapatkan bahwa kecepatan dan jumlah *node* memberikan pengaruh nyata terhadap *delay*, *throughput*, *packet loss* dan *PDR*.

5.3.4.2 Uji Beda Nyata Terkecil (BNT)

Uji BNT dilakukan untuk menguji nilai rata-rata untuk mengetahui kecepatan dan jumlah *node* mana yang terbaik. Pada uji BNT lingkungan *urban*, jumlah *node* dan kecepatan *node* akan dikategorikan menjadi 6 kategori sebagai berikut :

1. Kategori 1 untuk kecepatan *node* 8 m/s dan jumlah *node* 30
2. Kategori 2 untuk kecepatan *node* 8 m/s dan jumlah *node* 40
3. Kategori 3 untuk kecepatan *node* 8 m/s dan jumlah *node* 50
4. Kategori 4 untuk kecepatan *node* 18 m/s dan jumlah *node* 30
5. Kategori 5 untuk kecepatan *node* 18 m/s dan jumlah *node* 40
6. Kategori 6 untuk kecepatan *node* 18 m/s dan jumlah *node* 50.

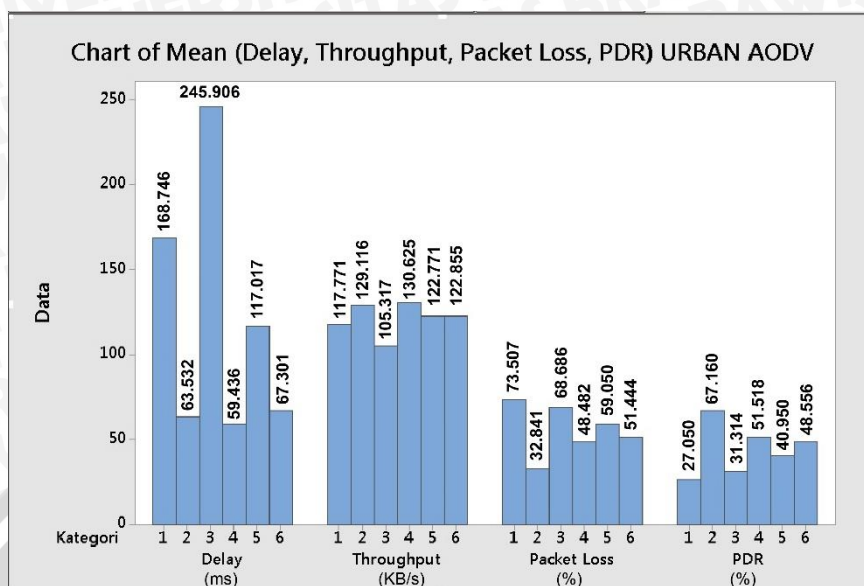
Pada uji BNT lingkungan *highway* jumlah *node* dan kecepatan *node* akan dikategorikan menjadi 3 kategori sebagai berikut :

1. Kategori 1 untuk jumlah *node* 10
2. Kategori 2 untuk jumlah *node* 20
3. Kategori 3 untuk jumlah *node* 30.

Hasil uji BNT digambarkan pada lampiran 6.

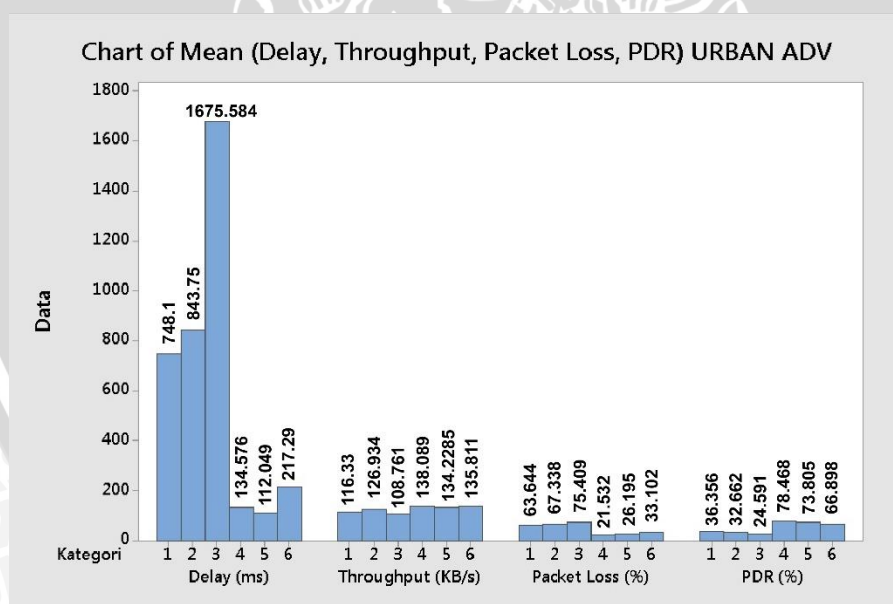
5.4 Analisis VANET

Dari hasil pengujian BNT dari parameter *delay*, *throughput*, *packet loss* dan *packet delivery ratio* maka didapatkan grafik seperti pada gambar 5.43, 5.44, 5.45 dan 5.46.



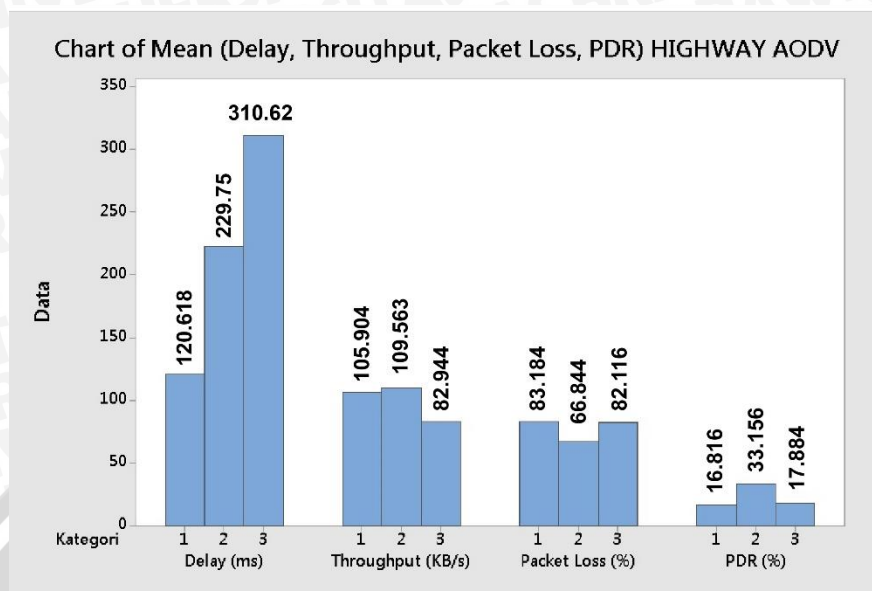
Gambar 5. 43 Grafik *delay, throughput, packet loss* dan PDR AODV pada urban

Sumber : [Hasil Pengujian]



