

**Implementasi Protokol *Routing Ad-Hoc On Demand*
Distance Vector (AODV) Pada Jaringan VANETS di JL.
Jend. Sudirman Purwokerto**

TUGAS AKHIR

Disusun sebagai syarat mata kuliah Tugas Akhir

Program Studi S1 Teknik Telekomunikasi

Disusun oleh:

Daffa Akhdan Afif/21101086

Abdul Fatah Amrulloh/21101088

Yusril Ardanis21101114



**Universitas
Telkom**

**Program Studi S1 Teknik Telekomunikasi
Direktorat Kampus Purwokerto
Universitas Telkom
Purwokerto
2025**

LEMBAR PENGESAHAN

Implementasi Protokol *Routing Ad-Hoc On Demand Distance Vector* (AODV) Pada Jaringan VANETS di Jl. Jend. Sudirman Purwokerto

Implementation of Ad-Hoc On Demand Distance Vector (AODV) Routing Protocol on VANETS Network in JL. Jend. Sudirman Purwokerto

Disusun oleh:

Daffa Akhdan Afif	(21101086)
Abdul Fatah Amrulloh	(21101088)
Yusril Ardanis	(21101114)

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagai syarat memperoleh gelar pada Program Studi S1 Teknik Telekomunikasi

Direktorat Kampus Purwokerto

Universitas Telkom

Purwokerto, 27 Mei 2025

Menyetujui

Pembimbing I,

Pembimbing II,

Eko Fajar Cahyadi,
S.T.,M.T.,Ph.D.
NIDN: 0616098703

Jafaruddin Gusti Amri Ginting,
S.T., M.T.
NIDN: 0620108901

Ketua Program Studi
Sarjana Teknik Telekomunikasi,

Zein Hanni Pradana, S.T., M.T.
NIP: 17900074

LEMBAR ORISINALITAS

NAMA : Daffa Akhdan Afif
NIM : 21101086
Alamat : Banjarsari RT03/RW04, Kec. Sukoharjo, Kab. Wonosobo
No Tlp/HP : 083894847874
E-mail : daffadan33@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinil saya sendiri, dengan judul :

Implementasi Protokol *Routing Ad-Hoc On Demand Distance Vector (AODV)* Pada Jaringan VANETS di Jl. Jend. Sudirman Purwokerto.

Implementation of Ad-Hoc On Demand Distance Vector (AODV) Routing Protocol on VANETS Network in JL. Jend. Sudirman Purwokerto.

Atas pernyataan ini, saya siap menanggung risiko / sanksi yang dijatuhkan pada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan tidak aslian karya ini.

Purwokerto, 4 Juni 2025

Yang menyatakan

Daffa Akhdan Afif

NIM 21101086

LEMBAR ORISINALITAS

NAMA : Abdul Fatah Amrulloh
NIM : 21101088
Alamat : Tuwel RT06/RW03, keb. bojong, kab. Tegal
No. Tlp/HP : 085135090075
E-mail : abdufatah2660502@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri, dengan judul:

Implementasi Protokol *Routing Ad-Hoc On Demand Distance Vector (AODV)* Pada Jaringan VANETS di Jl. Jend. Sudirman Purwokerto.

Implementation of Ad-Hoc On Demand Distance Vector (AODV) Routing Protocol on VANETS Network in JL. Jend. Sudirman Purwokerto.

Atas pernyataan ini, saya siap menanggung risiko / sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan tidak aslian karya ini.

Purwokerto, 4 Juni 2025

Yang menyatakan

Abdul Fatah Amrulloh

NIM 21101088

LEMBAR ORISINALITAS

NAMA : Yusril Ardanis
NIM : 21101114
Alamat : Karanggude kulon RT07/RW01,Kec. Karanglewas,Kab.
Banyumas
No. Tlp/HP : 087891486863
E-mail : yusrilardanis7@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinial saya sendiri dengan berjudul:

**Implementasi Protokol *Routing Ad-Hoc On Demand Distance Vector (AODV)*
Pada Jaringan VANETS di Jl. Jend. Sudirman Purwokerto.**

***Implementation of Ad-Hoc On Demand Distance Vector (AODV) Routing
Protocol on VANETS Network in JL. Jend. Sudirman Purwokerto.***

Atas pernyataan ini, saya siap menanggung risiko / sanksi yang di jatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan tidak aslian karya ini.

Purwokerto, 4 Juni 2025

Yang menyatakan

Yusril Ardanis

NIM 21101114

ABSTRAK

Kemacetan dan kecelakaan lalu lintas yang meningkat setiap tahun akibat pertumbuhan jumlah kendaraan di wilayah padat kendaraan menjadi permasalahan serius. Topik ini penting karena dibutuhkan solusi teknologi cerdas untuk meningkatkan efisiensi dan keselamatan lalu lintas. Salah satu pendekatan adalah dengan menerapkan komunikasi antar kendaraan (V2V) menggunakan jaringan Vehicular Ad-Hoc Network (VANET) dan protokol routing AODV yang bersifat reaktif. Penelitian ini menyimulasikan performa AODV menggunakan SUMO dan OMNeT++ dengan parameter evaluasi berupa Throughput, Packet Delivery Ratio (PDR), dan End-to-End Delay (EED). Solusi ini dikembangkan pada peta nyata dan divisualisasikan melalui dashboard berbasis web. Hasil simulasi menunjukkan bahwa peningkatan jumlah kendaraan dapat meningkatkan throughput hingga titik optimal tertentu, namun pada kepadatan tinggi, PDR mengalami penurunan dari 4,85% menjadi 4,19%. Kecepatan kendaraan berpengaruh terhadap kestabilan rute komunikasi, di mana semakin tinggi kecepatan menyebabkan kenaikan EED dari 0,0947 detik menjadi 0,1192 detik. Hal ini menunjukkan bahwa protokol AODV masih menghadapi tantangan dalam menjaga konektivitas yang stabil pada kondisi mobilitas tinggi dan topologi jaringan yang dinamis. Dengan demikian, kinerja AODV sangat dipengaruhi oleh jumlah dan kecepatan kendaraan, serta memerlukan optimasi lanjutan agar lebih adaptif dalam skenario lalu lintas perkotaan.

Kata Kunci: VANET, AODV, OMNeT++, SUMO, jumlah kendaraan, kecepatan, kinerja jaringan

ABSTRACT

Traffic congestion and accidents that increase every year due to the growth of the number of vehicles in densely populated areas are serious problems. This topic is important because intelligent technology solutions are needed to improve traffic efficiency and safety. One approach is to implement vehicle-to-vehicle (V2V) communication using a Vehicular Ad-Hoc Network (VANET) and a reactive AODV routing protocol. This study simulates the performance of AODV using SUMO and OMNeT++ with evaluation parameters in the form of Throughput, Packet Delivery Ratio (PDR), and End-to-End Delay (EED). This solution is developed on a real map and visualized through a web-based dashboard. The simulation results show that increasing the number of vehicles can increase throughput to a certain optimal point, but at high density, PDR decreases from 4.85% to 4.19%. Vehicle speed affects the stability of the communication route, where the higher the speed, the EED increases from 0.0947 seconds to 0.1192 seconds. This shows that the AODV protocol still faces challenges in maintaining stable connectivity under high mobility conditions and dynamic network topology. Thus, AODV performance is greatly influenced by the number and speed of vehicles, and requires further optimization to be more adaptive in urban traffic scenarios.

Keywords: *VANET, AODV, OMNeT++, SUMO, number of vehicles, speed, network performance*

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpah rahmat dan karunianya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Simulasi Protokol *Routing Ad-Hoc On Demand Distance Vector (AODV)* Pada Jaringan VANETS di Jl. Jend. Sudirman Purwokerto**” ini dengan baik.

Laporan tugas akhir ini disusun sebagai syarat untuk memperoleh gelar sarjana pada program studi S1 Teknik Telekomunikasi, Universitas Telkom Purwokerto. Adapun tujuan utama dalam penelitian ini adalah mengavaluasi performa protokol AODV dalam jaringan komunikasi kendaraan (VANETs), khususnya pada kondisi lalu lintas dinamis di kawasan Jl. Jend. Sudirman. Yang diharapkan, hasil dari penelitian ini dapat memberikan kontribusi dalam pengembangan sistem komunikasi antar kendaraan yang lebih efisien dan aman di masa mendatang.

Penulis berharap laporan tugas akhir ini dapat memberikan manfaat bagi pembaca, baik sebagai referensi akademik maupun sebagai kontribusi awal dalam pengembangan sistem komunikasi antar kendaraan di masa depan. Semoga hasil penelitian ini dapat mendorong penelitian lanjutan di bidang yang sama dan memberikan dampak positif bagi perkembangan teknologi transportasi cerdas di Indonesia.

Purwokerto, Mei 2025

UCAPAN TERIMA KASIH

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya atas segala bantuan dan bimbingan serta dukungan yang diperoleh dalam bentuk apa pun. Ucapan terima kasih ditujukan kepada :

1. Orang Tua yang selalu mendukung dan memberikan do'a kepada penulis dalam setiap proses penyusunan Tugas Akhir.
2. Dr. Tenia Wahyuningrum, S.Kom., M.T. selaku Direktur Telkom University Purwokerto.
3. Eko Fajar Cahyadi, S.T.,M.T.,Ph.D. selaku pembimbing pertama yang telah sabar serta memberikan pengetahuan baru kepada penulis.
4. Jafaruddin Gusti Amri Ginting, S.T., M.T. selaku pembimbing kedua yang turut memberikan masukan dan saran berharga bagi penyempurnaan tugas akhir ini.
5. (...) selaku penguji yang telah meluangkan waktu untuk memberikan masukan dan saran hasil penelitian tugas akhir.
6. Teman-teman yang telah menemani selama pengerjaan dan serta memberikan bantuan dan dukungan moral.
7. Daniel Baskara Putra selaku vokalis HINDIA yang telah menemani dalam proses pengerjaan tugas akhir dalam bentuk karya musiknya.
8. Serta semua pihak yang telah memberikan bantuan dan dukungan yang tidak dapat penulis sebutkan satu persatu dalam menyelesaikan Tugas Akhir ini.

Dalam penyusunan tugas akhir ini penulis menyadari masih banyak kekurangan dan kesalahan yang membuat tugas akhir ini jauh dari kata sempurna oleh karena itu sebelum penulis memohon maaf sebesar-besarnya. Penulis berharap laporan tugas akhir ini dapat bermanfaat dikemudian hari. Terima kasih dan mohon maaf.

Purwokerto, Mei 2025

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR ORISINALITAS	ii
LEMBAR ORISINALITAS	iii
LEMBAR ORISINALITAS	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR.....	vii
UCAPAN TERIMA KASIH.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan dan Manfaat.....	3
1.4 Batasan Masalah	4
1.5 Metode Penelitian.....	5
1.6 Jadwal Pelaksanaan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu	7
2.2 <i>Vehicle to Vehicle (V2V)</i>	11
2.3 <i>Vehicular Ad-Hoc Network (VANET)</i>	13
2.4 <i>Protokol Routing Ad-Hoc Demand Distance Vector (AODV)</i>	14
2.5 <i>Open Street Map (OSM)</i>	16
2.6 <i>Simulation of Urban Mobility (SUMO)</i>	16
2.7 OMNeT++	17
2.8 <i>Throughput</i>	18
2.9 <i>End to End Delay (EED)</i>	18
2.10 <i>Packet Loss</i>	19
2.11 <i>Packet Delivery Ratio (PDR)</i>	20
2.12 <i>Replit</i>	20
2.13 <i>Veins</i>	21

2.14	<i>Website</i>	21
BAB III	PERANCANGAN SISTEM	23
3.1	Desain Sistem.....	23
3.1.1	Diagram Blok	24
3.1.2	Fungsi dan Fitur	25
3.2	Desain Perangkat Keras	28
3.2.1	Spesifikasi Komponen	28
3.3	Desain Perangkat Lunak	29
3.4	Skenario Pengujian	36
BAB IV	HASIL DAN ANALISIS	38
4.1	Hasil Percobaan	38
4.1.1	Pengujian Skenario 1	42
4.1.2	Pengujian Skenario 2	44
4.2	Analisis.....	46
4.2.1	Analisis Hubungan Jumlah <i>Node</i> terhadap <i>Throughput</i>	47
4.2.2	Analisis Hubungan Jumlah <i>Node</i> Terhadap <i>Packet Delivery Ratio</i>	48
4.2.3	Analisis Hubungan Jumlah <i>Node</i> terhadap <i>End-to-End Delay</i>	49
4.2.4	Analisis Hubungan Kecepatan terhadap <i>Throughput</i>	50
4.2.5	Analisis Hubungan Kecepatan terhadap <i>Packet Delevery Ratio</i>	52
4.2.6	Analisis Hubungan Kecepatan terhadap <i>End-to-End Delay</i>	53
BAB V	SIMPULAN SARAN	56
5.1	Simpulan	56
5.2	Saran.....	56
DAFTAR PUSTAKA		58
LAMPIRAN		62

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Komunikasi V2V	12
Gambar 2.2 Komunikasi Pada VANETs [17].....	14
Gambar 2.3 Ilustrasi Pencarian Rute AODV	15
Gambar 2. 4 Tampilan OSM.....	16
Gambar 2.5 Tampilan SUMO	17
Gambar 2.6 Tampilan Awal OMNeT++.....	17
Gambar 3.1 Rancangan Sistem	23
Gambar 3.2 Blok Diagram	24
Gambar 3.3 Desain Perangkat Keras.....	28
Gambar 3.4 Alur Kerja Sistem	31
Gambar 4.1 Tampilan SUMO saat Simulasi	38
Gambar 4.2 Proses Pengiriman Paket Pada OMNeT++	39
Gambar 4. 3 <i>Log</i> Pengiriman Pesan antar <i>Node</i>	39
Gambar 4.4 Hasil Pengujian QoS Setelah Simulasi	40
Gambar 4.5 Hasil Pengukuran dalam Excel	40
Gambar 4.6 Hasil Simulasi OMNeT++ di Jl Sudirman Purwokerto	41
Gambar 4. 7 Tampilan <i>Website</i> Hasil Pengujian.....	42
Gambar 4. 8 Grafik Pengujian <i>Throghput</i> Skenario 1	43
Gambar 4. 9 Grafik Pengujian <i>Packet Delivery Ratio</i> Skenario 1	43
Gambar 4. 10 Grafik Pengujian <i>End-to-End Delay</i> Skenario 1	44
Gambar 4. 11 Grafik Pengujian <i>Throghput</i> Skenario 2.....	44
Gambar 4. 12 Grafik Pengujian <i>Packet Delivery Ratio</i> Skenario 2.....	45
Gambar 4. 13 Grafik Pengujian <i>End-to-End Delay</i> Skenario 2.....	45

DAFTAR TABEL

Tabel 1.1 Jadwal dan <i>Milestone</i>	6
Tabel 2.1 Penelitian Terdahulu	10
Tabel 2.2 Nilai Standarisasi <i>Throughput</i> [22]	18
Tabel 2.3 Nilai Standarisasi <i>End-to-end Delay</i> [24].....	19
Tabel 2.4 Standarisasi Nilai <i>Packet Loss</i>	19
Tabel 2. Nilai Standarisasi <i>Packet Delivery Ratio</i> [27].....	20
Tabel 3.1 Laptop ASUS	28
Tabel 3.2 Komputer <i>Remote</i>	29
Tabel 3.3 <i>Virtual Machine</i> Linux.....	29
Tabel 3.4 Spesifikasi Perangkat Lunak.....	30
Tabel 3.5 Skenario Pengujian	37

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Semakin dengan bertambahnya tahun, jumlah kendaraan juga terus meningkat setiap tahunnya, hal ini yang pastinya akan menimbulkan beberapa permasalahan bukan hanya karena polusi ditimbulkan tetapi kemacetan hingga kecelakaan lalu lintas yang akan terus bertambah di setiap tahunnya. Oleh karena itu diperlukan sistem yang bisa mengatur dan mengoptimalkan pergerakan kendaraan yaitu *vehicular ad hoc networks* (VANETs), yang akan kami simulasikan pada jaringan protokol *routing ad-hoc on demand distance vector* (AODV) Pada Jaringan VANETs di Jl. Jend. Sudirman Purwokerto yang dapat meningkatkan efisiensi serta keamanan transportasi. Kemajuan dalam bidang teknologi yang memudahkan manusia untuk saling berkomunikasi merupakan bagian dari VANETs .

Jl. Jend. Sudirman merupakan salah satu jalan utama dan tersibuk di kota Purwokerto, yang setiap harinya mengalami kenaikan volume kendaraan, terutama pada jam sibuk seperti pukul 07.00 dan 17.00 WIB. Padatnya arus lalu lintas di kawasan ini sering menyebabkan kemacetan dan meningkatkan risiko kecelakaan lalu lintas. Maka dari pada itu untuk mengatasi permasalahan tersebut, diperlukan penerapan teknologi cerdas yang mampu meningkatkan keselamatan dan efisiensi lalu lintas. Salah satu solusi potensi adalah implementasi komunikasi antar kendaraan atau *vehicle-to-vehicle* (V2V) melalui jaringan VANETs. Yang menggunakan protokol *routing* AODV, AODV merupakan jenis protokol *routing* distance vector yang termasuk klasifikasi protokol vector yang merupakan dalam klasifikasi protokol *routing* yang reaktif, yang berarti hanya untuk meminta suatu layanan ketika diperlukan [1]. Kendaraan dapat saling bertukar informasi secara dinamis dan *real-time*, seperti peringatan dini saat terjadi kecelakaan atau kondisi lalu lintas padat. Simulasi penerapan AODV dalam jaringan VANETs di kawasan Jl. Jend. Sudirman bertujuan untuk mengevaluasi kinerja protokol dalam meningkatkan keamanan dan kelancaran lalu lintas di area tersebut.

VANETs adalah sebuah sub-aplikasi dari mobile ad-hoc networks (MANET), yang mempunyai karakteristik yang berbeda dari MANET. Salah satu karakteristik VANETs yaitu mobilitas yang tinggi, VANETs merupakan jaringan wireless bersifat ad-hoc yang merupakan subkelas dari MANET dan memungkinkan komunikasi antar kendaraan atau kendaraan dengan perangkat yang diterapkan di jalan. Dengan tujuan yaitu meningkatkan keamanan pada jalan raya dan transportasi [2]. SUMO dikembangkan oleh Pusat Dirgantara Jerman pada tahun 2001. SUMO merupakan alat simulasi lalu lintas 2-Dimensi (2D) yang bersifat mikroskopis dan multi-moda yang mencakup kendaraan, truk, pejalan kaki, angkutan umum, sepeda, kereta api, kendaraan listrik, dan kombinasi berbagai moda. SUMO bersifat open-source, gratis, dan tersedia untuk semua orang. Fitur-fitur SUMO telah menjadikannya salah satu alat simulasi lalu lintas yang paling populer untuk penelitian [3]. Penelitian ini menggunakan simulator SUMO dan Omnet++, dengan skenario VANETs yang berupa peta yang berbentuk grid maupun peta nyata, kecepatan kendaraan, jumlah kendaraan, dan jumlah Road Side Unit (RSU) menjadi fitur yang mendukung kinerja simulasi VANETs. Adapun efisiensi protokol *routing* AODV yang diuji pada penelitian ini adalah *Throughput*, *End-to-End Delay* (EED), dan *Packet Delivery Ratio* (PDR), adapun sedikit penjelasan dari parameter yang di uji adalah *Throughput* merupakan parameter pengujian yang di ambil merupakan perubahan jumlah *node* [4]. *End-to-End Delay* merupakan parameter pengujian yang mengambil waktu yang digunakan oleh paket data agar sampai ditempat tujuan [5]. *Packet Delivery Ratio* merupakan jenis rasio yang di antara banyaknya paket yang diterima oleh tujuan dengan banyak paket yang dikirim oleh sumber [6].

1.2 Rumusan Masalah

Dari Penelitian ini pasti akan ada masalah yang akan di hadapi, dan berikut merupakan permasalahan yang di jadikan sebagai rumusan masalah:

1. Bagaimana pengaruh jumlah kendaraan terhadap kinerja protokol *routing* AODV pada jaringan VANETs di lingkungan perkotaan?

2. Bagaimana kecepatan kendaraan memengaruhi stabilitas koneksi dan efisiensi perutean dalam protokol *routing* AODV di VANETs?
3. Bagaimana variasi jumlah RSU memengaruhi performa protokol *routing* AODV pada kondisi topologi jaringan yang dinamis?
4. Bagaimana kinerja protokol *routing* AODV dalam kondisi mobilitas tinggi dan perubahan topologi jaringan yang cepat pada skenario perkotaan?

1.3 Tujuan dan Manfaat

Pada penelitian ini memiliki tujuan-tujuan kenapa di adakan penelitian Simulasi Protokol *Routing Ad-Hoc On Demand Distance Vector* (AODV) Pada Jaringan VANETs di Jl. Jend. Sudirman Purwokerto, serta memberitahukan beberapa manfaat penelitian ini jika di terapkan pada lalu lintas nyata terutama di perkotaan, berikut tujuan dan manfaatnya.

Tujuan Penelitian :

1. Menjalankan Simulasi Protokol *routing Ad-Hoc On Demand Distance Vector* (AODV) pada jaringan komunikasi kendaraan (VANETs) di kawasan Jl. Jend. Sudirman, Purwokerto
2. Menganalisis Performa protokol AODV dalam skenario lalu lintas nyata yang mencerminkan kondisi mobilitas tinggi, topologi dinamis, dan kepadatan kendaraan di jalur tersebut.
3. Menilai efektivitas AODV dalam menjaga kestabilan antara kendaraan V2V dan antara kendaraan dengan infrastruktur (V2I) di kawasan perkotaan.
4. Memvalidasi penggunaan *software* untuk membantu simulasi seperti SUMO dan *Network Simulator* sebagai tempat untuk menguji performa protokol *routing* pada jaringan VANETs berbasis lokasi nyata.

