

# AODV Routing Optimization in Wireless Mesh Networks Using SDN-Inspired Control and ETX-Driven Machine Learning Weight Adaptation

Mochamad Yusril<sup>1\*</sup>, Galura Muhammad Suranegara<sup>2</sup>

Sistem Telekomunikasi, Universitas Pendidikan Indonesia

[yusrilm875@upi.edu](mailto:yusrilm875@upi.edu)<sup>1</sup>, [galurams@upi.edu](mailto:galurams@upi.edu)<sup>2</sup>

## Article Info

### Article history:

Received 2026-04-07

Revised 2026-05-04

Accepted 2026-05-12

### Keyword:

Wireless Mesh Network (WMN),

AODV,

Software-Defined Networking

(SDN),

Expected Transmission Count

(ETX),

Machine Learning,

Gradient Boosting,

Quality of Service (QoS)

## ABSTRACT

Wireless Mesh Networks (WMNs) require adaptive routing to sustain Quality of Service (QoS) under dynamic conditions, yet conventional AODV is limited by its reliance on hop count, which does not reflect actual link quality. This study proposes a hybrid framework integrating AODV, SDN-inspired centralized control, and Machine Learning (ML)-based weight optimization using the Expected Transmission Count (ETX). The model was evaluated in NS-3 (v3.45) through a four-scenario ablation study with 30 repetitions per scenario and OLSR as a baseline. A Gradient Boosting model trained on 885 samples generated routing weights based on seven QoS-related features. The results show that the proposed method significantly improves performance compared to standard AODV, with throughput increasing by 32.71%, delay decreasing by 40.19%, and routing overhead reduced by 38.92%, all statistically significant. The model achieved high predictive accuracy ( $R^2 = 0.9929$ ) without overfitting, with ETX emerging as the most influential feature. Overall, the integration of SDN control and ML optimization enhances routing efficiency, stability, and adaptability in WMNs, offering strong potential for IoT and smart city applications.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. PENDAHULUAN

Wireless Mesh Network (WMN) berkembang sebagai solusi jaringan nirkabel yang fleksibel, skalabel, dan hemat biaya [1]. Pada WMN, setiap node berperan sebagai router yang meneruskan paket ke *node* lain sehingga membentuk topologi mesh yang bersifat *self-organizing* dan *self-healing* [2]. Dalam konteks ini, kinerja jaringan sangat dipengaruhi oleh mekanisme *routing*. Pemilihan jalur yang tidak optimal dapat menurunkan *throughput*, meningkatkan *delay*, serta meningkatkan *packet loss* [3].

Salah satu protokol *routing* yang sering digunakan pada WMN adalah *Ad hoc On-Demand Distance Vector* (AODV). Protokol ini bersifat reaktif dan membangun rute hanya ketika diperlukan melalui mekanisme *Route Discovery* menggunakan paket *Route Request* (RREQ) dan *Route Reply* (RREP) [4]. Namun, AODV menggunakan *hop count* sebagai metrik utama, yang tidak mampu merepresentasikan kualitas link secara akurat, terutama pada kondisi jaringan yang dinamis [5], [6]. Akibatnya, AODV cenderung memilih

jalur dengan jumlah hop minimum tanpa mempertimbangkan kualitas link secara menyeluruh, sehingga berpotensi menghasilkan rute dengan kualitas transmisi yang rendah dan berdampak pada penurunan performa jaringan secara keseluruhan.

Berbagai penelitian telah dilakukan untuk mengatasi keterbatasan tersebut. Salah satunya adalah penggunaan *Expected Transmission Count* (ETX) sebagai metrik kualitas *link*, yang mampu memperhitungkan tingkat keberhasilan transmisi paket dan terbukti meningkatkan *throughput* serta menurunkan *packet loss* dibandingkan pendekatan berbasis *hop count* [5]. Integrasi ETX ke dalam AODV juga menunjukkan peningkatan performa yang signifikan dalam berbagai skenario jaringan [4]. Selain itu, pendekatan *Software-Defined Networking* (SDN) telah diperkenalkan dalam WMN untuk menyediakan kontrol terpusat dengan visibilitas global terhadap kondisi jaringan, sehingga memungkinkan pengambilan keputusan *routing* yang lebih efisien melalui pemisahan *control plane* dan *data plane* [7], [8], [9]. Di sisi lain, pemanfaatan Machine Learning (ML)

dalam routing telah berkembang pesat, di mana model ML mampu mempelajari pola jaringan dan menghasilkan keputusan routing yang adaptif. Pendekatan berbasis *deep learning* terbukti meningkatkan *Packet Delivery Ratio* (PDR) dan throughput [10], sementara *reinforcement learning* mampu meningkatkan Quality of Service (QoS) melalui optimasi kebijakan routing secara dinamis [11], [12].

Namun demikian, studi literatur menunjukkan adanya beberapa keterbatasan yang masih belum teratasi. Sebagian besar penelitian hanya mengintegrasikan dua pendekatan, seperti ETX dengan AODV, SDN dengan routing, atau ML secara terpisah, tanpa menggabungkan ketiganya dalam satu kerangka kerja terpadu [13], [14]. Pendekatan berbasis ETX umumnya berfokus pada optimasi metrik tanpa dukungan kontrol terpusat, sementara implementasi SDN lebih menekankan manajemen jaringan tetapi belum mengakomodasi pembobotan metrik secara dinamis sesuai kondisi jaringan [15], [9]. Di sisi lain, pendekatan ML sering memodelkan routing sebagai masalah klasifikasi atau regresi tanpa secara langsung mengoptimalkan parameter routing secara adaptif [10], [11]. Selain itu, aspek validasi seperti analisis statistik yang komprehensif, termasuk interval kepercayaan dan ukuran efek, serta studi ablasi yang sistematis masih relatif terbatas dalam penelitian sebelumnya [16]. Tantangan lain yang belum banyak dieksplorasi adalah generalisasi metode pada jaringan berskala besar, kondisi mobilitas tinggi, serta implementasi pada lingkungan nyata.

Untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan kerangka *hybrid* terpadu yang mengintegrasikan AODV, kontrol terpusat terinspirasi SDN, serta optimasi bobot metrik berbasis ML menggunakan ETX. Berbeda dari pendekatan konvensional dengan bobot statis, metode ini mengoptimalkan bobot secara berbasis data melalui analisis *feature importance* dari model Gradient Boosting yang dilatih menggunakan data simulasi [17], [18]. Kontribusi utama penelitian ini meliputi: (1) perancangan arsitektur kontrol terpusat berbasis SDN sebagai *overlay* di atas AODV yang tetap mempertahankan mekanisme *route discovery* terdistribusi, dilengkapi alur keputusan routing lima tahap yang sistematis; (2) formulasi fungsi *routing score* multi-parameter berbasis ETX, *end-to-end delay*, dan *Packet Loss Ratio* (PLR) dengan bobot empiris hasil pembelajaran model ( $w_{\{ETX\}} = 0,5165$ ;  $w_{\{PLR\}} = 0,3579$ ;  $w_{\{delay\}} = 0,1257$ ); serta (3) validasi eksperimental melalui studi ablasi empat skenario dengan 30 pengulangan pada NS-3 versi 3.45, menggunakan *Optimized Link State Routing* (OLSR) sebagai baseline dan uji *Mann-Whitney U* pada tingkat kepercayaan 95%. Secara ringkas, hasil menunjukkan bahwa pendekatan yang diusulkan mampu meningkatkan throughput, menurunkan delay, serta mengurangi overhead secara signifikan dibandingkan AODV konvensional, dengan model yang memiliki akurasi tinggi tanpa indikasi overfitting. Temuan ini menegaskan bahwa integrasi SDN dan ML efektif dalam meningkatkan efisiensi, adaptivitas, dan stabilitas routing pada WMN dinamis.

## II. METODE

### A. Desain Ekperimen

Penelitian ini menggunakan NS-3 versi 3.45 sebagai *platform* simulasi karena mampu memodelkan jaringan nirkabel secara detail dan telah banyak digunakan dalam studi jaringan [19]. Untuk menjamin perbandingan yang adil, seluruh skenario dijalankan dengan parameter jaringan yang sama dan diulang 30 kali menggunakan *random seed* berbeda berbasis *seed* tetap 12345, sehingga hasil lebih *robust* terhadap variasi stokastik simulasi.

Topologi terdiri dari 20 *node* pada area  $300 \times 300$  m, merepresentasikan WMN berkepadatan sedang seperti lingkungan kampus atau perkotaan. Mobilitas *node* dimodelkan dengan *Random Waypoint Model* (RWM) pada kecepatan 1–5 m/s dan *pause time* 2 detik, sesuai dengan skenario mobilitas sedang yang umum digunakan dalam studi *Mobile Ad Hoc Network* (MANET)/WMN [20]. Komunikasi nirkabel menggunakan standar IEEE 802.11b dengan daya transmisi 20 dBm.

Trafik jaringan dihasilkan melalui empat aliran *User Datagram Protocol-Constant Bit Rate* (UDP-CBR) dari *node* sumber 0–4 ke tujuan 10–14, dengan laju 64 Kbps per aliran dan ukuran paket 512 *byte*. Simulasi berlangsung selama 100 detik, dengan 10 detik awal sebagai periode *warm-up* agar pengukuran dilakukan setelah jaringan mencapai kondisi stabil.

Sebagai baseline eksternal, OLSR disimulasikan menggunakan parameter jaringan dan traffic yang identik dengan skenario utama. Dengan demikian, perbedaan performa yang diamati dapat dikaitkan langsung dengan karakteristik protokol routing, bukan akibat variasi konfigurasi simulasi. Parameter lengkap disajikan pada Tabel I.

TABEL I  
PARAMETER SIMULASI NS-3.45

Parameter	Nilai
Simulator	NS-3 versi 3.45
Jumlah <i>Node</i>	20 <i>node</i>
Area Simulasi	$300 \times 300$ meter
Model Mobilitas	<i>Random Waypoint</i>
Kecepatan <i>Node</i>	1 – 5 m/s
<i>Pause Time</i>	2 detik
Standar WiFi	IEEE 802.11b
Tx Power	20 dBm
<i>Data Rate</i> per <i>Flow</i>	64 Kbps
Ukuran Paket	512 byte
Tipe <i>Traffic</i>	UDP CBR
Jumlah <i>Flow</i>	4 flow ( <i>node</i> 0–4 → 10–14)
Port	9000 – 9004
Durasi Simulasi	100 detik ( <i>warmup</i> 10 detik)
<i>Run</i> per Skenario	30 <i>run</i>
<i>Random Base Seed</i>	12345

### B. Skenario Simulasi

Penelitian ini mengevaluasi empat skenario routing menggunakan pendekatan studi ablasi yang sistematis, ditandai sebagai Skenario A hingga D. Setiap skenario

mengisolasi dan menambahkan satu komponen tambahan secara bertahap ke dalam kerangka routing, sehingga memungkinkan penilaian yang ketat terhadap kontribusi individual maupun kumulatif dari ETX, kontrol terpusat berbasis SDN, dan ML terhadap performa routing secara keseluruhan.