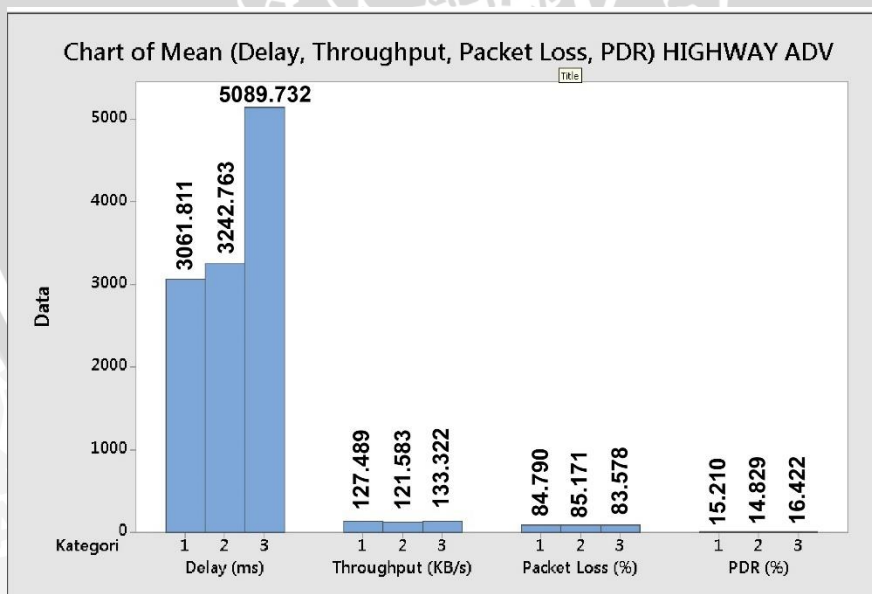
Gambar 5. 44 Grafik *delay, throughput, packet loss* dan PDR ADV pada urban

Sumber : [Hasil Pengujian]



Gambar 5. 45 Grafik delay, throughput, packet loss dan PDR AODV pada highway

Sumber : [Hasil Pengujian]



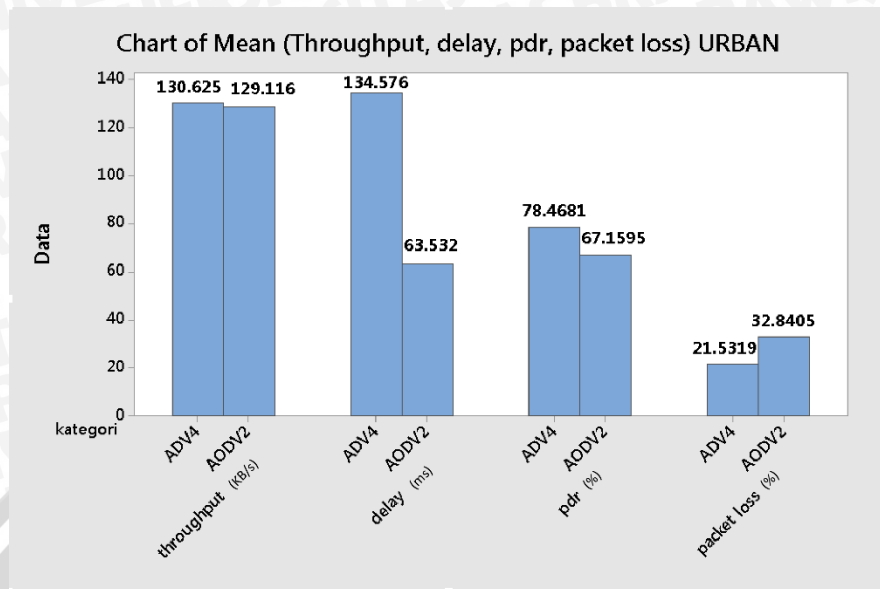
Gambar 5. 46 Grafik delay, throughput, packet loss dan PDR ADV pada highway

Sumber : [Hasil Pengujian]

Dari gambar 5.43, 5.44, 5.45 dan 5.46 dapat diketahui bahwa pada AODV untuk lingkungan *urban* dan *highway*, perbedaan jumlah *node* memberikan pengaruh yang signifikan untuk masing-masing parameter. Hal ini dapat dilihat dengan selisih nilai rata-rata yang besar pada masing-masing jumlah *node* untuk semua parameter. Sama halnya dengan perbedaan kecepatan *node*, AODV memberikan pengaruh yang signifikan untuk masing-masing parameter.

Pada ADV untuk masing-masing lingkungan *urban* dan *highway*, perbedaan jumlah *node* memberikan pengaruh yang sangat kecil untuk masing-masing parameter walaupun ada beberapa yang memberikan pengaruh yang tinggi pada *delay*. Sedangkan pada perbedaan kecepatan *node*, ADV memberikan pengaruh yang signifikan untuk masing-masing parameter.

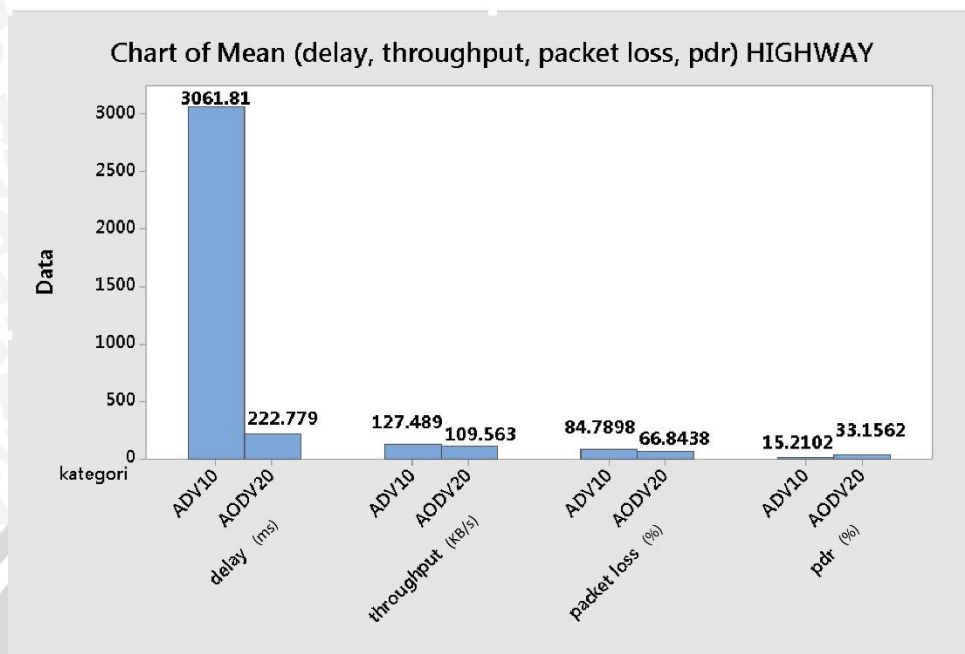
Dari semua parameter *delay*, *throughput* dan *packet loss* didapatkan bahwa pada lingkungan *urban*, protokol *routing* AODV memiliki kinerja paling baik pada jumlah *node* 40 dan kecepatan *node* 8 m/s. Sedangkan pada protokol *routing* ADV memiliki kinerja paling optimal pada jumlah *node* 30 dan kecepatan *node* 18 m/s. Untuk lingkungan *highway*, protokol *routing* AODV memiliki kinerja paling baik pada jumlah *node* 20. Begitu pula pada protokol *routing* ADV memiliki kinerja paling baik pada jumlah *node* 10.