Manfaat Penelitian:

1. Memberikan gambaran nyata mengenai kinerja protokol AODV dalam konteks lalu lintas di kawasan Jl. Jend. Sudirman, yang mana dapat dijadikan dasar pengembangan sistem komunikasi kendaraan di wilayah perkotaan.

2. Dapat memberikan acuan kepada instansi atau pengembang teknologi transportasi dalam merancang sistem komunikasi terutama V2V dan V2I yang andal dan responsif di wilayah perkotaan yang padat lalu lintas.
3. Memberikan informasi dan data yang mendukung agar dapat mengoptimalkan infrastruktur pendukung seperti *Road Side Unit* (RSU).
4. Memberikan contoh bahwa *software* simulasi berbasis SUMO dan *Network Simulator* dapat memproses analisis jaringan komunikasi kendaraan tanpa harus terjun langsung ke lapangan untuk melakukan pengujian, sehingga efisien dari segi waktu dan biaya.

1.4 Batasan Masalah

Pada penelitian ini memiliki rumusan masalah dan pastinya memiliki batasan masalah yang mana disusun berdasarkan rumusan masalah, berikut batasan masalahnya:

1. Penelitian dilakukan hanya pada skenario jaringan kendaraan di wilayah Jl. Jend. Sudirman. Purwokerto, dengan mempertimbangkan karakteristik lalu lintas kota seperti banyaknya persimpangan dan lampu lalu lintas.
2. Simulasi dilakukan menggunakan SUMO sebagai simulator mobilitas kendaraan, dan OMNeT++ untuk membaca data.
3. Jumlah kendaraan yang disimulasikan berkisaran 5 hingga 100 kendaraan, dengan mempresentasikan kepadatan lalu lintas ringan hingga padat di jalan Jl. Jend. Sudirman. Purwokerto.
4. Kecepatan kendaraan dalam simulasi memiliki rata-rata kecepatan 36km/H.
5. Jumlah *Road Side Unit* (RSU) yang digunakan dalam simulasi sebanyak 1 unit, yang di tempatkan pada jalan utama.
6. Protokol *routing* yang diuji dalam penelitian ini hanya *Ad Hoc On-Demand Distance Vector* (AODV), tanpa perbandingan langsung dengan protokol lainnya.
7. Topologi jaringan bersifat dinamis, dengan asumsi kendaraan dapat bergerak seusai dengan data lalu lintas.

8. Evaluasi performa protokol AODV akan difokuskan pada parameter seperti *Throughput*, *Packet Delivery Ratio* (PDR), dan *End-to-End Delay* (EED).

1.5 Metode Penelitian

Pada penelitian ini sangat mengandalkan data sekunder yang di dapat secara tidak langsung, seperti data yang diperoleh bukan melalui pengumpulan secara langsung oleh peneliti, Sumber data sekunder ini diperoleh dari berbagai literatur, Jurnal ilmiah, Artikel, dan situs-situs internet yang relevan dengan topik yang diteliti. Teknik pengumpulan data yang diterapkan dalam penelitian ini adalah studi literatur, dengan cara melakukan kajian mendalam terhadap buku, makalah ilmiah, jurnal, dan referensi daring lainnya guna memperoleh informasi yang terkait dengan konsep pengembangan sistem yang sedang diteliti.

1.6 Jadwal Pelaksanaan

Pada Tabel 1.1 berisi tentang jadwal pelaksanaan pengerjaan tugas akhir, dan beberapa *milestone* sebagai pencapaian pekerjaan.

Tabel 1.1 Jadwal dan *Milestone*

No.	Deskripsi Tahapan	Durasi	Tanggal Selesai	<i>Milestone</i>
1	Persiapan	2 Bulan	31 Okt 2024	Mendapatkan Judul.
2	Perencanaan	1 Bulan	30 Nov 2024	Mendapatkan <i>Software</i> dan <i>Protokol Routing</i> yang akan digunakan
3	Percobaan	3 Bulan	31 Jan 2025	Mendapatkan hasil simulasi SUMO dan OMNeT++
4	Implementasi dan Pengujian	1 Bulan	30 April 2025	Mendapatkan peta yang akan disimulasikan dan Mendapatkan hasil simulasi
5	Analisis dan Hasil	2 Bulan	30 April 2025	Penyelesaian Laporan

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian pertama [2], dilakukan analisis terhadap Dampak Serangan *Blackhole* dan *Jellyfish* pada protokol *routing Ad-Hoc On Demand Distance Vector* (AODV) di jaringan VANETs menggunakan model mobilitas acak. Metode yang digunakan berupa simulasi jaringan dengan NS2, mengonfigurasi jumlah *node*, jarak antar *node*, serta memasukkan *node* penyerang untuk menguji pengaruh serangan terhadap performa jaringan. Parameter yang dianalisis mencakup *Packet Delivery Ratio* (PDR), *throughput*, dan *delay*, baik dalam kondisi normal maupun saat terjadi serangan. Hasil penelitian menunjukkan bahwa serangan *Blackhole* dan *Jellyfish* secara signifikan menurunkan PDR dan *throughput*, serta meningkatkan *delay*, dengan performa jaringan terburuk tercatat saat terjadi serangan *Jellyfish*. Penelitian ini menegaskan pentingnya pengamanan *routing* dalam jaringan VANETs untuk menjaga kualitas layanan.

Pada penelitian kedua [7], tujuan penelitian untuk menganalisis kinerja protokol *routing* reaktif pada jaringan VANETs dengan model skenario jalan tol. Metode yang digunakan meliputi studi literatur, perancangan *environment* berbasis peta riil dari OpenStreetMap, serta simulasi menggunakan NS-2 untuk menguji performa protokol DSR dan AODV. Evaluasi dilakukan berdasarkan parameter *emd-to-end delay*, *packet delivery ratio*, dan *throughput*. Hasil penelitian menunjukkan bahwa protokol DSR konsisten memiliki performa lebih baik dibandingkan AODV, dengan *delay* lebih rendah, *packet delivery ratio* lebih tinggi, dan *throughput* yang lebih optimal. Temuan ini mengindikasikan bahwa DSR lebih cocok diterapkan pada skenario jaringan VANET dibandingkan AODV.

Pada penelitian ketiga [8], melakukan perbandingan dua protokol perutean, yaitu *Ad-Hoc On Demand Distance Vector* (AODV) dan *Software Defined Geographic Routing plus Recovery* (SDGR+R), dalam jaringan *Vehicular Ad Hoc Network* (VANETs) dengan studi kasus di lingkungan

Pelabuhan Lembar, Lombok. Metode yang digunakan adalah simulasi berbasis NS2, dengan integrasi konsep *Software Defined Networking* (SDN) untuk meningkatkan fleksibilitas dan efisiensi pengelolaan jaringan. Penelitian ini menguji performa kedua protokol dengan mengukur dua parameter utama, yaitu *latency* (keterlambatan pengiriman data) dan *packet delivery ratio* (PDR), untuk menilai efektivitas kondisi jaringan dinamis. Hasil penelitian ini menunjukkan bahwa protokol SDGR+R memberikan kinerja yang lebih unggul dibandingkan dengan AODV. Penelitian ini memberikan dampak positif penting mengenai pengembangan strategi *routing* yang andal dan adaptif untuk mendukung sistem transportasi cerdas.

Pada tahun 2021 Satyana et al. [9], melakukan penelitian dengan mengevaluasi kinerja protokol *routing Ad-Hoc On Demand Multipath Distance Vector* (AOMDV) dalam jaringan VANET di lingkungan perkotaan. Metode yang digunakan melibatkan simulasi dengan dua perangkat lunak, yaitu SUMO untuk merekayasa pergerakan lalu lintas kendaraan dan NS2 untuk mensimulasikan komunikasi jaringan. Evaluasi dilakukan dengan variasi kecepatan kendaraan dan variasi jumlah *node*, serta pengujian terhadap sejumlah parameter *Quality of Service* (QoS), seperti PDR, *Delay*, *Normalized Routing Load*, *Routing Overhead*, dan *Convergence Time*. Penelitian ini menunjukkan bahwa protokol AOMDV memiliki performa yang baik dalam berbagai skenario jaringan, serta menunjukkan kestabilan dan efisiensi yang menandakan bahwa protokol ini mampu mempertahankan kinerja optimal dalam kondisi jaringan VANETs yang dinamis dan padat.

Pada tahun 2020 Nurwarsito dan Setiawan [4], melakukan penelitian dengan melakukan perbandingan performansi tiga protokol *routing* pada jaringan MANET, yaitu AODV, AOMDV, dan DSR. Metode yang digunakan menggabungkan antara metode kuantitatif dengan metode kualitatif. Metode kuantitatif dilakukan dengan mengukur parameter kinerja seperti *throughput* dan *end-to-end delay*, sementara metode kualitatif memberikan analisis deskriptif terhadap fenomena yang diamati selama simulasi. Hasil penelitian menunjukkan bahwa protokol DSR memiliki performa terbaik dibandingkan protokol lainnya, dengan *throughput* tertinggi dengan nilai 215,36 Kbps dan

delay terendah dengan nilai 44,68 ms. Di sisi lain, AODV menunjukkan performa yang menurun seiring bertambahnya jumlah *node*, sedangkan AOMDV mengalami peningkatan *delay* akibat banyaknya duplikasi paket RREQ yang diterima. Studi ini memberikan pemahaman penting mengenai perbandingan kinerja protokol *routing* pada jaringan MANET dalam berbagai kondisi.

Pada penelitian keenam yang dilakukan oleh Ridwan pada tahun 2022 [10], penelitian ini menganalisis performa protokol 802.11p pada protokol *routing* AODV pada jaringan VANET menggunakan simulasi dengan perangkat lunak NS2. Metode yang diterapkan mencakup perancangan model jaringan, pengaturan parameter simulasi, serta pengukuran metrik QoS seperti *Delay*, PDR, *Throughput*, dan *Packet Drop Rate* dengan variasi jumlah kendaraan, yaitu 20 dan 30. Hasil penelitian menunjukkan bahwa peningkatan jumlah *node* memperbesar trafik dan *throughput* jaringan, sekaligus menurunkan *Packet Drop Rate*. *delay* mengalami sedikit kenaikan sekitar 4,5%, namun PDR meningkat secara signifikan sebesar 38%. Temuan ini mengindikasikan bahwa protokol AODV tetap mampu mempertahankan kinerja jaringan yang efisien meskipun dalam kondisi lalu lintas data yang lebih padat.

Chandra et al [11] melakukan penelitian yang membahas penerapan protokol *Geographic Source Routing* (GSR) pada komunikasi *Vehicle-to-Infrastructure* (V2I) di Jalan Sudirman, Kota Pekanbaru. Metode yang digunakan melibatkan survei lapangan untuk pengumpulan data koordinat, konversi file peta (map.xml dan map.cfg), dan implementasi simulasi menggunakan OMNeT++ dengan dukungan *library VeinsInet*. Penelitian menguji performa GSR melalui tiga skenario, yaitu variasi kecepatan kendaraan, kepadatan kendaraan, dan jumlah *Road Side Unit* (RSU), dengan parameter evaluasi meliputi *throughput*, PDR, dan *delay*. Hasil penelitian menunjukkan bahwa peningkatan jumlah RSU secara signifikan berdampak positif terhadap performa jaringan. *Throughput* tertinggi tercatat pada 4 RSU, sedangkan PDR terbaik diperoleh pada variasi 2 RSU. Secara keseluruhan, penelitian ini membuktikan bahwa GSR mampu memberikan kinerja yang

stabil dan efisien dalam jaringan V2I, dengan faktor kecepatan kendaraan dan jumlah RSU yang berpengaruh besar terhadap kinerja protokol.

Tabel 2.1 Penelitian Terdahulu

Penulis	Judul	Deskripsi
Alfarisi dan Nurwarsito [2]	Analisis Dampak Serangan <i>Blackhole</i> Dan <i>Jellyfish</i> Terhadap Protokol <i>Routing Ad-Hoc On Demand Distance Vector (Aodv)</i> Dengan Model Mobilitas <i>Random</i> Pada Vanet	Penelitian ini menunjukkan bahwa serangan <i>blackhole</i> dan <i>jellyfish</i> menurunkan performa, kestabilan, dan keamanan jaringan VANET berdasarkan pengujian QoS.
Dhiaulhaq [7]	Analisis Protokol <i>Routing Reactive</i> Pada Jaringan <i>Vehicular Ad Hoc Network (VANET)</i> Dengan Model Skenario Jalan Tol	Penelitian ini mensimulasikan jaringan VANET berbasis peta riil menggunakan protokol AODV dan DSR dengan standar IEEE 802.11p, serta merekomendasikan pengujian lanjutan dengan protokol lain dan tambahan elemen jaringan seperti gerbang tol dan rest area.
Ardhana et al. [8]	Evaluasi Kinerja Protokol Perutean AODV Dan SDGR+R Pada VANET Dengan Studi Kasus Pelabuhan Lembar	Penelitian ini mengevaluasi protokol AODV dan SDGR+R pada jaringan VANET berbasis SDN di Pelabuhan Lembar, menunjukkan bahwa integrasi SDN meningkatkan efisiensi dan keandalan komunikasi kendaraan.
Satyana et al. [9]	Studi Kinerja Protokol <i>Routing Ad-Hoc On Demand Multipath Distance Vector</i>	Penelitian ini mengevaluasi performa AOMDV pada VANET perkotaan berbasis simulasi NS2 dan SUMO, menunjukkan bahwa

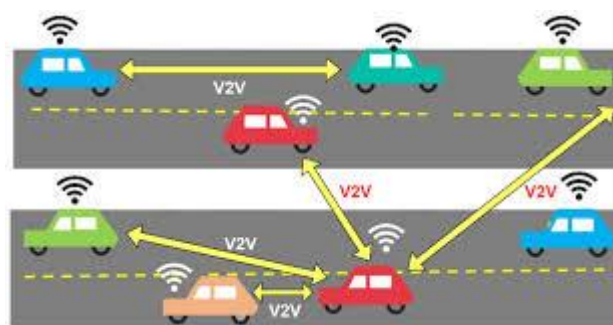
Penulis	Judul	Deskripsi
	(AOMDV) Dalam Lingkup <i>Vehicle Ad-Hoc Network</i> (VANET) Pada Daerah Perkotaan	AOMDV mampu mempertahankan kinerja baik pada berbagai jumlah <i>node</i> dan kecepatan kendaraan.
Nurwarsito dan Satriawan [4]	Performansi Protokol <i>Routing</i> AOMDV, DSR, Dan AODV Pada <i>Mobile Ad-Hoc Network</i> (MANET)	Penelitian ini membandingkan kinerja AOMDV, DSR, dan AODV pada MANET berdasarkan <i>throughput</i> dan <i>delay</i> melalui simulasi NS-2, serta merekomendasikan penelitian lanjutan dengan simulator dan skenario serangan berbeda.
Ridwan [10]	Analisa Performansi Protokol 802.11p Pada <i>Routing</i> AODV Di Jaringan VANET (<i>Vehicle Ad-Hoc Network</i>)	Penelitian ini menguji evaluasi kinerja AODV Penelitian dan menunjukkan AODV efektif pada jaringan VANET, dengan peningkatan <i>throughput</i> dan PDR meski trafik dan <i>delay</i> bertambah akibat bertambahnya jumlah <i>node</i> .
Chandra et al. [11]	Penerapan Geographic Source Routing Pada V2I Di Jalan Sudirman Kota Pekanbaru	Penelitian ini menunjukkan bahwa protokol GSR pada jaringan VANET memberikan kinerja optimal pada 50 kendaraan, kecepatan 50 km/jam, dan 2 RSU, serta mampu beradaptasi dalam berbagai kondisi jaringan.

2.2 *Vehicle to Vehicle* (V2V)

Komunikasi *vehicle to vehicle* (V2V) merupakan bentuk komunikasi nirkabel antar kendaraan yang bertujuan meningkatkan keselamatan dan kenyamanan bagi pengemudi maupun penumpang. Bentuk implementasi dari teknologi ini terdapat dalam jaringan kendaraan yang dikenal sebagai

Vehicle Ad-hoc Network (VANETs) [12]. Dalam sistem V2V, setiap kendaraan dapat saling bertukar informasi penting seperti kecepatan, arah, dan posisi secara *real-time*, guna meminimalkan risiko kecelakaan. Kendaraan-kendaraan ini terhubung melalui jaringan VANETs, di mana masing-masing mobil berfungsi sebagai *node* dalam sistem komunikasi nirkabel yang dinamis dan tersebar [13].

Komunikasi *Vehicle-to-Vehicle* (V2V) memungkinkan pertukaran data secara *real-time* antar kendaraan, seperti informasi lalu lintas, kondisi cuaca, dan situasi jalan lainnya. Teknologi ini berperan penting dalam pengoptimalan rute perjalanan, mengurangi kepadatan lalu lintas, serta meningkatkan keselamatan berkendara. Akan tetapi, terdapat beberapa tantangan yang ada sebelum V2V dapat diadopsi secara luas dalam sistem transportasi pintar. Tantangan tersebut mencakup tingginya biaya infrastruktur, kebutuhan akan protokol keamanan yang andal untuk mencegah serangan siber dan kebocoran data, serta persoalan interoperabilitas antar sistem V2V dari berbagai produsen kendaraan dan penyedia teknologi. Selain itu, masih diperlukan penelitian lebih lanjut untuk memahami integrasi optimal V2V dalam ekosistem lalu lintas cerdas agar manfaatnya dapat direalisasikan secara menyeluruh [14].



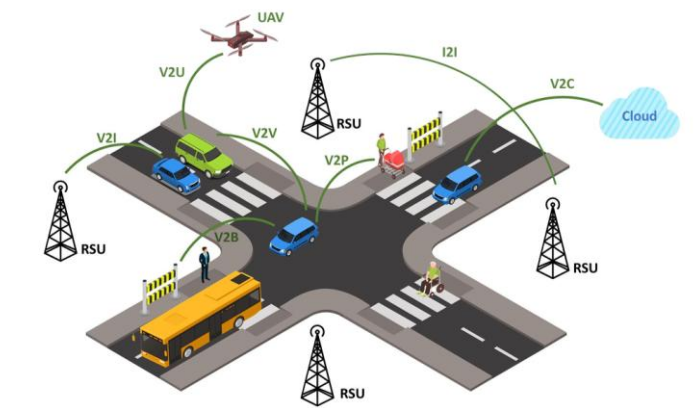
Gambar 2.1 Ilustrasi Komunikasi V2V

Gambar 2.1 merupakan gambaran ilustrasi komunikasi antar kendaraan. Pada sistem komunikasi V2V, digunakan teknologi radio Wi-Fi jarak pendek yang dikenal sebagai *Dedicated Short-Range Communication* (DSRC). Teknologi ini mengadopsi standar IEEE 802.11p dan mendukung kecepatan transmisi data antara 3 hingga 27 Mbps, tergantung pada kondisi lalu lintas

dan jaringan di jalan raya. Berdasarkan regulasi dari *Federal Communication Commission* (FCC), DSRC beroperasi pada spektrum frekuensi sebesar 75 MHz dengan pita frekuensi utama berada di 5,9 GHz. *Dedicated Short-Range Communication* (DSRC) dengan jangkauan pendek dan latensi rendah merupakan karakteristik penting dalam mendukung keamanan lalu lintas. Salah satu komponen utama dalam sistem ini adalah *On Board Unit* (OBU), yaitu perangkat yang dipasang pada kendaraan dan berfungsi untuk mengakses jaringan DSRC. OBU berperan penting dalam mendukung layanan komunikasi V2V serta meningkatkan keselamatan, efisiensi, dan kenyamanan dalam operasional kendaraan dan sistem transportasi secara keseluruhan.

2.3 *Vehicular Ad-Hoc Network* (VANET)

VANET adalah jenis jaringan *ad hoc* yang memanfaatkan kendaraan sebagai *node* dalam sistem komunikasinya. Tidak seperti jaringan konvensional yang bergantung pada infrastruktur seperti *access point*, jaringan *ad hoc* memungkinkan pertukaran data antar *node* secara langsung melalui media nirkabel. Setiap *node* dapat bergabung maupun keluar dari jaringan kapan saja, tergantung pada mobilitasnya [15]. Konsep dasar VANETs berasal dari MANET, yang memungkinkan komunikasi antar *node* tanpa infrastruktur tetap, dengan karakteristik mobilitas tinggi dan kapasitas *bandwidth* yang terbatas. Beberapa ciri khas VANETs meliputi perubahan kecepatan kendaraan yang kerap memicu terputusnya koneksi, serta variasi jenis protokol *routing* yang digunakan tergantung pada kondisi lalu lintas seperti jalan kota atau jalan tol. Selain mendukung komunikasi antar kendaraan maupun antara kendaraan dan infrastruktur, VANETs juga dirancang untuk meningkatkan keselamatan dan efisiensi lalu lintas secara keseluruhan [16].



Gambar 2.2 Komunikasi Pada VANETs [17]

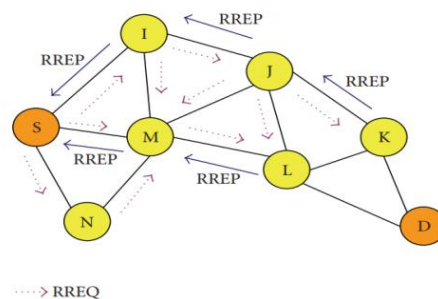
Dalam jaringan VANETs, terdapat berbagai bentuk komunikasi seperti pada Gambar 2.2 yang dirancang untuk meningkatkan keselamatan dan efisiensi transportasi. Pertama, komunikasi *Vehicle-to-Vehicle* (V2V) memungkinkan pertukaran langsung informasi antar kendaraan tanpa memerlukan infrastruktur tambahan. Kedua, komunikasi *Vehicle To Infrastructure* (V2I) atau *Vehicle To Roadside Unit* (V2R) menghubungkan kendaraan dengan infrastruktur jalan seperti RSU atau stasiun seluler untuk berbagi informasi dan akses internet. Ketiga, *Infrastructure-to-Infrastructure* (I2I) memfasilitasi pertukaran data antar infrastruktur di lokasi berbeda. Selain itu, terdapat komunikasi dengan pejalan kaki (V2P), hambatan jalan (V2B), *cloud* (V2C), drone (V2U), dan sensor (V2S). Setiap jenis komunikasi ini berperan penting dalam pertukaran data *real-time* guna mendukung sistem transportasi cerdas [17].

