

# PENCARIAN RUTE BERDASARKAN PERGERAKAN NODE TETANGGA PADA PROTOKOL ROUTING AODV UNTUK MENEMUKAN RUTE YANG HANDAL DI JARINGAN MANET

(ROUTE DISCOVERY BASED ON THE MOVEMENT OF NEIGHBORING NODES IN AODV ROUTING PROTOCOL TO FIND RELIABLE ROUTES IN MANET NETWORKS)

Novita Mabruka Azahra<sup>1)</sup>, Andy Hidayat Jatmika<sup>2)</sup>, dan Raphael Bianco Huwae<sup>3)</sup>

<sup>1, 2, 3)</sup> Program Studi Teknik Informatika, Fakultas Teknik Universitas Mataram

Jl. Majapahit No.62, Gomong, Kec. Selaparang, Kota Mataram, Nusa Tenggara Bar. 83125

e-mail: [novitaazahra404@gmail.com](mailto:novitaazahra404@gmail.com)<sup>1)</sup>, [andy@staff.unram.ac.id](mailto:andy@staff.unram.ac.id)<sup>2)</sup>, [raphael.bianco.huawe@unram.ac.id](mailto:raphael.bianco.huawe@unram.ac.id)<sup>3)</sup>

## ABSTRAK

Pergerakan node dalam jaringan *Mobile Ad Hoc Network* (MANET) sangat dinamis dan tidak terprediksi. Setiap node dalam jaringan ini bergerak bebas dengan kecepatan dan arah yang berbeda, yang menyebabkan perubahan topologi jaringan secara terus-menerus. Pergerakan node mempengaruhi jalur komunikasi antara node, karena koneksi antar node terputus atau terbentuk kembali sesuai dengan posisi node apakah masih dalam jangkauan komunikasi atau tidak. Selain itu, pergerakan node juga memengaruhi kinerja protokol routing digunakan dalam mencari rute, seperti *Ad hoc On-Demand Distance Vector* (AODV). Protokol routing AODV tidak memperhitungkan pergerakan node tetangganya dalam mencari rute, sehingga rute yang ditemukan bisa cepat terputus karena node-node yang terlibat dalam rute tersebut bergerak menjauh, keluar dari jangkauan sinyal komunikasi. Penelitian ini bertujuan untuk memperbaiki kinerja protokol routing AODV dengan mempertimbangkan pergerakan node dalam proses pencarian rute menggunakan algoritma *TimePrediction*. Pendekatan yang digunakan meliputi modifikasi pada pesan *Hello*, agar pesan tersebut juga menyertakan informasi posisi dan kecepatan node tetangga. Pesan *Hello* adalah paket yang dikirimkan secara berkala oleh setiap node dalam jaringan MANET untuk menginformasikan keberadaannya kepada node tetangga. Hasil penelitian menunjukkan bahwa dari variasi kecepatan node yang diberikan, protokol routing *TimePrediction-AODV* mampu memberikan rata-rata peningkatan *throughput* yang lebih baik sebesar 8.63% untuk 20 node, 29.10% untuk 40 node, dan 60.70% untuk 60 node, rata-rata peningkatan PDR yang lebih baik sebesar 6.06% untuk 20 node, 20.54% untuk 40 node, dan 27.50% untuk 60 node, dan rata-rata penurunan *end-to-end delay* yaitu sebesar 31.53% untuk 20 node, 28.86% untuk 40 node, dan 43.98% untuk 60 node dibandingkan dengan AODV standar.

**Kata Kunci:** AODV, MANET, pesan hello, pergerakan node, protokol routing.

## ABSTRACT

The movement of nodes in a *Mobile Ad Hoc Network* (MANET) is very dynamic and unpredictable. Each node in this network moves freely at different speeds and directions, which causes continuous changes in the network topology. The movement of nodes affects the communication path between nodes, as connections between nodes are lost or re-established according to the position of the node whether it is still within communication range or not. In addition, the movement of nodes also affects the performance of routing protocols used in finding routes, such as AODV (*Ad hoc On-Demand Distance Vector*). The AODV routing protocol does not take into account the movement of neighboring nodes in finding routes, so the route found can be quickly disconnected because the nodes involved in the route move away, out of range of the communication signal. This research aims to improve the performance of the AODV routing protocol by considering the movement of nodes in the route finding process using the *TimePrediction* algorithm. The approach used includes modifications to *Hello* messages, so that they also include position and velocity information of neighboring nodes. *Hello* messages are packets sent periodically by each node in a MANET network to inform neighboring nodes of its presence. The results show that from a given node speed variation, *TimePrediction-AODV* routing protocol is able to provide better average *throughput* improvement of 8.63% for 20 nodes, 29.10% for 40 nodes, and 60.70% for 60 nodes, better average PDR improvement of 6.06% for 20 nodes, 20.54% for 40 nodes, and 27.50% for 60 nodes, and average *end-to-end delay* reduction of 31.53% for 20 nodes, 28.86% for 40 nodes, and 43.98% for 60 nodes compared to standard AODV.

**Keywords:** AODV, MANET, hello messages, node movement, routing protocol.

## I. PENDAHULUAN

Pergerakan node dalam jaringan MANET (*Mobile Ad-hoc Network*) merupakan salah satu tantangan dan karakteristik utama yang membuat jaringan ini sangat unik dibandingkan jaringan kabel. Node pada jaringan MANET dapat berpindah dengan pola yang tidak terprediksi, seperti bergerak ke arah tertentu dengan kecepatan tetap atau berhenti sejenak sebelum melanjutkan pergerakan. Pergerakan node membuat topologi jaringan MANET terus berubah, sehingga menyebabkan koneksi antara node dapat terputus atau terbentuk kembali dengan cepat. Perubahan topologi ini menciptakan tantangan besar bagi protokol *routing*, yang harus mampu menyesuaikan rute dengan cepat agar komunikasi antar node tetap berjalan lancar[1].

Protokol *routing Ad hoc On-Demand Distance Vector* (AODV) adalah salah satu protokol *routing* dalam jaringan MANET yang termasuk dalam kategori protokol *reactive routing* dimana rute antara node hanya akan dibentuk ketika dibutuhkan, misalnya ketika ada data yang perlu dikirim. Saat sebuah node ingin mengirimkan data, node tersebut akan mengirimkan paket *route request* (RREQ) yang disiarkan ke semua node tetangganya. Node-node tetangga yang menerima RREQ akan meneruskan paket tersebut ke semua node tetangganya juga hingga mencapai node tujuan. Pada saat paket RREQ mencapai node tujuan, node tersebut akan mengirimkan paket *route reply* (RREP) kembali ke node sumber. Paket RREP berisi informasi tentang rute yang ditemukan[2].