Skenario A (*Baseline AODV*). Mengimplementasikan protokol AODV standar dengan metrik *hop count*. Pemilihan rute hanya mempertimbangkan jumlah lompatan, sehingga tidak merefleksikan kualitas tautan dan berpotensi menghasilkan rute dengan reliabilitas rendah [4].

Skenario B (AODV + ETX). Mengganti *hop count* dengan metrik ETX untuk mempertimbangkan probabilitas keberhasilan transmisi dua arah. Pendekatan ini meningkatkan keandalan rute dengan meminimalkan retransmisi, namun tetap bersifat terdistribusi tanpa visibilitas global [20].

Skenario C (AODV + SDN). Mengintegrasikan kontrol terpusat berbasis SDN yang menyediakan *global network view*. Meskipun metrik dasar masih *hop count*, keputusan routing menjadi lebih terkoordinasi dan adaptif terhadap kondisi jaringan secara menyeluruh [21].

Skenario D (Proposed Hybrid: AODV + ETX + SDN + ML). Menggabungkan ETX, kontrol SDN, dan optimasi bobot berbasis ML (*Gradient Boosting*). Model ML menentukan bobot optimal dari beberapa metrik (ETX, delay, dan PLR) untuk menghasilkan fungsi *routing score* yang lebih adaptif dan presisi [22].

*Baseline* Eksternal (OLSR). Sebagai pembandingan independen, digunakan *Optimized Link State Routing* (OLSR) yang bersifat proaktif. Evaluasi ini memberikan perspektif

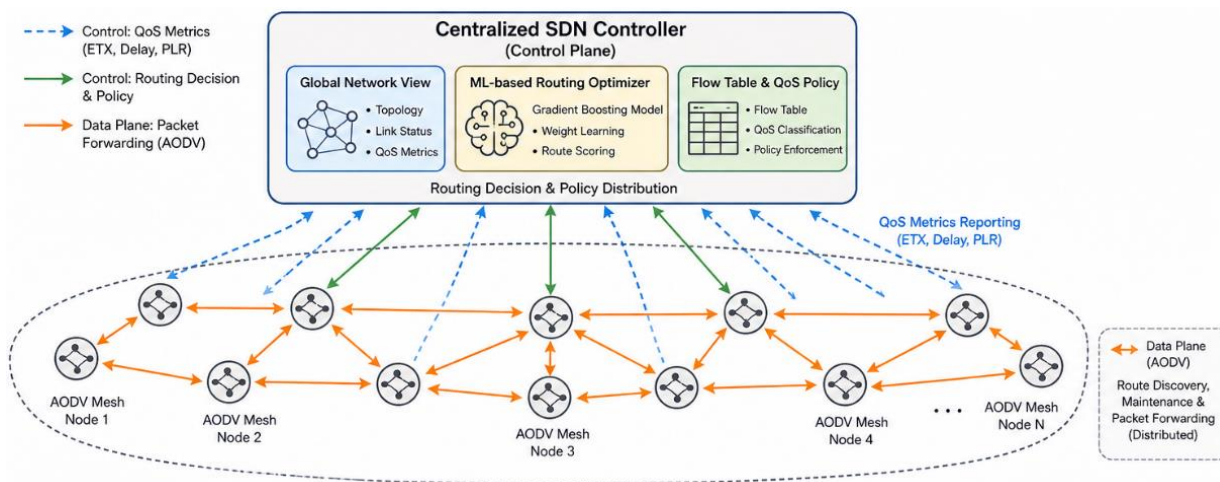
perbandingan antara pendekatan reaktif dan proaktif dalam kondisi simulasi yang identik [23].

### C. SDN-Inspired Centralized Control Architecture

Kerangka kerja yang diusulkan mengintegrasikan tiga lapisan yang saling melengkapi, yaitu: (1) protokol perutean reaktif AODV yang beroperasi pada bidang data (*data plane*), (2) pengendali terpusat yang terinspirasi dari SDN yang berfungsi pada bidang kendali (*control plane*), dan (3) modul ML yang bertanggung jawab terhadap optimasi bobot metrik berbasis data. Arsitektur berlapis ini dirancang untuk menggabungkan keunggulan perutean terdistribusi dengan pengambilan keputusan terpusat berbasis data.

Pendekatan yang terinspirasi dari SDN diimplementasikan sebagai lapisan kendali terpusat yang berada di atas protokol perutean AODV. Alih-alih mengadopsi implementasi SDN berbasis *OpenFlow* secara penuh, penelitian ini menekankan prinsip utama SDN, yaitu pemisahan antara bidang kendali dan bidang data, sehingga memungkinkan pengambilan keputusan perutean secara terpusat serta memiliki pandangan global terhadap jaringan [24]. Seluruh node tetap menjalankan AODV untuk proses penemuan rute dan penerusan paket pada bidang data, sementara sebuah node khusus sebagai SDN Controller mengagregasi metrik kualitas tautan meliputi ETX, *end-to-end delay*, dan *Packet Loss Ratio* (PLR) untuk membangun representasi kondisi jaringan secara menyeluruh serta menerapkan keputusan perutean yang telah dioptimalkan.

Gambar 1 menunjukkan arsitektur *Hybrid SDN-AODV* dengan *centralized control plane*, di mana SDN Controller berperan sebagai pusat pengambilan keputusan, sementara node lainnya berfungsi sebagai penerus data secara terdistribusi.



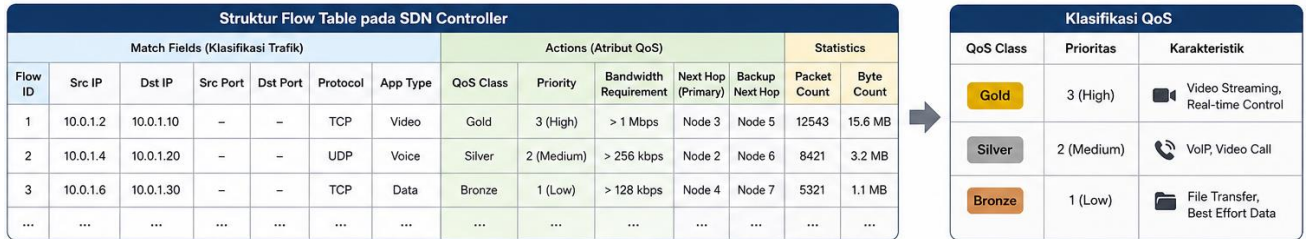
Gambar 1. Arsitektur Hybrid SDN-AODV dengan Centralized Control Plane

Dalam implementasinya, SDN Controller berperan sebagai entitas yang membangun *global network view* melalui pengumpulan metrik kualitas jaringan seperti ETX, *end-to-end delay*, dan PLR. Metrik-metrik ini dihitung secara

dinamis dan digunakan sebagai dasar evaluasi kualitas rute. Selain itu, controller juga mengelola *flow policies* dalam struktur *flow table* yang memetakan parameter komunikasi terhadap atribut QoS, seperti prioritas, kebutuhan *bandwidth*,

dan jenis aplikasi. Mekanisme ini memungkinkan proses klasifikasi trafik dilakukan secara sistematis dan terstruktur.

Struktur *flow table* dan mekanisme klasifikasi QoS pada SDN Controller ditunjukkan pada Gambar 2.



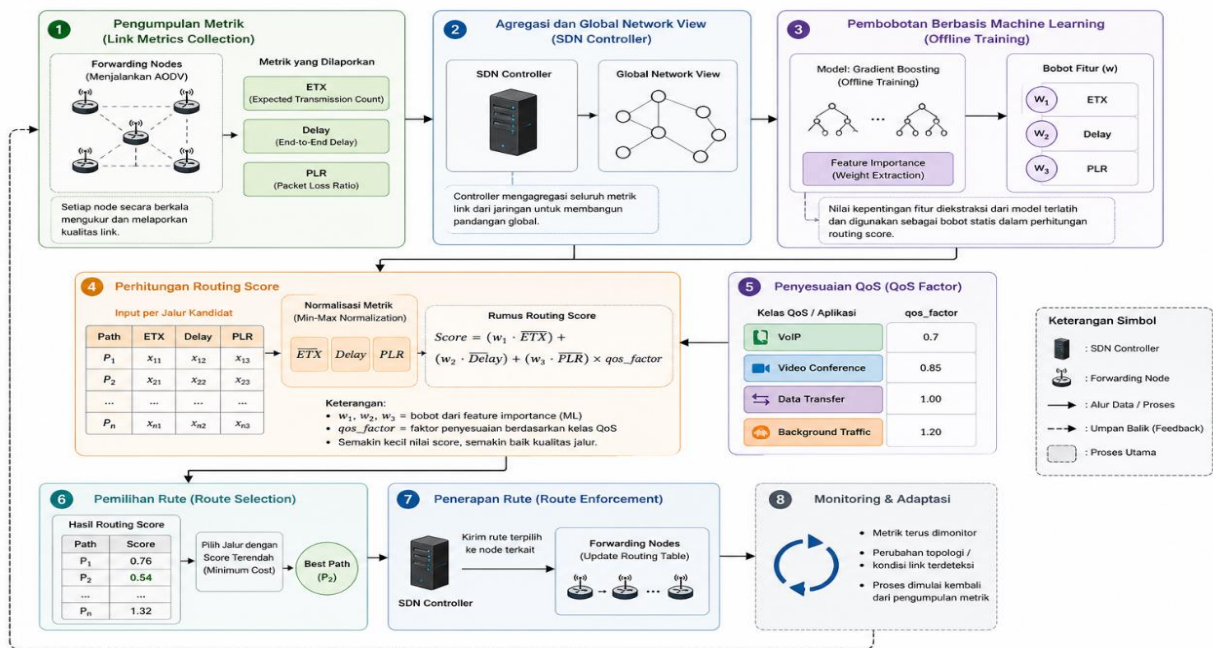
Gambar 2. Struktur *Flow Table* dan Klasifikasi QoS pada SDN Controller

Dalam skenario simulasi, satu *node* didedikasikan sebagai SDN Controller yang berfungsi sebagai pusat agregasi informasi dan pengambilan keputusan tanpa terlibat langsung dalam proses *packet forwarding*, sementara *node* lainnya bertindak sebagai *data forwarding nodes* yang menjalankan AODV secara terdistribusi. Berdasarkan informasi global yang dikumpulkan, controller melakukan evaluasi terhadap jalur yang tersedia menggunakan pendekatan berbasis skor perutean (*routing score*).

Proses pengambilan keputusan perutean dalam sistem ini mengikuti alur terstruktur yang terdiri atas beberapa tahap. Pertama, setiap *node* penerus secara berkala mengirimkan informasi kualitas tautan kepada SDN Controller melalui kanal kendali ringan. Selanjutnya, controller menghitung skor perutean komposit untuk setiap jalur kandidat

menggunakan fungsi pembobotan berbasis ML yang mempertimbangkan parameter QoS. Jalur dengan nilai skor terendah dipilih sebagai rute optimal, kemudian hasil keputusan tersebut didistribusikan kembali ke *node* terkait untuk memperbarui tabel perutean. Meskipun demikian, AODV tetap mempertahankan kemampuannya dalam menangani perutean ulang secara dinamis ketika terjadi kegagalan tautan atau perubahan topologi. Untuk meningkatkan adaptivitas, bobot metrik dalam fungsi penilaian ditentukan berdasarkan nilai kepentingan fitur yang diperoleh dari model *Gradient Boosting* yang telah dilatih secara *offline*.