2.4 Protokol *Routing Ad-Hoc Demand Distance Vector* (AODV)

Protokol *routing* berfungsi untuk mengirimkan data dari *node* pengirim menuju *node* penerima melalui jaringan yang melibatkan setidaknya satu *node* sebagai perantara. Proses komunikasi antar *node* menjadi aspek krusial dalam kinerja protokol ini. Protokol *routing* bertugas menghitung dan memilih jalur paling optimal secara sistematis berdasarkan informasi *routing* yang tersedia di setiap *node* [7].

Ad-Hoc On-Demand Distance Vector (AODV) merupakan salah satu protokol *routing* reaktif yang berfungsi dengan membentuk jalur komunikasi hanya saat dibutuhkan. Protokol ini dirancang untuk menjaga informasi rute

yang selalu diperbarui melalui proses penemuan rute dan pemeliharaan *routing table*. AODV bekerja dengan memperbarui *routing table* secara dinamis melalui proses pencarian dan pemeliharaan rute. Protokol ini mulai aktif ketika *node* sumber ingin mengirimkan data ke *node* tujuan atau membutuhkan informasi jalur tertentu. Terdapat dua komponen utama dalam AODV, yaitu penemuan rute (*route discovery*) dan pemeliharaan rute (*route maintenance*). Penemuan rute melibatkan pengiriman pesan *Route Request* (RREQ) dan penerimaan *Route Reply* (RREP), sedangkan pemeliharaan rute mencakup pembaruan jalur dan penanganan gangguan melalui pesan *Route Error* (RRER). [18].



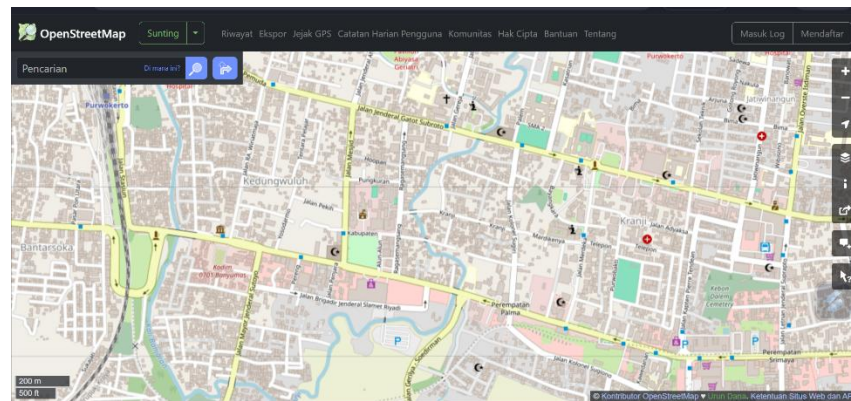
Gambar 2.3 Ilustrasi Pencarian Rute AODV

Protokol AODV memanfaatkan tiga jenis pesan utama, yaitu *Route Request* (RREQ), *Route Reply* (RREP), dan *Route Error* (RERR). Dalam tahap penemuan rute (*Route Discovery*), *node* sumber akan mengirimkan pesan RREQ secara *broadcast* yang memuat detail tujuan komunikasi. Setiap *node* yang menerima RREQ akan membentuk jalur balik (*reverse path*) menuju sumber dengan mencatat *node* pengirim sebelumnya. Proses ini berlanjut hingga pesan mencapai *node* tujuan. Jika *node* penerima mengetahui jalur ke tujuan, maka akan membalas dengan mengirimkan RREP. Sebaliknya, jika tidak mengetahui rute, RREQ akan terus diteruskan. Untuk mencegah terjadinya duplikasi, *node* yang menerima RREQ yang sama akan mengabaikannya. Setelah mencapai tujuan, *node* tujuan mengirimkan RREP kembali melalui jalur balik yang telah terbentuk sebelumnya. Selain itu, AODV memiliki fitur pemeliharaan rute (*Route Maintenance*). Jika terjadi kegagalan koneksi atau gangguan jaringan, pesan RERR akan dikirimkan

melalui jalur yang ada untuk memberi tahu node terkait mengenai kegagalan tersebut agar informasi rute dapat diperbarui. [18].

2.5 *Open Street Map (OSM)*

OpenStreetMap (OSM) merupakan *platform* pemetaan digital bersifat *open source* yang memungkinkan siapa saja untuk mengakses, mengedit, dan memperbarui data peta secara gratis guna menciptakan peta yang lebih detail, akurat, dan selalu terkini. Berbeda dengan layanan peta komersial seperti Google Maps, OSM memiliki keunggulan dalam hal kelengkapan data karena terdapat lebih dari dua juta kontributor terdaftar yang secara aktif melakukan pembaruan setiap saat. OSM jugadapat digunakan untuk memberikan informasi navigasi di wilayah tertentu [19].

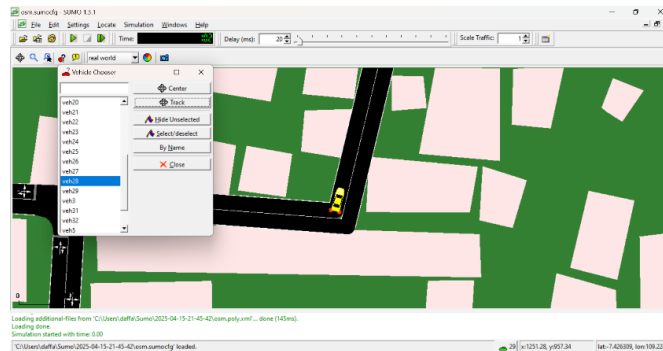


Gambar 2. 4 Tampilan OSM

2.6 *Simulation of Urban Mobility (SUMO)*

SUMO adalah simulator keadaan lalu lintas *real-time* yang digunakan secara luas dalam penelitian jaringan *ad-hoc* kendaraan (VANETs). Simulator ini banyak dimanfaatkan dalam berbagai proyek, seperti pengembangan strategi mengemudi otomatis maupun sistem manajemen transportasi. Melalui SUMO, pengguna dapat mengamati perilaku kendaraan secara langsung selama proses simulasi berlangsung. SUMO merupakan *tools* yang sangat cocok digunakan dalam penelitian yang berkaitan dengan lalu lintas kendaraan, terutama di area dengan tingkat kepadatan tinggi [20]. Perangkat lunak ini mulai dikembangkan pada tahun 2001 dan diluncurkan sebagai proyek *open-source* pada tahun 2002 oleh Daniel Krajzewicz, Eric Nikolay, serta Michael Behrisch. Dalam simulasi VANET, SUMO berperan penting

dalam membentuk model kendaraan atau *node* berdasarkan skenario dan kondisi yang telah ditentukan sebelumnya[19].



Gambar 2.5 Tampilan SUMO

2.7 OMNeT++

OMNeT++ merupakan perangkat lunak simulasi yang digunakan untuk memodelkan dan menganalisis komunikasi dalam jaringan kendaraan (VANETs). OMNeT++ memungkinkan pengujian protokol *routing* yang berfungsi mengatur jalur pengiriman data antar kendaraan dalam kondisi jaringan yang dinamis. Penggunaan OMNeT++ dapat diperkuat dengan integrasi bersama SUMO untuk menggambarkan kondisi lalu lintas nyata. Hasil simulasi memungkinkan analisis terhadap kinerja protokol *routing* berdasarkan metrik seperti PDR, *throughput*, dan *delay*. OMNeT++ menjadi aplikasi penting untuk mendukung pengembangan sistem komunikasi antar kendaraan secara efisien dan akurat.



Gambar 2.6 Tampilan Awal OMNeT++

2.8 *Throughput*

Throughput merupakan salah satu parameter penting dalam jaringan komunikasi yang menggambarkan seberapa banyak data yang berhasil dikirim dan diterima dalam jangka waktu tertentu. Dalam konteks jaringan, *throughput* diartikan sebagai kecepatan transfer data efektif yang diukur dalam satuan bit per detik (bps). Sistem jaringan akan mencatat jumlah total paket data yang berhasil diterima oleh *node* tujuan selama periode waktu tertentu, kemudian membaginya dengan durasi waktu tersebut untuk memperoleh nilai *throughput*. Dengan demikian, *throughput* menunjukkan efisiensi jaringan dalam memproses dan menyampaikan data secara sukses dari sumber ke tujuan [21].

Tabel 2.2 Nilai Standarisasi *Throughput* [22]

Kategori	<i>Throughput</i>	Indeks
Sangat Bagus	>2,1 Mbps	4
Bagus	1,2 Mbps – 2,1 Mbps	2
Sedang	700 Kbps – 1200 Kbps	3
Buruk	0 – 338 Kbps	1

Perhitungan *throughput* dapat dilakukan dengan menggunakan rumus:

$$\textit{Throughput} = \frac{\textit{Jumlah data yang diterima}}{\textit{Waktu pengiriman data}} \quad (1)$$

Nilai *throughput* memberikan gambaran tentang efisiensi jaringan dalam memindahkan data. Semakin tinggi *throughput*, semakin cepat dan efisien suatu jaringan dalam mentransfer data. *Throughput* yang rendah menunjukkan adanya masalah dalam jaringan, seperti kemacetan atau paket yang hilang, yang menyebabkan transfer data berjalan lambat.

2.9 *End to End Delay (EED)*

Delay pada jaringan komunikasi merujuk pada lamanya waktu yang diperlukan untuk mentransmisikan paket data dari *node* pengirim ke *node* penerima. Salah satu bentuk *delay* tersebut adalah *End-to-End Delay*, yaitu rata-rata waktu yang dibutuhkan oleh sebuah paket untuk melewati seluruh rute jaringan hingga mencapai tujuan akhirnya [23].

Tabel 2.3 Nilai Standarisasi *End-to-end Delay* [24]

Kategori	Delay	Indeks
Sangat Bagus	<150 ms	4
Bagus	151-300 ms	2
Sedang	301-450 ms	3
Buruk	>450 ms	1

Perhitungan *delay* dapat dilakukan dengan rumus:

$$End - to - end\ delay = \frac{Total\ delay}{Jumlah\ paket\ yang\ diterima} \quad (2)$$

Waktu tunda ini dipengaruhi oleh jumlah *node*, proses penemuan rute, dan antrian transmisi. Jika jumlah *node* yang terlibat semakin banyak atau jarak antar *node* semakin jauh, maka *delay* cenderung meningkat. Selain itu, proses pengiriman dan balasan rute seperti RREQ dan RREP juga memengaruhi lama *delay* [5].

2.10 Packet Loss

Packet loss merupakan kondisi dalam jaringan di mana sebagian paket data yang dikirim tidak berhasil mencapai tujuan. Kondisi ini bisa disebabkan oleh berbagai hal, seperti kemacetan lalu lintas data (*congestion*), tabrakan antar paket (*collision*), gangguan pada sinyal, atau masalah pada perangkat keras jaringan. Kehilangan paket ini dapat menurunkan kinerja jaringan karena membutuhkan proses pengiriman ulang data, yang pada akhirnya membebani sistem dan mengurangi efisiensi komunikasi. *Packet loss* juga dapat menyebabkan peningkatan *jitter*, yaitu ketidakstabilan waktu pengiriman antar paket, yang berdampak buruk pada layanan yang membutuhkan koneksi stabil seperti *streaming* video, panggilan suara, dan bermain game *online* [25].

Tabel 2.4 Standarisasi Nilai *Packet Loss*

Kategori	Packet Loss	Indeks
Sangat Bagus	0 %	4
Bagus	3 %	2
Sedang	15 %	3

Kategori	Packet Loss	Indeks
Buruk	25 %	1

Perhitungan *packet loss* dapat dilakukan dengan menggunakan perumusan:

$$Packet\ Loss = \frac{Paket\ data\ dikirim - Paket\ data\ diterima}{Paket\ data\ dikirim} \times 100\% \quad (3)$$

2.11 Packet Delivery Ratio (PDR)

Packet Delivery Ratio (PDR) adalah metrik penting yang digunakan untuk mengukur efektivitas protokol *routing* dalam jaringan komunikasi, terutama pada jaringan *ad hoc* seperti *Vehicular Ad-Hoc Network* (VANET). PDR menunjukkan sejauh mana paket data yang dikirim dari *node* sumber berhasil diterima oleh *node* tujuan, terutama dalam kondisi jaringan yang terus berubah akibat mobilitas tinggi [26]. Perhitungan PDR dapat dilakukan dengan rumus:

$$PDR = \frac{jumlah\ paket\ yang\ diterima}{jumlah\ paket\ yang\ dikirim} \times 100\% \quad (4)$$

Nilai PDR yang tinggi mencerminkan keandalan protokol *routing* dalam menyampaikan data secara efisien, meskipun dalam kondisi jaringan yang kompleks. Sebaliknya, nilai PDR yang rendah menandakan adanya masalah seperti gangguan sinyal, rute yang rusak, atau mobilitas *node* yang tinggi, yang menyebabkan kehilangan paket selama transmisi.

Tabel 2.5 Nilai Standarisasi *Packet Delivery Ratio* [27]

Kategori	Packet Delivery Ratio (%)	Indeks
Sanget Bagus	95%-100%	4
Bagus	80%-95%	3
Sedang	50%-80%	2
Buruk	<50%	1

2.12 Replit

Replit adalah sebuah IDE *Integrated Development Environment* (IDE) yang berbasis web dan bertujuan mempermudah pengguna dalam menulis serta mengeksekusi kode secara langsung lewat *browser*. Platform ini sangat

bermanfaat dalam kegiatan belajar dan pengembangan proyek pemrograman, khususnya dalam bahasa seperti *Python*, karena tidak memerlukan instalasi *software* tambahan di perangkat yang digunakan. Dengan menggunakan koneksi internet, siapa pun dapat menggunakan Replit dari berbagai perangkat. Keunggulan lainnya adalah adanya fitur kolaborasi yang memungkinkan banyak pengguna mengerjakan kode bersama secara waktu nyata (*real-time*), hal ini membuat replit sangat cocok untuk kerja tim kecil atau pembelajaran bersama. Sebelumnya platform ini dikenal dengan nama Repl.it, lalu pada tahun 2021 berubah menjadi Replit dengan alamat situs baru, yaitu Replit.com. Replit mendukung lebih dari 50 bahasa pemrograman seperti HTML dan JavaScript, serta mampu membantu membuat prototipe API dan aplikasi web dengan mudah, Replit menjadi pilihan efisien dan praktis bagi siapa saja yang ingin belajar atau mengembangkan proyek secara fleksibel langsung dari *browser* [28].

2.13 *Veins*

Veins adalah *library* dan perangkat bantu untuk OMNeT++ yang digunakan dalam simulasi kendaraan jalan raya yang saling berkomunikasi. *Veins* mendukung penelitian di bidang *Vehicular Ad Hoc Networks* (VANETs) maupun *Intelligent Transportation Systems* (ITS), dan didistribusikan sebagai perangkat lunak *open source*, sehingga bebas untuk diunduh, dimodifikasi, dan digunakan. *Library* ini mencakup simulasi lengkap untuk kendaraan yang berkomunikasi, terutama mobil dan truk menggunakan teknologi nirkabel berbasis WLAN, seperti standar IEEE 802.11, WAVE, ETSI ITS-G5, dan ARIB T-109. *Veins* juga dapat digunakan untuk memodelkan *node* bergerak lain seperti pejalan kaki, sepeda, kereta, hingga drone (UAV), serta untuk teknologi komunikasi lain seperti LTE dan *Visible Light Communication* (VLC) [29].

2.14 *Website*

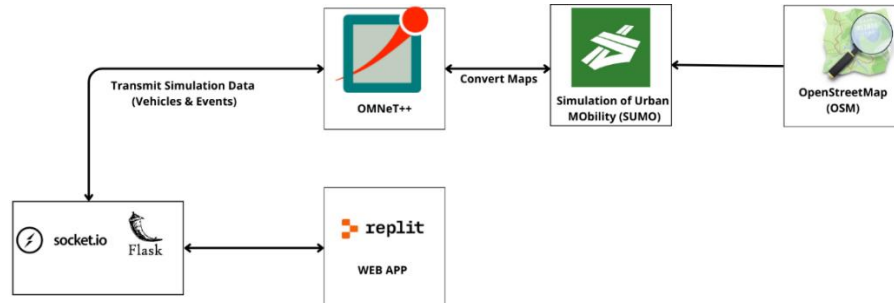
Website merupakan kumpulan halaman web yang saling terhubung dan dilengkapi dengan berbagai file pendukung seperti gambar, video, serta dokumen digital lainnya yang tersimpan di dalam web server. Umumnya,

website diakses melalui jaringan internet. Secara umum, *website* tersusun atas folder dan file yang berisi perintah atau fungsi tertentu, seperti untuk menampilkan informasi maupun mengelola data. Halaman-halaman web ini biasanya ditulis menggunakan format *Hyper Text Markup Language* (HTML) dan diakses melalui protokol HTTP atau HTTPS, yang bertugas mentransfer data dari server ke pengguna melalui aplikasi browser [30].

BAB III

PERANCANGAN SISTEM

3.1 Desain Sistem



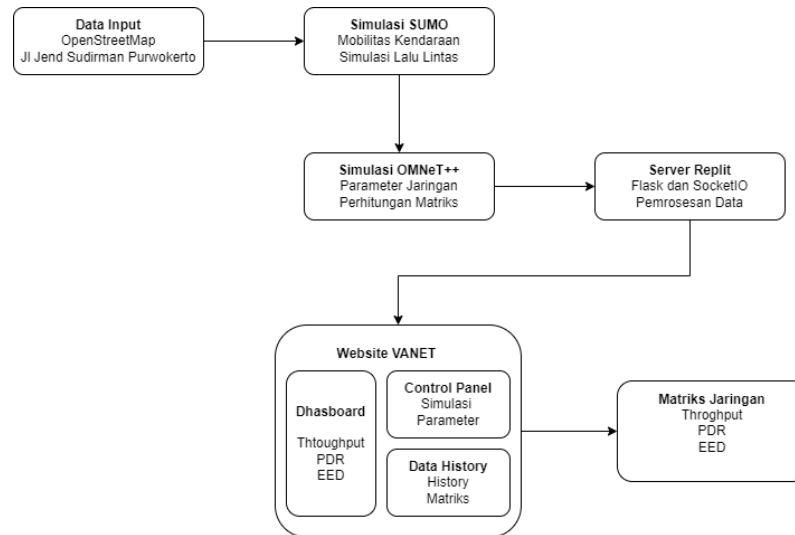
Gambar 3.1 Rancangan Sistem

Gambar 3.1 Rancangan Rancangan Sistem ini dirancang untuk mensimulasikan komunikasi antar kendaraan VANETs berbasis data lalu lintas realistis menggunakan empat komponen utama: OSM, SUMO, OMNeT++/Veins, dan sebuah *Website* untuk menampilkan hasil simulasi. Proses dimulai dengan OSM sebagai sumber data peta mentah yang kemudian dikonversi menggunakan *tool netconvert* dari SUMO menjadi format peta jaringan *.ned.xml*. Peta ini digunakan dalam SUMO untuk membuat simulasi pergerakan kendaraan. Hasil dari sumo ini dihubungkan ke OMNeT++ melalui protokol *Traffic Control Interface (TraCI)* sehingga pergerakan kendaraan dalam simulasi jaringan dapat sinkron dengan simulasi lalu lintas.

OMNeT++ bersama *framework veins* melakukan simulasi komunikasi antar kendaraan *Vehicle-to-Vehicle (V2V)* maupun komunikasi kendaraan antar infrastruktur *Vehicle-to-Infrastructure (V2I)*. Dalam proses ini, sistem akan menghasilkan data statistik seperti *throughput*, PDR, dan EED. Data tersebut disimpan atau dikirimkan melalui backend script atau API untuk kemudian ditampilkan secara *real-time* maupun dalam bentuk log *history* sebuah *Website dashboard*. *Website* ini menampilkan grafik dan informasi prforma jaringan sehingga dapat digunakan sebagai analisis hasil simulasi VANET secara visual dan interaktif.

3.1.1 Diagram Blok

Proses simulasi mobilitas kendaraan dan analisis jaringan dijelaskan melalui blok diagram berikut, yang menggambarkan alur dari pengambilan data hingga visualisasi hasil simulasi.



Gambar 3.2 Blok Diagram

Gambar 3.2 Blok menjelaskan Menggambarkan sebuah sistem simulasi mobilitas kendaraan yang terintegrasi dengan analisis jaringan komunikasi. Proses dimulai dengan pengambilan data geografis dari OSM yang menggambarkan wilayah tertentu Jl Jend Sudirman Purwokerto. Data ini kemudian digunakan untuk membangun model simulasi pergerakan kendaraan di area tersebut.

Simulasi pergerakan kendaraan ini dilakukan menggunakan SUMO, sebuah perangkat lunak yang memungkinkan pemodelan pergerakan kendaraan secara realistis, menggambarkan pola lalu lintas yang ada di dunia nyata. Dengan SUMO, sistem dapat mensimulasikan dinamika lalu lintas, seperti kepadatan kendaraan, kecepatan, dan jalur yang dilalui.

Setelah simulasi mobilitas selesai, langkah berikutnya melibatkan penggunaan OMNeT++, yang berfokus pada analisis jaringan komunikasi yang terkait dengan simulasi kendaraan tersebut. OMNeT++ digunakan untuk menghitung parameter jaringan penting seperti *throughput*, PDR, dan EED. Analisis ini penting untuk menilai kinerja sistem komunikasi kendaraan dalam jaringan VANETs.