Dalam menemukan rute, protokol *routing* AODV tidak mempertimbangkan pergerakan node tetangga ketika sebuah node melakukan *broadcast* pengiriman RREQ ke semua node tetangga. Rute yang ditemukan bisa saja cepat terputus diakibatkan node-node yang berpartisipasi pada rute tersebut bergerak saling menjauh keluar dari radius sinyal komunikasi[3].

Terdapat beberapa penelitian yang telah dilakukan pada sebuah protokol *routing* dengan mempertimbangkan pemilihan node tetangga dalam mencari rute. Pada penelitian [4], penulis melakukan penelitian pada protokol *routing Dynamic Source Routing* (DSR) dengan tujuan memilih node tetangga terbaik untuk menemukan rute yang handal. Pemilihan node tetangga terbaik

dihitung berdasarkan pada 3 (tiga) kriteria yaitu rasio *bandwidth*, *round trip time*, dan rasio *packet loss*. Modifikasi protokol *routing* dilakukan dengan memanfaatkan paket Hello untuk memperoleh informasi ketiga kriteria dari node tetangga. Kinerja jaringan diukur menggunakan parameter *throughput*, *delay*, dan *packet delivery ratio* (PDR) yang dilakukan pada 25, 50, dan 100 node. Hasil penelitian menunjukkan bahwa modifikasi pada protokol *routing* DSR menghasilkan kinerja yang lebih unggul dibanding protokol *routing* DSR standar. Namun dari kedua protokol *routing* memberikan pola kinerja yang sama yaitu ketika jumlah kepadatan node sedikit, hasil yang diberikan belum optimal dikarenakan tidak dilakukan perhitungan arah pergerakan node apakah bergerak menjauh atau tidak dari node sumber.

Penelitian [5], penulis melakukan analisis pengaruh model mobilitas node terhadap kinerja protokol *routing* AODV dalam jaringan MANET dengan skenario yang melibatkan 10 node dalam area 500x500 m<sup>2</sup>. Kinerja diukur dengan parameter seperti *latency*, *throughput*, *jitter*, dan *packet loss*. Hasil penelitian menunjukkan bahwa protokol AODV memberikan nilai *packet loss* dan *latency* yang cukup tinggi dari skenario simulasi yang diberikan. Pada penelitian tersebut tidak dilakukan perhitungan prediksi arah pergerakan node ketika proses pencarian rute.

Dalam penelitian [6], penulis menganalisis protokol AODV pada jaringan MANET untuk mengevaluasi bagaimana variasi parameter jaringan mempengaruhi *delay*. Pengujian dilakukan dengan skenario yang melibatkan 25, 50, dan 100 node, kecepatan node 20 m/s dan 50 m/s. Hasil penelitian menunjukkan bahwa *delay* pada protokol AODV meningkat signifikan ketika jumlah node dan kecepatan bertambah, terutama pada kecepatan 50 m/s dan lebih dari 100 node.

Penelitian ini bertujuan untuk meningkatkan kinerja protokol *routing* AODV dengan memasukkan aspek pergerakan node dalam proses pencarian rute. Pendekatan yang digunakan melibatkan modifikasi pada pesan Hello, sehingga pesan tersebut juga memuat informasi posisi dan kecepatan node tetangga. Pesan Hello adalah sebuah paket yang secara berkala dikirimkan oleh setiap node dalam jaringan MANET untuk memberitahukan keberadaannya kepada node-node tetangganya. Informasi ini akan

dimanfaatkan untuk memprediksi durasi keberadaan node dalam jangkauan komunikasi, sehingga memungkinkan pemilihan rute yang lebih stabil dengan risiko putus yang lebih kecil.

Penelitian ini menggunakan *Network* simulator NS-2.35 yang dijalankan pada sistem operasi *Linux Ubuntu* 14.04 di *VirtualBox* untuk melakukan simulasi dan modifikasi protokol *routing*. Protokol *routing* AODV standar digunakan sebagai pembanding untuk mengevaluasi kinerja versi AODV yang telah dimodifikasi. Modifikasi ini memanfaatkan algoritma *TimePrediction* dengan pendekatan yang mengubah pesan Hello, sehingga pesan tersebut turut mencakup informasi mengenai posisi dan kecepatan node tetangga. Parameter kinerja protokol *routing* yang digunakan meliputi *end-to-end delay*, *throughput*, dan *packet delivery ratio* (PDR).

## II. STUDI PUSTAKA

Pada penelitian [7], penulis memodifikasi protokol OLSR di MANET untuk meningkatkan ketahanannya terhadap mobilitas node. Interval pesan Hello yang awalnya statis (2 detik) diubah menjadi variabel berbasis mobilitas, dengan tujuan memprediksi waktu keluarnya node dari jangkauan tetangga. Simulasi menggunakan 30 node dengan model *Random Waypoint* menunjukkan peningkatan signifikan pada parameter *packet loss* dan *delay-sum* dibandingkan OLSR standar. Pendekatan ini juga disarankan untuk diterapkan pada protokol AODV, yang diketahui rentan terhadap mobilitas tinggi.

Penelitian [8], penulis menganalisis kinerja MEDSR-LET, sebuah modifikasi dari protokol DSR yang menggunakan algoritma *Link Expiration Time* (LET) untuk meningkatkan efisiensi pencarian rute. Simulasi menggunakan 50 node dalam area 1000x1000 m<sup>2</sup> dan kecepatan node 30 m/s, 50 m/s, dan 70 m/s. Hasil penelitian menunjukkan bahwa pada kecepatan yang tinggi, terjadi peningkatan *delay* serta penurunan *throughput* dan *Packet Delivery Ratio* (PDR). Simulasi memperlihatkan bahwa pada kecepatan 70 m/s, *throughput* tercatat hanya 19,92 kbps, sementara PDR turun menjadi 93,71%.

Pada penelitian [9], penulis menganalisis pengaruh model mobilitas node terhadap kinerja protokol *routing* DYMO. Penelitian ini mengukur

parameter *packet delivery ratio* (PDR), *throughput*, *end-to-end delay*, dan *packet loss*. Simulasi menggunakan 30 hingga 120 node, tergantung pada skenario seperti penambahan jumlah node, perluasan area simulasi, dan peningkatan kecepatan maksimum. Hasil penelitian menunjukkan bahwa RWP lebih unggul dibandingkan RD dalam skenario area kecil, dengan rata-rata PDR mencapai 98,60%, lebih tinggi dibandingkan RD yang hanya 96,16%. Namun, dalam kondisi jumlah node yang besar, RD menghasilkan *end-to-end delay* yang lebih rendah karena distribusi nodenya lebih merata. Secara keseluruhan, RWP lebih cocok untuk area simulasi yang kecil dengan jarak antar node lebih pendek, sedangkan RD lebih efektif untuk mengurangi penumpukan node di area yang lebih luas.