Mekanisme perhitungan *routing score* berbasis ETX, *delay*, dan PLR ditunjukkan pada Gambar 3.



Gambar 3. Mekanisme Perhitungan *Routing Score*. Berbasis ETX, *Delay*, dan PLR

Secara keseluruhan, pendekatan ini memastikan bahwa keputusan perutean tetap responsif terhadap kondisi jaringan yang dinamis, sekaligus memanfaatkan visibilitas global yang

dimiliki oleh SDN Controller untuk menghasilkan rute yang lebih optimal dan adaptif terhadap kebutuhan QoS [25].

D. *Baseline Comparison Protocol: OLSR*

Untuk memberikan konteks terhadap kinerja metode yang diusulkan di luar empat skenario ablation, protokol OLSR disertakan sebagai baseline eksternal tambahan [26]. OLSR merupakan protokol perutean *link-state* yang bersifat proaktif, yang mempertahankan tabel perutean yang selalu diperbarui melalui penyebaran berkala pesan *Hello* dan paket *Topology Control* (TC). Perilaku proaktif ini secara mendasar berbeda dengan mekanisme penemuan rute berbasis permintaan (*on-demand*) pada AODV, sehingga menjadikan OLSR sebagai referensi yang relevan untuk mengevaluasi *trade-off* antara strategi perutean reaktif dan proaktif dalam lingkungan WMN yang dinamis.

Penting untuk ditekankan bahwa OLSR tidak termasuk dalam desain studi ablation (Skenario A–D). Sebaliknya, OLSR berperan sebagai baseline pembandingan independen yang dievaluasi menggunakan parameter simulasi yang sama (Tabel I) guna memastikan perbandingan yang adil dan tidak bias. Pendekatan ini memungkinkan metode yang diusulkan dievaluasi tidak hanya terhadap variasi ablation internalnya, tetapi juga terhadap alternatif protokol perutean proaktif yang telah mapan.

#### E. Formulasi ETX dan Routing Score

##### 1. Perhitungan ETX

ETX merupakan metrik kualitas tautan yang mengkuantifikasi jumlah rata-rata transmisi yang dibutuhkan agar satu paket berhasil diterima oleh node tujuan, dengan mempertimbangkan kondisi tautan pada kedua arah komunikasi [27]. Metrik ini pertama kali diperkenalkan untuk meningkatkan efisiensi pemilihan rute pada jaringan nirkabel multi-hop dengan mempertimbangkan probabilitas keberhasilan transmisi paket [27], [28]. ETX diformulasikan sebagai berikut:

$$ETX = \frac{1}{(d_f \times d_r)} \quad (1)$$

di mana:

- $d_f$  adalah *forward delivery ratio*, yaitu rasio jumlah paket yang berhasil diterima *node* tujuan terhadap jumlah paket yang dikirim *node* sumber pada arah maju, dengan rentang nilai  $d_f \in (0,1]$
- $d_r$  adalah *reverse delivery ratio*, yaitu rasio jumlah paket ACK yang berhasil diterima kembali oleh *node* sumber terhadap jumlah paket ACK yang dikirim *node* tujuan pada arah balik, dengan rentang nilai  $d_r \in (0,1]$

Berdasarkan Persamaan (1), nilai ETX memiliki batas bawah teoritis  $ETX = 1$ , yang dicapai ketika  $d_f = d_r = 1$ , yakni kondisi tautan ideal di mana seluruh paket berhasil dikirim dan diterima tanpa retransmisi. Seiring menurunnya kualitas tautan yang ditandai dengan berkurangnya nilai  $d_f$  atau  $d_r$ , nilai ETX meningkat secara tidak linear menuju tak hingga ( $ETX \rightarrow \infty$ ). Secara fisik, nilai ETX yang besar mengindikasikan bahwa tautan bersangkutan memerlukan banyak retransmisi, sehingga tidak efisien untuk dijadikan jalur pengiriman data. Dalam penelitian ini, nilai ETX dihitung per *flow*

menggunakan statistik paket yang diperoleh dari modul *FlowMonitor* NS-3, di mana  $d_f$  dan  $d_r$  diturunkan dari rasio paket yang diterima terhadap paket yang dikirim pada masing-masing arah komunikasi.

Untuk mencegah pembagian dengan nol dalam kondisi tautan yang sepenuhnya gagal ( $d_f = 0$  atau  $d_r = 0$ ), nilai ETX dibatasi dengan ambang batas maksimum  $ETX_{max} = 999$  yang digunakan sebagai indikator tautan tidak valid. Entri dengan nilai ETX = 999 selanjutnya difilter dan dikecualikan dari proses pelatihan model ML maupun perhitungan *routing score*, sebagaimana diterapkan pada tahap pra-pemrosesan dataset.

##### 2. Fungsi Routing Score Berbasis ML

*Routing score* merepresentasikan estimasi biaya kualitas suatu rute secara keseluruhan, yang dihitung menggunakan fungsi biaya linear berbobot (*weighted cost function*). Bobot pada fungsi ini diperoleh dari analisis *feature importance* model *Gradient Boosting* [17], sehingga mencerminkan kontribusi relatif masing-masing metrik jaringan terhadap kualitas rute berdasarkan data. Fungsi *routing score* dirumuskan sebagai berikut:

$$\text{Score} = \text{qos}_{factor} \times (0,0291 \times ETX_{norm} + 0,3567 \times Delay_{norm} + 0,6142 \times PLR_{norm}) \quad (2)$$

di mana bobot yang diperoleh dari *feature importance* model *Gradient Boosting* adalah:

- $w_{ETX} = 0,5165$  (kontribusi ETX terhadap kualitas rute)
- $w_{delay} = 0,1257$  (kontribusi *delay* terhadap kualitas rute)
- $w_{PLR} = 0,3579$  (kontribusi PLR terhadap kualitas rute)

Dengan  $w_{ETX} + w_{delay} + w_{PLR} = 1,0000$

Ketiga komponen metrik dinormalisasi ke rentang  $[0, 1]$  menggunakan normalisasi *min-max* sebagai berikut:

$$ETX_{norm} = \frac{ETX - ETX_{min}}{ETX_{max} - ETX_{min}} \quad (3)$$

$$Delay_{norm} = \frac{Delay - Delay_{min}}{Delay_{max} - Delay_{min}} \quad (4)$$

$$PLR_{norm} = \frac{PLR}{100} \quad (5)$$

Faktor QoS ( $\text{qos}_{factor}$ ) diintegrasikan ke dalam fungsi *score* untuk memberikan prioritas yang berbeda berdasarkan kelas layanan, sebagaimana didefinisikan berikut:

$$\text{qos}_{factor} = \begin{cases} 0,8 & \text{jika kelas } HIGH \text{ (real-time, VoIP)} \\ 1,0 & \text{jika kelas } MEDIUM \text{ (data umum)} \\ 1,2 & \text{jika kelas } LOW \text{ (background traffic)} \end{cases} \quad (6)$$

Nilai *score* yang lebih kecil menunjukkan kualitas rute yang lebih baik, sehingga fungsi ini berperan sebagai fungsi biaya (*cost function*) dalam proses pemilihan rute, dengan nilai *score* terendah diprioritaskan oleh kontroler SDN untuk diterapkan pada *flow table*. Kelas *HIGH* memperoleh nilai  $qos_{factor}$  terkecil sehingga ambang seleksi rute menjadi lebih ketat, yang cocok untuk layanan *real-time* yang sensitif terhadap latensi. Sebaliknya, kelas *LOW* memperoleh nilai  $qos_{factor}$  terbesar yang memungkinkan toleransi terhadap kualitas rute yang lebih rendah, sesuai dengan karakteristik *background traffic* yang tidak sensitif terhadap latensi.

## F. Training Pelatihan Model Machine Learning

### 1. Dataset dan Pra-pemrosesan

Dataset yang digunakan dalam penelitian ini diperoleh secara langsung dari hasil simulasi NS-3, di mana setiap baris merepresentasikan satu pengamatan kondisi jaringan pada satu *run* simulasi untuk satu kelas QoS tertentu. Dataset mentah terdiri dari 1.280 sampel yang mencakup tiga kelas QoS (*HIGH*, *MEDIUM*, *LOW*). Tahap pra-pemrosesan dilakukan melalui tiga langkah berurutan sebagai berikut.

Pertama, penyaringan data tidak valid: seluruh entri dengan nilai ETX = 999 — yang merepresentasikan kondisi tautan yang sepenuhnya gagal dan tidak dapat digunakan untuk *routing* — dikecualikan dari dataset. Langkah ini menghasilkan 885 sampel valid yang digunakan dalam seluruh proses pelatihan dan evaluasi [29].

Kedua, pengkodean variabel kategorik: variabel kelas QoS yang bersifat nominal dikodekan menjadi variabel ordinal numerik menggunakan pemetaan *LOW* = 0, *MEDIUM* = 1, dan *HIGH* = 2, sehingga dapat diproses oleh algoritma *ensemble* berbasis pohon keputusan.

Ketiga, normalisasi variabel target: nilai *routing score* dinormalisasi ke rentang [0, 1] menggunakan normalisasi *min-max* (Persamaan 3), untuk memastikan skala target yang konsisten dan mempermudah interpretasi fungsi biaya antar kelas QoS.

### 2. Fitur Input dan Variabel Target

Tujuh fitur input digunakan: *etx*, *df*, *dr*, *delay\_ms*, *plr\_pct*, *overhead\_pct*, dan *qos\_encoded*. Variabel target adalah *routing score* ter-normalisasi, yang merepresentasikan kualitas rute (semakin kecil semakin baik). Pemilihan fitur yang representatif sangat berpengaruh terhadap performa model, karena karakteristik dataset dan distribusi fitur merupakan faktor utama dalam keberhasilan pembelajaran ML [30].

### 3. Algoritma dan Konfigurasi Model

Dataset yang telah melalui pra-pemrosesan dibagi menggunakan skema *hold-out* 80:20, menghasilkan 708 sampel pelatihan dan 177 sampel pengujian dengan *random state* tetap (42) untuk memastikan reproduktibilitas eksperimen [31]. Dua algoritma *ensemble* berbasis pohon keputusan dilatih dan dibandingkan secara sistematis:

- *Random Forest Regressor*: membangun sejumlah pohon keputusan secara paralel melalui teknik *bagging* dan pemilihan fitur acak pada setiap simpul. Agregasi prediksi dari banyak pohon yang independen bertujuan mengurangi varians dan meningkatkan stabilitas model. Konfigurasi yang digunakan:  $n_{estimators} = 100$ ,  $random\_state = 42$ .
- *Gradient Boosting Regressor*: membangun pohon keputusan secara sekuensial, di mana setiap pohon berikutnya diarahkan untuk memperbaiki kesalahan residual pohon sebelumnya melalui optimasi fungsi kerugian secara bertahap. Pendekatan ini umumnya menghasilkan bias yang lebih rendah dibandingkan *Random Forest*. Konfigurasi yang digunakan:  $n_{estimators} = 100$ ,  $random\_state = 42$ .