Data yang diperoleh dari simulasi dan analisis jaringan ini kemudian diolah dan disajikan melalui server *Replit* yang menggunakan Flask dan SocketIO, yang memfasilitasi pengolahan dan penyajian data secara *real-time*. Hasil akhirnya adalah sebuah *Website* VANETs yang menyajikan *Dashboard* informasi. *Dashboard* ini menampilkan metrik-metrik kunci seperti *throughput*, PDR, dan EED, serta memungkinkan pengguna untuk mengakses data secara *real-time* dan melihat riwayat data untuk analisis lebih lanjut.

Secara umum, sistem ini tidak hanya menyediakan simulasi mobilitas kendaraan, tetapi juga menganalisis bagaimana jaringan komunikasi kendaraan VANETs berfungsi dalam kondisi lalu lintas yang dinamis, sehingga memungkinkan perencanaan dan optimisasi infrastruktur komunikasi di area tersebut.

3.1.2 Fungsi dan Fitur

Sistem simulasi yang dirancang dalam penelitian ini mengintegrasikan berbagai alat dan metode untuk memberikan gambaran menyeluruh mengenai mobilitas kendaraan dan kinerja jaringan komunikasi dalam lingkungan perkotaan. Proses ini melibatkan tiga komponen utama: SUMO untuk simulasi mobilitas kendaraan, OMNeT++ untuk analisis jaringan komunikasi, dan *Website* VANETs untuk visualisasi hasil pengujian. Berikut adalah penjelasan tentang fungsi dan fitur dari masing-masing komponen dalam sistem ini.

1. Pengambilan Data dari *OpenStreetMap*

Fungsi:

Pengambilan Data Geografis Dasar: Fungsi pertama adalah untuk memperoleh data geografis dari *OpenStreetMap* untuk area Jl. Jend. Sudirman Purwokerto. Data ini mencakup informasi tentang jaringan jalan, persimpangan, dan infrastruktur lainnya yang akan menjadi dasar simulasi pergerakan kendaraan dan analisis jaringan.

Fitur:

Integrasi Data Geografis: Mengimpor peta dan data jalan dari *OpenStreetMap* yang mencakup jalan, zona hijau, dan bangunan di

wilayah yang akan disimulasikan.

Persiapan Data untuk Simulasi: Data geografis ini menjadi fondasi utama untuk membangun model pergerakan kendaraan dan simulasi jaringan.

2. Simulasi Mobilitas Kendaraan dengan *Simulation of Urban Mobility* (SUMO)

Fungsi:

Simulasi Pergerakan Kendaraan: SUMO digunakan untuk mensimulasikan pergerakan kendaraan di jalan-jalan yang telah dianalisis sebelumnya. Kendaraan bergerak sesuai dengan pola lalu lintas yang realistis, menggambarkan kondisi jalan dan kepadatan kendaraan di area tersebut.

Fitur:

Visualisasi Lalu Lintas: Tampilan grafis simulasi lalu lintas, yang menunjukkan pergerakan kendaraan di atas peta wilayah perkotaan berdasarkan data dari *OpenStreetMap*.

Analisis Lalu Lintas: SUMO mengoptimalkan pola lalu lintas dan pergerakan kendaraan untuk analisis lebih lanjut mengenai dinamisnya kendaraan di jalan.

3. Simulasi Menggunakan OMNeT++

Fungsi:

Simulasi Jaringan Komunikasi: Setelah simulasi mobilitas kendaraan selesai, OMNeT++ digunakan untuk menganalisis komunikasi antara kendaraan dalam jaringan VANET. Proses ini meliputi perhitungan parameter jaringan seperti *throughput*, PDR, dan EED.

Fitur:

Simulasi Jaringan VANET: Memodelkan komunikasi antara *node* dalam jaringan VANET yang mencakup kendaraan dan infrastruktur jalan.

Pengukuran Kinerja Jaringan: Melakukan pengukuran terhadap berbagai parameter teknis yang mempengaruhi kinerja jaringan, termasuk pengiriman paket dan latensi komunikasi.

4. Pengolahan Data Menggunakan Server *Replit* (Flask dan SocketIO)

Fungsi:

Pengolahan Data Simulasi dan Jaringan: Data yang dihasilkan dari simulasi kendaraan dan analisis jaringan kemudian diproses melalui server yang dibangun menggunakan Flask dan SocketIO. Ini memungkinkan pengolahan data secara *real-time* dan integrasi antara simulasi kendaraan dan komunikasi jaringan.

Fitur:

Pengolahan data *Real-Time*: Flask dan SocketIO memungkinkan pengolahan data secara langsung dan interaktif, mempersiapkan data untuk visualisasi di *Website*.

Real-Time Data Integration: Menghubungkan simulasi kendaraan dan analisis jaringan secara langsung, memungkinkan pemantauan hasil secara *real-time*.

5. Visualisasi Hasil Simulasi Melalui *Website* VANET

Fungsi:

Visualisasi Hasil Simulasi Mobilitas dan Jaringan: *Website* VANET berfungsi sebagai antarmuka untuk menampilkan hasil simulasi mobilitas kendaraan dan analisis jaringan. Parameter seperti *throughput*, PDR, dan EED dapat dipantau secara langsung dalam bentuk grafik dan tabel.

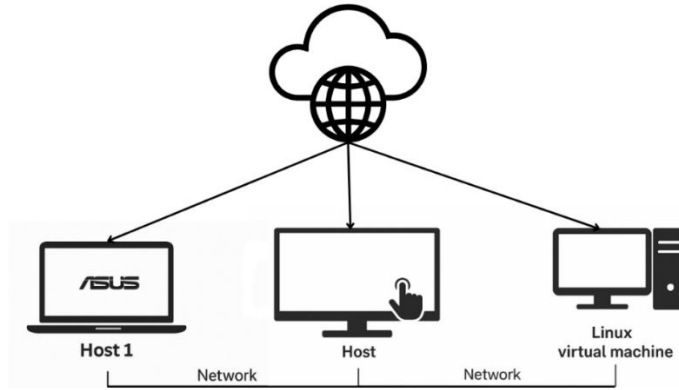
Fitur:

Dashboard Interaktif: Menyediakan tampilan grafis dengan parameter kunci jaringan seperti *throughput*, PDR, dan EED dalam grafik deret waktu yang mudah dipahami.

Control Panel Real-Time: Memungkinkan pengguna untuk mengontrol dan mengakses simulasi jaringan secara langsung, serta memantau parameter secara *real-time*.

Data History: Fitur ini memungkinkan pengguna untuk melihat dan membandingkan data historis dari simulasi sebelumnya, memberikan wawasan lebih dalam tentang perubahan performa jaringan dari waktu ke waktu.

3.2 Desain Perangkat Keras



Gambar 3.3 Desain Perangkat Keras

Gambar 3.3 topologi jaringan yang ditampilkan menggambarkan arsitektur koneksi berbasis *cloud* yang menghubungkan tiga perangkat berbeda melalui infrastruktur internet. *Host 1* berupa laptop ASUS terhubung ke jaringan, sedangkan *host* kedua merupakan komputer *desktop*, dan *host* terakhir adalah mesin virtual Linux terpasang pada *desktop*. *Cloud* yang terpampang di bagian atas menandakan konektivitas internet yang memungkinkan komunikasi dan pertukaran data antar ketiga perangkat tersebut, menciptakan infrastruktur jaringan yang fleksibel dan terintegrasi.

3.2.1 Spesifikasi Komponen

Untuk menjalankan SUMO dan OMNeT++, penelitian ini menggunakan dua perangkat keras berbeda:

Host 1 - Laptop ASUS (*Remote Client*)

Digunakan sebagai laptop utama untuk mengakses *Host 2* melalui jaringan. Spesifikasinya adalah:

Tabel 3.1 Laptop ASUS

No	Perangkat keras	Minimum Spec
1	Processor	Intel(R) Celeron(R) N4500 @ 1.10GHz
2	RAM	8 GB (7.78 GB usable)
3	ROM	Windows 11 Home, 64-bit
4	VGA	Tidak ada

Host 2 – Komputer Remote (*Remote Server*)

Komputer berspesifikasi tinggi untuk menjalankan dan mengelola *Virtual Machine Linux*. Spesifikasinya adalah:

Tabel 3.2 Komputer *Remote*

No	Perangkat keras	Minimum Spec
1	Processor	11TH Gen Intel(R) Core(TM) i7-11700F @ 2.50GHz
2	RAM	128 GB (128 GB usable)
3	ROM	64-bit Oprating System, x64-based processor
4	VGA	Touch Support with 10 Touch Points

Komputer ketiga Virtual Machine Linux

Mesin virtual yang berjalan di *Host 2*, dan dapat diakses dari *Host 1* dan *Host 2* melalui jaringan. Spesifikasinya adalah:

Tabel 3.3 Virtual *Machine* Linux

No	Perangkat keras	<i>Minimum Spec</i>
1	Processor	2 core (1 inti per processor)
2	RAM	4 GB
3	ROM	20 GB
4	VGA	<i>Auto detect</i>

Konfigurasi ini memungkinkan penggunaan sumber daya komputasi yang optimal untuk menjalankan simulasi SUMO dan OMNeT++ dengan efisien.

3.3 Desain Perangkat Lunak

Pengembangan sistem VANET menuntut integrasi perangkat lunak mutakhir yang mampu memodelkan interaksi komunikasi kompleks antar kendaraan dengan tingkat akurasi tinggi. Perancangan arsitektur perangkat lunak menjadi elemen kritis dalam membangun sistem simulasi yang andal dan presisi. Berikut akan dipaparkan detail komponen teknologi yang digunakan dalam simulasi VANET, mengungkap keragaman teknologi dan pendekatan *framework* yang dimanfaatkan untuk menghasilkan simulasi komunikasi kendaraan yang mendalam dan komprehensif.

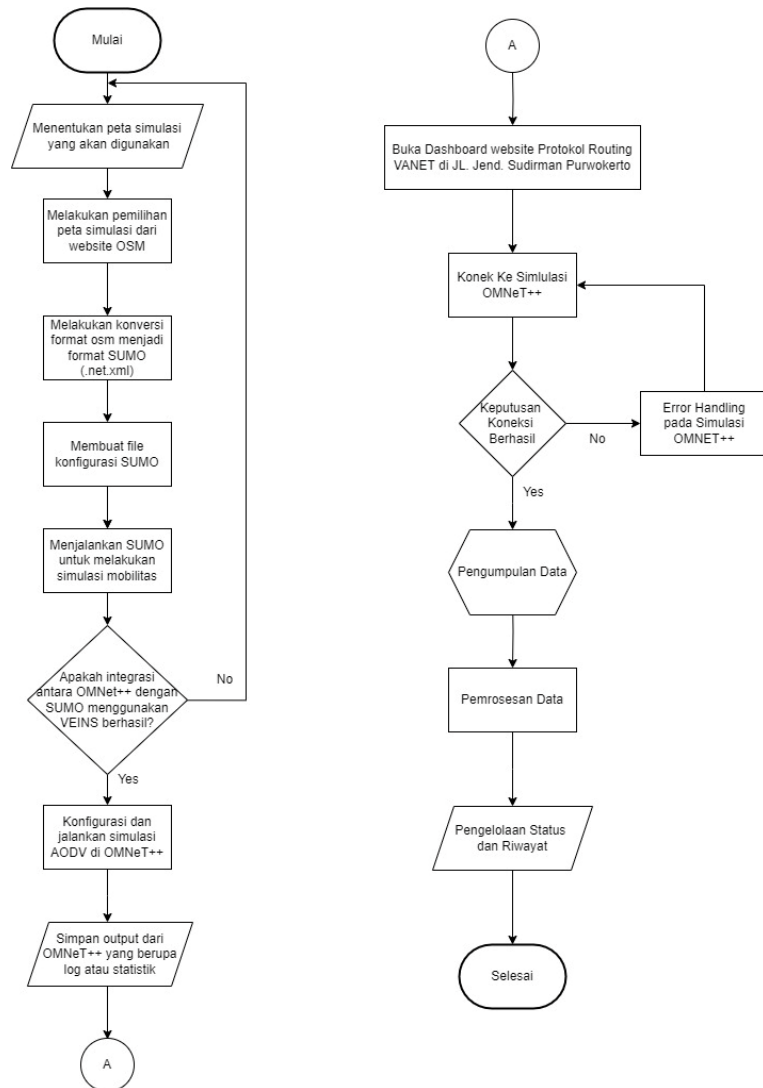
Tabel 3.4 Spesifikasi Perangkat Lunak

Komponen	Nama Aplikasi	Bahasa Pemrogramman	<i>Framework</i>	Versi
Pemetaan	OpenStreetMap (OSM)	<i>Python</i>	OSMnx	1.3.0
Simulasi Mobilitas	Simulation of Urban Mobility (SUMO)	C++	-	1.14.1
Simulasi Jaringan	OMNeT++ dengan <i>veins</i>	C++	<i>Veins</i> 5.2	5.0
Server Backend	Replit	<i>Python</i>	Flask, SocketIO	2.1.0
Frontend Web	<i>Website</i> VANET	JavaScript	React.js	18.2

Pada Tabel 3.4 Spesifikasi Perangkat Lunak adalah Sistem VANET yang dirancang mengintegrasikan berbagai perangkat lunak canggih untuk mensimulasikan komunikasi antar kendaraan secara komprehensif. Dimulai dengan *OpenStreetMap* menggunakan *Python* dan *library OSMnx* versi 1.3.0, sistem mampu mengekstraksi data geografis dan jaringan jalan dengan presisi tinggi. Simulasi mobilitas kendaraan dilakukan melalui SUMO dengan bahasa pemrograman C++, memungkinkan pemodelan pergerakan kendaraan yang realistis dalam lingkungan perkotaan.

Komponen kritis selanjutnya adalah OMNeT++ dengan *Veins* versi 5.2, yang menggunakan C++ untuk mensimulasikan komunikasi jaringan kendaraan secara mendalam. *Backend* sistem dibangun menggunakan *Replit* dengan *Python*, *Framework* Flask dan SocketIO versi 2.1.0, yang memfasilitasi pemrosesan data dan komunikasi *real-time*. Tahap akhir melibatkan pengembangan *Website* VANETs menggunakan *JavaScript* dengan *React.js* versi 18.2, yang menyediakan antarmuka interaktif untuk visualisasi dan kontrol simulasi. Keseluruhan arsitektur perangkat lunak dirancang untuk menghasilkan simulasi VANETs yang akurat, responsif, dan dapat memberikan wawasan mendalam tentang komunikasi antar

kendaraan.



Gambar 3.4 Alur Kerja Sistem

Gambar 3.4 menunjukkan alur kerja sistem simulasi VANET berbasis integrasi SUMO dan OMNeT++ yang dikaitkan dengan *Dashboard* pemantauan berbasis web. Proses dimulai dengan penentuan dan pemilihan peta simulasi dari platform OSM, kemudian dikonversi ke format yang sesuai dengan SUMO (.net.xml), diikuti dengan pembuatan file konfigurasi dan pelaksanaan simulasi mobilitas. Setelah itu, data hasil simulasi SUMO diintegrasikan dengan OMNeT++ menggunakan *Veins* untuk menjalankan simulasi protokol AODV, dan *output* yang dihasilkan disimpan dalam bentuk log atau statistik. Proses berlanjut pada *Dashboard* web yang berfungsi untuk menghubungkan ke simulasi OMNeT++,

memeriksa koneksi, dan jika berhasil, mengumpulkan serta memproses data yang diterima. Data ini kemudian dikelola untuk keperluan pemantauan status dan penyimpanan riwayat, sehingga menghasilkan sistem pemantauan performa jaringan VANET yang *real-time* dan historis.

VANETs dalam penelitian ini berfokus pada pemodelan mobilitas kendaraan dan jaringan komunikasi di sepanjang Jl. Jend. Sudirman, Purwokerto, dengan pendekatan terintegrasi menggunakan SUMO, OMNeT++, dan *framework Vehicles in Network Simulation (Veins)*. Simulasi ini memanfaatkan data geospasial dunia nyata dan pola mobilitas kendaraan yang realistis untuk mengevaluasi metrik kinerja jaringan seperti *Throughput*, PDR, dan EED. Selain itu, dikembangkan sistem pemantauan berbasis web untuk memvisualisasikan metrik-metrik tersebut secara *real-time*, memberikan analisis komprehensif tentang kinerja VANET.

Proses simulasi dimulai dengan menyiapkan jaringan jalan menggunakan data geospasial dari OSM. File OSM, ‘map_pwt.osm’, dikonversi menjadi file jaringan yang kompatibel dengan SUMO, yaitu ‘map_pwt.net.xml’, menggunakan perintah ‘netconvert’:

```
Netconvert -osm-files map_pwt.osm -output-file map_pwt.net.xml -
geometry.remove -roundabouts.guess -ramps.guess -junctions.join -
tls.guess-signals -tls.discard-simple -tls.join
```

Perintah ini mengoptimalkan jaringan jalan dengan menghapus detail geometri yang tidak diperlukan, mendeteksi bundaran, mengenali jalur masuk/keluar, menggabungkan persimpangan yang berdekatan, dan mengatur sistem lampu lalu lintas untuk memastikan aliran lalu lintas yang realistis. Selanjutnya, perjalanan kendaraan dibuat secara acak menggunakan skrip ‘randomTrips.py’:

```
randomTrips.py -n map_pwt.nwt.xml -e 1000 -o map_pwt.trips.xml
```

Skrip ini menghasilkan perjalanan kendaraan acak dalam periode simulasi 1000 detik, dengan hasil disimpan dalam file ‘map_pwt.trips.xml’. Perjalanan ini kemudian dikonversi menjadi rute kendaraan menggunakan perintah ‘duarouter’:

```
duarouter -n map_pwt.net.xml route-files map_pwt.trips.xml -o
map_pwt.rou.xml --ignore-errors
```

File ‘map_pwt.rou.xml’ yang dihasilkan berisi rute detail untuk setiap kendaraan, mencakup titik awal, tujuan, dan jalur yang dipilih berdasarkan jaringan jalan. File konfigurasi SUMO (`sumocfg``) kemudian dibuat untuk mengoordinasikan simulasi:

```
<configuration>
  <input>
    <net-file value="map_pwt.net.xml"/>
    <route-files value="new_routes.rou.xml"/>
  </input>
  <time>
    <begin value="0"0"/>
    <end value="10000"/>
  </time>
  <output>
    <fcd-output value="fcd.xml"/>
  </output>
</configuration>
```

Konfigurasi ini menentukan file jaringan (‘map_pwt.net.xml’) dan rute (‘new_routes.rou.xml’), mengatur durasi simulasi dari 0 hingga 10.000 detik, dan menghasilkan Floating Car Data (FCD) ke file ‘fcd.xml’ untuk analisis lebih lanjut. FCD mencakup posisi dan kecepatan kendaraan, yang penting untuk integrasi dengan OMNeT++ melalui *Traffic Control Interface* (TRaCI).

Pada OMNeT++, simulasi dikonfigurasi menggunakan file ‘omnetpp.ini’, yang mendefinisikan parameter kunci untuk jaringan komunikasi VANETs:

```
Debug-on-errors = true
Print-undisposed = true
```

```

Sim-time-limit = 500s
**.scalar-recording = true
**.vector-recording = true
*.playgroundSizeX = 2500m
*.playgroundSizeY = 2500m
*.playgoroundSizeZ = 500m
*.manager.updateInterval = 1s
*.manager.host = "localhost"
*.manager.port = 9999
*.manager.autoShutdown = true
*.manager.lunchConfig = xmldoc("pwt.laumchd.xml")
*.rou[0].mobility.x = 1600
*.rsu[0].mobility.y = 755
*.rsu[0].mobility.z = 3
*.rsu[*].applType = "TRaCIDemoRSU11p"
*.rsu[*].app.headerLength = 80bit
*.rsu[*].appl.sendBeacons = false
*.rsu[*].appl.dataOnSch = false
*.rsu[*].appl.beaconInterval = 1s
*.rsu[*].appl.beaconUserPriority = 7
*.rsu[*].appl.dataUserPriority = 5
*.rsu[*].nic.phy80211p.antennaOffistZ = 0m
*.node[*].veinsmobility.set.HostSpeed = false
*.node[*].veinsmobility.speed = 40km/h
*.node[0].veinsmobility.x = 0
*.node[0].veinsmobility.y = 0
*.node[0].veinsmobility.z = 0
[Config WithBeaconing]
*.rsu[*].appl.sendBeacons = true
*.node[*].appl.dataOnSch = true
[Config Experiment16]
*.*.nic.mac.1609_4.useServiceChannel = true

```

```
*.node[*].appl.dataOnSch = true
*.rsu[*].app.dataOnSch = true
[Config Experiment16]
description = "80 vehicles with max speed 40 km/h"
*.*.nic.mac1609_4.useServiceChannel = false
*.node[*].appl.dataOnSch = false
*.rsu[*].appl.dataOnSch = false
*.node[*].appl.sendBeacons = true
*.rsu[*].appl.sendBeacons = true
```

Konfigurasi ini mengaktifkan *debugging*, membatasi simulasi hingga 500 detik, dan mencatat data skalar dan vektor untuk analisis. Area simulasi diatur seluas 2500m x 2500m x 50m, dengan posisi kendaraan diperbarui setiap detik melalui TRaCI. *Roadside Unit (RSU)* ditempatkan pada koordinat (1600m, 755m, 3m) dan dikonfigurasi untuk komunikasi V2I menggunakan aplikasi TRaCIDemoRSU11p. Kendaraan bergerak dengan kecepatan konstan 40 km/jam, dengan posisi awal ditentukan secara manual. Konfigurasi juga mencakup opsi untuk mengaktifkan *beaconing* dan beralih antara *Control Channel (CCH)* dan *Service Channel (SCH)* untuk transmisi data. Konfigurasi Experiment16 mensimulasikan 80 kendaraan dengan *beaconing* diaktifkan tetapi membatasi komunikasi pada CCH.