Gambar 5. 47 Perbandingan kinerja AODV dan ADV yang paling baik pada lingkungan urban

Sumber : [Hasil Pengujian]

Pada gambar 5.47 di atas diketahui bahwa untuk *delay* dan *throughput* pada lingkungan urban, AODV jauh lebih baik daripada ADV. Sedangkan pada *packet loss* dan *packet delivery ratio* ADV lebih baik daripada AODV walaupun perbedaannya tidak signifikan sehingga dapat dikatakan protokol routing AODV lebih baik daripada ADV pada lingkungan urban



Gambar 5. 48 Perbandingan kinerja AODV dan ADV yang paling baik pada lingkungan *highway*

Sumber : [Hasil Pengujian]

Pada gambar 5.48 di atas diketahui bahwa untuk *delay*, *packet loss* dan *packet delivery ratio* pada lingkungan *highway*, AODV jauh lebih baik daripada ADV. Sedangkan pada *throughput* ADV lebih baik daripada AODV walaupun perbedaannya tidak signifikan sehingga dapat dikatakan protokol *routing* AODV lebih baik daripada ADV pada lingkungan *highway*.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Pada lingkungan *urban* dan *highway*, AODV dan ADV memiliki pengaruh yang signifikan terhadap perbedaan kecepatan node
2. Pada lingkungan *urban* dan *highway*, AODV tidak memiliki pengaruh signifikan terhadap perbedaan jumlah node tetapi ADV memiliki pengaruh signifikan terhadap perbedaan jumlah node
3. Pada lingkungan *urban*, protokol *routing* AODV memiliki kinerja paling baik pada jumlah *node* 40 dan kecepatan *node* 8 m/s, sedangkan pada protokol *routing* ADV memiliki kinerja paling optimal pada jumlah *node* 30 dan kecepatan *node* 18 m/s terhadap parameter delay, throughput, packet loss dan packet delivery ratio. Pada lingkungan *highway*, protokol *routing* AODV memiliki kinerja paling baik pada jumlah *node* 20, sedangkan pada protokol *routing* ADV memiliki kinerja paling baik pada jumlah *node* 10 terhadap parameter delay, throughput, packet loss dan packet delivery ratio

6.2 Saran

Saran untuk pengembangan sistem VANET selanjutnya antara lain :

1. Untuk pengembangan lebih lanjut dapat diujikan dengan menggunakan paket data streaming supaya dapat diketahui performansi VANET
2. Untuk pengembangan lebih lanjut perlu ditambahkan RSU agar proses *routing* lebih efektif
3. Untuk penelitian selanjutnya dapat dilakukan dengan membandingkan protokol *routing* berbasis posisi seperti GPSR.

DAFTAR PUSTAKA

1. a., hadi, M. Z. S., Amran, H. & R, N. P., 2011. Analisis Performansi Routing AODV pada Jaringan VANET.
2. Arijit, M. et al., 2014. AODV Routing Protocol Modification With Stack For VANET In City Scenarios. *International Journal Emerging Technologies In Computational and Applied Sciences (IJECTAS)*, p. 128.
3. Boppana, R. V. & Konduru, S. P., 2001. An Adaptive Distance Vector Algorithm for Mobile , Ad Hoc Networks. *IEEE*, pp. 1753-1762.
4. Dalal, K., Chaudhary, P. & Dahiya, P., 2012. Performance Evaluation of AODV and ADV Protocols in VANET Scenarios. *IJCTA*, Februari, Volume 3 (1), pp. 50-55.
5. Dhamayanti, Y. & Hendrantoro, G., 2013. Analisis Perbandingan Kinerja Protokol Dynamic Source Routing dan Ad hoc On-demand Distance Vector Pada Mobile Ad hoc Network Untuk Sistem Komunikasi Taktis Kapal Perang. *Jurnal Ilmiah Elite Elektro*, Maret, 4(1), pp. 5-10.
6. Faikah, A., Munadi, R. & Yovita, L. V., t.thn. Analisis Performansi Routing Protokol GPSR, GyTAR, dan B-MFR Pada VANET Untuk Inter Vehicle Communication.
7. Mogi, I. K. A. & Wibisono, W., 2012. VANET Untuk Solusi Komunikasi Data Di Kawasan Pariwisata Bali. 22 10.
8. Muslim, A., 2007. Analisis Codec dan Payload Pada Micronet dan CISCO Pada Jaringan VPN_MPLS. *Jurnal Teknologi Informasi DINAMIK*, Volume XIII, pp. 109-121.
9. Rios, X. C. et al., 2011. Performance Evaluation of Realistic Scenarios for Vehicular Ad hoc Networks With CityMob and NCTUns. Juni.

10. S S, F. D. et al., t.thn. Analisis Performansi Protokol Routing AODV dan DSR pada MANET.
11. Saha, S., Roy, U. & Sinha, D. D., 2013. Comparative Study of Ad-Hoc Protocols in MANET and VANET. *Research India Publications*, Volume 8, pp. 1083-1096.
12. Wan, S.-Y., Chou, C.-L., Lin, C.-C. & Huang, C.-H., 2010. The Protocol Developer Manual for the NCTUNS 6.0 Network Simulator and Emulator. 15 January.
13. Zeadally, S. et al., 2010. vehicular ad hoc Networks (VANETS) : Status, Results, and Challenges.



LAMPIRAN

Lampiran 1. Hasil Delay (ms)

AODVUrban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	90,12	80,2	208,18
2	182,83	70,6	192,15
3	178,6	62,93	283,22
4	172,99	67,62	356,04
5	182,83	48,94	229,66
6	163,44	57,89	274,68
7	155,16	58,09	208,62
8	192,83	73,04	203,12
9	190,06	58	280,48
10	178,6	58,01	222,91
Rata-rata	177,482	63,532	245,906

AODVUrban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	70,64	105,24	69,96
2	64,31	98,82	46,18
3	60,64	106,13	57,42
4	74,6	100,91	73,47
5	54,1	122,2	136
6	28,5	119,01	43,18
7	64,77	105,32	51,13
8	69,2	133,39	89,2
9	43,3	110,02	59,57
10	64,3	169,13	46,9
Rata-rata	59,436	117,017	67,301

ADVUrban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	915,42	965,53	2312,18
2	706,27	843,17	2380,15
3	915,42	843,17	1186,47
4	706,27	843,11	1701,47
5	706,27	843,17	1186,47
6	706,27	843,17	2083,29
7	706,27	727,92	1187,02
8	706,27	843,17	1279,37
9	706,27	843,11	2252,95
10	706,27	841,98	1186,47
Rata-rata	748,1	843,75	1675,58

ADVUrban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	130,36	111,95	198,27
2	130,35	111,96	127,92
3	130,87	111,94	130,52
4	172,05	111,95	132,47
5	130,35	111,8	484,14
6	130,36	111,96	321,14
7	130,35	112,52	368,24
8	130,34	111,96	130
9	130,38	111,98	151
10	130,35	112,47	129,2
Rata-rata	134,576	112,049	217,29

AODV Highway

no percobaan	node 10	node 20	node 30
1	108,67	219,82	312,04
2	112,5	238,01	312,04
3	108,77	205,41	299,36
4	107,74	239,88	312,13
5	112,47	320,59	276,99
6	108,94	239,89	312,02
7	205,01	221,49	312,02
8	108,66	175,34	305,64
9	124,74	263,23	351,94
10	108,68	173,84	312,02
Rata-rata	120,618	229,75	310,62

ADV Highway

no percobaan	node 10	node 20	node 30
1	3061,01	3437,94	5556,42
2	3065,4	3148,13	5558,32
3	3061,03	3082,27	3353,11
4	3059,47	2994,55	5.564,26
5	3062,21	3065,4	3353,11
6	3063,35	3437,95	5556,42
7	3061,14	3955,07	5556,42
8	3062,48	3092,98	5760,93
9	3061,01	3148,13	5556,43
10	3061,01	3065,21	5556,43
Rata-rata	3061,81	3242,76	5089,73

Lampiran 2. Hasil *Throughput* (KB/s)

AODV *Urban* Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	121,7	128,11	113,81
2	117,6	128,28	109,93
3	117,53	130,31	101,5
4	115,39	129,58	96,76
5	117,6	129,07	101,25
6	117,53	128,95	105,41
7	118,41	130,98	104,82
8	116,25	129,96	110,33
9	118,17	127,96	103,73
10	117,53	127,96	105,63
Rata-rata	117,771	129,116	105,317