Penelitian [10], penulis mengusulkan protokol *Mobility-Aware Routing Algorithm* (MARA) untuk meningkatkan keandalan rute di jaringan MANET. Protokol ini memodifikasi AODV dengan menambahkan metrik kecepatan node, arah pergerakan, dan energi residu pada paket RREQ. Pemilihan node dilakukan dengan memastikan kecepatan di bawah ambang batas, bergerak ke arah tujuan, dan memiliki energi yang cukup, sehingga mengurangi risiko putus rute dan masalah *broadcast storm*. Simulasi menggunakan 10–70 node dan kecepatan hingga 50 m/s bahwa MARA memberikan kinerja lebih baik dibandingkan AODV, MAD, dan MAE. MARA mencapai PDR lebih tinggi, *throughput* lebih baik, dan *overhead* lebih rendah, meski pada kepadatan tinggi, *end-to-end delay* meningkat karena tabrakan paket.

Pada penelitian [11], penulis menganalisis kinerja protokol DSR (*routing Dynamic Source Routing*) dan DSDV (*Destination-Sequenced Distance-Vector*) dengan empat model mobilitas node, yaitu *Random Waypoint*, *Random Point Group Mobility* (RPGM), *Freeway*, dan *Manhattan*. Penelitian ini menggunakan 20, 40, 60, 80, dan 100 node, serta pengujian pada 50 node untuk jumlah hops yang bervariasi. Hasil penelitian menunjukkan bahwa DSR lebih unggul dibandingkan DSDV dalam berbagai kondisi. Pada skenario jumlah hops, DSR mencatat *throughput* rata-rata 254.08 bytes/unit time untuk kurang dari 5 hops dan 193.92 bytes/unit time untuk lebih dari 5 hops, jauh lebih tinggi dibandingkan DSDV yang hanya mencapai 123.84 bytes/unit time dan 24.96 bytes/unit time.

Pada kepadatan node rendah, kinerja terbatas karena minimnya koneksi, tetapi dengan peningkatan jumlah node, *throughput* meningkat hingga pada kepadatan tinggi terjadi penurunan kinerja akibat tabrakan paket.

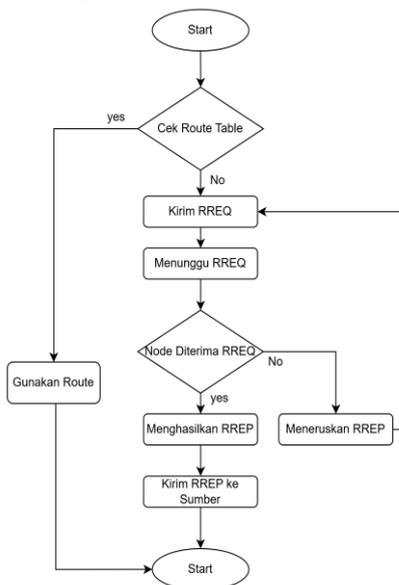
Berdasarkan penelitian tersebut, peneliti akan mengukur kinerja algoritma *TimePrediction* yang dimodifikasi dan menganalisis statistik untuk menilai peningkatan kinerja dalam memprediksi keberadaan node. Pendekatan ini lebih inovatif dibandingkan studi sebelumnya yang hanya fokus pada pemilihan node atau analisis mobilitas tanpa prediksi pergerakan.

### III. METODE PENELITIAN

Bagian bab ini, menjelaskan tahapan-tahapan penelitian yang meliputi mekanisme kerja protokol AODV standar, algoritma prediksi pergerakan node tetangga, pemanfaatan pesan Hello, serta analisis kinerja jaringan berdasarkan parameter yang telah ditentukan.

#### A. Mekanisme Kerja Protokol AODV Standar

AODV termasuk dalam protokol *routing* reaktif yang hanya bekerja ketika terdapat permintaan dari node sumber untuk mencari jalur pengiriman pesan ke node tujuan[2]. Protokol ini berfokus pada pencarian jalur tanpa *loop* dan memastikan jalur yang ditemukan merupakan rute terpendek menuju node tujuan.

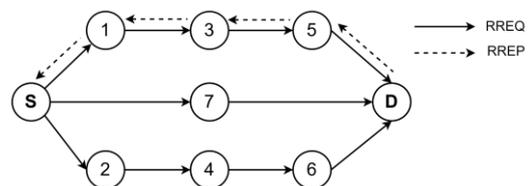


Gambar 1. Flowchart Route Discovery AODV Standar

Berikut penjelasan Gambar 1:

1. Proses *Route Discovery* dimulai ketika sebuah node ingin menemukan jalur untuk mengirim suatu data ke tujuan.
2. Node akan mengecek tabel rutenya terlebih dahulu untuk melihat apakah sudah ada jalur ke tujuan. Kalau ternyata rutenya sudah tersedia, node bisa langsung menggunakannya tanpa perlu melanjutkan proses ini.
3. Kalau tabel rute tidak punya jalur ke tujuan, node akan membuat dan mengirimkan RREQ (*Route Request*) ke jaringan. Setelah itu, node akan menunggu respons dari node lain.
4. Begitu RREQ diterima oleh node lain, ada dua kemungkinan. Kalau node itu punya rute ke tujuan atau ternyata dia adalah node tujuan, maka dia akan membuat RREP (*Route Reply*). Tapi kalau tidak, dia akan meneruskan RREQ ke node lainnya sampai ada yang bisa menemukan rute.
5. Setelah rute ditemukan, node tujuan atau node yang tahu jalur ke tujuan akan mengirimkan RREP kembali ke sumber. RREP ini berisi informasi rute yang dibutuhkan.
6. Node sumber menerima RREP, dan rute pun siap digunakan. Setelah itu, pengiriman data ke tujuan bisa dimulai.
7. Proses ini selesai setelah rute berhasil ditemukan dan digunakan untuk komunikasi. Jalur yang sudah ditemukan kini bisa dipakai untuk pengiriman data.

Proses *Route Discovery* pada AODV memungkinkan node untuk menemukan jalur yang optimal guna mengirimkan data. Setelah rute ditemukan dan RREP diterima, komunikasi dapat dilanjutkan dengan efisien. Proses ini memastikan konektivitas meskipun jaringan dinamis.