Framework Veins mengintegrasikan SUMO dan OMNeT++ dengan menyinkronkan data mobilitas kendaraan dari SUMO dengan jaringan komunikasi di OMNeT++. TRaCI memfasilitasi pertukaran data secara *real-time*, memungkinkan OMNeT++ untuk memperbarui posisi *node* dan mensimulasikan protokol *routing* seperti AODV. Simulasi ini memodelkan parameter jaringan seperti jangkauan sinyal, interferensi, dan penundaan transmisi, dengan kinerja dievaluasi menggunakan metrik seperti PDR, EED, dan *Throughput*.

Sistem pemantauan berbasis web, yang dibangun menggunakan Flask dan SocketIO, memvisualisasikan hasil simulasi secara *real-time*. Sistem ini terdiri dari *Dashboard* yang menampilkan grafik langsung untuk

Throughput, PDR, dan EED, Panel Kontrol untuk mengelola parameter simulasi, dan halaman Riwayat untuk meninjau data sebelumnya. Server Flask memproses data mentah dari simulasi OMNeT++, memperbarui variabel global, dan mengirimkan metrik ke klien yang terhubung melalui SocketIO. Alur kerja dimulai dengan inisialisasi server, diikuti dengan pembuatan koneksi ke simulasi OMNeT++. Data mentah diproses untuk mengekstrak metrik utama, yang kemudian ditampilkan di antarmuka web tanpa memerlukan penyegaran halaman manual. Data historis disimpan untuk analisis perbandingan, dan sistem mendukung perintah untuk memulai, menghentikan, atau mengatur ulang simulasi.

Pengaturan simulasi terintegrasi ini menyediakan *platform* yang kuat untuk mengevaluasi kinerja VANET dalam kondisi lalu lintas realistis, dengan antarmuka web yang meningkatkan kemampuan untuk memantau dan menganalisis perilaku jaringan secara dinamis.

3.4 Skenario Pengujian

Skenario pengujian dalam simulasi ini dirancang menggunakan aplikasi simulator SUMO dan OMNeT++ yang dihubungkan melalui integrasi menggunakan *VEINS*. Terdapat dua skenario utama yang dirancang untuk mengukur performa jaringan VANET dengan menggunakan protokol *routing* yang diuji. Skenario Pengujian 1 berfokus pada pengaruh jumlah kendaraan terhadap kualitas layanan (QoS). Dalam skenario ini, jumlah kendaraan divariasikan sebanyak 10, 25, dan 50 unit, dengan kecepatan konstan 35 km/jam serta hanya menggunakan 1 *Road Side Unit* (RSU).

Skenario Pengujian kedua dirancang untuk menganalisis dampak variasi kecepatan *node* terhadap performa jaringan. Tiga variasi kecepatan yang diuji adalah 30 km/jam, 40 km/jam, dan 50 km/jam, dengan jumlah kendaraan berjumlah 35 dan 1 RSU yang digunakan. Semua skenario dilakukan untuk mengevaluasi performa jaringan berdasarkan tiga parameter QoS utama, yaitu *Throughput*, *Packet Delivery Ratio* (PDR), dan *End-to-End Delay*. Kedua skenario ini dirancang untuk memberikan gambaran menyeluruh terhadap pengaruh parameter mobilitas dan

infrastruktur terhadap kinerja protokol *routing* pada jaringan VANET.

Tabel 3.5 Skenario Pengujian

Skenario	Parameter 1	Parameter 2	Parameter Pengujian
Skenario Pengujian 1	Jumlah Kendaraan 10	Kecepatan kendaraan berada di 35 Km/h dengan jumlah RSU satu	Pengukuran QoS berupa <i>Throughput</i> , <i>Packet Delivery Ratio</i> dan <i>end-to-end delay</i>
	Jumlah Kendaraan 25		
	Jumlah Kendaraan 50		
Skenario Pengujian 2	Variasi Kecepatan <i>node</i> 30 KM/h	Pengujian dilakukan dengan jumlah <i>node</i> kendaraan 35 dengan jumlah RSU satu	Pengukuran QoS berupa <i>Throghput</i> , <i>Packet Dekivery Rasio</i> dan <i>End to End Delay</i>
	Variasi Kecepatan <i>node</i> 40		
	Variasi Kecepatan <i>node</i> 50		

BAB IV HASIL DAN ANALISIS

4.1 Hasil Percobaan

Pada tahap hasil percobaan ini, proses simulasi jaringan menjadi titik puncak dari keseluruhan metodologi yang telah dirancang. Integrasi antara *Simulation of Urban Mobility*, OMNeT++, dan pengembangan *Website* merupakan langkah kritis dalam memvisualisasikan dan menganalisis dinamika pergerakan dan komunikasi dalam lingkungan perkotaan yang kompleks. Melalui pendekatan simulasi yang komprehensif ini, penelitian mampu menghadirkan gambaran nyata tentang interaksi dan performa jaringan dalam konteks mobilitas perkotaan.



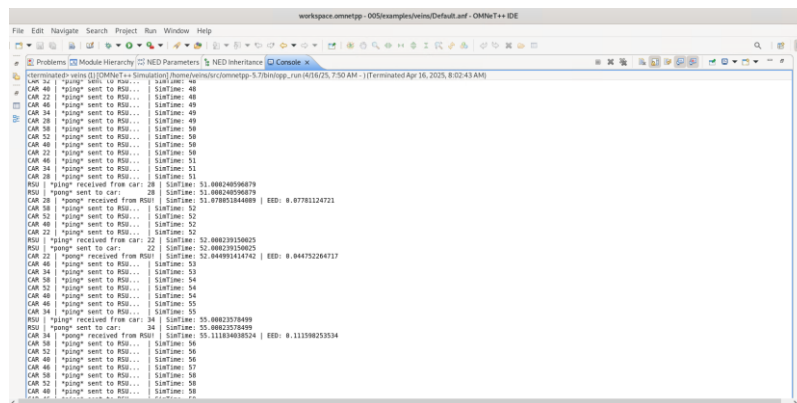
Gambar 4.1 Tampilan SUMO saat Simulasi

Gambar 4.1 Tampilan SUMO saat menampilkan tampilan simulasi lalu lintas menggunakan *software* SUMO, yang memperlihatkan jaringan jalan berwarna hitam dengan persimpangan dan bundaran, dilengkapi area hijau sebagai zona taman serta blok merah muda dan kuning untuk merepresentasikan bangunan. Beberapa kendaraan terlihat bergerak pada jalan, dengan informasi teknis di bagian bawah menunjukkan data *loading* dan versi SUMO 1.20 yang digunakan untuk menganalisis dan mengoptimalkan pola lalu lintas di area perkotaan tersebut.



Gambar 4.2 Proses Pengiriman Paket pada OMNeT++

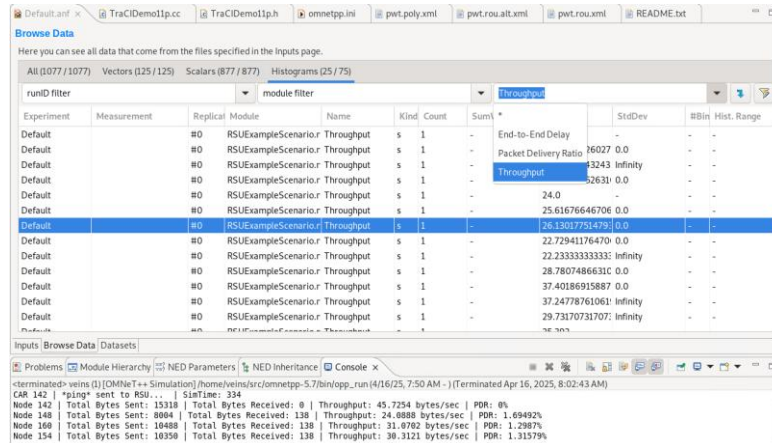
Gambar 4.2 Proses Pengiriman Paket pada **OMNeT++** menampilkan visualisasi proses pengiriman paket data pada simulator OMNeT++, dengan jaringan kompleks yang direpresentasikan oleh garis-garis merah membentuk topologi area perkotaan atau jaringan komunikasi. Bagian atas gambar menunjukkan tampilan grafis simulasi dengan beberapa *node* berwarna biru yang tampak tersebar di jaringan, sementara bagian bawah menampilkan *log* pengiriman paket dalam format teks yang berisi informasi detail seperti waktu *event*, ID modul, *interface*, dan status pengiriman dengan berbagai parameter teknis. Simulasi ini kemungkinan digunakan untuk menganalisis performa protokol komunikasi dalam skenario jaringan perkotaan.



Gambar 4.3 Log Pengiriman Pesan antar Node

Gambar 4.3 Log Pengiriman Pesan antar *Node* dalam sebuah sistem jaringan. Log ini mencatat aktivitas komunikasi dengan detail *timestamp*, *node* pengirim dan penerima, serta status pengiriman pesan. Terlihat beberapa baris Log yang menampilkan pola komunikasi berurutan dengan

format yang konsisten, menunjukkan proses pertukaran informasi yang berjalan di antara *node* dengan *node* dalam jaringan. *Log* ini berfungsi sebagai catatan penting untuk analisis performa komunikasi, pemecahan masalah, dan verifikasi bahwa semua pesan telah terkirim dengan benar antar *node* dalam sistem yang sedang diuji.



Gambar 4.4 Hasil Pengujian QoS Setelah Simulasi

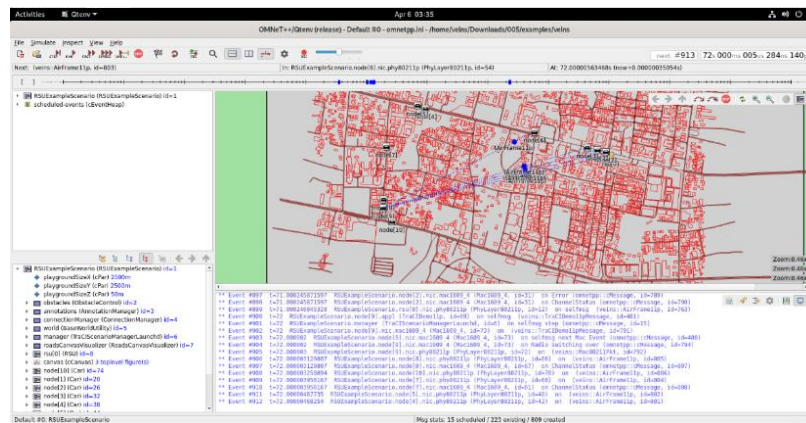
Gambar 4.4 Hasil Pengujian QoS Setelah Simulasi jaringan. Pada bagian atas terdapat tabel dengan parameter *Throughput* sebagai metrik utama yang diukur, menunjukkan kinerja komunikasi antar *node*. Nilai *Throughput* tercatat bervariasi antara 22-37 kbps dengan beberapa nilai ditandai "*infinity*", mengindikasikan laju data yang sangat baik pada koneksi tersebut. Bagian bawah gambar menampilkan ringkasan statistik simulasi dengan total paket yang dikirim/diterima beserta *Throughput* rata-rata 36.3121 *bytes/sec*. Data ini memberikan evaluasi komprehensif tentang performa jaringan yang telah disimulasikan.

Experiment	Measurement	Replication	Module	Name	Kind	Count	SumWeights	Mean	StdDev	#Bins	Hist. Range
Default	#0		RSUExampleScenario.node[0].appl	End-to-End Delay	s	1	0.05944326626	0.05944326626	0.0000	-	-
Default	#0		RSUExampleScenario.node[0].appl	Throughput	s	1	-	36.0000	0.0000	-	-
Default	#0		RSUExampleScenario.node[0].appl	Packet Delivery Ratio	s	1	0.0555555555555556	0.0555555555555556	0.0000	-	-
Default	#0		RSUExampleScenario.node[2].appl	End-to-End Delay	s	1	0.07781124721	0.07781124721	0.0000	-	-
Default	#0		RSUExampleScenario.node[2].appl	Throughput	s	1	34.027397260274	34.027397260274	0.0000	-	-
Default	#0		RSUExampleScenario.node[2].appl	Packet Delivery Ratio	s	1	0.0555555555555556	0.0555555555555556	0.0000	-	-
Default	#0		RSUExampleScenario.node[1].appl	End-to-End Delay	s	1	0.044752264717	0.044752264717	0.0000	-	-
Default	#0		RSUExampleScenario.node[1].appl	Throughput	s	1	35.432432432432	35.432432432432	Infinity	-	-
Default	#0		RSUExampleScenario.node[1].appl	Packet Delivery Ratio	s	1	0.052631578947368	0.052631578947368	Infinity	-	-
Default	#0		RSUExampleScenario.node[3].appl	End-to-End Delay	s	1	0.111598253534	0.111598253534	Infinity	-	-
Default	#0		RSUExampleScenario.node[3].appl	Throughput	s	1	32.684210526316	32.684210526316	0.0000	-	-
Default	#0		RSUExampleScenario.node[3].appl	Packet Delivery Ratio	s	1	0.0555555555555556	0.0555555555555556	0.0000	-	-
Default	#0		RSUExampleScenario.node[6].appl	End-to-End Delay	s	1	0.023705234548	0.023705234548	0.0000	-	-
Default	#0		RSUExampleScenario.node[6].appl	Throughput	s	1	-	24.0000	0.0000	-	-
Default	#0		RSUExampleScenario.node[6].appl	Packet Delivery Ratio	s	1	0.434027778	0.434027778	0.0000	-	-
Default	#0		RSUExampleScenario.node[12].appl	End-to-End Delay	s	1	0.135037248461	0.135037248461	Infinity	-	-
Default	#0		RSUExampleScenario.node[12].appl	Throughput	s	1	25.616766467066	25.616766467066	0.0000	-	-
Default	#0		RSUExampleScenario.node[12].appl	Packet Delivery Ratio	s	1	0.032258064516129	0.032258064516129	0.0000	-	-
Default	#0		RSUExampleScenario.node[11].appl	End-to-End Delay	s	1	0.089108266468	0.089108266468	Infinity	-	-
Default	#0		RSUExampleScenario.node[11].appl	Throughput	s	1	26.130177514793	26.130177514793	0.0000	-	-
Default	#0		RSUExampleScenario.node[11].appl	Packet Delivery Ratio	s	1	0.03125	0.03125	0.0000	-	-
Default	#0		RSUExampleScenario.node[13].appl	End-to-End Delay	s	1	0.01892171813	0.01892171813	Infinity	-	-
Default	#0		RSUExampleScenario.node[13].appl	Throughput	s	1	22.729411764706	22.729411764706	0.0000	-	-
Default	#0		RSUExampleScenario.node[13].appl	Packet Delivery Ratio	s	1	0.035714285714286	0.035714285714286	0.0000	-	-
Default	#0		RSUExampleScenario.node[15].appl	End-to-End Delay	s	1	0.060651269944	0.060651269944	Infinity	-	-
Default	#0		RSUExampleScenario.node[15].appl	Throughput	s	1	22.233333333333	22.233333333333	Infinity	-	-
Default	#0		RSUExampleScenario.node[15].appl	Packet Delivery Ratio	s	1	0.03448275862069	0.03448275862069	0.0000	-	-

Gambar 4.5 Hasil Pengukuran dalam Excel

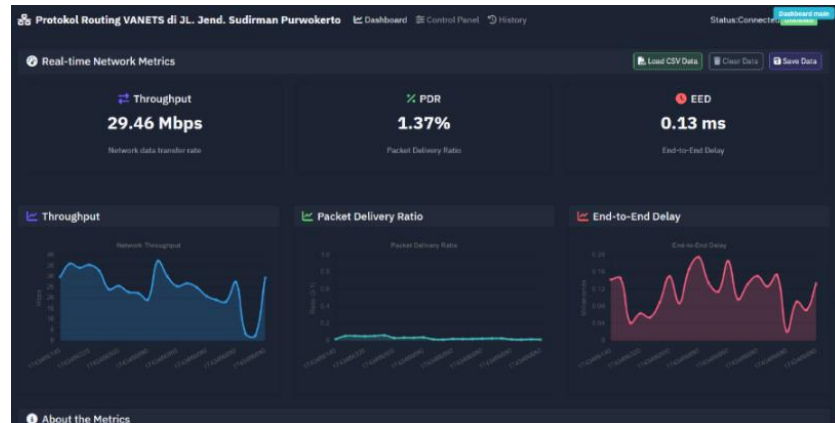
Gambar 4.5 Hasil Pengukuran dalam Excel menampilkan hasil

pengukuran kinerja jaringan yang telah diekspor ke format Excel untuk analisis lebih lanjut. Tabel ini menyajikan parameter *QoS* secara lengkap dengan kolom-kolom yang terstruktur: *Experiment*, *Measurement*, *Publication*, *Module*, dan *Name*. Data mencakup nilai *End-to-End Delay*, *Packet Delivery Ratio*, dan *Throughput* untuk berbagai skenario pengujian. Terlihat variasi nilai *Throughput* antara 22-35 kbps, dengan beberapa nilai "infinity" pada *End-to-End Delay*. *Packet Delivery ratio* umumnya berada pada rentang 0.6-1.0, menunjukkan tingkat keberhasilan pengiriman data yang baik. Representasi data dalam format Excel ini memudahkan perbandingan dan analisis statistik lebih lanjut.



Gambar 4.6 Hasil Simulasi OMNeT++ di Jl Sudirman Purwokerto

Gambar 4.6 Hasil Simulasi OMNeT++ di Jl Sudirman Purwokerto menampilkan hasil simulasi jaringan menggunakan OMNeT++ yang dilakukan di area Jalan Sudirman, Purwokerto. Visualisasi ini terdiri dari dua bagian utama: peta jalan (atas) dan *log* simulasi (bawah). Pada bagian peta, terlihat jaringan jalan berwarna merah yang mewakili infrastruktur Jl. Sudirman dengan beberapa *node* komunikasi yang ditandai. Panel kiri menampilkan hierarki komponen simulasi, sementara bagian bawah menunjukkan *log* aktivitas komunikasi antar *node* dengan kode warna yang berbeda untuk jenis pesan yang dikirimkan. Simulasi ini memberikan gambaran visual tentang bagaimana komunikasi antar kendaraan (VANET) beroperasi dalam kondisi lalu lintas realistis di area perkotaan Purwokerto.



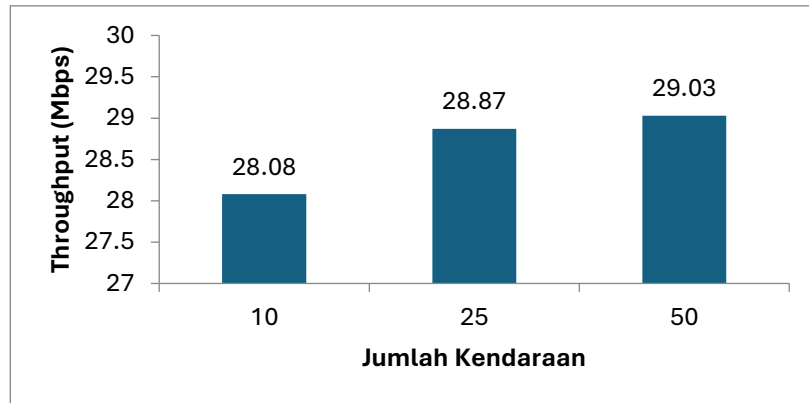
Gambar 4.7 Tampilan Website Hasil Pengujian

Gambar 4.7 Tampilan Website Pengujian menampilkan antarmuka *Website* monitoring untuk hasil pengujian protokol *routing* VANETS di Jl. Jend. Sudirman Purwokerto. *Dashboard* interaktif ini menampilkan metrik jaringan *real-time* dengan tiga parameter utama: *Throughput* (29.46 Mbps), PDR sebesar 1.37%, dan EED 0.13 ms. Ketiga parameter tersebut divisualisasikan dalam bentuk grafik deret waktu yang menunjukkan fluktuasi performa jaringan selama periode pengujian. *Website* ini dihosting pada platform *Replit* untuk memudahkan kolaborasi dan pembagian hasil pengujian secara *online* kepada anggota tim. Tampilan yang informatif dan responsif memungkinkan pemantauan kualitas jaringan VANETS secara efektif.

4.1.1 Pengujian Skenario 1

1. Parameter *Throughput*

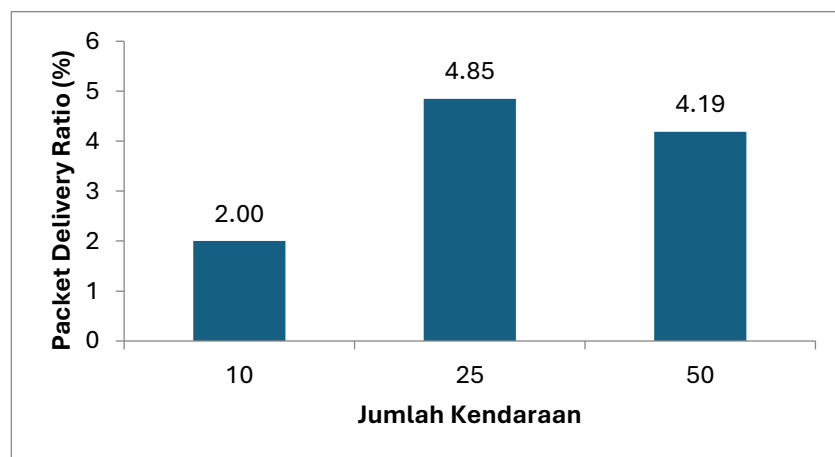
Gambar 4. 8 merupakan hasil pengujian parameter *throughput* pada skenario 1, dengan variasi jumlah *node* yang digunakan yaitu 10, 25, dan 50 *node* dengan menggunakan 1 RSU. Berdasarkan **Gambar 4. 8** hasil pengujian diperoleh saat terjadi pelonjakan *node* yang signifikan akan mempengaruhi parameter *throughput* yang di dapat, berdasarkan hasil tersebut dapat diketahui bahwa semakin bertambahnya *node* akan berpengaruh kepada parameter *throughput*.