AODV *Urban* Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	123,61	122,8	120,66
2	130,96	120,56	124,2
3	130,83	125,56	128,59
4	129,49	123,59	122,83
5	131,18	122,75	112,53
6	130,59	121,33	126
7	130,77	118,54	120,93
8	130,95	126,55	123,36
9	133,9	121,16	123,89
10	133,97	124,87	125,56
Rata-rata	130,625	122,771	122,855

ADV Urban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	114,05	133,33	126,54
2	116,9	127,52	126,37
3	114,05	127,52	96,69
4	116,9	127,52	117,24
5	116,9	127,52	96,69
6	116,9	127,52	115,71
7	116,9	115,85	96,69
8	116,9	127,52	94,72
9	116,9	127,52	120,27
10	116,9	127,52	96,69
Rata-rata	116,33	126,934	108,761

ADV Urban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	138,6	134,79	127
2	138,6	134,79	137,48
3	138,5	134,79	138,55
4	133,59	134,79	138,67
5	138,6	133,94	135,04
6	138,6	134,79	135,76
7	138,6	130,025	136,7
8	138,6	134,79	137,52
9	138,6	134,79	134,47
10	138,6	134,79	136,92
Rata-rata	138,089	134,229	135,811

AODV Highway

no percobaan	node 10	node 20	node 30
1	106,1	107,11	84,83
2	104,4	115,33	84,83
3	106,1	111,42	80,55
4	107,37	103,31	84,83
5	104,4	108,73	78,84
6	107,07	104,51	84,83
7	104,96	107,27	83,34
8	106,1	116,28	80,19
9	106,44	104,78	89,33
10	106,1	116,89	77,87
Rata-rata	105,904	109,563	82,944

ADV Highway

no percobaan	node 10	node 20	node 30
1	127,18	125,43	132,63
2	127,02	109,42	132,63
3	127,18	127,58	132,63
4	127,24	114,56	132,63
5	127,18	130,96	139,55
6	130,31	125,43	132,63
7	127,18	116,59	132,63
8	127,24	125,48	132,63
9	127,18	109,42	132,63
10	127,18	130,96	132,63
Rata-rata	127,489	121,583	133,322

Lampiran 3. Hasil Packet Loss (%)

AODV Urban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	72,00602021	33,36	61,84
2	74,17764021	34,15	66,63581199
3	74,19067107	31,215	72,83348518
4	75,7676521	32,015	71,85517392
5	74,17764021	31,29	73,50073391
6	74,19067107	33,03	64,38741811
7	67,80456706	35,93	69,65918098
8	74,05328858	32,425	67,54758534
9	74,5130372	32,495	69,32189761
10	74,19067107	32,495	69,27772676
Rata-rata	73,50718588	32,8405	68,68590138

AODV Urban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	40,86	49,12831437	46,795
2	41,5518381	65,96525195	55,98459728
3	44,14	61,14729745	43,09063893
4	59,91206375	58,10516727	53,02043679
5	50,1082194	60,30991847	44,67723386
6	53,90909571	59,94352562	57,05843395
7	48,7	63,33867414	55,65319901
8	52,02580125	55,76151541	56,21161784
9	52,0614475	61,20632995	47,51063297
10	41,5518381	55,59109081	54,43663442
Rata-rata	48,48203038	59,04970854	51,44384251

ADV Urban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	73,13253012	63,74201046	74,745
2	61,2720403	67,26775076	74,625
3	73,13253012	67,26775076	72,05221175
4	61,2720403	67,26775076	77,395
5	61,2720403	67,26775076	72,05221175
6	61,2720403	67,26775076	80,3
7	61,2720403	71,4957667	72,05221175
8	61,2720403	67,26775076	84,38755767
9	61,2720403	67,26775076	74,43
10	61,2720403	67,26775076	72,05221175
Rata-rata	63,64413827	67,33797833	75,40914047

ADV Urban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	19,97350603	24,96313772	71,03671042
2	19,97350603	24,96313772	19,82236007
3	20,03839338	24,96313772	19,88541084
4	35,49281984	24,96313772	19,94366197
5	19,97350603	28,75	47,40577756
6	19,97350603	24,96313772	38,28149061
7	19,97350603	28,54736842	40,87854924
8	19,97350603	24,96313772	19,66937518
9	19,97350603	25,13361045	34,60390356
10	19,97350603	29,73999416	19,49092955
Rata-rata	21,53192615	26,19497993	33,1018169

AODV Highway

no percobaan	node 10	node 20	node 30
1	83,22498811	70,27851393	82,22896376
2	83,57380688	65,23273399	82,22896376
3	83,22498811	66,07330367	84,4354072
4	83,02415306	67,855	82,22896376
5	83,57380688	68,395	82,66033254
6	83,07700439	67,855	82,22896376
7	84,61497807	67,875	82,22896376
8	83,22498811	65,46327316	82,68544129
9	81,08051876	71,20752427	78,00774194
10	83,22498811	58,20291015	82,22896376
Rata-rata	83,18442205	66,84382592	82,11627055

ADV Highway

no percobaan	node 10	node 20	node 30
1	84,78907135	84,43768997	82,66566573
2	84,81030578	86,71530628	82,66566573
3	84,78907135	84,43768997	87,22878815
4	84,7819932	85,07925396	82,66566573
5	84,78907135	84,43768997	87,22878815
6	84,78907135	84,43768997	82,66566573
7	84,78907135	86,583728	82,66566573
8	84,7819932	84,43104671	82,66566573
9	84,78907135	86,71530628	82,66566573
10	84,78907135	84,43768997	82,66566573
Rata-rata	84,78977916	85,17130911	83,57829022

Lampiran 4. Hasil *Packet Delivery Ratio* (%)

AODV Urban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	27,99397979	66,64	38,16
2	25,82235979	65,85	33,36418801
3	25,80932893	68,785	27,16651482
4	24,2323479	67,985	28,14482608
5	25,82235979	68,71	26,49926609
6	25,80932893	66,97	35,61258189
7	32,19543294	64,07	30,34081902
8	25,94671142	67,575	32,45241466
9	25,4869628	67,505	30,67810239
10	25,80932893	67,505	30,72227324
Rata-rata	27,049553	67,1595	31,31409862

AODV Urban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	59,14	50,87168563	53,205
2	58,4481619	34,03474805	44,01540272
3	55,86	38,85270255	56,90936107
4	40,08793625	41,89483273	46,97956321
5	49,8917806	39,69008153	55,32276614
6	46,09090429	40,05647438	42,94156605
7	51,3	36,66132586	44,34680099
8	47,97419875	44,23848459	43,78838216
9	47,9385525	38,79367005	52,48936703
10	58,4481619	44,40890919	45,56336558
Rata-rata	51,51796962	40,95029146	48,55615749

ADV Urban Kecepatan 8 m/s

no percobaan	node 30	node 40	node 50
1	26,86746988	36,25798954	25,255
2	38,7279597	32,73224924	25,375
3	26,86746988	32,73224924	27,94778825
4	38,7279597	32,73224924	22,605
5	38,7279597	32,73224924	27,94778825
6	38,7279597	32,73224924	19,7
7	38,7279597	28,5042333	27,94778825
8	38,7279597	32,73224924	15,61244233
9	38,7279597	32,73224924	25,57
10	38,7279597	32,73224924	27,94778825
Rata-rata	36,35586173	32,66202167	24,59085953

ADV Urban Kecepatan 18 m/s

no percobaan	node 30	node 40	node 50
1	80,02649397	75,03686228	28,96328958
2	80,02649397	75,03686228	80,17763993
3	79,96160662	75,03686228	80,11458916
4	64,50718016	75,03686228	80,05633803
5	80,02649397	71,25	52,59422244
6	80,02649397	75,03686228	61,71850939
7	80,02649397	71,45263158	59,12145076
8	80,02649397	75,03686228	80,33062482
9	80,02649397	74,86638955	65,39609644
10	80,02649397	70,26000584	80,50907045
Rata-rata	78,46807385	73,80502007	66,8981831