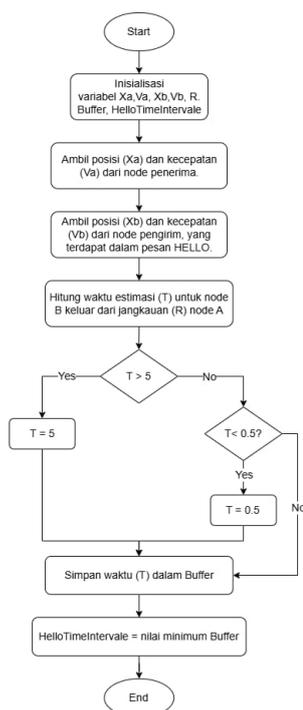
Gambar 2. Route Discovery AODV

Pada gambar 2 menggambarkan proses *Route Discovery* pada protokol AODV, yang digunakan dalam menemukan jalur antara node sumber (*source*, S) dan node tujuan (*destination*, D) di jaringan *ad hoc*. Proses dimulai ketika node sumber (S) ingin mengirimkan data ke tujuan tetapi tidak memiliki jalur di tabel *routing*-nya. Node

sumber kemudian membuat dan mengirimkan paket permintaan rute (*Route Request*, RREQ) ke seluruh node tetangganya menggunakan *broadcast*. Paket ini diterima oleh node perantara seperti 1, 2, dan 7, yang kemudian meneruskannya ke tetangga masing-masing. Misalnya, node 1 meneruskan RREQ ke 3, dan node 3 meneruskannya ke 5, hingga akhirnya mencapai node tujuan (D). Ketika node tujuan (D) menerima RREQ pertama, ia merespons dengan mengirimkan paket respons rute (*Route Reply*, RREP) secara *unicast* melalui jalur yang sama dengan jalur masuk RREQ. Paket RREP ini dikirim kembali ke S melalui jalur terbaik yang ditemukan selama propagasi RREQ. Setelah RREP diterima oleh S, jalur dianggap siap digunakan, dan pengiriman data dari S ke D dapat dimulai. Proses *Route Discovery* ini memastikan konektivitas yang efisien dan optimal meskipun jaringan bersifat dinamis. Hal ini karena rute hanya ditemukan saat diperlukan (*on-demand*)[12].

**B. Algoritma Prediksi Pergerakan Node Tetangga**

Algoritma prediksi pergerakan node tetangga dirancang untuk memperkirakan waktu sebuah node akan keluar dari jangkauan komunikasi. Algoritma ini memanfaatkan data posisi dan kecepatan yang disertakan dalam pesan HELLO yang telah dimodifikasi. Estimasi ini digunakan untuk menyesuaikan interval pengiriman pesan HELLO, memungkinkan protokol *routing* beradaptasi secara dinamis terhadap perubahan mobilitas jaringan.



Gambar 3. *Flowchart* Pergerakan Node Tetangga

Pada gambar 2, menjelaskan alur proses perhitungan waktu estimasi node tetangga yang keluar dari jangkauan node utama dalam jaringan MANET. Diawali dengan inisialisasi variabel-variabel, seperti posisi dan kecepatan node penerima ( $X_a, V_a$ ) dan node pengirim ( $X_b, V_b$ ), jangkauan komunikasi ( $R$ ), *buffer*, serta interval waktu pesan Hello (*HelloTimeIntervale*). Setelah itu, posisi ( $X_a$ ) dan kecepatan ( $V_a$ ) node penerima diambil sebagai data awal, diikuti oleh pengambilan posisi ( $X_b$ ) dan kecepatan ( $V_b$ ) node pengirim yang terdapat dalam pesan Hello. Selanjutnya, dilakukan perhitungan waktu estimasi ( $T$ ) untuk menentukan kapan node pengirim akan keluar dari jangkauan node penerima[7].

Jika waktu estimasi yang dihitung lebih dari 5, maka nilai  $T$  disesuaikan menjadi 5. Namun, jika nilai  $T$  kurang dari 0.5, maka nilainya diatur menjadi 0.5. Apabila tidak memenuhi kedua kondisi tersebut, nilai  $T$  tetap menggunakan hasil perhitungan tersebut, nilai  $T$  yang telah disesuaikan kemudian disimpan dalam *buffer*. Terakhir, interval waktu pengiriman pesan Hello diperbarui berdasarkan nilai minimum yang ada di *buffer*. Proses ini bertujuan untuk mengoptimalkan frekuensi pengiriman pesan Hello agar komunikasi antar-node tetap efisien dan stabil.

**C. Pesan Hello**

Pesan Hello digunakan untuk menjaga konektivitas antar-node dan mendeteksi keberadaan node tetangga dalam jaringan MANET. Setiap node secara rutin mengirimkan pesan "HELLO" ke node di sekitarnya untuk memberi tahu tentang keberadaannya dan status koneksinya. Pesan tersebut mencakup informasi seperti alamat pengirim, daftar node tetangga yang terdeteksi, dan kondisi koneksi. Dengan data yang diterima dari pesan "HELLO", node dapat mendeteksi perubahan dalam jaringan, seperti saat ada node yang keluar dari jangkauan atau node baru yang bergabung. Informasi ini membantu memperbarui tabel *routing* dan menjaga jalur komunikasi tetap stabil[13].

Efisiensi penggunaan pesan "HELLO" sangat mempengaruhi performa jaringan. Interval pengiriman pesan yang tepat bisa membantu menjaga kestabilan koneksi, mengurangi kehilangan data, dan mempertahankan *throughput*. Namun, pengaturan interval yang terlalu sering bisa menambah beban pada jaringan,

sementara interval yang terlalu jarang dapat memperlambat deteksi perubahan konektivitas. Oleh karena itu, mengatur interval yang optimal sangat penting agar protokol *routing* dalam jaringan MANET tetap berjalan dengan efisien, terutama di jaringan dengan mobilitas node yang tinggi[14].

Protokol AODV memungkinkan setiap node untuk mengirim pesan Hello secara rutin kepada tetangganya untuk menyebarkan informasi tentang keberadaan mereka melalui deteksi link dan penemuan tetangganya. Dengan demikian, node dapat mengetahui posisi dan informasi kecepatan dari tetangganya, format pesan Hello telah dimodifikasi dengan menambahkan enam bidang baru yang ditunjukkan pada Tabel 1, yaitu koordinat posisi node (Xpos, Ypos, Zpos) dan kecepatan node di masing-masing sumbu (Vx, Vy, Vz) disertakan[7].

Tabel 1. Struktur Pesan Hello

Reserved		Htime	Willingness
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
Xpos		yPos	
zPos		Vx	
Vy		Vz	

Koordinat-koordinat ini dapat digunakan untuk memperkirakan waktu yang diperlukan oleh node pengirim sebelum keluar dari jangkauan node penerima yang menerima pesan Hello. Ketika sebuah node menerima pesan Hello dari tetangganya, penginderaan link dilakukan oleh node tersebut, dan informasi link akan disusun serta disimpan. Agar penginderaan link menjadi lebih efektif, interval pengiriman pesan Hello disesuaikan menggunakan algoritma tertentu, memungkinkan analisis terhadap kinerja dan stabilitas jaringan sesuai dengan konsep yang diusulkan.