Gambar 4. 8 Grafik Pengujian *Throghput* Skenario 1

2. Parameter *Packet Delivery Ratio*

Gambar 4.9 menunjukkan hasil pengujian parameter *Packet Delivery Rasio* pada protokol routing AODV dalam jaringan VANET dengan skenario kecepatan kendaraan tetap 35 km/jam dan jumlah RSU sebanyak 1 unit. Berdasarkan grafik, PDR tercatat paling rendah sebesar 2,00% ketika jumlah kendaraan adalah 10, dan meningkat secara signifikan menjadi 4,85% ketika jumlah kendaraan bertambah menjadi 25. Namun, ketika jumlah kendaraan mencapai 50, PDR sedikit menurun menjadi 4,19%.

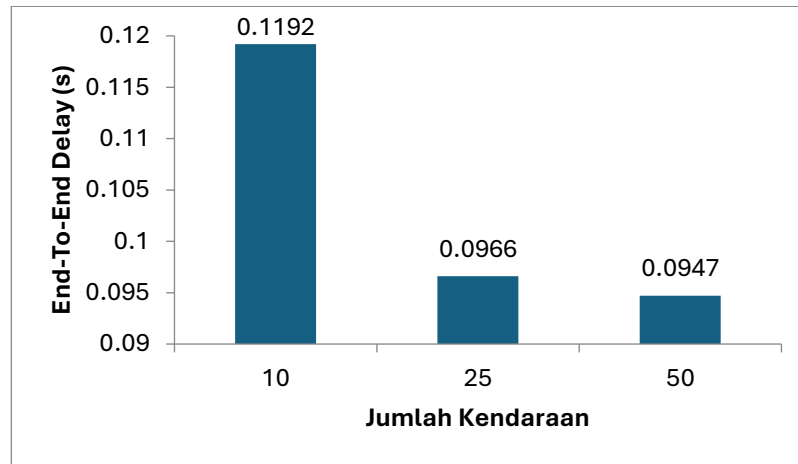


Gambar 4.9 Grafik Pengujian *Packet Delevry Ratio* Skenario 1

3. Parameter *End-to-End Delay*

Gambar 4.10 merupakan hasil pengujian parameter *End-to-End Delay* pada skenario 1, dengan variasi jumlah *node* yang digunakan yaitu 10, 25, dan 50 *node* dengan menggunakan 1 RSU. Berdasarkan **Gambar**

4.10 dapat diketahui bahwa nilai pengujian parameter *end-to-end delay* mengalami penurunan disetiap penambahan *node*.

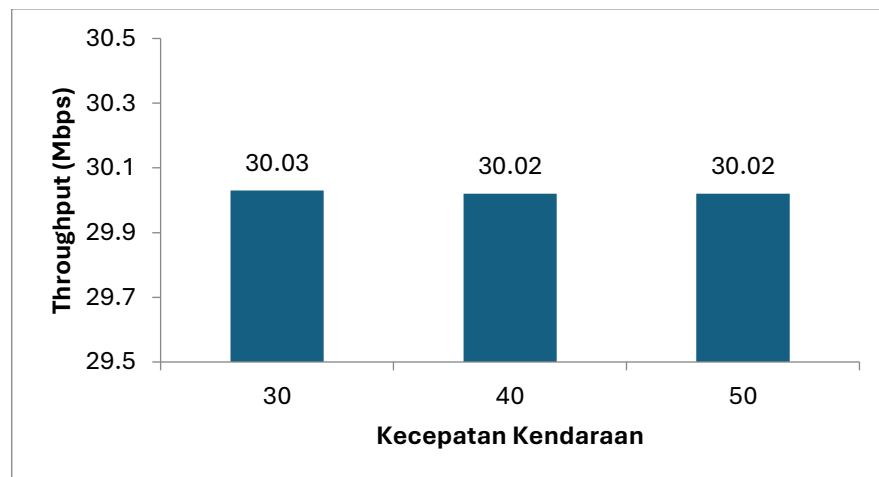


Gambar 4.10 Grafik Pengujian *End-to-End Delay* Skenario 1

4.1.2 Pengujian Skenario 2

1. Parameter *Throughput*

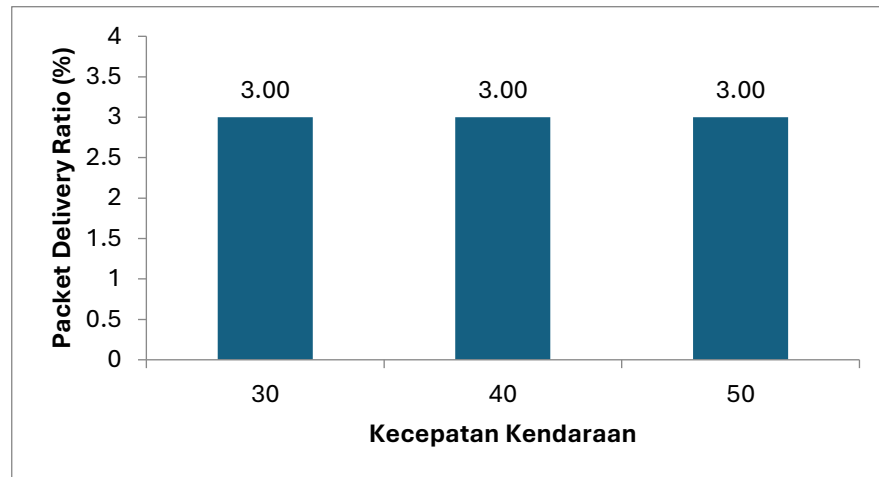
Gambar 4. 11 menunjukkan hasil pengujian parameter *throughput* pada Protokol *Routing* dalam jaringan VANET dengan skenario variasi kecepatan kendaraan sebesar 30 km/jam, 40 km/jam, dan 50 km/jam. Dari grafik terlihat bahwa perubahan kecepatan kendaraan tidak memberikan pengaruh yang signifikan terhadap nilai *throughput*, di mana *throughput* relatif stabil pada kisaran 30,02–30,03 Mbps.



Gambar 4. 11 Grafik Pengujian *Throghput* Skenario 2

2. Parameter *Packet Delivery Ratio*

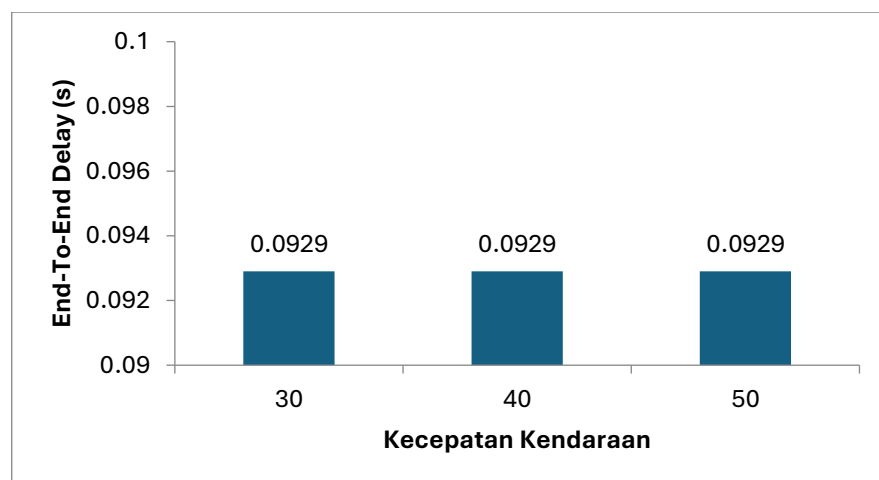
Gambar 4. 12 merupakan hasil pengujian parameter *Packet Delivery Ratio* pada skenario 2, dengan variasi kecepatan pergerakan *node* yang digunakan yaitu 30, 40, dan 50 Km/h dengan menggunakan 1 RSU.



Gambar 4. 12 Grafik Pengujian *Packet Delivery Ratio* Skenario 2

3. Parameter *End-to-End Delay*

Gambar 4. 13 menunjukkan hasil pengujian parameter *End-to-End Delay* protokol *routing* AODV dalam jaringan VANET dengan variasi kendaraan 30 km/jam, 40 km/jam, 50 km/jam. Berdasarkan grafik, nilai *delay* yang dihasilkan pada ketiga variasi kecepatan adalah 0,0929 detik, yang berarti tidak terjadi perubahan nilai *delay* meskipun kecepatan kendaraan meningkat.



Gambar 4. 13 Grafik Pengujian *End-to-End Delay* Skenario 2

4.2 Analisis

Simulasi jaringan menggunakan SUMO, OMNeT++, dan *Website* monitoring menjadi tahapan penting untuk mengukur performa komunikasi antar *node* dalam jaringan VANET. Melalui pendekatan simulasi yang komprehensif, analisis dilakukan untuk memahami bagaimana parameter *End-to-End Delay*, *Throughput*, dan PDR berinteraksi dalam skenario jaringan perkotaan. Berikut adalah hasil analisis dari setiap tahap simulasi:

1. Proses simulasi jaringan dalam penelitian ini merupakan titik puncak dari keseluruhan metodologi yang telah diterapkan. Dengan mengintegrasikan SUMO, OMNeT++, dan pengembangan *Website*, simulasi ini memungkinkan pemahaman yang lebih mendalam mengenai dinamika pergerakan dan komunikasi dalam lingkungan perkotaan yang kompleks. Melalui pendekatan simulasi yang holistik ini, penelitian berhasil menggambarkan secara nyata interaksi dan performa jaringan dalam konteks mobilitas perkotaan, yang sangat relevan untuk mengoptimalkan pola lalu lintas serta komunikasi dalam area perkotaan.
2. Simulasi dilakukan menggunakan OMNeT++ untuk memberikan gambaran mengenai pengiriman paket data dalam jaringan komunikasi perkotaan. Jaringan kompleks yang terlihat menunjukkan topologi perkotaan dengan distribusi *node* yang tersebar, serta proses pengiriman pesan yang mencakup berbagai parameter teknis. *Log* pengiriman paket yang tercatat dalam format teks memberikan informasi mendalam terkait waktu, status pengiriman, serta modul yang terlibat. Analisis terhadap *log* ini sangat penting untuk mengevaluasi performa protokol komunikasi yang digunakan, serta untuk memastikan bahwa komunikasi antar *node* berjalan dengan efektif dan sesuai dengan skenario jaringan yang diuji.
3. Pengujian kualitas layanan *Quality of Service* (QoS) pada jaringan yang disimulasikan memberikan insight terkait performa komunikasi antar *node*. Dengan parameter seperti *throughput*, PDR, dan *end-to-end delay*, pengujian ini memberikan gambaran jelas mengenai kinerja

jaringan secara keseluruhan. Hasil pengujian menunjukkan variasi nilai *throughput* yang cukup signifikan, dengan nilai rata-rata yang memadai untuk memastikan komunikasi yang efisien dalam kondisi lalu lintas yang dinamis. Selain itu, representasi data dalam format Excel memudahkan perbandingan antar skenario pengujian, serta memungkinkan analisis lebih lanjut untuk menentukan keberhasilan pengiriman paket dan kestabilan jaringan yang diuji.

4.2.1 Analisis Hubungan Jumlah *Node* terhadap *Throughput*

Berdasarkan Gambar 4. 8 hasil simulasi parameter *Throughput* pada protocol AODV diuji dengan tiga variasi jumlah kendaraan (*node*), yaitu 10, 25, dan 50 *node*, dengan kecepatan konstan 35Km/h dan jumlah RSU yang tersedia adalah 1. Pada saat jumlah *node* sebanyak 10 kendaraan, *throughput* tercatat sebesar 28,08 Mbps. Nilai ini meningkat menjadi 28,87 Mbps saat jumlah *node* ditambah menjadi 25, dan kembali mengalami peningkatan menjadi 29,03 Mbps ketika jumlah *node* mencapai 50.

Peningkatan *throughput* yang sejalan dengan bertambahnya jumlah kendaraan menunjukkan bahwa protokol AODV mampu mengelola lalu lintas data dengan baik meskipun dalam kondisi jaringan yang lebih padat. Hal ini mengindikasikan bahwa keberadaan lebih banyak *node* memberikan peluang lebih besar dalam pembentukan rute alternatif dan perbaikan rute secara dinamis. Dengan kata lain, meningkatnya densitas jaringan memberikan keuntungan dalam hal konektivitas dan efisiensi rute, sehingga jumlah data yang berhasil dikirim dalam jaringan juga meningkat.

Dengan bertambahnya jumlah kendaraan dalam jaringan VANET, protokol AODV memperoleh lebih banyak pilihan jalur atau rute untuk mengirimkan data, yang pada akhirnya meningkatkan peluang terbentuknya jalur komunikasi yang lebih stabil dan efisien sehingga *throughput* meningkat. Penambahan jumlah *node* juga meningkatkan tingkat konektivitas antar kendaraan, yang berarti kemungkinan terputusnya rute menjadi lebih kecil karena adanya banyak *node* cadangan yang siap menggantikan saat *node* keluar dari jangkauan. Konektivitas yang tinggi ini mengurangi gangguan serta retransmisi data, dan turut

memperbaiki kinerja *throughput*. Selain itu, semakin banyaknya *node* memungkinkan distribusi beban lalu lintas data menjadi lebih merata ke berbagai rute, sehingga mengurangi kepadatan atau kemacetan pada satu jalur tertentu dan meningkatkan efisiensi transmisi data.

Secara keseluruhan, protokol AODV menunjukkan performa *throughput* yang cukup stabil dan cenderung meningkat terhadap penambahan jumlah kendaraan dalam skenario mobilitas sedang (35 Km/h) dengan satu RSU.

4.2.2 Analisis Hubungan Jumlah Node Terhadap *Packet Delivery Ratio*

Berdasarkan **Gambar 4. 12** hasil pengujian parameter *Packet Delivery Ratio* (PDR) dengan skenario kecepatan kendaraan konstan sebesar 35 km/jam dan jumlah RSU sebanyak 1 unit, diperoleh nilai PDR untuk variasi jumlah *node* sebanyak 10, 25, dan 50 kendaraan masing-masing sebesar 2,00%, 4,85%, dan 4,19%. Hasil ini menunjukkan bahwa peningkatan jumlah *node* dari 10 ke 25 kendaraan memberikan dampak positif terhadap kinerja protokol AODV, yang ditandai dengan meningkatnya nilai PDR secara signifikan. Hal ini terjadi karena penambahan jumlah kendaraan dalam jaringan memperbesar kemungkinan terbentuknya rute komunikasi yang lebih stabil dan efisien, sehingga lebih banyak paket data berhasil dikirim dan diterima tanpa mengalami kehilangan.

Namun, ketika jumlah *node* ditingkatkan lebih lanjut menjadi 50 kendaraan, nilai PDR justru mengalami penurunan dari 4,85% menjadi 4,19%. Penurunan ini dapat disebabkan oleh meningkatnya kepadatan jaringan yang mengakibatkan terjadinya interferensi antar *node*, peningkatan jumlah tabrakan paket (*packet collision*), dan tingginya beban lalu lintas data yang harus ditangani oleh RSU yang hanya berjumlah 1. Selain itu, sifat reaktif dari protokol AODV menyebabkan *overhead routing* bertambah seiring bertambahnya *node*, sehingga proses penemuan dan pemeliharaan rute menjadi lebih kompleks dan menimbulkan *delay* serta potensi kehilangan paket yang lebih besar.

Secara keseluruhan, hasil pengujian parameter ini menunjukkan nilai

yang sangat rendah dan belum mencerminkan performa yang baik untuk jaringan komunikasi seperti VANET. Salah satu faktor penyebabnya adalah penggunaan hanya 1 RSU, yang menimbulkan hambatan ketika banyak kendaraan mencoba mengaksesnya secara bersamaan. Selain itu, kondisi lingkungan yang sangat dinamis membuat jalur komunikasi antar kendaraan cepat berubah, sementara protokol AODV memiliki keterbatasan dalam membangun rute secara cepat, sehingga berdampak pada rendahnya nilai PDR yang dihasilkan.

Dari hasil ini dapat disimpulkan bahwa hubungan antar jumlah *node* dan nilai PDR bersifat tidak linier. Peningkatan jumlah *node* memang dapat meningkatkan konektivitas jaringan hingga titik tertentu, namun jika jumlah *node* terlalu banyak, justru akan menimbulkan kemacetan informasi dan konflik dalam pengiriman data. Dengan demikian, terdapat jumlah *node* yang optimal sekitar 25 *node* untuk menghasilkan kinerja protokol AODV yang paling efisien dalam skenario VANET dengan kecepatan dan jumlah RSU yang tetap.

4.2.3 Analisis Hubungan Jumlah Node terhadap End-to-End Delay

Berdasarkan Gambar 4.10 hasil pengujian terhadap *End-to-End Delay* menggunakan protokol *routing* AODV pada jaringan VANET, dengan konfigurasi kecepatan kendaraan konstan sebesar 35 km/jam dan hanya melibatkan 1 unit RSU, diperoleh hasil *delay* masing-masing sebesar 0,1192 detik untuk 10 kendaraan, 0,0966 detik untuk 25 kendaraan, dan 0,0947 detik untuk 50 kendaraan. Dari hasil tersebut, terlihat bahwa peningkatan jumlah *node* berdampak terhadap penurunan nilai *delay*. Hal ini menunjukkan bahwa kepadatan kendaraan atau *node* dalam jaringan memberikan pengaruh yang cukup besar terhadap efektivitas waktu pengiriman paket data dari sumber ke tujuan.

Pada kondisi dengan jumlah *node* sebanyak 10, jaringan berada dalam keadaan yang kurang padat, sehingga jarak antar kendaraan cukup berjauhan. Situasi ini menyebabkan keterbatasan dalam membentuk jalur komunikasi yang stabil, serta mempersulit protokol AODV dalam menemukan rute yang optimal. Akibatnya, waktu tempuh pengiriman data

menjadi lebih lama karena proses pencarian rute membutuhkan waktu lebih panjang. Hal ini tercermin dari nilai *delay* tertinggi yang terjadi pada skenario ini, yaitu 0,1192 detik.

Ketika jumlah kendaraan ditingkatkan menjadi 25, kondisi jaringan menjadi lebih padat. Jarak antar *node* semakin dekat, sehingga memungkinkan lebih banyak rute alternatif yang dapat digunakan. Pada situasi ini, protokol AODV lebih cepat dalam membentuk rute karena memiliki lebih banyak pilihan *node* untuk dijadikan perantara (relay). Efisiensi jalur komunikasi pun meningkat, dan berdampak pada penurunan nilai *delay* menjadi 0,0966 detik. Artinya, pengiriman data dari sumber ke tujuan menjadi lebih cepat dan efektif.

Peningkatan jumlah *node* lebih lanjut menjadi 50 kendaraan menghasilkan nilai *delay* yang kembali menurun, meskipun tidak terlalu signifikan, yaitu menjadi 0,0947 detik. Hal ini menunjukkan bahwa setelah mencapai kepadatan tertentu, penambahan jumlah *node* tidak lagi memberikan pengaruh besar terhadap penurunan *delay*. Bahkan, jika tidak dikelola dengan baik, kondisi ini bisa berisiko menimbulkan overhead *routing* yang lebih besar serta potensi gangguan lain seperti tabrakan paket akibat lalu lintas data yang terlalu padat.

Secara keseluruhan, hasil pengujian ini menunjukkan bahwa peningkatan jumlah kendaraan dalam jaringan VANET berdampak positif terhadap penurunan End-to-End *Delay*, setidaknya hingga kepadatan jaringan mencapai titik optimal. Protokol AODV cenderung bekerja lebih efisien dalam kondisi jaringan yang cukup padat karena rute komunikasi dapat dibentuk dengan lebih cepat dan fleksibel. Namun demikian, penting untuk memperhatikan faktor pendukung lainnya seperti jumlah RSU, kapasitas kanal komunikasi, dan manajemen lalu lintas data, terutama jika jaringan diperluas lebih jauh. Dalam pengujian ini, satu RSU masih mampu menangani komunikasi hingga 50 *node* tanpa menyebabkan lonjakan *delay* yang signifikan.

4.2.4 Analisis Hubungan Kecepatan terhadap *Throughput*

Gambar 4. 11 Grafik Pengujian *Throughput* Skenario 2 yang

ditampilkan menunjukkan grafik hasil pengujian parameter QoS berupa *throughput* dalam jaringan VANET yang menggunakan protokol *routing* AODV. Pengujian ini dilakukan dengan konfigurasi yang melibatkan 35 *node* kendaraan dan satu RSU dengan variasi kecepatan kendaraan sebesar 30 km/jam, 40 km/jam 50 km/jam. Dari grafik terlihat bawah nilai *throughput* yang dihasilkan sangat konsisten, yakni sebesar 30,03 Mbps untuk kecepatan 30 km/jam, dan masing-masing 30,02 Mbps untuk kecepatan 40 km/jam dan 50 km/jam. Perbedaan antar nilai *throughput* tersebut sangat kecil dan dapat dikatakan tidak signifikan, yang mengindikasikan bahwa variasi kecepatan dalam rentang tersebut tidak memberikan pengaruh besar terhadap kinerja *throughput* protokol AODV dalam kondisi pengujian ini.

Throughput merupakan parameter yang menggambarkan jumlah data yang berhasil ditransmisikan melalui jaringan dalam satuan waktu, dan menjadi salah satu indikator penting dalam mengukur kinerja jaringan. Nilai *throughput* yang stabil di kisaran 30 Mbps menunjukkan bahwa protokol AODV mampu mempertahankan kemampuan transmisi data yang baik meskipun terjadi perubahan kecepatan kendaraan. Artinya, dalam skenario ini, pergerakan *node* kendaraan dari 30 hingga 50 km/jam tidak menyebabkan gangguan berarti terhadap kelancaran arus data di jaringan. Hal ini bisa jadi karena meskipun AODV bersifat reaktif dan cenderung sensitif terhadap perubahan topologi, pada kecepatan yang relatif sedang seperti dalam pengujian ini, dinamika jaringan masih dapat diakomodasi dengan cukup baik oleh mekanisme *route discovery* dan *maintenance* milik AODV.