AODV Highway

no percobaan	node 10	node 20	node 30
1	16,77501189	29,72148607	17,77103624
2	16,42619312	34,76726601	17,77103624
3	16,77501189	33,92669633	15,5645928
4	16,97584694	32,145	17,77103624
5	16,42619312	31,605	17,33966746
6	16,92299561	32,145	17,77103624
7	15,38502193	32,125	17,77103624
8	16,77501189	34,53672684	17,31455871
9	18,91948124	28,79247573	21,99225806
10	16,77501189	41,79708985	17,77103624
Rata-rata	16,81557795	33,15617408	17,88372945

ADV Highway

no percobaan	node 10	node 20	node 30
1	15,21092865	15,56231003	17,33433427
2	15,18969422	13,28469372	17,33433427
3	15,21092865	15,56231003	12,77121185
4	15,2180068	14,92074604	17,33433427
5	15,21092865	15,56231003	12,77121185
6	15,21092865	15,56231003	17,33433427
7	15,21092865	13,416272	17,33433427
8	15,2180068	15,56895329	17,33433427
9	15,21092865	13,28469372	17,33433427
10	15,21092865	15,56231003	17,33433427
Rata-rata	15,21022084	14,82869089	16,42170978

Lampiran 5. Uji analysis of variance (anova)

Uji anova AODV terhadap *delay*

Hasil uji anova

Lingkungan	Protokol Routing	Parameter	Keterangan
Urban	AODV	<i>delay</i>	signifikan
		<i>throughput</i>	signifikan
		<i>packet loss</i>	signifikan
		PDR	signifikan
	ADV	<i>delay</i>	signifikan
		<i>throughput</i>	signifikan
		<i>packet loss</i>	signifikan
		PDR	signifikan
Highway	AODV	<i>delay</i>	signifikan
		<i>throughput</i>	signifikan
		<i>packet loss</i>	signifikan
		PDR	signifikan
	ADV	<i>delay</i>	signifikan
		<i>throughput</i>	signifikan
		<i>packet loss</i>	signifikan
		PDR	signifikan

Lampiran 6. Uji Beda Nyata Terkecil (BNT)

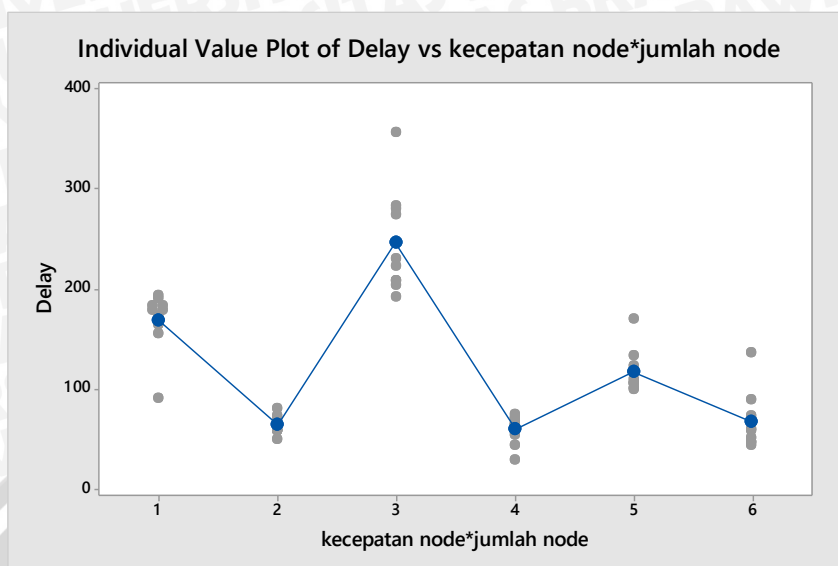
Uji BNT Lingkungan *Urban*

Kategori lingkungan *packet delivery ratio* :

1. Kategori 1 untuk kecepatan *node* 8 m/s dan jumlah *node* 30
2. Kategori 2 untuk kecepatan *node* 8 m/s dan jumlah *node* 40
3. Kategori 3 untuk kecepatan *node* 8 m/s dan jumlah *node* 50
4. Kategori 4 untuk kecepatan *node* 18 m/s dan jumlah *node* 30
5. Kategori 5 untuk kecepatan *node* 18 m/s dan jumlah *node* 40
6. Kategori 6 untuk kecepatan *node* 18 m/s dan jumlah *node* 50.

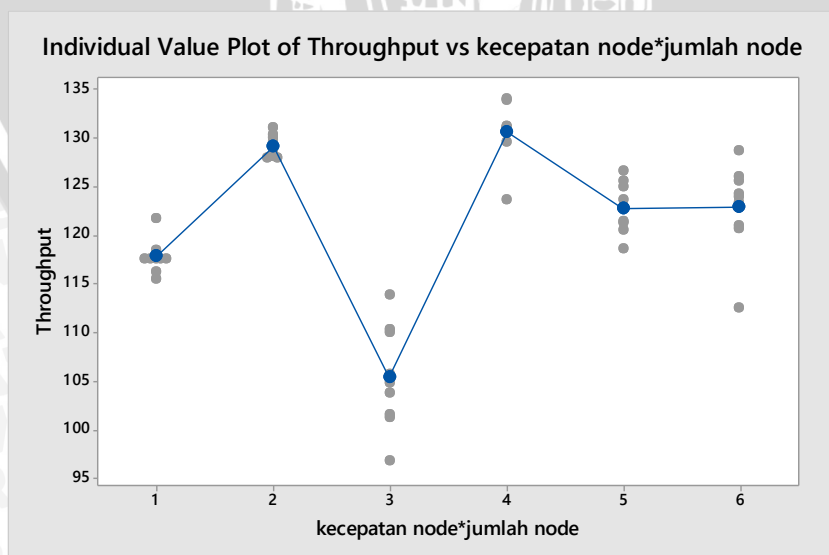
Uji BNT AODV terhadap *delay*

Kategori	N	Mean	Grouping
3	10	245.9	A
1	10	168.75	B
5	10	117.02	C
6	10	67.30	D
2	10	63.53	D
4	10	59.44	D



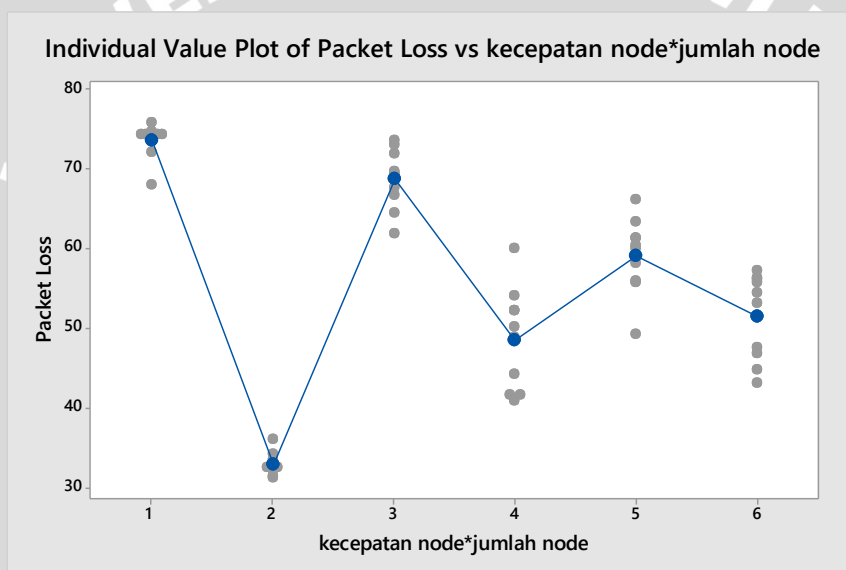
Uji BNT AODV terhadap *throughput*

Kategori	N	Mean	Grouping
4	10	130.625	A
2	10	129.116	A
6	10	122.85	B
5	10	122.771	B
1	10	117.771	C
3	10	105.32	D