Kinerja kedua model dievaluasi menggunakan tiga metrik kuantitatif: koefisien determinasi ( $R^2$ ) untuk mengukur proporsi varians data yang dijelaskan oleh model, *Mean Absolute Error* (MAE) untuk mengukur rata-rata kesalahan prediksi secara absolut, dan *Root Mean Square Error* (RMSE) untuk mengukur kesalahan prediksi dengan penalti lebih besar pada deviasi yang besar [31]. Untuk mengurangi risiko *overfitting* dan memvalidasi kemampuan generalisasi model, metode *5-fold cross-validation* diterapkan pada keseluruhan dataset, yang memungkinkan evaluasi kinerja yang lebih robust melalui pemisahan data yang berulang dengan partisi berbeda. Model dengan performa terbaik pada seluruh metrik evaluasi dipilih sebagai model akhir untuk tahap ekstraksi bobot.

### G. Optimasi Bobot Berbasis Machine Learning

Model ML digunakan untuk mengoptimalkan bobot metrik dalam *routing score*, bukan menggantikan mekanisme *routing*. *Feature importance* diekstrak, lalu bobot untuk *etx*, *delay\_ms*, dan *plr\_pct* dinormalisasi:

$$w_i = \frac{importance_i}{\sum_{j \in \{etx, delay, plr\}} importance_j} \quad (7)$$

Fitur *df* dan *dr* tidak digunakan karena sudah terwakili dalam ETX, sementara *overhead\_pct* dan *qos\_encoded* diakomodasi melalui parameter  $qos_{factor}$ . Pendekatan ini meningkatkan adaptivitas dibanding bobot statis [34], [35].

### H. QoS-Aware Modeling

Untuk mengakomodasi persyaratan layanan yang berbeda, parameter QoS dimasukkan ke dalam model dalam bentuk variabel yang disebut *qos\_encoded*, yang memungkinkan model untuk mempelajari hubungan antara metrik jaringan dan kualitas layanan yang diharapkan. Pendekatan ini konsisten dengan studi terbaru di mana model ML mengintegrasikan fitur terkait QoS dan klasifikasi trafik untuk meningkatkan kinerja jaringan dan proses pengambilan keputusan [32].

Lebih lanjut, pemodelan semacam ini memungkinkan sistem untuk menghasilkan bobot yang lebih kontekstual terhadap kelas QoS tertentu, seiring dengan pendekatan

perutean sadar QoS (*QoS-aware routing*) modern yang semakin mengandalkan diferensiasi berbasis kelas dan mekanisme adaptif untuk menangani permintaan trafik yang heterogen [33]. Sebagai hasilnya, proses pemilihan rute dapat disesuaikan dengan kebutuhan spesifik aplikasi, seperti meminimalkan latensi untuk layanan *real-time* atau memaksimalkan keandalan untuk transmisi data kritis, yang merupakan tujuan utama dalam strategi perutean berbasis QoS dan berbasis AI terbaru.

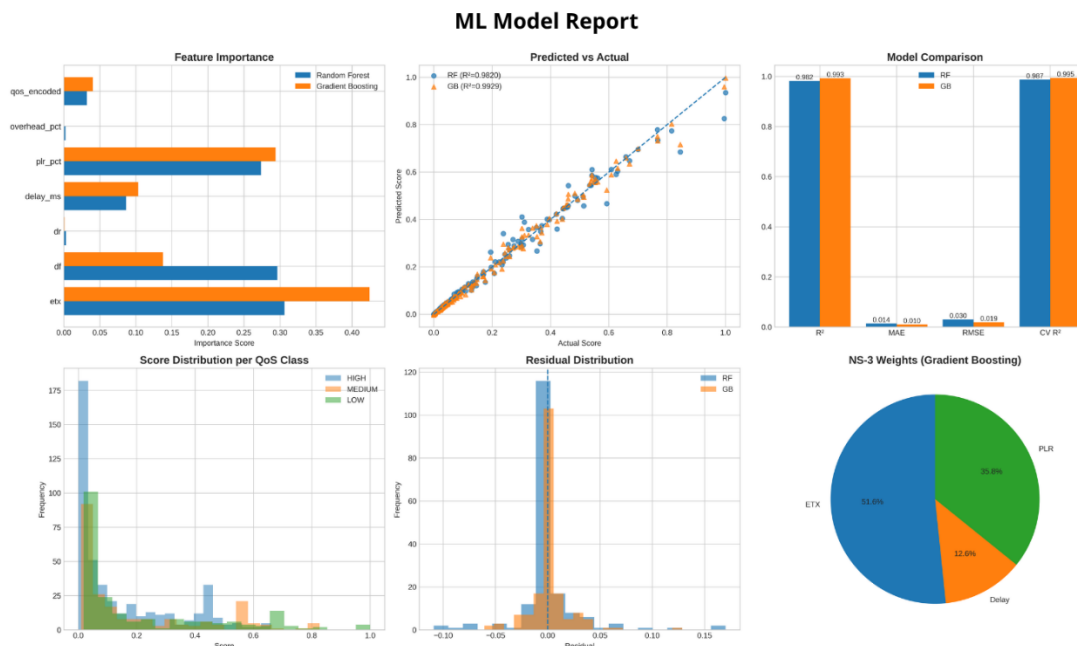
### III. HASIL DAN PEMBAHASAN

#### A. Hasil Evaluasi Model Machine Learning

##### 1. Perbandingan Kinerja *Random Forest* dan *Gradient Boosting*

TABEL II  
PERBANDINGAN PERFORMA MODEL MACHINE LEARNING

Metrik	RF	GB	Target
R <sup>2</sup> (data uji)	0,9820	0,9929	≥ 0,70
MAE (data uji)	0,0137	0,0102	-
RMSE (data uji)	0,0303	0,0190	-
CV R <sup>2</sup> rata-rata (5-fold)	0,9871	0,9945	-
CV R <sup>2</sup> std. deviasi	±0,0042	±0,0012	-
R <sup>2</sup> (data latih)	0,9979	0,9992	-
Selisih Train–Test R <sup>2</sup>	0,0159	0,0062	-



Gambar 4. Evaluasi model ML: perbandingan *Random Forest* dan *Gradient Boosting*, hubungan *predicted vs actual*, distribusi residual, *feature importance*, serta bobot QoS untuk implementasi NS-3

Gambar 4 menunjukkan bahwa *Gradient Boosting* (GB) secara konsisten mengungguli *Random Forest* (RF) pada seluruh metrik. Hal ini selaras dengan Tabel II, di mana GB mencapai R<sup>2</sup> data uji sebesar 0,9929 (lebih tinggi dari RF = 0,9820), serta menghasilkan error yang lebih rendah (MAE = 0,0102; RMSE = 0,0190). Stabilitas model juga lebih baik, ditunjukkan oleh CV R<sup>2</sup> = 0,9945 ± 0,0012, dibandingkan RF yang memiliki deviasi lebih besar. Selisih Train–Test R<sup>2</sup> yang lebih kecil pada GB (0,0062) mengindikasikan generalisasi yang lebih kuat dan *overfitting* yang minimal.

Visual *Predicted vs Actual* memperlihatkan titik GB lebih rapat di sekitar garis diagonal, sementara distribusi residual lebih sempit dan terpusat di nol, menegaskan akurasi dan konsistensi prediksi. Dari sisi interpretasi, *feature importance* menunjukkan ETX sebagai faktor dominan, diikuti PLR dan delay. Hal ini diperkuat oleh

bobot NS-3 (ETX 51,6%; PLR 35,8%; delay 12,6%), yang menegaskan bahwa kualitas link menjadi determinan utama QoS.

##### 2. Analisis Nilai R<sup>2</sup> Tinggi dan Validasi terhadap *Overfitting*

Nilai R<sup>2</sup> yang melebihi 0,99 pada model *Gradient Boosting* memerlukan analisis kritis untuk memastikan bahwa nilai tersebut bukan merupakan artefak *overfitting* atau *data leakage*. Tiga bukti metodologis berikut menunjukkan bahwa nilai R<sup>2</sup> tinggi ini merupakan cerminan karakteristik data yang sesungguhnya.

Pertama, selisih antara R<sup>2</sup> data latih dan data uji pada model *Gradient Boosting* hanya sebesar **0,0062** (0,9992 vs. 0,9929). Selisih yang sangat kecil ini mengindikasikan bahwa model tidak menghafal data latih secara berlebihan dan mampu melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

Kedua, hasil *5-fold cross-validation* menunjukkan  $CV R^2 = 0,9945 \pm 0,0012$  dengan standar deviasi yang sangat kecil. Konsistensi performa di seluruh *fold* memberikan bukti kuat bahwa performa tinggi bukan merupakan hasil kebetulan pada satu partisi data tertentu.

Ketiga, nilai  $R^2$  tinggi dapat dijelaskan secara kausal: variabel *routing score* merupakan fungsi deterministik yang diturunkan langsung dari kombinasi metrik jaringan (ETX, *delay*, PLR) melalui Persamaan (2). Korelasi yang sangat kuat antara fitur *input* dan variabel target bersifat inheren secara teoritis, bukan merupakan indikasi *data leakage*. Pembagian data latihan dan uji dilakukan secara eksplisit sebelum proses pelatihan sehingga tidak ada informasi dari data uji yang bocor ke proses pelatihan.

TABEL III  
FEATURE IMPORTANCE DAN BOBOT SCORING FUNCTION NS-3

Fitur	Importance (GB)	Bobot NS-3	Interpretasi
etx	0,4241 (42,4%)	w_etx = 0,5165	Gabungan $df \times dr$ — biaya transmisi
plr_pct	0,2939 (29,4%)	w_plr = 0,3579	Kerugian paket — dampak langsung QoS
df	0,1376 (13,8%)	—	<i>Forward delivery ratio</i> — kualitas link maju
delay_ms	0,1032 (10,3%)	w_delay = 0,1257	Latensi jalur — indikator QoS <i>real-time</i>
qos_encoded	0,0403 (4,0%)	—	Kelas QoS (HIGH / MEDIUM / LOW)
dr	0,0009 (0,1%)	—	<i>Reverse delivery ratio</i> — link balik
overhead_pct	0,0001 (0,0%)	—	<i>Routing overhead</i> — kontribusi minimal

#### 4. Performa Model per Kelas QoS

Evaluasi tambahan dilakukan secara terpisah per kelas QoS untuk memvalidasi kemampuan generalisasi model. Model menunjukkan kinerja yang konsisten dengan  $R^2 > 0,996$  pada seluruh kelas yang ditunjukkan oleh Tabel IV. Nilai MAE yang sedikit lebih tinggi pada kelas *LOW* (0,0063) dibandingkan kelas *HIGH* (0,0044) dapat dijelaskan oleh variabilitas metrik yang lebih tinggi pada *background traffic* yang memiliki toleransi lebih besar terhadap fluktuasi kondisi jaringan.