Namun demikian, tetap perlu dicermati bahwa pengujian ini dilakukan dalam lingkungan yang hanya melibatkan 1 RSU dan 35 *node* kendaraan. Jumlah RSU yang terbatas dapat membatasi cakupan komunikasi dan memperbesar ketergantungan pada komunikasi antar kendaraan langsung (V2V). Meski demikian, hasil *throughput* yang tinggi dan stabil mengindikasikan bahwa, setidaknya untuk skenario ini, densitas *node* masih mencukupi untuk membentuk rute-rute komunikasi yang

berkelanjutan. Selain itu, pengaturan parameter simulasi seperti jangkauan sinyal, durasi pengiriman data, serta kondisi lalu lintas simulasi juga kemungkinan besar turut mendukung kestabilan *throughput* yang diperoleh.

Secara keseluruhan, hasil pada grafik ini memperlihatkan bahwa protokol AODV masih mampu memberikan performa yang baik dari sisi *throughput* dalam jaringan VANET dengan jumlah *node* terbatas dan hanya 1 RSU, bahkan ketika kecepatan kendaraan mengalami peningkatan. Meskipun AODV bukan protokol yang paling unggul dalam menghadapi kondisi jaringan dengan mobilitas tinggi, hasil ini menunjukkan bahwa dalam rentang kecepatan yang tidak terlalu ekstrem, performa *throughput*-nya masih bisa dijaga dengan baik. Untuk pengujian di masa mendatang, akan lebih representatif apabila dilakukan variasi terhadap jumlah *node*, jumlah RSU, serta skenario mobilitas yang lebih kompleks untuk melihat batas performa AODV secara lebih menyeluruh dalam lingkungan VANET.

4.2.5 Analisis Hubungan Kecepatan terhadap *Packet Delivery Ratio*

Gambar 4. 12 *Grafik Pengujian Packet Delivery Ratio Skenario 2* pada Skenario 2 menunjukkan performa protokol AODV dalam jaringan VANET dengan parameter kualitas layanan berupa PDR yang diukur dalam satuan persen. Pengujian ini dilakukan dengan melibatkan 35 *node* kendaraan dan 1 RSU, serta menggunakan variasi kecepatan kendaraan sebesar 30 km/jam, 40 km/jam, dan 50 km/jam. Dari grafik terlihat bahwa nilai PDR pada ketiga tingkat kecepatan tersebut sama, yakni sebesar 3%. Hal ini mengindikasikan bahwa perubahan kecepatan kendaraan dalam rentang yang diuji tidak berpengaruh signifikan terhadap efektivitas pengiriman paket menggunakan protokol AODV. PDR sebesar 3% termasuk rendah, yang berarti hanya sebagian kecil paket data yang berhasil mencapai tujuan, sehingga performa jaringan dalam kondisi tersebut bisa dikategorikan tidak optimal.

Konsistensi nilai PDR yang rendah di semua variasi kecepatan kendaraan menandakan bahwa faktor utama yang memengaruhi kinerja

jaringan kemungkinan bukan berasal dari mobilitas semata. Hal ini bisa disebabkan oleh jumlah *node* yang terbatas atau penyebarannya yang kurang merata dalam area simulasi, yang menyebabkan terbentuknya jalur komunikasi yang tidak stabil. Selain itu, jumlah RSU yang hanya 1 juga sangat membatasi jangkauan komunikasi dan fungsinya sebagai penghubung antar *node*. Dalam jaringan VANET, RSU memiliki peran penting dalam memperkuat dan memperluas jangkauan komunikasi. Ketika jumlah RSU sangat terbatas, *node* kendaraan terpaksa mengandalkan komunikasi langsung antar kendaraan (V2V), yang berisiko terganggu karena tingginya dinamika topologi akibat pergerakan kendaraan.

Sifat protokol AODV yang reaktif juga turut berkontribusi terhadap rendahnya PDR. AODV hanya membentuk rute saat diperlukan dan rute tersebut bisa cepat menjadi tidak valid dalam kondisi pergerakan *node* yang dinamis seperti di VANET. Akibatnya, rute yang telah dibuat bisa terputus sebelum proses pengiriman data selesai, sehingga banyak paket yang hilang ditengah jalan. Meskipun kecepatan dalam skenario ini tidak terlalu tinggi, yaitu maksimum 50 km/jam, tetap saja perubahan posisi *node* secara cepat dapat menyebabkan rute komunikasi menjadi tidak stabil. Berdasarkan hasil pengujian ini, terlihat bahwa AODV kurang ideal digunakan dalam skenario jaringan seperti ini. Untuk meningkatkan performa jaringan VANET, disarankan untuk menambah jumlah *node* kendaraan guna meningkatkan kepadatan jaringan, menambah jumlah RSU agar komunikasi lebih stabil, serta mempertimbangkan penggunaan protokol *routing* lain yang lebih sesuai untuk mobilitas tinggi, seperti protokol berbasis posisi. Selain itu, perlu juga dilakukan penyesuaian terhadap parameter simulasi seperti jangkauan transmisi, interval pengiriman paket, dan waktu simulasi agar hasil yang diperoleh lebih representatif terhadap kondisi nyata di lapangan.

4.2.6 Analisis Hubungan Kecepatan terhadap End-to-End Delay

Gambar 4. 13 *Grafik Pengujian End-to-End Delay Skenario 2 Delay* Skenario 2 yang ditampilkan merupakan grafik hasil pengukuran parameter QoS berupa *End-to-End Delay* dalam jaringan VANET dengan

menggunakan protokol AODV. Pengujian dilakukan dengan melibatkan 35 *node* kendaraan dan 1 RSU, sementara variasi kecepatan kendaraan yang digunakan dalam simulasi adalah 30 km/h, 40 km/h, dan 50 km/h. Dari hasil grafik dapat dilihat nilai *End-to-End Delay* yang dihasilkan pada 3 variasi kecepatan tersebut sama, yaitu sebesar 0,0929 detik. Hal ini menunjukkan bahwa perubahan kecepatan kendaraan dalam rentang yang diuji tidak memberikan pengaruh signifikan terhadap waktu tunda yang dibutuhkan oleh paket data untuk sampai dari sumber ke tujuan.

End-to-End Delay merupakan parameter yang sangat penting dalam menilai kualitas layanan jaringan, khususnya dalam aplikasi *real-time* seperti sistem komunikasi antar kendaraan yang sensitif terhadap latensi. Nilai *delay* yang relatif rendah dan stabil pada ketiga skenario ini mengindikasikan bahwa protokol AODV mampu mempertahankan efisiensi dalam proses pengiriman data meskipun terjadi perubahan mobilitas kendaraan. Kemampuan AODV untuk menjaga stabilitas waktu tunda ini bisa jadi disebabkan oleh tingkat kepadatan jaringan yang masih mencukupi untuk menjaga konektivitas antar *node*, serta kecepatan kendaraan yang tergolong sedang sehingga perubahan topologi jaringan tidak terjadi secara ekstrem.

Selain itu, keberadaan 1 RSU dalam skenario ini meskipun terbatas, tetap memberikan kontribusi terhadap kestabilan komunikasi, terutama jika *node-node* kendaraan berada dalam jangkauan komunikasi yang saling terhubung secara langsung atau melalui RSU. Hal ini mengurangi kemungkinan terjadinya penundaan yang disebabkan oleh pencarian rute baru secara berulang. Di sisi lain, sifat AODV yang reaktif, yakni hanya membentuk rute saat dibutuhkan, dapat berpotensi menyebabkan *delay* yang lebih tinggi pada kondisi mobilitas tinggi. Namun, pada skenario ini, kondisi mobilitas yang relatif moderat tidak memicu gangguan yang cukup besar terhadap proses *routing* sehingga *delay* dapat tetap dijaga pada level yang rendah dan stabil.

Secara keseluruhan, grafik ini menunjukkan bahwa dalam jaringan VANET dengan konfigurasi 35 *node* dan 1 RSU, protokol AODV masih

mampu memberikan performa yang baik dari sisi *End-to-End Delay*, bahkan ketika kecepatan kendaraan mengalami peningkatan. Nilai *delay* yang konsisten sebesar 0,0929 detik menunjukkan bahwa AODV cukup andal dalam menjaga waktu tempuh pengiriman data dari pengirim ke penerima. Untuk pengujian lebih lanjut, akan sangat berguna untuk melakukan variasi terhadap jumlah *node*, penambahan RSU, serta uji dalam kondisi kepadatan lalu lintas dan kecepatan yang lebih ekstrem untuk mengevaluasi batas toleransi AODV terhadap fluktuasi kondisi jaringan VANET. Hasil ini juga menunjukkan bahwa dalam konteks pengiriman data yang sensitif terhadap latensi, protokol AODV masih dapat diandalkan pada kondisi tertentu, selama dinamika jaringan tidak terlalu ekstrem dan kepadatan *node* cukup untuk menjaga konektivitas.

BAB V

SIMPULAN SARAN

5.1 Simpulan

Dapat disimpulkan pengaruh jumlah kendaraan kinerja protokol routing AODV pada jaringan VANETs di lingkungan perkotaan menunjukkan peningkatan *throughput* seiring bertambahnya jumlah *node* hingga mencapai titik optimal tertentu.

Peningkatan jumlah kendaraan menunjukkan peningkatan *throughput* hingga mencapai titik optimal, yaitu sebesar 30,02 Mbps. Namun, ketika jumlah kendaraan terlalu padat, seperti pada skenario dengan 100 kendaraan, performa jaringan mulai menurun, yang ditandai dengan penurunan nilai *Packet Delivery Ratio* (PDR) dari 4,85% menjadi 4,19% akibat meningkatnya interferensi dan kepadatan lalu lintas komunikasi.

Selain itu, peningkatan kecepatan kendaraan turut mempengaruhi kestabilan koneksi dan efisiensi perutean. Semakin tinggi kecepatan kendaraan, semakin sulit bagi protokol AODV untuk menjaga jalur komunikasi yang stabil, yang berdampak pada peningkatan *End-to-End Delay* (EED) dari 0,0947 detik menjadi 0,1192 detik.

Dari penelitian ini menunjukkan bahwa performa protokol routing AODV sangat dipengaruhi oleh jumlah dan kecepatan kendaraan. Diperlukan upaya optimasi lebih lanjut agar protokol ini mampu bekerja secara lebih efisien dalam menghadapi mobilitas tinggi dan dinamika jaringan VANETs di lingkungan perkotaan.

5.2 Saran

Bagi para peneliti yang ini mengembangkan penelitian ini lebih lanjut terkait simulasi protokol AODV pada jaringan VANET di kawasan perkotaan, terdapat beberapa saran dari peneliti yang dapat dipertimbangkan :

1. Dalam penelitian berikutnya disarankan untuk melakukan identifikasi masalah yang lebih mendalam terkait potensi interferensi antar *node* serta dampak dari penambahan jumlah kendaraan terhadap stabilitas rute. Karena hal ini dapat dilakukan dengan memperluas variasi jumlah

node, kecepatan kendaraan, dan kondisi lingkungan simulasi agar lebih mendekati situasi nyata.

2. Mengingat pesatnya perkembangan teknologi komunikasi kendaraan yang terus berkembang, peneliti menyarankan sebaiknya pengujian protokol tidak hanya berfokus pada AODV, tetapi juga membandingkan dengan protokol lainnya seperti DSR, OLSR, dan GPSR. Implementasi RSU yang lebih canggih dan pengujian dengan jaringan 5G atau teknologi komunikasi V2X juga perlu dipertimbangkan agar memperkaya hasil analisis.
3. Berdasarkan pada hasil pengujian, penurunan nilai PDR pada jumlah *node* yang tinggi menunjukkan adanya masalah kemacetan pada jaringan dan keterbatasan pada protokol AODV dalam mengelola rute pada kondisi padat. Oleh karena itu peneliti menyarankan untuk pengembangan di segi algoritma *routing* yang lebih adaptif, lalu penambahan RSU agar memperluas cakupan komunikasi, dan serta pengaturan pada parameter jaringan secara dinamis supaya dapat menjadi fokus penelitian berikutnya untuk mengatasi kelemahan tersebut.

DAFTAR PUSTAKA

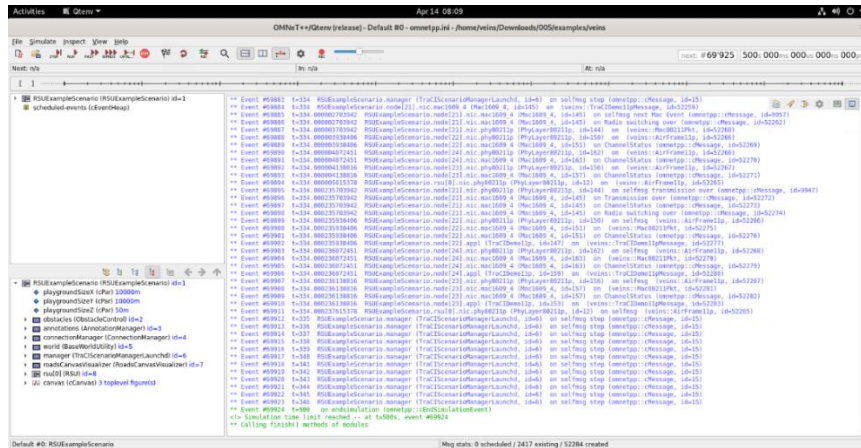
- [1] F. Aisya, P. H. Trisnawan, and R. A. Siregar, “Rekayasa Traffic Pada MANET Berdasarkan Tingkat Energi Di Suatu Node,” vol. 1, no. 1, pp. 1–6, 2017.
- [2] M. Alfarisi and H. Nurwarsito, “Analisis Dampak Serangan Blackhole dan Jellyfish terhadap Protokol Routing *Ad-Hoc on Demand Distance Vector* (AODV) dengan Model Mobilitas Random pada VANET,” ... *Teknol. Inf. dan Ilmu ...*, vol. 6, no. 5, pp. 2054–2063, 2022, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10994>
- [3] A. Mohammadi, P. Y. Park, M. Nourinejad, M. S. B. Cherakkatil, and H. S. Park, “SUMO2Unity: An Open-Source Traffic Co-Simulation Tool to Improve Road Safety,” *IEEE Intell. Veh. Symp. Proc.*, no. June, pp. 2523–2528, 2024, doi: 10.1109/IV55156.2024.10588571.
- [4] H. Nuwarsito and M. Arhingga Setiawan, “Performansi Protokol Routing Aomdv, Dsr, Dan Aodv Pada Mobile Ad-Hoc Network (Manet),” *Eng. Sci.*, vol. 6, no. 1, pp. 887–894, 2020.
- [5] A. J. T. Segara and A. Wijayanto, “Optimisasi Mobile Ad-Hoc Network dengan Algoritma Particle Swarm Optimization,” *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 468, 2021, doi: 10.30865/mib.v5i2.2636.
- [6] A. Wijayanto, A. J. T. Segara, and F. D. Adhinata, “Perancangan Mobile Ad-Hoc Network Menggunakan Optimasi Routing AOMDV,” *J. Media Inform. Budidarma*, vol. 5, no. 4, p. 1605, 2021, doi: 10.30865/mib.v5i4.3352.
- [7] M. S. Dhiaulhaq, “Analisis Protokol Routing *Reactive* pada Jaringan *Vehicular Ad Hoc Network* (VANET) dengan Model Skenario Jalan Tol,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 2, pp. 944–952, 2023, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [8] V. Y. P. Ardhana, E. S. Manapa, T. W. Sagala, Y. A. Sihaan, and E. A. M. Sampetoding, “Evaluasi Kinerja Protokol Perutean AODV dan SDGR+R pada VANET dengan Studi Kasus Pelabuhan Lembar,” *JTIM J. Teknol. Inf. dan Multimed.*, vol. 2, no. 1, pp. 59–67, 2020, doi: 10.35746/jtim.v2i1.76.

- [9] A. A. Satyana, P. H. Trisnawan, and R. Primananda, "Studi Kinerja Protokol Routing Ad-Hoc On Demand Multipath Distance Vector (AOMDV) dalam Lingkup *Vehicular Ad-Hoc Network* (VANET) pada Daerah Perkotaan," vol. 5, no. 12, pp. 5553–5561, 2021.
- [10] A. Ridwan, "Analisa Performansi Protokol 802.11P Pada Routing AODV di Jaringan VANET (*Vehicular Ad-hoc Network*)," *J. Sains dan Inform.*, vol. 8, no. 1, pp. 2598–5841, 2022, doi: 10.34128/jsi.v8i1.392.
- [11] H. Chandra, L. O. Sari, and E. Safrianti, "Penerapan Geographic Source Routing Pada V2I di Jalan Sudirman Kota Pekanbaru," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 312–323, 2023, doi: 10.57152/malcom.v3i2.957.
- [12] S. J. Raya, A. B. Rahmadani, V. Suryani, and G. B. Satrya, "JARINGAN VANET UNTUK KOMUNIKASI VEHICLE TO VEHICLE (STUDI KASUS Latar Belakang *Vehicular Ad-hoc Network* (VANet) merupakan tipe khusus dari Mobile," pp. 0–6, 2013.
- [13] K. Kamarullah, E. Endroyono, and W. Wirawan, "Optimasi Cross Layer Untuk Protokol Dynamic Source Routing Pada Komunikasi Antar Kendaraan Berbasis *Vehicular Ad-Hoc Networks* (VANETs)," *J. Tek. ITS*, vol. 6, no. 2, pp. 443–448, 2017, doi: 10.12962/j23373539.v6i2.23771.
- [14] K. B. Yogha Bintoro, SDH Permana, Ade Syahputra, Yaddarabullah, and Budi Arifitama, "V2V Communication in Smart Traffic Systems: Current status, challenges and future perspectives," *J. Process.*, vol. 19, no. 1, pp. 21–31, 2024, doi: 10.33998/processor.2024.19.1.1524.
- [15] R. Marten, D. P. Kartikasari, and F. A. Bakhtiar, "Analisis Pengaruh Node Density pada *Vehicular Ad Hoc Network* (VANET) dengan menggunakan Protokol Routing Dynamic Source Routing (DSR) di Model Jalan Perkotaan," vol. 6, no. 11, pp. 5123–5128, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [16] R. Z. Akbar, Istikmal, and Sussi, "Performance Analysis FSR and DSR Routing Protocol in VANET with V2V and V2I Models," *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2020*, pp. 158–163, 2020, doi: 10.1109/ISRITI51436.2020.9315367.

- [17] A. Hozouri, A. Mirzaei, S. RazaghZadeh, and D. Yousefi, “An overview of VANET vehicular networks,” 2023, [Online]. Available: <http://arxiv.org/abs/2309.06555>
- [18] H. Tria Ananda, “ANALISIS KINERJA ROUTING PROTOCOL AODV DAN AOMDV TERHADAP JELLYFISH DELAY VARIANCE ATTACK PADA *VEHICULAR AD-HOC NETWORK* (VANET),” 2023.
- [19] R. Aji Pratama, L. Rosselina, D. Sulistyowati, R. Fitri Sari, and R. Harwahyu, “Performance Evaluation on VANET Routing Protocols in the Way Road of Central Jakarta using NS-3 and SUMO,” *Proc. - 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Secur. Age Digit. Disruption, iSemantic 2020*, pp. 280–285, 2020, doi: 10.1109/iSemantic50169.2020.9234202.
- [20] S. Pande, R. Sadakale, and N. V. K. Ramesh, “Performance analysis of AODV routing protocol in VANET using NS-2 and SUMO,” *CEUR Workshop Proc.*, vol. 2889, pp. 144–154, 2021.
- [21] A. Ananda, F. W. Ginting, K. Putri, K. Lahagu, and S. K. Halawa, “Analisis Kualitas Layanan Jaringan Internet Wireless Lan Pada Layanan Indihome,” *J. Ilm. Multidisiplin Ilmu Komput.*, vol. 1, no. 1, pp. 24–30, 2023, doi: 10.61674/jimik.v1i1.111.
- [22] H. Anom, S. Aji, and A. C. Prasetyo, “Evaluasi Kinerja Jaringan WiFi Mahasiswa : Analisis Throughput , Delay , Jitter , dan Packet loss,” vol. 8, no. 1, pp. 23–27, 2024.
- [23] M. H. Ridwan, A. Solehudin, and C. Rozikin, “Analisis Quality of Service (Qos) Jaringan Wireless Dengan Penerapan Pcq (Studi Kasus : Kantor Kecamatan Kemang),” *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3293–3309, 2024.
- [24] S. Puspitasari, A. Subardono, and R. Adrian, “Pengujian dan Analisis Quality of Service Mobile Ad-Hoc Network (MANET) Menggunakan Protokol Routing Babel Pada Raspberry Pi,” *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 2, pp. 79–87, 2019, doi: 10.25077/teknosi.v5i2.2019.79-87.
- [25] st Fahmi, nd Yulita Salim, and rd Ramdan Satra, “Analisis Quality of Service Menggunakan Delay, Packet Loss, Jitter dan Mean Opinon Score

- pada Voice Over IP,” *Pros. SAKTI (Seminar Ilmu Komput. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 93–96, 2018, [Online]. Available: <https://e-journals.unmul.ac.id/index.php/SAKTI/article/view/1828>
- [26] A. J. T. Segara, A. Wijayanto, M. A. Gustalika, and A. D. Ramadhani, “Implementasi Mobile Ad-Hoc Network Pada Daerah Pasca Bencana Dengan Protokol DSR,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 4, p. 834, 2022, doi: 10.30865/jurikom.v9i4.4508.
- [27] D. Priambodo, “ANALISIS PERBANDINGAN KINERJA ROUTING PROTOCOL OLSR DAN DSDV MENGGUNAKAN RANDOM WALK,” 2024.
- [28] S. I. Jannah, “PENGEMBANGAN MEDIA PEMBELAJARAN BERBASIS WEB MENGGUNAKAN WEB DESIGNER REPLIT PADA TOPIK IKATAN KIMIA,” 2024.
- [29] C. Sommer *et al.*, *Veins: The open source vehicular network simulation framework*. 2019. doi: 10.1007/978-3-030-12842-5_6.
- [30] Y. Wahyudin and D. N. Rahayu, “Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review,” *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 15, no. 3, pp. 26–40, 2020, doi: 10.35969/interkom.v15i3.74.