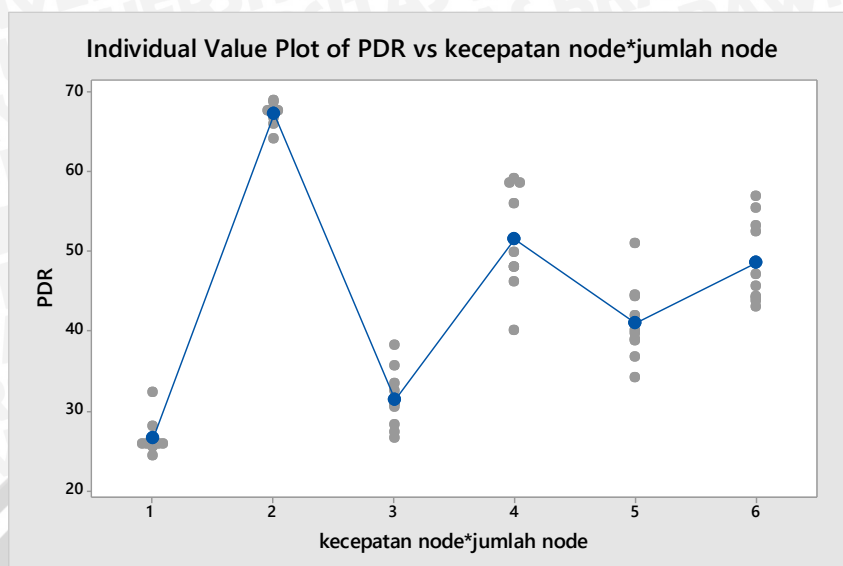
Uji BNT AODV terhadap *packet loss*

Kategori	N	Mean	Grouping
1	10	73.507	A
3	10	68.69	B
5	10	59.05	C
6	10	51.44	D
4	10	48.48	D
2	10	32.841	E



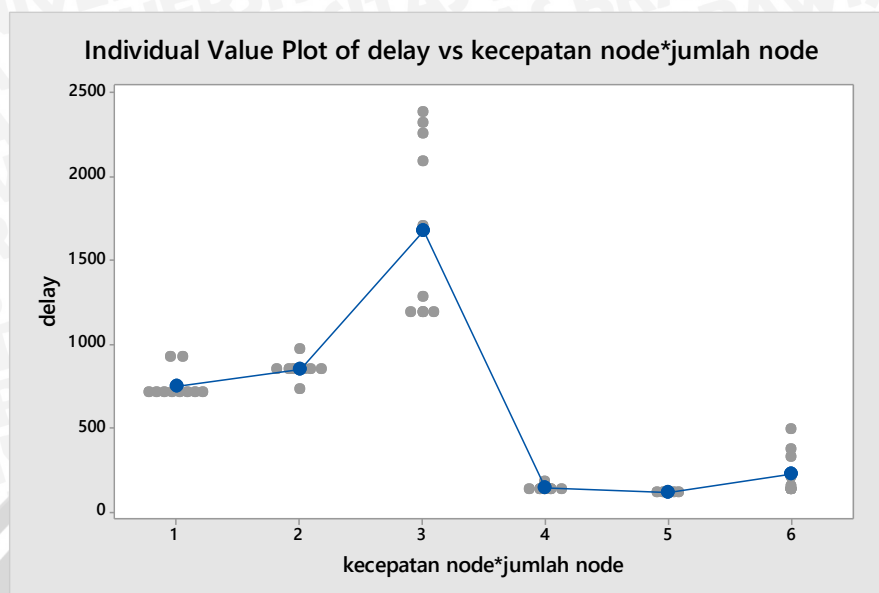
Uji BNT AODV terhadap *packet delivery ratio*

Kategori	N	Mean	Grouping
2	10	67.159	A
4	10	51.52	B
6	10	48.56	B
5	10	40.95	C
3	10	31.31	D
1	10	26.493	E



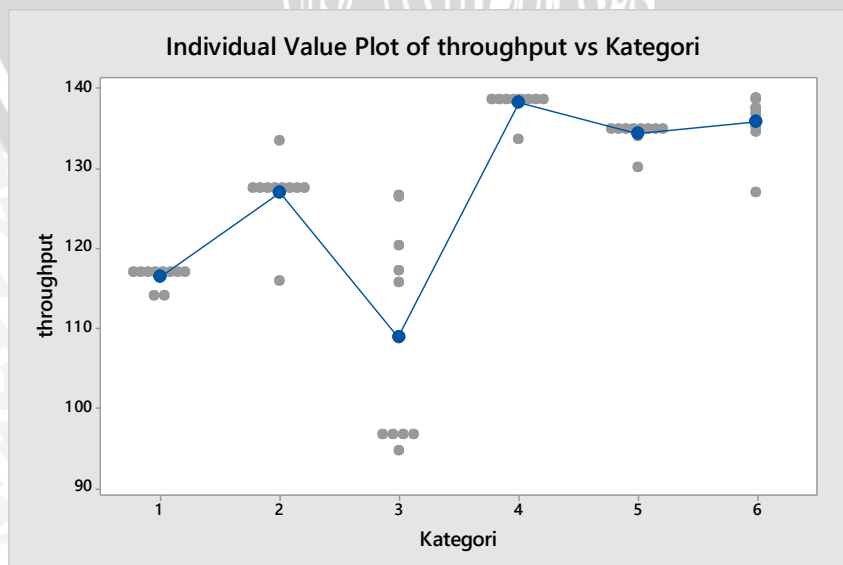
Uji BNT ADV terhadap delay

Kategori	N	Mean	Grouping
3	10	1676	A
2	10	843.8	B
1	10	748.1	B
6	10	217.3	C
4	10	134.58	C
5	10	112.049	C



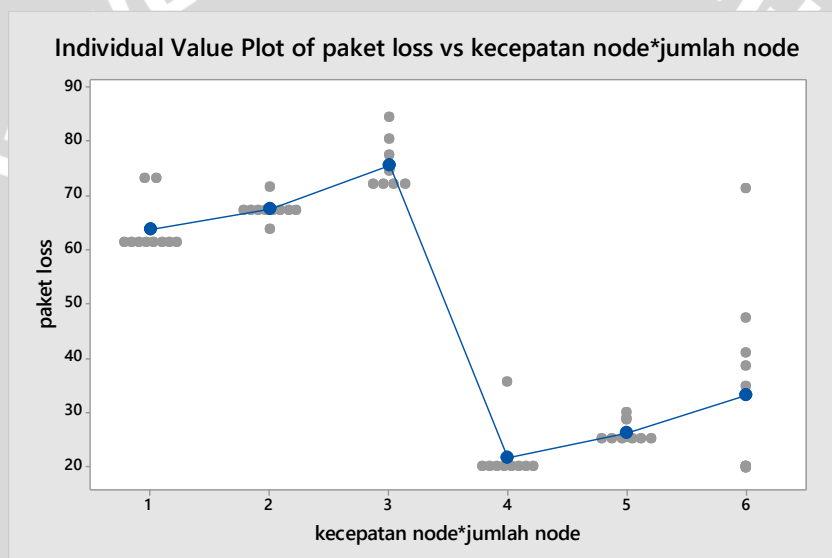
Uji BNT ADV terhadap *throughput*

Kategori	N	Mean	Grouping
4	10	138.089	A
6	10	135.81	A
5	10	134.229	A
2	10	126.93	B
1	10	116.330	C
3	10	108.76	D