TABEL IV  
PERFORMA MODEL GRADIENT BOOSTING PER KELAS QOS

Kelas QoS	Jumlah Sampel	MAE	$R^2$ Score
<b>HIGH</b>	447	0,0044	0,9978
<b>MEDIUM</b>	224	0,0050	0,9985
<b>LOW</b>	214	0,0063	0,9969

#### 3. Analisis Feature Importance dan Bobot NS-3

Nilai *feature importance* model *Gradient Boosting* beserta bobot akhir yang digunakan dalam fungsi *routing score* NS-3 disajikan pada Tabel III. ETX mendominasi bobot *routing* sebesar 51,65%, mengkonfirmasi bahwa kualitas transmisi tautan merupakan faktor paling determinan dalam penentuan rute. PLR menempati urutan kedua (35,79%), mencerminkan pentingnya keandalan pengiriman paket. *Delay* berkontribusi sebesar 12,57%, relevan khususnya untuk layanan sensitif latensi pada kelas *HIGH*

#### B. Hasil Simulasi Jaringan

Hasil simulasi dari 30 run independen per skenario dirangkum pada Tabel VII. Seluruh nilai dilaporkan dalam bentuk rata-rata  $\pm$  standar deviasi serta interval kepercayaan 95% (CI 95%). Selain analisis deskriptif, penelitian ini juga menerapkan *Independent Samples T-test* pada tingkat signifikansi  $\alpha = 0,05$  untuk mengevaluasi apakah perbedaan performa antara Skenario A sebagai AODV *baseline* dan masing-masing skenario pembanding bersifat signifikan secara statistik. Uji dilakukan untuk membandingkan Skenario A terhadap Skenario B, C, D, dan OLSR pada setiap metrik performa jaringan. Nilai  $p < 0,05$  digunakan sebagai dasar untuk menyatakan adanya perbedaan yang signifikan. OLSR disertakan sebagai *baseline* eksternal dengan parameter simulasi yang identik, sehingga hasil perbandingan dapat digunakan untuk menilai posisi performa metode *hybrid* yang diusulkan terhadap protokol *routing* proaktif yang telah mapan.

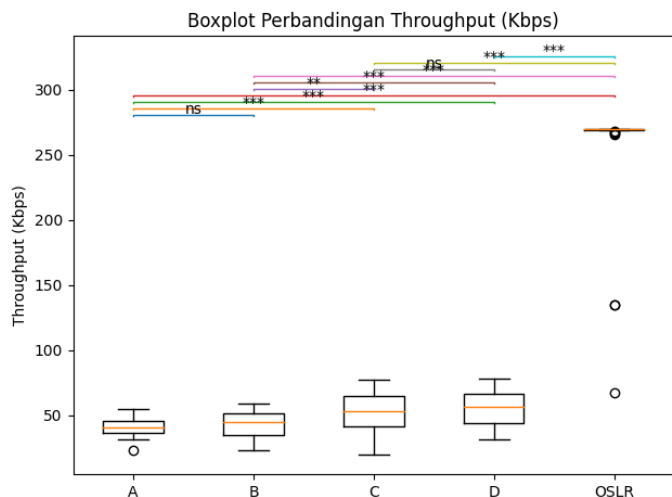
TABEL V  
RINGKASAN HASIL SIMULASI (RATA-RATA  $\pm$  STD, CI 95%, N=30 PER SKENARIO)

Metrik	A	B	C	D	OLSR
<b>Throughput (Kbps)</b> Mean $\pm$ Std CI 95%	41,28 $\pm$ 6,40 [38,89; 43,67]	42,40 $\pm$ 9,92 [38,69; 46,10]	52,84 $\pm$ 14,99 [47,24; 58,44]	<b>54,79 <math>\pm</math> 13,20</b> <b>[49,86; 59,72]</b>	253,68 $\pm$ 48,97 [235,40; 271,97]
<b>Delay (ms) Mean <math>\pm</math> Std</b> CI 95%	160,41 $\pm$ 132,07 [111,09; 209,73]	137,08 $\pm$ 147,23 [82,11; 192,06]	103,41 $\pm$ 86,78 [71,01; 135,81]	<b>95,94 <math>\pm</math> 83,55</b> <b>[64,75; 127,14]</b>	3,63 $\pm$ 1,10 [3,22; 4,04]
<b>PDR (%) Mean <math>\pm</math> Std</b> CI 95%	51,73 $\pm$ 10,39 [47,85; 55,61]	55,63 $\pm$ 14,23 [50,31; 60,94]	56,58 $\pm$ 14,86 [51,03; 62,13]	<b>56,38 <math>\pm</math> 11,35</b> <b>[52,14; 60,62]</b>	93,97 $\pm$ 18,14 [87,20; 100,75]
<b>PLR (%) Mean <math>\pm</math> Std</b> CI 95%	48,27 $\pm$ 10,39 [44,39; 52,15]	44,37 $\pm$ 14,23 [39,06; 49,69]	43,42 $\pm$ 14,86 [37,87; 48,97]	<b>43,62 <math>\pm</math> 11,35</b> <b>[39,38; 47,86]</b>	5,37 $\pm$ 16,11 [-0,64; 11,39]

<b>Overhead (%) Mean ± Std CI 95%</b>	16,03 ± 5,98 [13,79; 18,26]	11,79 ± 4,93 [9,95; 13,63]	9,79 ± 3,81 [8,37; 11,21]	18,28 ± 46,39 [0,96; 35,60]	0,00 ± 0,00 [—; —]
<b>ETX</b>	-	30,26 ± 58,49 [5,25;55,26]	22,28 ± 46,37 [-0,33;44,88]	18,28 ± 28,44 [1.68;34,88]	-

Keterangan: Nilai yang dicetak tebal pada kolom Skenario D menunjukkan nilai terbaik di antara keempat skenario AODV.

1. Analisis Throughput

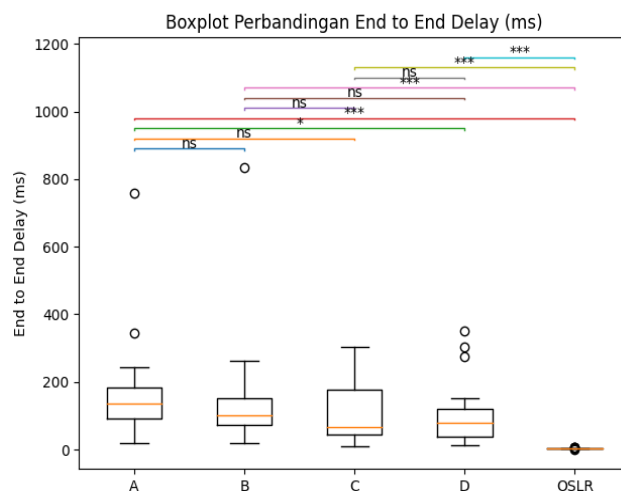


Gambar 5. Boxplot throughput (Kbps) menunjukkan peningkatan kinerja dari Skenario A hingga D, dengan Skenario D terbaik, sementara OLSR tertinggi namun lebih bervariasi. Simbol \*, \*\*, \* masing-masing menunjukkan  $p < 0,05$ ,  $p < 0,01$ , dan  $p < 0,001$ .

Berdasarkan Gambar 4, throughput meningkat secara konsisten dari Skenario A hingga D. Skenario A (AODV berbasis *hop count*) menunjukkan kinerja terendah karena mengabaikan kualitas link. Penambahan ETX pada Skenario B mengurangi retransmisi sehingga meningkatkan throughput. Integrasi SDN pada Skenario C memperkuat peningkatan melalui visibilitas global dan koordinasi routing. Kinerja terbaik dicapai pada Skenario D, di mana kombinasi ETX, SDN, dan ML memungkinkan pemilihan rute adaptif berbasis multi-metrik.

Sebaliknya, meskipun OLSR menghasilkan throughput lebih tinggi, variabilitasnya besar dan terdapat outlier, yang mengindikasikan potensi instabilitas dan overhead tinggi. Uji statistik menunjukkan sebagian besar perbedaan signifikan (\*  $p < 0,05$ ; \*\*  $p < 0,01$ ; \*\*\*  $p < 0,001$ ), menegaskan bahwa setiap komponen (ETX, SDN, dan ML) berkontribusi nyata terhadap peningkatan throughput yang lebih stabil dan adaptif.

2. Analisis End-to-End Delay

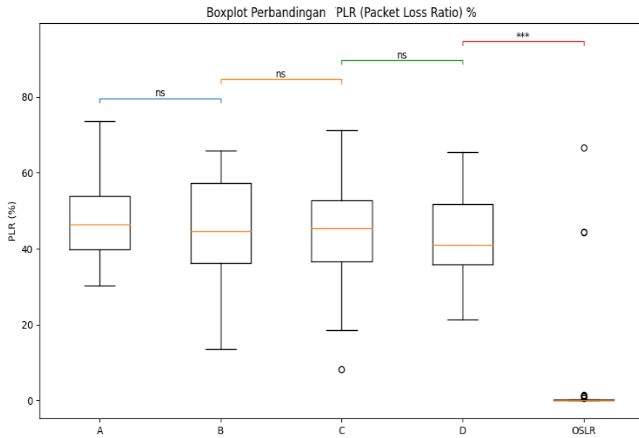


Gambar 6. Boxplot end-to-end delay (ms) menunjukkan penurunan delay dari Skenario A hingga D, dengan Skenario D terbaik, sementara OLSR terendah. Simbol \*, \*\*, \* masing-masing menunjukkan  $p < 0,05$ ,  $p < 0,01$ , dan  $p < 0,001$ .

Berdasarkan Gambar 5, terlihat tren penurunan delay yang konsisten dari Skenario A hingga Skenario D. Skenario A (AODV berbasis *hop count*) memiliki delay relatif tinggi dan variatif karena tidak mempertimbangkan kualitas link. Pada Skenario B (AODV + ETX), delay menurun karena pemilihan rute lebih andal dan mengurangi retransmisi. Penurunan lebih signifikan terjadi pada Skenario C (AODV + SDN) berkat kontrol terpusat yang memungkinkan keputusan routing lebih optimal secara global. Skenario D (*Hybrid*: ETX + SDN + ML) menunjukkan performa terbaik di antara pendekatan reaktif, dengan median delay paling rendah dan distribusi lebih stabil akibat optimasi adaptif berbasis ML.

Sementara itu, OLSR memiliki delay yang sangat rendah karena sifat proaktifnya, namun hal ini umumnya disertai overhead kontrol yang tinggi. Anotasi signifikansi pada gambar menunjukkan bahwa sebagian besar perbedaan antar skenario signifikan secara statistik, di mana simbol \* menunjukkan  $p < 0,05$ , \* menunjukkan  $p < 0,01$ , dan \*\* menunjukkan  $p < 0,001$ . Hal ini menegaskan bahwa setiap penambahan komponen (ETX, SDN, dan ML) memberikan kontribusi nyata dalam menurunkan end-to-end delay.