#### D. Parameter Kinerja Jaringan

##### 1) Throughput

*Throughput* mengukur seberapa banyak data yang berhasil ditransfer dalam jaringan dalam jangka waktu tertentu, biasanya dinyatakan dalam satuan bit per detik (*bps*). Tingginya nilai *throughput* menunjukkan bahwa jaringan berfungsi dengan efisien dan mampu mengirimkan data dengan cepat. Sebaliknya, nilai *throughput* yang rendah seringkali mengindikasikan adanya hambatan dalam jaringan, seperti kemacetan, gangguan sinyal,

atau topologi jaringan yang kurang optimal, yang dapat berujung pada keterlambatan signifikan atau kegagalan pengiriman data[1]. *Throughput* dapat dihitung setelah proses transmisi data selesai dan diperoleh menggunakan rumus berikut:

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

##### 2) End-to-End Delay

*Delay* mengacu pada waktu yang dibutuhkan oleh paket data untuk bergerak dari sumber ke tujuan dalam jaringan. Semakin tinggi nilai delay, semakin lama waktu yang dibutuhkan untuk mengirimkan data. *Delay* umumnya diukur dalam satuan milidetik (ms). Faktor-faktor yang memengaruhi terjadinya *delay* meliputi kepadatan jaringan, penggunaan server yang berlebihan, serta jarak antar titik koneksi dalam jaringan. Nilai *delay* yang lebih kecil menunjukkan kualitas jaringan yang lebih baik, karena data dapat dikirimkan dengan lebih cepat[15]. Rata-rata *delay* dapat dihitung dengan menggunakan rumus berikut:

$$\text{Delay} = \frac{\sum(\text{waktu terima} - \text{waktu kirim})}{\text{jumlah paket yang diterima}} \quad (2)$$

##### 3) PDR

PDR (*Packet Delivery Ratio*) mengukur tingkat keberhasilan pengiriman paket dalam jaringan dengan membandingkan jumlah paket yang berhasil untuk diterima oleh node tujuan dalam jangka waktu mengirimkan data. Semakin tinggi nilai PDR, semakin baik kinerja jaringan, karena menunjukkan bahwa lebih banyak paket yang berhasil sampai tujuan. PDR ini menggambarkan efektivitas protokol *routing* dalam mengirimkan paket dengan sukses[15]. PDR dapat dihitung menggunakan rumus berikut untuk menilai performa jaringan dalam hal keandalan pengiriman data:

$$\text{PDR} = \frac{\text{Jumlah paket yang diterima}}{\text{Jumlah paket yang dikirim}} \times 100\% \quad (3)$$

## IV. HASIL DAN PEMBAHASAN

### A. Parameter Skenario Simulasi

Penelitian ini menggunakan *Network Simulator* (NS-2) versi 2.35, yang diinstal pada sistem operasi *Linux Ubuntu* versi 14.04, untuk mensimulasikan jaringan MANET dengan

berbagai skenario. Modifikasi pada protokol *routing* AODV dilakukan dengan mengubah modul AODV yang sudah ada di NS-2, menggunakan bahasa pemrograman C++. Script simulasi jaringan ditulis dalam bahasa Tool Command Language (TCL) dan dijalankan di NS-2 versi 2.35. Hasil simulasi berupa file trace dengan ekstensi *.tr*, yang kemudian diproses dengan bahasa pemrograman AWK untuk memperoleh nilai *throughput*, *End-to-End Delay*, dan *Packet Delivery Ratio* (PDR).

Simulasi dilakukan dengan berbagai jumlah node, yaitu 20, 40, dan 60 node, yang ditempatkan secara acak di area 1000x1000 meter persegi. Setiap node bergerak dengan kecepatan yang ditentukan, dan posisi serta kecepataannya diperbarui secara dinamis selama simulasi. Interval pengiriman pesan Hello disesuaikan berdasarkan algoritma prediksi pergerakan node tetangga. Durasi simulasi ditetapkan selama 200 detik untuk mengevaluasi kinerja protokol AODV standar dan protokol AODV yang telah dimodifikasi.

Tabel I. Parameter Skenario Simulasi

Versi <i>Network Simulator</i>	NS-2.35
Jumlah kepadatan node	20, 40, 60 node
Kecepatan node	15, 35, 55, 75 (m/s)
Luas area simulasi	1000*1000 m <sup>2</sup>
Protokol <i>routing</i>	AODV
Model Mobilitas	Random Waypoint
Waktu simulasi	200 s
Model antena	Omni-directional
MAC protokol	IEEE 802.11
Pola trafik	CBR

### B. Hasil dan Analisis

Bagian ini menjelaskan hasil simulasi kinerja protokol *routing* AODV dan *TimePrediction-AODV* yang telah dilakukan menggunakan simulator jaringan NS-2 versi 2.35. Nilai-nilai dari parameter uji tersebut didapatkan dari file *trace* dengan menggunakan bahasa pemrograman AWK.

```
Novi@Novi-VirtualBox:~/TA_Novi$ awk -f delay-pdr-throughput.awk aodv-hasil.tr
Packet Delivery Ratio : 97%
End to End Delay      : 147
Average Throughput[kbps] : 1944
```

Gambar 4. Hasil Simulator 20 Node

Pada gambar 4 menunjukkan hasil implementasi uji coba untuk parameter PDR, *end-to-end delay*, dan *throughput* pada 20 node dengan kecepatan 15 m/s menggunakan protokol *routing TimePredictionAODV*. Ketiga parameter tersebut dihitung dari file *trace (.tr)* yang dihasilkan oleh simulasi NS-2, menggunakan skrip AWK. Pada Gambar 4, tercatat *throughput*

sebesar 1994 kbps, *end-to-end delay* sebesar 147 ms, dan PDR sebesar 97%. Hasil pengujian untuk 20 node dengan kecepatan 35 m/s, 55 m/s dan 75 m/s dapat dilihat pada Gambar 7, Gambar 10 dan Gambar 13.

```
Novi@Novi-VirtualBox:~/TA_Novi$ awk -f delay-pdr-throughput.awk aodv-hasil.tr
Packet Delivery Ratio : 94%
End to End Delay      : 213
Average Throughput[kbps] : 1673
```

Gambar 5. Hasil Simulator 40 Node

Pada gambar 5 ditampilkan hasil uji coba *throughput*, *end-to-end delay*, dan PDR pada 40 node dengan kecepatan 15 m/s menggunakan protokol *routing TimePrediction-AODV*. Nilai-nilai parameter tersebut dihitung menggunakan skrip AWK dari file *trace (.tr)* yang dihasilkan selama simulasi pada NS-2. Berdasarkan Gambar tersebut, *throughput* mencapai 1673 kbps, *end-to-end delay* sebesar 213 ms, dan PDR sebesar 94%. Untuk pengujian dengan kecepatan 35 m/s, 55 m/s dan 75 m/s untuk 40 node hasilnya disajikan pada Gambar 8, Gambar 11 dan Gambar 14.