LAMPIRAN



Tampilan Log Simulasi OMNeT++ pada Proyek "RSUExampleScenario"

Experiment	Measurement	Replica	Module	Name	Value
Default	#0	#0	RSUExampleScenario.r	Simulation durati	69.0
Default	#0	#0	RSUExampleScenario.r	Total Bytes Sent	2346.0
Default	#0	#0	RSUExampleScenario.r	Total Bytes Recei	138.0
Default	#0	#0	RSUExampleScenario.r	Total Packets Ser	18.0
Default	#0	#0	RSUExampleScenario.r	Total Packets Rec	1.0
Default	#0	#0	RSUExampleScenario.r	Throughput (byte-	36.0
Default	#0	#0	RSUExampleScenario.r	PDR	0.0555555555555556
Default	#0	#0	RSUExampleScenario.r	busyTime	1.817333333333333E-4
Default	#0	#0	RSUExampleScenario.r	ReceivedUnicastI	1.0
Default	#0	#0	RSUExampleScenario.r	ReceivedBroadca	29.0
Default	#0	#0	RSUExampleScenario.r	SentPackets	17.0
Default	#0	#0	RSUExampleScenario.r	SentAcknowledge	0.0

Tampilan Data Statistik dari Simulasi "RSUExampleScenario" di OMNeT++

Experiment	Measurement	Replica	Module	Name	Count	Mean	StdDev
Default	#0	#0	RSUExampleScenario.r	posx	60	1076.036873175	296.9072931993996
Default	#0	#0	RSUExampleScenario.r	posy	60	616.9055661326	71.333234275078
Default	#0	#0	RSUExampleScenario.r	speed	59	17.48938757467	10.611106684311551
Default	#0	#0	RSUExampleScenario.r	posx	63	1075.460430812	287.03863370271046
Default	#0	#0	RSUExampleScenario.r	posy	63	614.2882763729	72.36939223169601
Default	#0	#0	RSUExampleScenario.r	posx	62	1085.0210921911	292.22509408591264
Default	#0	#0	RSUExampleScenario.r	posy	62	618.3714870265	71.2600129007668
Default	#0	#0	RSUExampleScenario.r	acceleration	58	0.481472678496	2.032739328895954
Default	#0	#0	RSUExampleScenario.r	co2emission	58	6.966300505241	5.722995728446518
Default	#0	#0	RSUExampleScenario.r	speed	62	16.47597738278	9.529682424945966
Default	#0	#0	RSUExampleScenario.r	speed	61	16.95195480856	9.742661705256763
Default	#0	#0	RSUExampleScenario.r	acceleration	61	0.438588461391	1.9138107760654206

Statistik Vektor Parameter Posisi, Kecepatan, Akselerasi, pada Modul RSUExampleScenario di OMNeT++

Lampiran 1 *source code* Antena.xml pada simulasi OMNET++

```

<root>
  <Antenna type="SampledAntenna1D" id="monopole">
    <!-- monopole antenna on roof -->
    <!-- samples taken from "Effects of Antenna Characteristics and
    Placements on a Vehicle-to-Vehicle Channel Scenario" -->
      <parameter name="samples" type="string" value="-0.888 -0.942 -
      1.109 -1.29 -1.543 -1.717 -1.898 -1.902 -1.979 -2.018 -2.18 -2.336 -
      2.354 -2.287 -2.181 -2.008 -1.837 -1.667 -1.538 -1.553 -1.687 -1.819 -
      1.921 -1.977 -1.902 -1.768 -1.672 -1.741 -1.888 -2.167 -2.304 -2.326 -
      2.114 -1.838 -1.53 -1.36 -1.275 -1.331 -1.524 -1.759 -2.046 -2.212 -
      2.251 -2.04 -1.732 -1.519 -1.476 -1.579 -1.713 -1.775 -1.73 -1.585 -
      1.423 -1.339 -1.263 -1.433 -1.62 -1.857 -1.973 -2.059 -2.114 -2.097 -
      1.991 -1.95 -1.865 -1.865 -1.736 -1.606 -1.371 -1.17 -0.986 -0.893"/>
    </Antenna>
    <Antenna type="SampledAntenna1D" id="panorama">
      <!-- monopole antenna on panorama glass roof -->
      <!-- samples taken from "Influence of car panorama glass roofs on
      Car2Car communication" -->
        <parameter name="samples" type="string" value="-14.962 -14.531
        -14.035 -15.912 -13.103 -11.064 -9.902 -4.728 -6.49 -4.516 -2.66 -0.206
        -1.223 2.692 3.219 2.568 3.52 5.896 6.006 6.384 5.405 5.279 5.243
        5.433 3.03 2.296 1.664 0.618 1.708 -0.457 1.822 -0.799 1.658 2.735
        0.948 0.622 1.156 2.046 1.655 2.611 1.335 -0.108 1.857 0.207 1.221
        1.316 2.706 3.575 5.188 7.051 5.599 6.507 7.22 6.805 6.252 6.154 4.16
        2.247 3.291 2.866 -1.093 -0.769 -2.331 -4.004 -5.806 -4.79 -10.014 -
        12.566 -15.903 -14.306 -11.265 -14.368"/>
        </Antenna>
        <Antenna type="SampledAntenna1D" id="patch">
          <!-- patch antenna on side mirrors -->
          <!-- samples taken from "Effects of Antenna Characteristics and
          Placements on a Vehicle-to-Vehicle Channel Scenario" -->
            <parameter name="samples" type="string" value="5.061 5.016
            4.961 4.804 4.577 4.334 3.999 3.549 3.027 2.456 1.833 1.045 0.25 -
            0.579 -1.516 -2.48 -3.459 -4.348 -5.361 -6.195 -7.24 -8.236 -9.2 -10.345
            -11.326 -12.302 -13.302 -13.833 -14.209 -14.186 -13.844 -13.361 -
            12.824 -12.239 -11.901 -11.588 -11.487 -11.523 -11.733 -12.162 -
            12.529 -13.15 -13.723 -14.141 -14.289 -14.069 -13.538 -12.537 -11.764
            -10.582 -9.617 -8.646 -7.65 -6.601 -5.695 -4.599 -3.598 -2.608 -1.64 -
            0.741 0.079 0.958 1.62 2.36 2.911 3.515 3.962 4.313 4.593 4.808 4.957
            5.041"/>
            </Antenna>
          </root>

```

Lampiran 2 *source code* config.xml pada simulasi OMNET++

```
<root>
  <AnaLog ueModels>
    <AnaLog          ueModel          type="SimplePathlossModel"
thresholding="true">
      <parameter name="alpha" type="double" value="2.0"/>
    </AnaLog ueModel>
    <AnaLog          ueModel          type="SimpleObstacleShadowing"
thresholding="true">
      <obstacles>
        <type id="building" db-per-cut="9" db-per-meter="0.4" />
      </obstacles>
    </AnaLog ueModel>
  </AnaLog ueModels>
  <Decider type="Decider80211p">
    <!-- The center frequency on which the phy listens-->
    <parameter          name="centerFrequency"          type="double"
value="5.890e9"/>
  </Decider>
</root>
```

Lampiran 3 *source code* erlangen.launchd.xml pada simulasi OMNET++

```
<!-- debug config-->
<launch>
  <copy file="pwt.net.xml" />
  <copy file="pwt.rou.xml" />
  <copy file="pwt.poly.xml" />
  <copy file="pwt.sumo.cfg" type="config" />
</launch>
```

Lampiran 4 *source code* pwt.rou.alt.xml pada simulasi OMNET++

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- generated on 2025-03-24 14:34:15 by Eclipse SUMO duarouter
Version 1.11.0
<configuration          xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/duarouterCo
nfiguration.xsd">

  <input>
    <net-file value="pwt.net.xml"/>
    <route-files value="myflow.xml"/>
  </input>
```

```

<output>
  <output-file value="pwt.rou.xml"/>
  <alternatives-output value="pwt.rou.alt.xml"/>
</output>

<processing>
  <randomize-flows value="true"/>
</processing>

</configuration>
-->

<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/routes_file.xsd">
  <vehicle id="flow1.0" depart="7.58" departLane="random"
color="red">
    <routeDistribution last="0">
      <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
    </routeDistribution>
  </vehicle>
  <vehicle id="flow1.1" depart="9.51" departLane="random"
color="red">
    <routeDistribution last="0">
      <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
    </routeDistribution>
  </vehicle>
  <vehicle id="flow1.2" depart="9.74" departLane="random"
color="red">
    <routeDistribution last="0">
      <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1

```

```

189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow1.3" depart="12.69" departLane="random"
color="red">
  <routeDistribution last="0">
    <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow2.0" depart="15.39" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow2.1" depart="17.24" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3

```

```

189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>
</vehicle>
<vehicle id="flow1.4" depart="35.35" departLane="random"
color="red">
<routeDistribution last="0">
<route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
</routeDistribution>
</vehicle>
<vehicle id="flow2.2" depart="38.16" departLane="random"
color="green">
<routeDistribution last="0">
<route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>
</vehicle>
<vehicle id="flow2.3" depart="55.65" departLane="random"
color="green">
<routeDistribution last="0">
<route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1

```

```

196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flowl2.4" depart="56.85" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flowl2.5" depart="66.58" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4

```

```

190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow1.5" depart="75.18" departLane="random"
color="red">
  <routeDistribution last="0">
    <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow1.6" depart="75.85" departLane="random"
color="red">
  <routeDistribution last="0">
    <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow1.7" depart="83.72" departLane="random"
color="red">
  <routeDistribution last="0">
    <route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow2.6" depart="84.41" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1

```

```

189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>
</vehicle>
<vehicle id="flow1.8" depart="88.65" departLane="random"
color="red">
<routeDistribution last="0">
<route cost="40.20" probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
</routeDistribution>
</vehicle>
<vehicle id="flow2.7" depart="91.84" departLane="random"
color="green">
<routeDistribution last="0">
<route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>
</vehicle>
<vehicle id="flow2.8" depart="94.34" departLane="random"
color="green">

```

```

<routeDistribution last="0">
  <route      cost="120.02"      probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flow1.9" depart="99.33" departLane="random"
color="red">
  <routeDistribution last="0">
    <route      cost="40.20"      probability="1.00000000"
edges="891797537#1 891797537#2 891797537#3 1093735059
561406371 561406369 438929822 189173770#0 189173770#1
189173770#2 189173770#3 189173770#4 189173770#5 1146029535#1
1146029535#2 1146029535#3 189173514#0 189173514#1
189173514#2 189173514#3 189173514#4 189173514#5 189173514#6
189173514#7 189173514#8 189173514#9"/>
    </routeDistribution>
  </vehicle>
  <vehicle id="flow2.9" depart="100.02" departLane="random"
color="green">
    <routeDistribution last="0">
      <route      cost="120.02"      probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
    </routeDistribution>
  </vehicle>

```

```

</vehicle>
<vehicle id="flowI2.10" depart="103.99" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flowI2.11" depart="104.17" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
  </routeDistribution>
</vehicle>
<vehicle id="flowI2.12" depart="113.40" departLane="random"
color="green">
  <routeDistribution last="0">
    <route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369

```

```

438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>
</vehicle>
<vehicle id="flowI2.13" depart="115.58" departLane="random"
color="green">
<routeDistribution last="0">
<route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>
</vehicle>
<vehicle id="flowI2.14" depart="119.78" departLane="random"
color="green">
<routeDistribution last="0">
<route cost="120.02" probability="1.00000000"
edges="774902101#3 774902101#4 189170975#0 189170975#1
189170975#2 189170975#3 189170975#4 196047601#0 196047601#1
196047601#2 196047601#3 196047601#4 196047601#5 891797537#1
891797537#2 891797537#3 1093735059 561406371 561406369
438929822 189173770#0 189173770#1 189173770#2 189173770#3
189173770#4 189173770#5 1146029535#1 1146029535#2
1146029535#3 189173514#0 189173514#1 189173514#2 189173514#3
189173514#4 189173514#5 189173514#6 189173514#7 189173514#8
189173514#9 189186805#0 189186805#1 189184438#0 189184438#1
190403446#0 190403446#1 190403446#2 190403446#3 190403446#4
190403446#5 190403446#6 190403446#7 180244618#0 180244618#1
1146029534#13 961903477#0 961903477#1"/>
</routeDistribution>

```

```
</vehicle>
</routes>
```

Lampiran 5 *source code* omnet_interface.py untuk koneksi simulasi pada Website

```
import socket
import threading
import json
import time
import Log ging
import random # For demonstration purposes only

Log ger = Log ging.getLog ger(__name__)

# Configuration
DEFAULT_HOST = 'localhost'
DEFAULT_PORT = 4242
BUFFER_SIZE = 1024

# Global variables
_connection = None
_collector_thread = None
_collecting = False
_callback = None

class OMNeTPPConnection:
    """Connection handler for OMNeT++/Veins simulation."""

    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.socket = None
        self.connected = False

    def connect(self):
        """Connect to the OMNeT++/Veins simulation."""
        try:
            # In a real implementation, this would establish a connection to
            OMNeT++
            # For this demonstration, we'll simulate a connection
            Log ger.info(f"Connecting to OMNeT++ at {self.host}:{self.port}")

            # Simulate connection attempt
            # In a real implementation, this would use socket.connect()
```

```

        # self.socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
        # self.socket.connect((self.host, self.port))

        # Simulate successful connection
        self.connected = True
        Log ger.info("Connected successfully to OMNeT++")
        return True
    except Exception as e:
        Log ger.error(f"Failed to connect to OMNeT++: {e}")
        self.connected = False
        return False

    def disconnect(self):
        """Disconnect from the OMNeT++/Veins simulation."""
        try:
            if self.connected:
                # In a real implementation, this would close the socket
                # if self.socket:
                #     self.socket.close()

                self.connected = False
                Log ger.info("Disconnected from OMNeT++")
            return True
        except Exception as e:
            Log ger.error(f"Error disconnecting from OMNeT++: {e}")
            return False

    def send_command(self, command, params=None):
        """Send a command to the OMNeT++/Veins simulation."""
        if not self.connected:
            raise ConnectionError("Not connected to OMNeT++")

        try:
            # In a real implementation, this would send a command to
            OMNeT++
            # command_data = json.dumps({"command": command,
            "params": params})
            # self.socket.sendall(command_data.encode())
            # response = self.socket.recv(BUFFER_SIZE).decode()

            # Simulate command response
            Log ger.info(f"Sending command: {command} with params:
            {params}")

```

```

# Simulate different command responses
if command == "start":
    response = {"status": "started"}
elif command == "stop":
    response = {"status": "stopped"}
elif command == "pause":
    response = {"status": "paused"}
elif command == "resume":
    response = {"status": "resumed"}
elif command == "set_params":
    response = {"status": "params_set", "params": params}
else:
    response = {"status": "unknown_command"}

Log ger.info(f"Received response: {response}")
return response
except Exception as e:
    Log ger.error(f"Error sending command to OMNeT++: {e}")
    raise

def receive_data(self):
    """Receive data from the OMNeT++/Veins simulation."""
    if not self.connected:
        raise ConnectionError("Not connected to OMNeT++")

    try:
        # In a real implementation, this would receive data from OMNeT++
        # data = self.socket.recv(BUFFER_SIZE).decode()

        # Simulate received data
        # Generate random values for demonstration
        timestamp = time.time()
        Throughput = random.uniform(0.5, 10.0) # Mbps
        pdr = random.uniform(0.6, 1.0) # 60-100%
        eed = random.uniform(5, 100) # ms

        data = {
            "timestamp": timestamp,
            "metrics": {
                "Throughput": Throughput,
                "pdr": pdr,
                "eed": eed
            },
            "node_info": {
                "id": random.randint(1, 10),

```

```

        "position": {
            "x": random.uniform(0, 1000),
            "y": random.uniform(0, 1000)
        }
    }
}

    return data
except Exception as e:
    Log ger.error(f"Error receiving data from OMNeT++: {e}")
    raise

def connect(host=DEFAULT_HOST, port=DEFAULT_PORT):
    """Connect to the OMNeT++/Veins simulation."""
    global _connection

    if _connection and _connection.connected:
        Log ger.warning("Already connected to OMNeT++")
        return True

    _connection = OMNeTPPConnection(host, port)
    return _connection.connect()

def disconnect():
    """Disconnect from the OMNeT++/Veins simulation."""
    global _connection, _collecting, _collector_thread

    if not _connection or not _connection.connected:
        Log ger.warning("Not connected to OMNeT++")
        return True

    # Stop data collection
    _collecting = False
    if _collector_thread:
        _collector_thread.join(2.0) # Wait for the collector thread to finish

    return _connection.disconnect()

def send_command(command, params=None):
    """Send a command to the OMNeT++/Veins simulation."""
    global _connection

    if not _connection or not _connection.connected:
        raise ConnectionError("Not connected to OMNeT++")

```

```

return _connection.send_command(command, params)

def _data_collector():
    """Thread function to collect data from the simulation."""
    global _connection, _collecting, _callback

    Log ger.info("Starting data collection")

    while _collecting:
        try:
            if _connection and _connection.connected:
                data = _connection.receive_data()
                if _callback:
                    _callback(data)

                # Sleep to avoid overwhelming the system
                # In a real implementation, this would depend on the data rate
                time.sleep(1.0)
            except Exception as e:
                Log ger.error(f"Error in data collection: {e}")
                time.sleep(5.0) # Wait before retrying

    Log ger.info("Data collection stopped")

def start_data_collection(callback):
    """Start collecting data from the simulation."""
    global _connection, _collecting, _collector_thread, _callback

    if not _connection or not _connection.connected:
        raise ConnectionError("Not connected to OMNeT++")

    if _collecting:
        Log ger.warning("Data collection already started")
        return

    _callback = callback
    _collecting = True
    _collector_thread = threading.Thread(target=_data_collector)
    _collector_thread.daemon = True
    _collector_thread.start()

    Log ger.info("Data collection started")

def stop_data_collection():
    """Stop collecting data from the simulation."""

```

```

global _collecting, _collector_thread

if not _collecting:
    Log ger.warning("Data collection not started")
    return

_collecting = False
if _collector_thread:
    _collector_thread.join(2.0)

```

Lampiran 6 *source code* omnet_interface.py data_processor.py memproses data yang diterima *Template* HTML di folder *templates* digunakan untuk UI pada *Website*

```

import time
import Log ging

Log ger = Log ging.getLog ger(__name__)

def process_raw_data(raw_data):
    """
    Process raw data from OMNeT++/Veins simulation.

    Args:
        raw_data (dict): Raw data from OMNeT++/Veins simulation

    Returns:
        dict: Processed data with metrics
    """
    try:
        # Extract metrics from raw data
        metrics = raw_data.get("metrics", {})
        timestamp = raw_data.get("timestamp", time.time())
        node_info = raw_data.get("node_info", {})

        # Process Throughput (convert to Mbps if necessary)
        Throughput = metrics.get("Throughput", 0.0)

        # Process PDR (ensure it's a percentage between 0 and 1)
        pdr = metrics.get("pdr", 0.0)

        # Process EED (End-to-End Delay in milliseconds)
        eed = metrics.get("eed", 0.0)

        # Create processed data structure
        processed_data = {

```

```

        "timestamp": timestamp,
        "Throughput": Throughput,
        "pdr": pdr,
        "eed": eed,
        "node_info": node_info
    }

    # Add formatted timestamp for display
    processed_data["formatted_time"] = time.strftime("%H:%M:%S",
time.localtime(timestamp))

    return processed_data
except Exception as e:
    Log ger.error(f"Error processing data: {e}")
    # Return empty data structure in case of error
    return {
        "timestamp": time.time(),
        "formatted_time": time.strftime("%H:%M:%S"),
        "Throughput": 0.0,
        "pdr": 0.0,
        "eed": 0.0,
        "node_info": {}
    }

def calculate_statistics(data_series):
    """
    Calculate statistics for a series of data points.

    Args:
        data_series (list): List of data points

    Returns:
        dict: Statistics (min, max, avg, etc.)
    """
    if not data_series:
        return {
            "min": 0,
            "max": 0,
            "avg": 0,
            "median": 0
        }

    # Sort the data for median calculation
    sorted_data = sorted(data_series)

```

```

# Calculate statistics
min_val = min(data_series)
max_val = max(data_series)
avg_val = sum(data_series) / len(data_series)

# Calculate median
n = len(sorted_data)
if n % 2 == 0:
    median_val = (sorted_data[n//2 - 1] + sorted_data[n//2]) / 2
else:
    median_val = sorted_data[n//2]

return {
    "min": min_val,
    "max": max_val,
    "avg": avg_val,
    "median": median_val
}

def process_historical_data(simulation_data):
    """
    Process historical simulation data for comparison.

    Args:
        simulation_data (dict): Dictionary of simulation runs

    Returns:
        dict: Processed data for comparison
    """
    result = {}

    for sim_name, sim_data in simulation_data.items():
        Throughput_stats = calculate_statistics(sim_data.get("Throughput",
[]))
        pdr_stats = calculate_statistics(sim_data.get("pdr", []))
        eed_stats = calculate_statistics(sim_data.get("eed", []))

        result[sim_name] = {
            "Throughput": Throughput_stats,
            "pdr": pdr_stats,
            "eed": eed_stats,
            "data_points": len(sim_data.get("Throughput", []))
        }

    return result

```