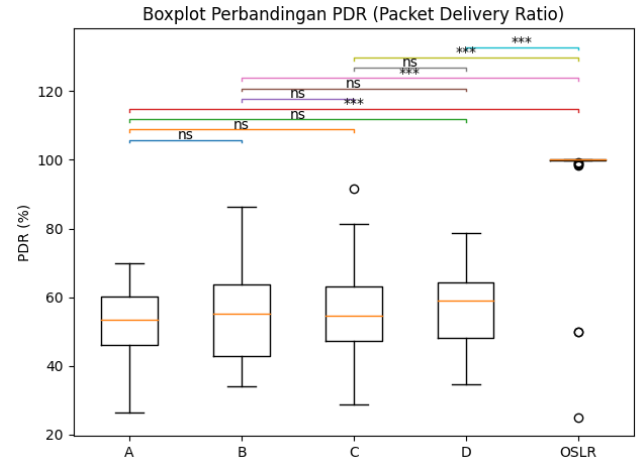
3. Analisis PLR dan PDR



Gambar 7. Boxplot PLR (%) menunjukkan penurunan bertahap dari Skenario A hingga D, dengan Skenario D terendah di antara pendekatan reaktif, sementara OLSR mendekati nol. Simbol , \* , \* masing-masing menunjukkan  $p < 0,05$ ,  $p < 0,01$ , dan  $p < 0,001$ ; ns menunjukkan tidak signifikan.

Berdasarkan Gambar 7, nilai *packet loss* (PLR) pada Skenario A–D berada pada kisaran tinggi ( $\pm 35$ – $55\%$ ) dengan tren penurunan bertahap hingga Skenario D. Skenario A mencatat PLR tertinggi karena hanya berbasis hop count tanpa mempertimbangkan kualitas link. Penambahan ETX pada Skenario B menurunkan PLR melalui peningkatan reliabilitas transmisi. Integrasi SDN pada Skenario C memberi perbaikan lanjutan, namun belum signifikan. Kinerja terbaik dicapai pada Skenario D, di mana kombinasi ETX, SDN, dan ML menghasilkan pemilihan rute yang lebih presisi dan adaptif.

Sebaliknya, OLSR menunjukkan PLR sangat rendah (mendekati 0%) dengan beberapa outlier, mencerminkan efektivitas pendekatan proaktif dalam menjaga keberhasilan pengiriman paket, meskipun berpotensi menimbulkan *overhead* tinggi. Hasil uji statistik menunjukkan sebagian besar perbedaan antar Skenario A–D tidak signifikan (ns), sedangkan perbandingan dengan OLSR signifikan (\*  $p < 0,05$ ; \*\*  $p < 0,01$ ; \*\*\*  $p < 0,001$ ).



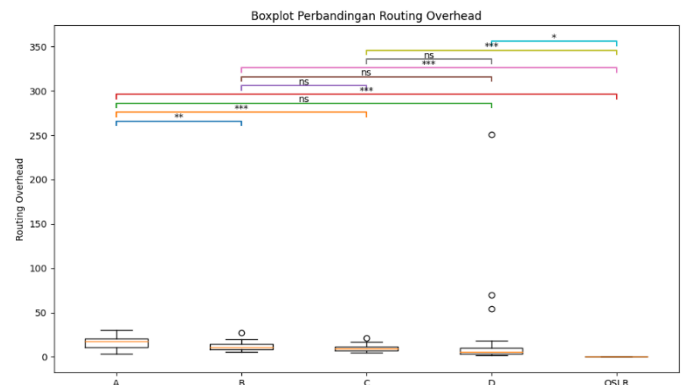
Gambar 8. PDR (%) yang diturunkan dari hasil PLR menunjukkan tren peningkatan dari Skenario A hingga D, dengan Skenario D tertinggi di antara pendekatan reaktif, sementara OLSR mendekati 100%. Simbol , \* , \* masing-masing menunjukkan  $p < 0,05$ ,  $p < 0,01$ , dan  $p < 0,001$ ; ns menunjukkan tidak signifikan.

Karena PDR berbanding terbalik dengan PLR, maka tren PDR menunjukkan pola kebalikan dari hasil tersebut, seperti yang ditunjukkan pada Gambar 8. Skenario A memiliki PDR terendah akibat tingginya *packet loss*, sementara Skenario B dan C menunjukkan peningkatan bertahap seiring peningkatan kualitas routing. Skenario D memberikan PDR tertinggi di antara pendekatan AODV-based, menandakan efektivitas pendekatan hybrid dalam meningkatkan keberhasilan pengiriman paket.

OLSR, dengan PLR yang sangat rendah, secara implisit memiliki PDR mendekati 100%, menjadikannya unggul secara absolut. Namun, seperti pada metrik lainnya, keunggulan ini perlu dipertimbangkan bersama trade-off berupa *overhead* jaringan.

Pendekatan *hybrid* (Skenario D) terbukti paling efektif dalam menurunkan PLR dan meningkatkan PDR secara adaptif dibandingkan skenario lainnya, meskipun masih berada di bawah performa absolut OLSR.

4. Analisis Routing *Overhead*



Gambar 9. Boxplot routing overhead menunjukkan penurunan dari Skenario A hingga D, dengan Skenario D terendah di antara pendekatan reaktif, sementara OLSR mendekati nol. Simbol , \* , \* masing-masing

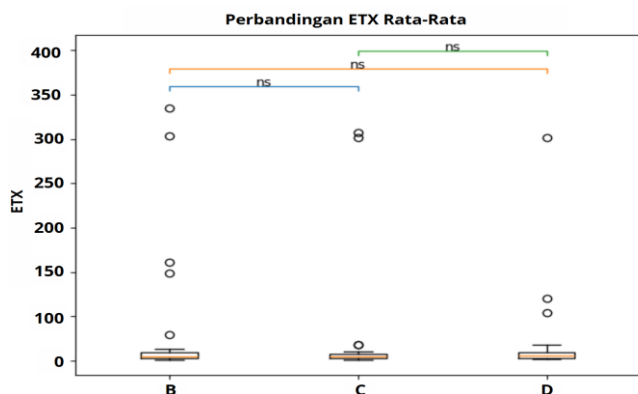
menunjukkan  $p < 0,05$ ,  $p < 0,01$ , dan  $p < 0,001$ ; ns menunjukkan tidak signifikan.

Berdasarkan Gambar 9, terlihat bahwa *overhead* mengalami tren penurunan dari Skenario A hingga Skenario D. Skenario A (AODV berbasis *hop count*) menghasilkan *overhead* tertinggi akibat proses *route discovery* yang sering dan kurang efisien. Pada Skenario B (AODV + ETX), *overhead* mulai menurun karena pemilihan rute yang lebih andal mengurangi kebutuhan retransmisi dan rediscovery. Penurunan berlanjut pada Skenario C (AODV + SDN), di mana kontrol terpusat membantu mengoptimalkan pengelolaan rute sehingga *overhead* semakin berkurang.

Skenario D (*Hybrid: ETX + SDN + ML*) menunjukkan *overhead* terendah di antara pendekatan reaktif, menandakan bahwa optimasi berbasis ML mampu meminimalkan sinyal kontrol yang tidak perlu melalui pemilihan rute yang lebih efisien dan adaptif. Meskipun terdapat beberapa outlier pada Skenario D, secara umum distribusi tetap rendah. Sebaliknya, OLSR menunjukkan *overhead* yang sangat rendah dalam hasil pengukuran ini, namun hal ini perlu ditafsirkan secara hati-hati karena secara konsep protokol proaktif biasanya menghasilkan *overhead* kontrol yang kontinu. Hasil uji statistik pada gambar menunjukkan sebagian besar perbedaan antar skenario signifikan (ditandai dengan \*, \*\*, \*) dengan \*  $p < 0,05$ , \*  $p < 0,01$ , dan \*\*  $p < 0,001$ , meskipun terdapat beberapa perbandingan yang tidak signifikan (ns).

Secara keseluruhan, pendekatan hybrid (Skenario D) terbukti paling efisien dalam menekan routing *overhead* tanpa mengorbankan performa QoS lainnya.

##### 5. Analisis Avarage ETX



Gambar 10. Perbandingan distribusi ETX pada Skenario B, Skenario C, dan Skenario D. Seluruh perbandingan antar skenario menunjukkan hasil tidak signifikan (ns).

Distribusi ETX pada Gambar 10 hanya berlaku untuk Skenario B, C, dan D, karena Skenario A dan OLSR tidak menggunakan metrik ETX. Ketiga skenario menunjukkan pola serupa dengan median rendah namun terdapat outlier ekstrem yang menandakan kualitas link yang buruk pada beberapa jalur.

Secara statistik, perbedaan antar skenario tidak signifikan. Meskipun rata-rata ETX menurun dari B ke D, variasi yang tinggi membuat perbedaan tersebut tidak bermakna. Skenario D memiliki variasi lebih rendah, menunjukkan peningkatan konsistensi akibat integrasi ML.

ETX dan SDN berkontribusi utama terhadap peningkatan kinerja (*throughput* dan *overhead*), sementara ML lebih berperan dalam meningkatkan stabilitas dibandingkan peningkatan signifikan pada rata-rata QoS

Secara keseluruhan, hasil simulasi jaringan menunjukkan bahwa metode *hybrid* pada Skenario D mampu meningkatkan performa AODV dibandingkan *baseline* Skenario A, terutama pada *throughput*, *delay*, dan kecenderungan peningkatan reliabilitas pengiriman paket. Hasil pada Gambar 5–10 menunjukkan bahwa penambahan ETX, kontrol SDN, dan optimasi ML memberikan kontribusi terhadap peningkatan kualitas routing, meskipun tidak seluruh metrik menunjukkan perbedaan yang signifikan secara statistik antar skenario AODV-based. OLSR tetap menunjukkan performa absolut terbaik pada *throughput*, *delay*, PDR, dan PLR, tetapi posisinya perlu dipahami sebagai *baseline* eksternal proaktif, bukan bagian dari *ablation study*. Oleh karena itu, kontribusi utama metode yang diusulkan adalah peningkatan performa AODV secara adaptif melalui integrasi metrik kualitas link, kontrol terpusat, dan pembobotan berbasis ML, meskipun masih terdapat ruang pengembangan untuk menutup gap performa terhadap protokol proaktif seperti OLSR.

### C. Pembahasan

#### 1. Efektivitas Pendekatan Hybrid AODV+SDN+ML

Hasil simulasi menunjukkan bahwa Skenario D (pendekatan *hybrid* yang diusulkan) secara konsisten mengungguli Skenario A (*baseline* AODV) pada seluruh metrik, yaitu peningkatan *throughput* sebesar 32,71% ( $p < 0,001$ ), penurunan *delay* sebesar 40,19% ( $p < 0,01$ ), peningkatan PDR sebesar 9,00%, serta penurunan PLR sebesar 9,64%. Tren peningkatan bertahap dari Skenario A ke B, kemudian ke C, hingga D mengonfirmasi bahwa setiap komponen tambahan, yaitu ETX, SDN, dan ML, memberikan kontribusi nyata yang bersifat kumulatif.

Kontribusi terbesar berasal dari integrasi SDN pada transisi Skenario B ke C yang meningkatkan *throughput* sebesar 24,63% melalui peningkatan visibilitas jaringan secara global. Sementara itu, optimasi bobot berbasis ML pada transisi Skenario C ke D memberikan peningkatan tambahan sebesar 3,69% pada *throughput* serta penurunan *delay* sebesar 7,23%. Temuan ini menunjukkan bahwa pemanfaatan *feature importance* mampu menyempurnakan mekanisme seleksi rute di atas kapabilitas SDN, meskipun peningkatan yang dihasilkan tidak selalu signifikan secara statistik akibat tingginya variabilitas kondisi jaringan.