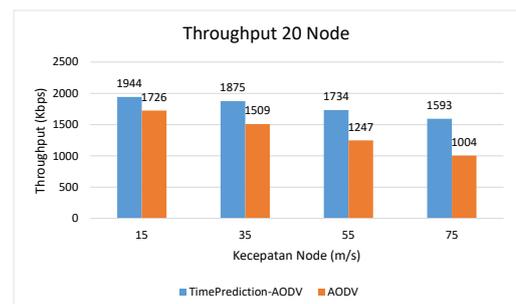
```
Novi@Novi-VirtualBox:~/TA_Novi$ awk -f delay-pdr-throughput.awk aodv-hasil.tr
Packet Delivery Ratio : 85%
End to End Delay      : 317
Average Throughput[kbps] : 988
```

Gambar 6. Hasil Simulator 60 Node

Pada Gambar 6 ditampilkan hasil uji coba untuk parameter *throughput*, *end-to-end delay*, dan PDR yang dilakukan pada 60 node dengan kecepatan 25 m/s menggunakan protokol *routing TimePredictionAODV*. Parameter tersebut diperoleh melalui pengolahan file *trace (.tr)* hasil simulasi NS-2 menggunakan skrip AWK. Berdasarkan Gambar, *throughput* tercatat sebesar 2100 kbps, *end-to-end delay* mencapai 152 ms, dan PDR berada di angka 85%. Hasil untuk kecepatan 35 m/s, 55 m/s dan 75 m/s untuk 60 node dapat dilihat pada Gambar 9, Gambar 12 dan Gambar 15.

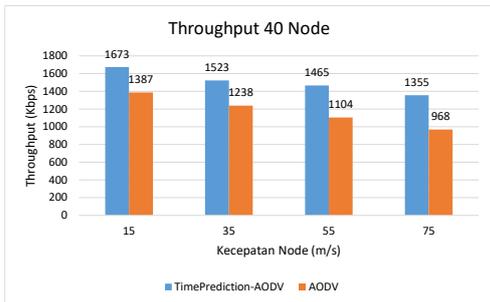
Hasil simulasi disajikan dalam bentuk grafik berdasarkan parameter uji *throughput*, *End-to-End Delay*, dan PDR.

#### 1) Throughput



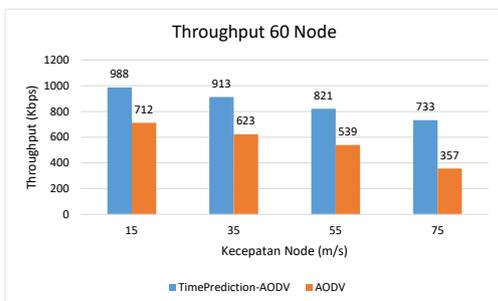
Gambar 7. Grafik *Throughput* 20 Node

Pada Gambar 7, menunjukkan hasil simulasi *throughput* untuk 20 node. Protokol *routing TimePrediction-AODV* menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan protokol *routing AODV* standar, meskipun *throughput* menurun seiring meningkatnya kecepatan node. Pada kecepatan node 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, *throughput TimePrediction-AODV* secara berturut-turut lebih tinggi dibandingkan AODV sebesar 12.63%, 18.87%, 28.88%, dan 58.73%. Hal ini disebabkan oleh kemampuan *TimePrediction-AODV* dalam mengelola perubahan topologi jaringan yang dinamis sehingga dapat mengurangi kehilangan paket data dan mempertahankan efisiensi jaringan.



Gambar 8. Grafik *Throughput* 40 Node

Pada Gambar 8, menunjukkan hasil simulasi *throughput* untuk 40 node. Protokol *routing TimePrediction-AODV* tetap lebih baik dibandingkan protokol *routing AODV* standar, meskipun *throughput* mengalami penurunan seiring dengan meningkatnya kecepatan node. Pada kecepatan node 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, *throughput TimePrediction-AODV* secara berturut-turut lebih tinggi dibandingkan AODV sebesar 20.58%, 23.05%, 33.03%, dan 40.63%. Keunggulan ini disebabkan oleh kemampuan *TimePrediction-AODV* dalam meminimalkan kehilangan paket dan memaksimalkan efisiensi jaringan, terutama dalam kondisi dengan perubahan topologi jaringan yang dinamis akibat peningkatan kecepatan node.

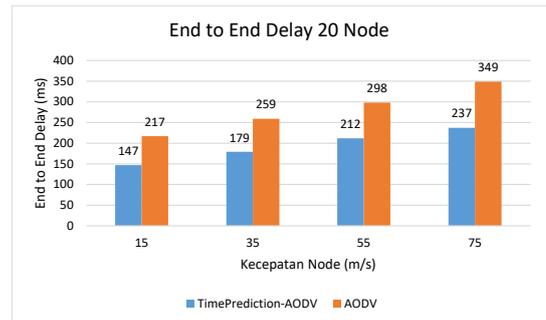


Gambar 9. Grafik *Throughput* 60 Node

Pada Gambar 9, menunjukkan hasil simulasi *throughput* untuk 60 node. Protokol *routing TimePrediction-AODV* mengalami peningkatan yang signifikan dibandingkan dengan protokol

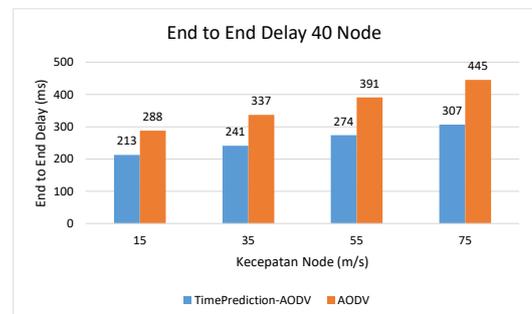
*routing AODV* standar seiring dengan meningkatnya kecepatan node. Untuk tingkat kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, secara berturut-turut *throughput* meningkat sebesar 38.77%, 46.48%, 52.50%, dan 105.31%. Hal ini disebabkan oleh kemampuan *TimePrediction-AODV* untuk memprediksi pergerakan node tetangga secara lebih akurat, sehingga mengurangi pemborosan *bandwidth* selama proses penemuan rute dan meningkatkan keandalan dalam jaringan MANET.