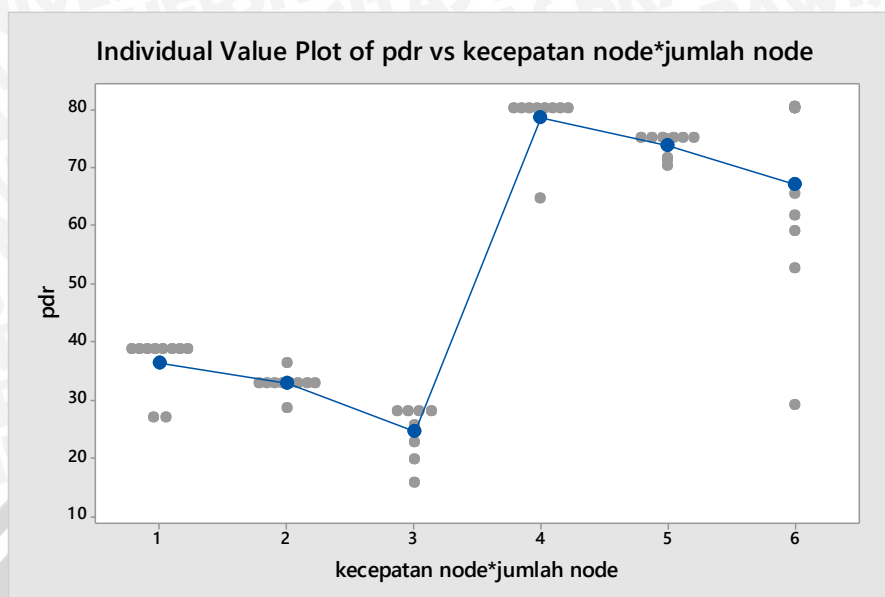
Uji BNT ADV terhadap *packet loss*

Kategori	N	Mean	Grouping
3	10	75.41	A
2	10	67.338	B
1	10	63.64	B
6	10	33.10	C
5	10	26.195	C D
4	10	21.53	D



Uji BNT ADV terhadap *packet delivery ratio*

Kategori	N	Mean	Grouping
4	10	78.47	A
5	10	73.805	A B
6	10	66.90	B
1	10	36.36	C
2	10	32.662	C
3	10	24.59	D



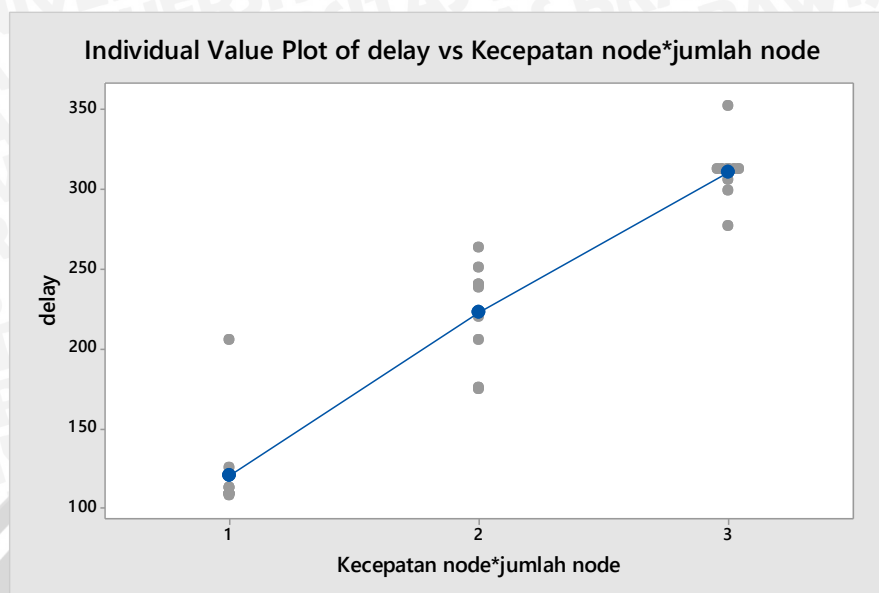
Uji BNT Lingkungan Highway

Kategori lingkungan *packet delivery ratio* :