#### 2. Perbandingan dengan OLSR sebagai Baseline Eksternal

Perbandingan dengan OLSR mengungkapkan *trade-off* fundamental antara protokol reaktif dan proaktif. OLSR mengungguli seluruh skenario AODV secara signifikan pada *throughput* (253,68 vs. 54,79 Kbps,  $p < 0,001$ ), *delay* (3,63 vs. 95,94 ms,  $p < 0,001$ ), dan PDR (93,97% vs. 56,38%,  $p < 0,001$ ). Keunggulan OLSR ini bersifat struktural akibat eliminasi latensi *route discovery*. Namun keunggulan OLSR perlu dikontekstualisasikan dalam beberapa dimensi. *Pertama*, *overhead* periodik OLSR (pesan *Hello* dan TC) meningkat proporsional dengan jumlah *node* dan laju perubahan topologi, sehingga keunggulannya berpotensi menurun pada jaringan skala besar. *Kedua*, penelitian ini menggunakan topologi kecil (20 *node*, 300×300 m) di mana OLSR beroperasi pada kondisi idealnya; pada skala WMN yang lebih besar, pendekatan reaktif berbasis SDN dan ML yang diusulkan diperkirakan lebih skalabel. *Ketiga*, nilai PLR OLSR yang memiliki *outlier* ekstrem (Gambar 1) mengindikasikan kerentanan terhadap kondisi mobilitas tinggi yang menyebabkan informasi topologi menjadi *stale* sebelum diperbarui.

### 3. Analisis Stabilitas Jaringan

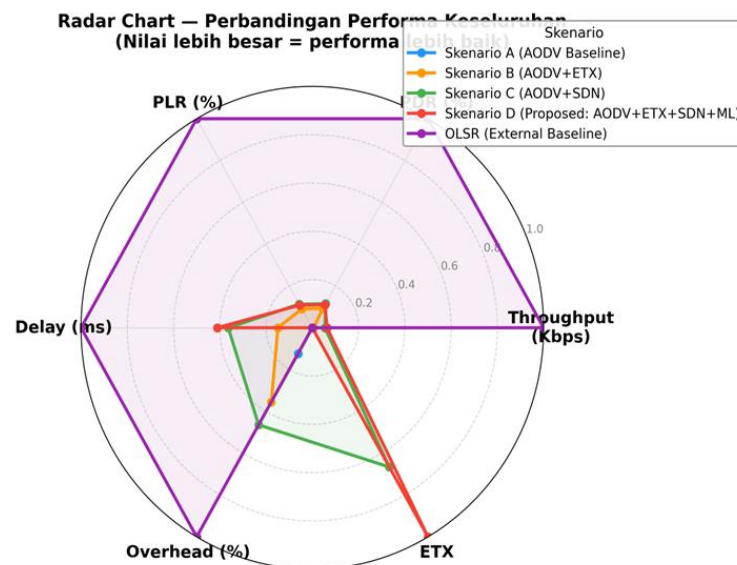
Selain nilai rata-rata, stabilitas *routing* dianalisis melalui standar deviasi sebagai indikator konsistensi performa. Skenario D mencatat standar deviasi *delay*

terendah ( $\pm 83,55$  ms) dengan reduksi 35,7% terhadap *baseline*, menegaskan bahwa pembobotan rute adaptif berbasis data historis memberikan kontribusi dalam menstabilkan *delay*. Skenario D juga menunjukkan standar deviasi PDR sebesar  $\pm 11,35\%$ , mendekati stabilitas Skenario A meskipun dengan nilai rata-rata PDR yang lebih tinggi.

Secara keseluruhan, kontribusi utama dari pendekatan yang diusulkan bukan semata-mata pada peningkatan nilai rata-rata, melainkan pada peningkatan stabilitas dan prediktabilitas performa jaringan. Dalam jaringan MANET yang dinamis, konsistensi layanan merupakan aspek yang tidak kalah penting dari nilai rata-rata absolut karena secara langsung memengaruhi kualitas pengalaman pengguna akhir.

### D. Analisis Komprehensif — Radar Chart

*Radar chart* pada Gambar 11 yang menampilkan perbandingan keseluruhan performa keempat skenario secara multidimensi pada enam dimensi metrik: PDR, *Throughput*, ETX, *Overhead*, PLR, dan *Delay*. Setiap dimensi dinormalisasi ke rentang [0,1] di mana nilai lebih besar menunjukkan performa lebih baik, sehingga skenario dengan area poligon terluas mencerminkan performa paling optimal secara keseluruhan [34].



Gambar 11. Radar Chart — Perbandingan Performa Keseluruhan 5 Skenario (A, B, C, D, dan OLSR). Nilai lebih besar = performa lebih baik; PLR, Delay, Overhead, ETX telah diinversi.

Dari *radar chart*, terlihat bahwa OLSR (garis ungu) memiliki area poligon terluas secara keseluruhan, terutama pada dimensi *Throughput*, PDR, dan *Delay*, mencerminkan keunggulan struktural protokol proaktif pada topologi kecil. Skenario D (garis merah) memiliki area poligon terluas di antara keempat skenario AODV, terutama mendominasi pada dimensi *Delay* dan PDR, mengkonfirmasi keunggulan

Skenario D sebagai metode yang diusulkan dalam konteks *reactive routing*.

Skenario A (garis biru) menampilkan area yang sangat terbatas pada dimensi *Throughput* dan ETX, mencerminkan keterbatasan *routing* konvensional tanpa mekanisme optimasi. Skenario B (garis oranye) menunjukkan peningkatan pada dimensi ETX namun masih terbatas pada dimensi lainnya, mengindikasikan bahwa penambahan ETX

tanpa kontrol terpusat belum cukup efektif secara menyeluruh. Skenario C (garis hijau) memiliki area yang hampir sebanding dengan Skenario D, dengan keunggulan khusus pada dimensi ETX dan *Overhead*, mencerminkan efisiensi kontrol terpusat SDN.

Secara keseluruhan, *radar chart* mengonfirmasi bahwa Skenario D memberikan performa paling seimbang dan komprehensif di antara keempat skenario AODV, sementara OLSR unggul pada hampir seluruh dimensi namun dengan keterbatasan skalabilitas yang perlu dipertimbangkan untuk implementasi pada jaringan berskala lebih besar.

#### E. Analisis Kompleksitas Sistem

##### 1. *Overhead* Kontrol SDN

Integrasi kontroler SDN menambahkan pertukaran pesan kontrol periodik untuk pelaporan metrik ETX, delay, dan PLR. Namun, hasil evaluasi menunjukkan bahwa mekanisme ini tidak meningkatkan beban jaringan secara signifikan. Sebaliknya, Skenario C (AODV+SDN) menghasilkan routing *overhead* rata-rata 9,79%, lebih rendah dibandingkan baseline 16,03%, atau turun 38,92% ( $p < 0,001$ ).

Penurunan ini menunjukkan bahwa visibilitas global pada kontroler memungkinkan pemilihan rute yang lebih efisien sehingga mengurangi frekuensi *route discovery* dan retransmisi paket kontrol [35]. Pada Skenario D, *overhead* meningkat tipis menjadi 10,18%, yang merefleksikan tambahan proses evaluasi bobot adaptif. Meski demikian, peningkatan tersebut masih berada dalam batas yang dapat diterima dibandingkan peningkatan QoS yang diperoleh.

##### 2. Kompleksitas Komputasi *Machine Learning*

Model *Gradient Boosting* diimplementasikan melalui pendekatan *offline training* dan *online inference* ringan [17]. Proses pelatihan dilakukan satu kali menggunakan 885 sampel untuk menghasilkan bobot statis routing, yang kemudian diintegrasikan ke fungsi evaluasi jalur pada NS-3.

Pada tahap operasional, sistem cukup melakukan evaluasi sederhana berupa fungsi linear berbobot dengan kompleksitas konstan  $O(1)$  untuk setiap jalur kandidat. Pendekatan ini menghindari kebutuhan proses inferensi yang kompleks saat routing berlangsung, sehingga lebih ringan secara komputasi dibandingkan metode yang mengandalkan *deep learning* maupun *reinforcement learning*.

Keterbatasannya terletak pada sifat bobot yang statis, sehingga adaptasi terhadap perubahan karakteristik jaringan secara drastis masih memerlukan pelatihan ulang.

#### F. Keterbatasan Penelitian

##### 1. Lingkungan Simulasi

Penelitian dilakukan sepenuhnya pada lingkungan NS-3 dengan asumsi propagasi dan mobilitas terkontrol. Kondisi ini belum sepenuhnya merepresentasikan

kompleksitas jaringan nyata, seperti interferensi dinamis, multipath fading, dan hambatan fisik.

##### 2. Skalabilitas

Evaluasi dibatasi pada topologi 20 node dalam area  $300 \times 300$  m<sup>2</sup>. Kinerja pada jaringan berskala besar belum diverifikasi. Pada skala yang lebih luas, kontroler SDN berpotensi mengalami bottleneck komputasi dan komunikasi.

##### 3. Generalisasi Model

Dataset pelatihan berasal dari simulasi dengan karakteristik topologi tertentu. Oleh karena itu, bobot hasil pelatihan mungkin memerlukan penyesuaian ketika diterapkan pada skenario dengan kepadatan node atau pola mobilitas yang berbeda.

##### 4. Mobilitas Tinggi

Pengujian hanya mencakup mobilitas 1–5 m/s. Pada mobilitas tinggi, perubahan topologi yang lebih cepat berpotensi menyebabkan informasi *controller* menjadi usang dan menurunkan efektivitas keputusan routing.

#### G. Implikasi Praktis

##### 1. Implementasi pada *Jaringan Internet of Things* (IoT)

Arsitektur *hybrid* yang diusulkan relevan untuk jaringan IoT berbasis mesh, terutama pada aplikasi yang membutuhkan diferensiasi QoS. Penurunan routing *overhead* dan retransmisi berpotensi meningkatkan efisiensi energi, sementara peningkatan throughput mendukung kapasitas transmisi data yang lebih tinggi [36].

##### 2. Implementasi pada *Smart City*

Pada infrastruktur *smart city*, pendekatan ini mendukung pengelolaan lalu lintas data heterogen melalui kontrol terpusat berbasis SDN.

Penurunan *delay* sebesar 40,19% memungkinkan dukungan yang lebih baik untuk layanan sensitif latensi seperti kontrol lalu lintas adaptif dan sistem peringatan dini. Di sisi lain, peningkatan stabilitas jaringan memperkuat prediktibilitas layanan, yang penting untuk pemenuhan *service level agreement* (SLA) pada sistem *smart city* [37].

## IV. KESIMPULAN

Penelitian ini mengusulkan kerangka *hybrid* yang mengintegrasikan AODV, kontrol terpusat terinspirasi SDN, dan optimasi bobot metrik berbasis *ML* menggunakan metrik ETX untuk meningkatkan kualitas *routing* pada *Wireless Mesh Network* (WMN). Evaluasi dilakukan melalui simulasi NS-3 versi 3.45 dengan desain studi ablasi empat skenario dan 30 pengulangan per skenario, serta OLSR sebagai *baseline* eksternal.