2) *End-to-End Delay*



Gambar 10. Grafik *End-to-End Delay* 20 Node

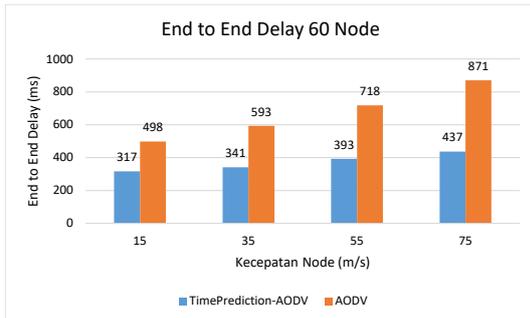
Pada Gambar 10, menunjukkan hasil simulasi *End-to-End Delay* untuk 20 node, terlihat bahwa nilai *End-to-End Delay* berkurang pada berbagai tingkat kecepatan node dengan menggunakan protokol *routing TimePrediction-AODV*. Pada kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, secara berturut-turut mengalami penurunan *End-to-End Delay* sebesar 32,26%, 30,89%, 28,86%, dan 32,09% dibandingkan dengan protokol *routing AODV* standar. Hal ini karena *TimePrediction-AODV* mampu memprediksi perubahan topologi jaringan lebih akurat, sehingga mengurangi keterlambatan transmisi. Namun, kedua protokol menunjukkan pola serupa, yaitu peningkatan *End-to-End Delay* seiring meningkatnya kecepatan node.



Gambar 11. Grafik *End-to-End Delay* 40 Node

Pada Gambar 11, menunjukkan hasil simulasi *End-to-End Delay* untuk 60 node, terlihat bahwa nilai *End-to-End Delay* berkurang pada berbagai tingkat kecepatan node dengan menggunakan protokol *routing*

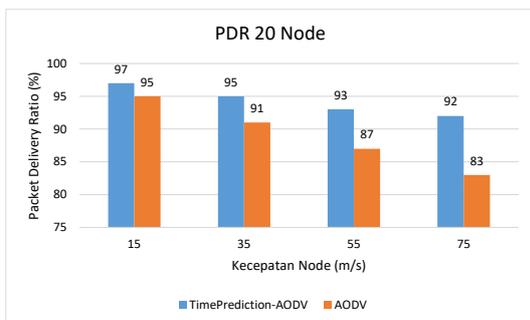
*TimePrediction-AODV*. Pada kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, secara berturut-turut terjadi penurunan *End-to-End Delay* sebesar 26,04%, 28,49%, 29,92%, dan 30,89% dibandingkan dengan protokol *routing AODV* standar. *TimePrediction-AODV* mampu mengirim data lebih cepat daripada *AODV* standar berkat efisiensi prediksi perubahan topologi. Meskipun demikian, *End-to-End Delay* tetap meningkat seiring bertambahnya kecepatan node pada kedua protokol.



Gambar 12. Grafik *End-to-End Delay* 60 Node

Pada Gambar 12, menunjukkan hasil simulasi *End-to-End Delay* untuk 60 node, terlihat bahwa nilai *End-to-End Delay* berkurang pada berbagai tingkat kecepatan node dengan menggunakan protokol *routing TimePrediction-AODV*. Pada kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, secara berturut-turut terjadi penurunan *End-to-End Delay* sebesar 36,35%, 42,51%, 45,26%, dan 49,83% dibandingkan dengan protokol *routing AODV* standar. *TimePrediction-AODV* lebih unggul dalam mengurangi delay dibandingkan *AODV* standar berkat prediksi topologi yang akurat. Namun, peningkatan *End-to-End Delay* tetap terjadi pada kedua protokol dengan meningkatnya kecepatan node.

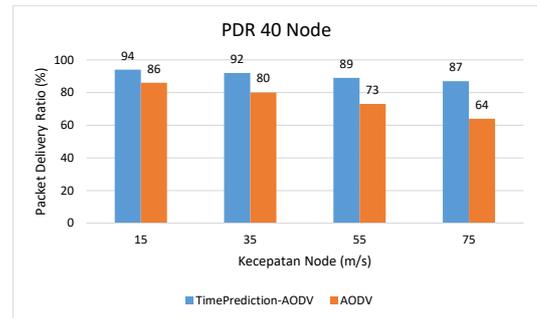
### 3) PDR



Gambar 13. Grafik *PDR* 20 Node

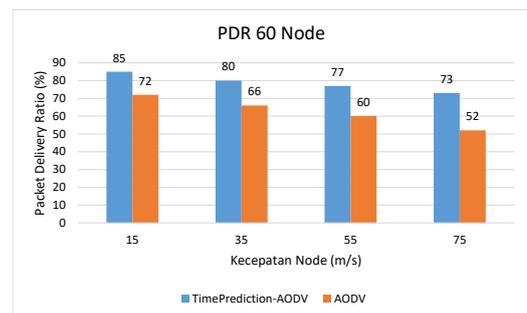
Pada Gambar 13, terlihat bahwa nilai *PDR* untuk jaringan dengan 20 node cenderung lebih

tinggi pada protokol *TimePrediction-AODV* dibandingkan dengan protokol *AODV* standar, khususnya pada berbagai tingkat kecepatan node. Pada kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, *TimePrediction-AODV* mencapai *PDR* 97%, 95%, 93%, dan 92%, sedangkan protokol *AODV* standar hanya mencapai nilai *PDR* sebesar 95%, 91%, 87%, dan 83%. Hal ini menunjukkan peningkatan sebesar 2%, 4%, 6%, dan 9%. Peningkatan ini terjadi karena *TimePrediction-AODV* lebih efisien memprediksi pergerakan node, sehingga menemukan rute handal dan mengurangi paket hilang dibandingkan *AODV* standar.



Gambar 14. Grafik *PDR* 40 Node

Pada Gambar 14, terlihat bahwa nilai *PDR* untuk jaringan dengan 40 node cenderung lebih tinggi ketika menggunakan protokol *TimePrediction-AODV* dibandingkan dengan protokol *AODV* standar, terutama pada berbagai tingkat kecepatan node. Pada kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, *TimePrediction-AODV* mencatat *PDR* sebesar 94%, 92%, 89%, dan 87%. Sedangkan, protokol *AODV* standar hanya mencapai *PDR* sebesar 86%, 80%, 73%, dan 64%, dengan peningkatan berturut-turut sebesar 8%, 12%, 16%, dan 23%. Keunggulan ini berasal dari efisiensi *TimePrediction-AODV* dalam memprediksi pergerakan node, memastikan rute handal, dan mengurangi kehilangan paket, dibandingkan *AODV* standar yang kurang adaptif.