4. Kategori 1 untuk jumlah *node* 10
5. Kategori 2 untuk jumlah *node* 20
6. Kategori 3 untuk jumlah *node* 30.

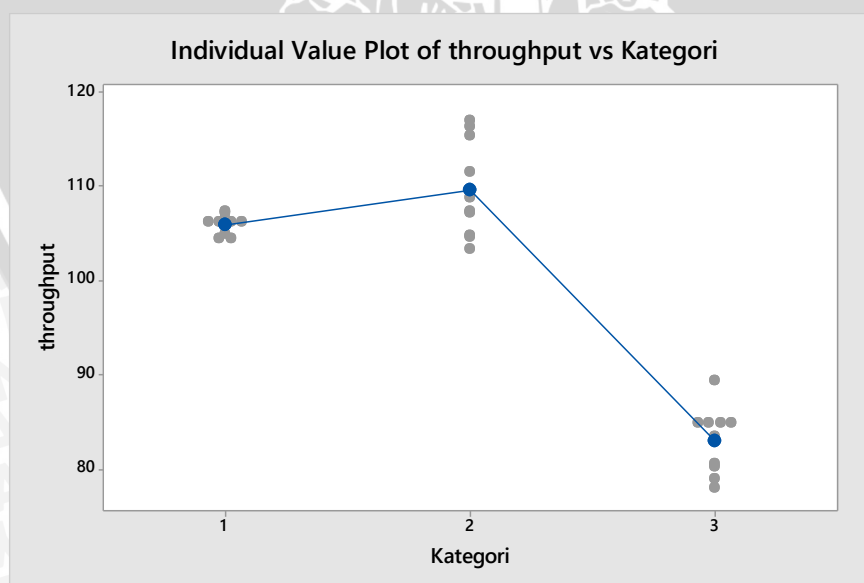
Uji BNT AODV terhadap *delay*

Kategori	N	Mean	Grouping
3	10	310.62	A
2	10	222.78	B
1	10	120.62	C



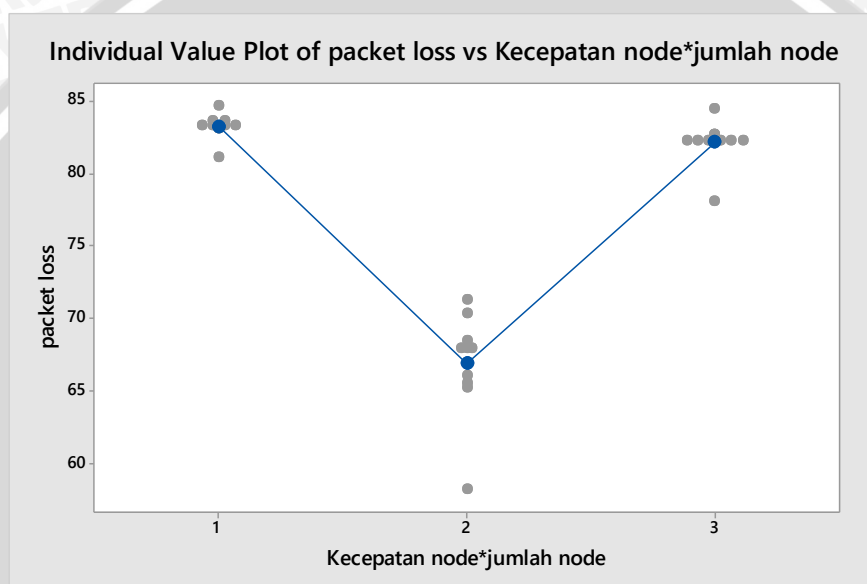
Uji BNT AODV terhadap *throughput*

Kategori	N	Mean	Grouping
2	10	109.56	A
1	10	105.904	B
3	10	82.94	C



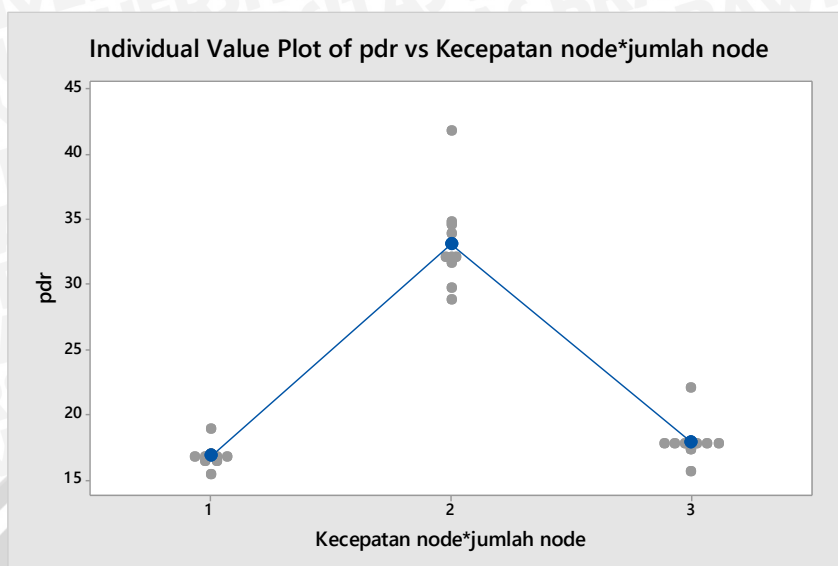
Uji BNT AODV terhadap *packet loss*

Kategori	N	Mean	Grouping
1	10	83.184	A
3	10	82.116	A
2	10	66.84	B



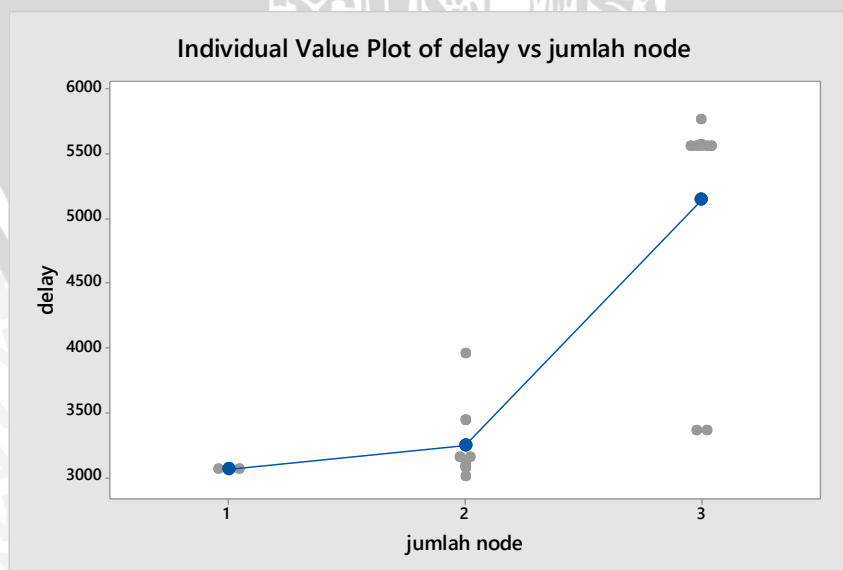
Uji BNT AODV terhadap *packet delivery ratio*

Kategori	N	Mean	Grouping
2	10	33.16	A
3	10	17.884	B
1	10	16.816	B



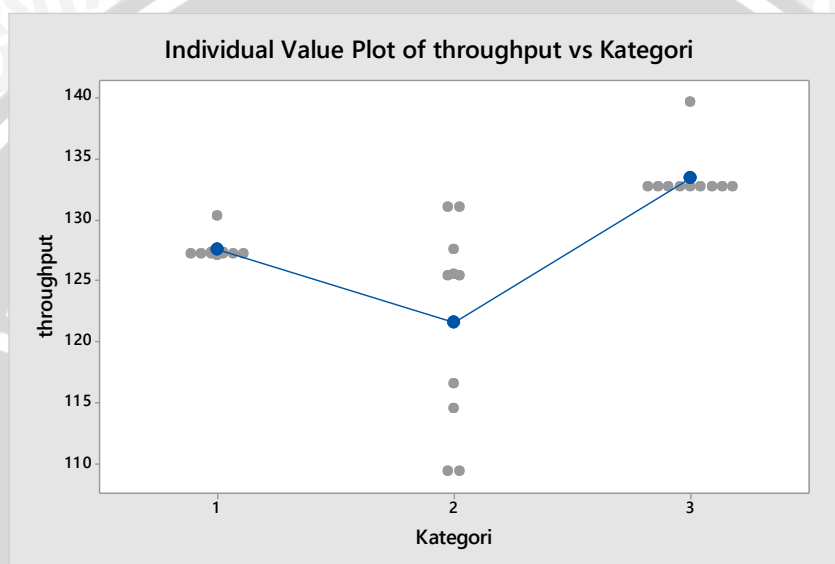
Uji BNT ADV terhadap delay

kategori	N	Mean	Grouping
3	10	5137	A
2	10	3242.8	B
1	10	3061.81	B



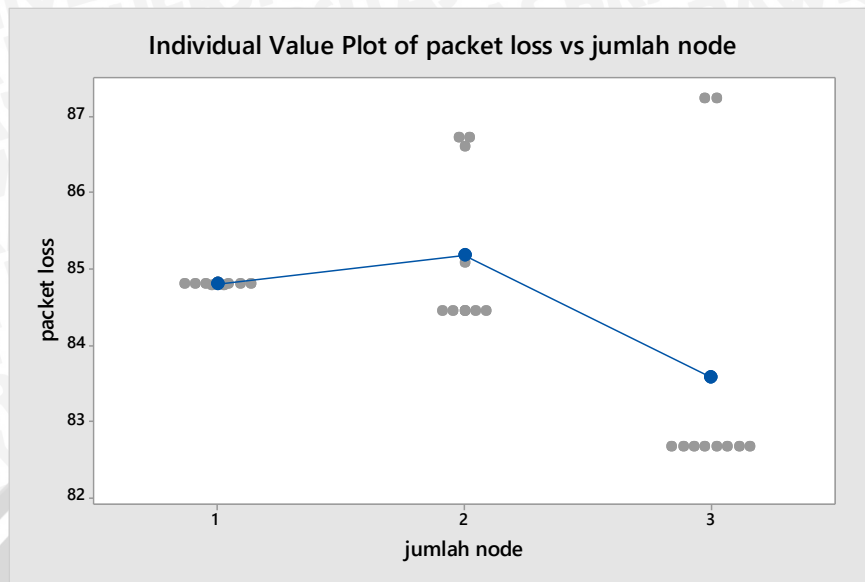
Uji BNT ADV terhadap *throughput*

Kategori	N	Mean	Grouping
3	10	133.322	A
1	10	127.489	B
2	10	121.58	C



Uji BNT ADV terhadap *packet loss*

kategori	N	Mean	Grouping
2	10	85.171	A
1	10	84.7898	A
3	10	83.578	B



dap packet delivery ratio

Mean	Grouping
16.422	A
5.2102	B
14.829	B

kategori	N	Mean	Grouping
3	10	16.422	A
1	10	15.2102	B
2	10	14.829	B