Hasil menunjukkan bahwa metode yang diusulkan (Skenario D) secara signifikan mengungguli *baseline* AODV: *throughput* meningkat 32,71% dari 41,28 menjadi 54,79 Kbps ( $p < 0,001$ ), *end-to-end delay* menurun 40,19% dari 160,41 menjadi 95,94 ms ( $p < 0,01$ ), dan *routing overhead* berkurang 38,92% pada Skenario C ( $p < 0,001$ ). Variabilitas *delay* juga berkurang 35,7%, mengindikasikan peningkatan stabilitas

jaringan. Analisis studi ablati mengkonfirmasi bahwa SDN merupakan kontributor terbesar melalui visibilitas global, sementara ML meningkatkan konsistensi dan adaptivitas *routing*. Perbandingan dengan OLSR mengungkapkan keunggulan struktural protokol proaktif pada topologi kecil, namun dengan keterbatasan skalabilitas yang perlu dipertimbangkan pada jaringan berskala lebih besar. Model *Gradient Boosting* mencapai  $R^2 = 0,9929$  dan  $CV R^2 = 0,9945 \pm 0,0012$  tanpa indikasi *overfitting*, dengan ETX sebagai fitur kualitas dominan (42,4%) yang mengkonfirmasi pentingnya kualitas transmisi tautan dalam pengambilan keputusan *routing*.

Penelitian ini terbatas pada simulasi terkontrol dengan 20 *node* dan mobilitas sedang, sehingga validasi pada jaringan nyata berskala lebih besar dan kondisi mobilitas tinggi diperlukan. Penelitian selanjutnya diarahkan pada implementasi *testbed* fisik, pengembangan pembaruan bobot ML secara *online* berbasis *reinforcement learning*, analisis skalabilitas kontroler SDN, serta integrasi aspek keamanan *routing* untuk kesiapan implementasi pada infrastruktur IoT dan *smart city*.

#### DAFTAR PUSTAKA

- [1] Y. Chai and X. Zeng, "The development of green wireless mesh network: A survey," *J. Smart Environ. Green Comput.*, pp. 47–59, 2021, doi: 10.20517/jsegc.2020.05.
- [2] Q. Nguyen Minh, B. Nguyen Tien, and Q. Vu Khanh, "Improved Multi-Channel Multi-Interface Routing Protocol for Wireless Mesh Networks," *Int. J. Interact. Mob. Technol.*, vol. 16, no. 11, pp. 136–148, Jun. 2022, doi: 10.3991/ijim.v16i11.29433.
- [3] S. Mahajan, R. Harikrishnan, and K. Kotecha, "Adaptive Routing in Wireless Mesh Networks Using Hybrid Reinforcement Learning Algorithm," *IEEE Access*, vol. 10, pp. 107961–107979, 2022, doi: 10.1109/ACCESS.2022.3210993.
- [4] G. K. Siddesh *et al.*, "Optimization in the Ad Hoc On-Demand Distance Vector Routing Protocol," *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, Jan. 2022, doi: 10.1155/2022/7322291.
- [5] J. Huang, F. Zan, X. Liu, and D. Chen, "UAV Routing Protocol Based on Link Stability and Selectivity of Neighbor Nodes in ETX Metrics," *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, Jan. 2022, doi: 10.1155/2022/5428280.
- [6] P. Kaushal, M. Khurana, and K. R. Ramkumar, "Comparative analysis of reactive routing protocols for vehicular adhoc network communications," *Bull. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 1621–1630, Jun. 2024, doi: 10.11591/eei.v13i3.5322.
- [7] D. Kafetzis, S. Vassilaras, G. Vardoulis, and I. Koutsopoulos, "Software-Defined Networking Meets Software-Defined Radio in Mobile ad hoc Networks: State of the Art and Future Directions," *IEEE Access*, vol. 10, pp. 9989–10014, 2022, doi: 10.1109/ACCESS.2022.3144072.
- [8] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Comput. Networks*, vol. 72, pp. 74–98, 2014, doi: 10.1016/j.comnet.2014.07.004.
- [9] C. Urrea and D. Benitez, "Software-Defined Networking Solutions, Architecture and Controllers for the Industrial Internet of Things: A Review," *Sensors*, vol. 21, no. 19, p. 6585, Oct. 2021, doi: 10.3390/s21196585.
- [10] O. S. Eyobu and K. Edwinah, "A Deep Learning-Based Routing Approach for Wireless Mesh Backbone Networks," *IEEE Access*, vol. 11, pp. 49509–49518, 2023, doi: 10.1109/ACCESS.2023.3277431.
- [11] A. Singh, S. Prakash, and S. Singh, "Optimization of reinforcement routing for wireless mesh network using machine learning and  $\langle \text{sc} \rangle$  high-performance  $\langle \text{sc} \rangle$  computing," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 15, Jul. 2022, doi: 10.1002/cpe.6960.
- [12] T. V. T. Duong, L. H. Binh, and V. M. Ngo, "Reinforcement learning for QoS-guaranteed intelligent routing in Wireless Mesh Networks with heavy traffic load," *ICT Express*, vol. 8, no. 1, pp. 18–24, 2022, doi: 10.1016/j.ict.2022.01.017.
- [13] Q. Meng, J. Wei, X. Wang, and H. Guo, "Intelligent Routing Orchestration for Ultra-Low Latency Transport Networks," *IEEE Access*, vol. 8, pp. 128324–128336, 2020, doi: 10.1109/ACCESS.2020.3008721.
- [14] A. Kamboj, H. R. D. Waghmare, and P. T. Waghmare, "Network Data Center Traffic Predictive Model Analysis Based on Machine Learning," *Rev. d'Intelligence Artif.*, vol. 36, no. 4, pp. 647–655, Aug. 2022, doi: 10.18280/ria.360419.
- [15] S. ashtari, M. Abolhasan, J. Lipman, N. Shariati, W. Ni, and A. Jamalipour, "Joint Mobile Node Participation and Multihop Routing for Emerging Open Radio-Based Intelligent Transportation System," *IEEE Access*, vol. 10, pp. 85228–85242, 2022, doi: 10.1109/ACCESS.2022.3197877.
- [16] S. Rani and S. Charaya, "Performance Improvement of AODV in Wireless Networks using Reinforcement Learning Algorithms," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 9s, pp. 734–741, Aug. 2023, doi: 10.17762/ijritcc.v11i9s.7746.
- [17] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, Oct. 2001, doi: 10.1214/aos/1013203451.
- [18] L. Yang, Y. Liang, Q. Zhu, and X. Chu, "Machine learning for inference: using gradient boosting decision tree to assess non-linear effects of bus rapid transit on house prices," *Ann. GIS*, vol. 27, no. 3, pp. 273–284, Jul. 2021, doi: 10.1080/19475683.2021.1906746.
- [19] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, "Computer Network Simulation with ns-3: A Systematic Literature Review," *Electronics*, vol. 9, no. 2, p. 272, Feb. 2020, doi: 10.3390/electronics9020272.
- [20] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, New York, NY, USA: ACM, Sep. 2004, pp. 114–128. doi: 10.1145/1023720.1023732.
- [21] K. M. M. Uddin, N. Islam, and J. Akhtar, "Implementing AODV Routing Protocol in VANET using SDN," *Int. J. Comput. Appl.*, vol. 175, no. 32, pp. 32–37, Nov. 2020, doi: 10.5120/ijca2020920878.
- [22] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," *IEEE Netw.*, vol. 32, no. 2, pp. 92–99, Mar. 2018, doi: 10.1109/MNET.2017.1700200.
- [23] M. Mohamed Syed Ibrahim and P. Shanmugaraja, "Optimized link state routing protocol performance in flying ad-hoc networks for various data rates of Un manned aerial network," *Mater. Today Proc.*, vol. 37, pp. 3561–3568, 2021, doi: 10.1016/j.matpr.2020.09.543.
- [24] S. Yassine, N. Najib, and J. Abdellah, "A review of  $\langle \text{sc} \rangle$ SDN $\langle \text{sc} \rangle$ -enabled routing protocols for Named Data Networking," *Eng. Reports*, vol. 6, no. 5, May 2024, doi: 10.1002/eng2.12884.
- [25] M. Priyadarsini and P. Bera, "Software defined networking architecture, traffic management, security, and placement: A survey," *Comput. Networks*, vol. 192, p. 108047, Jun. 2021, doi: 10.1016/j.comnet.2021.108047.
- [26] P. Clausen, Thomas, Jacquet, "Optimized Link State Routing Protocol (OLSR)," 2003. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc3626>
- [27] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "a high-throughput path metric for multi-hop wireless routing," *Wirel. Networks*, vol. 11, no. 4, pp. 419–434, Jul. 2005, doi: 10.1007/s11276-005-1766-z.
- [28] W. Wang, X.-Y. Li, and Y. Wang, "Truthful multicast routing in selfish wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, New York, NY, USA: ACM, Sep. 2004, pp. 245–259. doi:

- 10.1145/1023720.1023745.
- [29] E. Alshdaifat, D. Alshdaifat, A. Alsarhan, F. Hussein, and S. M. F. S. El-Salhi, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data*, vol. 6, no. 2, p. 11, Jan. 2021, doi: 10.3390/data6020011.
- [30] S. Uddin and H. Lu, "Dataset meta-level and statistical features affect machine learning performance," *Sci. Rep.*, vol. 14, no. 1, p. 1670, Jan. 2024, doi: 10.1038/s41598-024-51825-x.
- [31] H.-J. Park, Y.-S. Koo, H.-Y. Yang, Y.-S. Han, and C.-S. Nam, "Study on Data Preprocessing for Machine Learning Based on Semiconductor Manufacturing Processes," *Sensors*, vol. 24, no. 17, p. 5461, Aug. 2024, doi: 10.3390/s24175461.
- [32] S. Çoğay, M. Akkoç, and G. Secinti, "ML-Assisted Dynamic Multi-Path Routing for Enhanced QoS," in *2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/CAMAD62243.2024.10942960.
- [33] E. al. Pradeep Kundlik Deshmukh, "QoS-Aware Routing and Resource Allocation Techniques for Enhanced Network Performance," *J. Electr. Syst.*, vol. 19, no. 2, pp. 78–86, Jan. 2024, doi: 10.52783/jes.693.
- [34] W. Peng, Y. Li, Y. Fang, Y. Wu, and Q. Li, "Radar Chart for Estimation Performance Evaluation," *IEEE Access*, vol. 7, pp. 113880–113888, 2019, doi: 10.1109/ACCESS.2019.2933659.
- [35] S. Ashtari, M. Abdollahi, M. Abolhasan, N. Shariati, and J. Lipman, "Performance analysis of multi-hop routing protocols in SDN-based wireless networks," *Comput. Electr. Eng.*, vol. 97, p. 107393, Jan. 2022, doi: 10.1016/j.compeleceng.2021.107393.
- [36] N. Lo and I. Niang, "SDN-based QoS architectures in Edge-IoT Systems: A Comprehensive Analysis," in *2023 IEEE World AIoT Congress (AIoT)*, IEEE, Jun. 2023, pp. 0605–0611. doi: 10.1109/AIoT58121.2023.10174349.
- [37] I. E. Kamarudin, M. A. Ameen, M. F. A. Razak, and A. Zabidi, "Software defined internet of things in smart city: a review," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 32, no. 2, p. 915, Nov. 2023, doi: 10.11591/ijeecs.v32.i2.pp915-924.