Gambar 15. Grafik *PDR* 60 Node

Pada Gambar 15, terlihat bahwa nilai PDR untuk jaringan dengan 60 node lebih tinggi ketika menggunakan protokol *TimePrediction-AODV* dibandingkan dengan protokol AODV standar pada berbagai tingkat kecepatan node. Pada kecepatan 15 m/s, 35 m/s, 55 m/s, dan 75 m/s, protokol *TimePrediction-AODV* masing-masing menghasilkan PDR sebesar 85%, 80%, 77%, dan 73%. Sementara itu, protokol AODV standar hanya mencapai PDR sebesar 72%, 66%, 60%, dan 52%, hal ini menunjukkan peningkatan PDR sebesar 13%, 14%, 17%, dan 21%. *TimePrediction-AODV* unggul berkat prediksi pergerakan node, membuatnya lebih efisien dalam menemukan rute dan mengurangi kehilangan paket, sementara AODV standar melemah pada topologi dinamis.

## V. KESIMPULAN

Berdasarkan hasil pengujian dari skenario simulasi jaringan yang telah dilakukan, dapat disimpulkan bahwa:

1. Protokol *routing TimePrediction-AODV* memberikan peningkatan *throughput* yang lebih baik pada berbagai variasi kecepatan node yang diberikan yaitu pada kecepatan 15, 35, 55, dan 75 m/s, dengan rata-rata kenaikan sebesar 8,63% untuk 20 node, 29,10% untuk 40 node, dan 60,70% untuk 60 node. Peningkatan PDR rata-rata juga lebih baik, yaitu 6,06% untuk 20 node, 20,54% untuk 40 node, dan 27,50% untuk 60 node. Selain itu, protokol ini menunjukkan penurunan rata-rata *end-to-end delay* sebesar 31,53% untuk 20 node, 28,86% untuk 40 node, dan 43,98% untuk 60 node, dibandingkan dengan AODV standar.
2. Pada kecepatan rendah yaitu 15 m/s, perbedaan kinerja kedua protokol *routing* tidak terlalu signifikan, hal ini disebabkan karena ketika mencari rute, posisi dan arah node masih berada dalam jangkauan komunikasi.
3. Algoritma *TimePrediction* yang diterapkan pada kerangka protokol *routing* AODV baru dapat memberikan kinerja yang cukup signifikan pada kecepatan node 35 m/s, 55 m/s, dan 75 m/s, hal ini dikarenakan dalam mencari rute, kedua protokol *routing* menggunakan metode yang berbeda pada skenario kecepatan node yang makin tinggi.

## DAFTAR PUSTAKA

- [1] T. K. Priyambodo, D. Wijayanto, and M. S. Gitakarma, "Performance optimization of MANET Networks through routing protocol analysis," *Computers*, vol. 10, no. 1, pp. 1–13, 2021, doi: 10.3390/computers10010002.
- [2] M. Sadiq, A. Majid, A. Naveed, M. Arif, M. Asif, and A. U. Khattak, "Effects of Nodes Mobility on the Energy Consumption of Routing Protocols (AODV, DSDV) in Adhoc Networks," *Pakistan J. Sci. Ind. Res. Ser. A Phys. Sci.*, vol. 65, no. 1, pp. 74–79, 2022, doi: 10.52763/PJSIR.PHYS.SCI.65.1.2022.74.79.
- [3] S. Deepika, N. Nishanth, and A. Mujeeb, "An Assessment of Recent Advances in AODV Routing Protocol Path Optimization Algorithms for Mobile Ad hoc Networks," *Proc. 4th Int. Conf. Microelectron. Signals Syst. ICMSS 2021*, pp. 1–6, 2021, doi: 10.1109/ICMSS53060.2021.9673632.
- [4] Andy Hidayat Jatmika, S. P. Waskito, and Ariyan Zubaidi, "Implementasi Algoritma Pemilihan Node Tetangga Terbaik Pada Protokol Routing DSR di Jaringan MANET," *J. Comput. Sci. Informatics Eng.*, vol. 6, no. 1, pp. 64–72, 2022, doi: 10.29303/jcosine.v6i1.435.
- [5] A. Agussalim, "Analisis Pengaruh Mobilitas Node Pada Kinerja Aodv Protokol Menggunakan Emulator Mininet-Wifi," *SCAN J. Teknol. Inf. dan Komun.*, vol. 16, no. 3, pp. 26–30, 2021, doi: 10.33005/scan.v16i3.2818.
- [6] D. T. Nugrahadhi, M. R. Faisal, L. Triyasmono, and M. Janawi, "Dampak dari Parameter Variasi Koneksi, Node dan Kecepatan Node Terhadap Delay pada Routing Protocol AODV dan BATMAN Jaringan MANET," *J. Komputasi*, vol. 8, no. 2, pp. 10–22, 2020, doi: 10.23960/komputasi.v8i2.2675.
- [7] H. Berradi, A. Habbani, N. Mouchfiq, and M. Souidi, "Improvement of olsr protocol using the hello message scheme based on neighbors mobility," *J. Commun.*, vol. 15, no. 7, pp. 551–557, 2020, doi: 10.12720/jcm.15.7.551-557.
- [8] A. H. Jatmika, N. Alamsyah, and R. B. Huwae, "Dampak Variasi Kecepatan Node Terhadap Protokol Routing Medsr-Let Pada Jaringan Manet," *J. Teknol. Informasi, Komputer, dan Apl. (JTika)*, vol. 5, no. 2, pp. 217–224, 2023, doi: 10.29303/jtika.v5i2.350.
- [9] M. W. Anjani and H. Nurwarsito, "Pengaruh Pergerakan Node Pada Protokol Routing Dynamic Manet On Demand (DYMO) Dalam Mobile Ad-Hoc Network (MANET)," vol. 5, no. 1, pp. 105–114, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [10] C. Z. Sirmollo and M. A. Bitew, "Mobility-Aware Routing Algorithm for Mobile Ad Hoc Networks," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/6672297.
- [11] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, "Impact of node mobility on MANET routing protocols models," *J. Digit. Inf. Manag.*, vol. 5, no. 1,

pp. 19–24, 2021.

- [12] I. Alameri, J. Komarkova, and T. Al-Hadhrami, “Fuzzy-based optimization of AODV routing for efficient route in wireless mesh *Networks*,” *PeerJ Comput. Sci.*, vol. 9, pp. 1–25, 2023, doi: 10.7717/PEERJ-CS.1508.
- [13] R. Rahmadani, “Analisis Kinerja Protocol OLSR Pada Jaringan Ad Hoc,” *INFORMATICS Educ. Prof. J. Informatics*, vol. 8, no. 1, p. 78, 2023, doi: 10.51211/itbi.v8i1.2513.
- [14] M. N. Wicaksono, D. P. Kartikasari, and ..., “Deteksi Sinkhole Attack pada MANET dengan Protokol Routing AODV Menggunakan Perbedaan Sequence Number,” ... *Teknol. Inf. dan ...*, vol. 4, no. 1, pp. 250–260, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6891>
- [15] A. Wahab *et al.*, “SEDERHANA MENGGUNAKAN CISCO PACKET TRACER,” vol. 8, no. 6, pp. 12100–12107, 2024.