Teknik: Jurnal Ilmu Teknik dan Informatika
Volume 6 Nomor 2 Oktober 2026
e-ISSN: 2798-2513; p-ISSN: 2808-8751, Hal *172-188*
DOI: https://doi.org/10.51903/teknik.v6i1.1283
Available online at: https://journal.stiestekom.ac.id/index.php/TEKNIK

Komparasi Arsitektur Deep Learning Berbasis Attention untuk Prediksi Quality of Service (QoS) Jaringan

Yehezkiel Haganta Tarigan[1], Zulfahmi Indra[2], M. Zikry Sahendra Chan[3], Angelina S. Saragih[4]
[1-3]Ilmu Komputer, Fakultas Matematika dan Pengetahuan Alam, Universitas Negeri Medan

Korespondensi penulis: angelina25saragih@gmail.com

Abstract. *Quality of Service (QoS) is an important indicator in computer network management as it directly affects the quality of services experienced by users. Accurate QoS prediction can assist network administrators in making proactive decisions and improving network performance. This study compares the performance of four deep learning architectures, namely Conv1D, LSTM, Conv1D+Attention, and LSTM+Attention, for time-series QoS prediction. The dataset consists of 95,611 observations collected from a GNS3-based network simulation environment, including three QoS parameters: traffic_in_bps, traffic_out_bps, and latency_ms. Prior to model training, the data were processed using Min-Max Scaling and a sliding window approach to generate sequential inputs. Model performance was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). Experimental results show that the LSTM+Attention model achieved the best performance with an MAE of 0.003224, an RMSE of 0.013578, and an R^2 of 0.943872. Meanwhile, the Conv1D model demonstrated competitive performance with results close to the best-performing model. Further analysis indicates that the attention mechanism improved the performance of the LSTM architecture but did not provide similar benefits when applied to Conv1D. These findings suggest that the effectiveness of attention mechanisms is influenced by the characteristics of the underlying architecture used for QoS prediction*.
Keywords: *Deep learning architecture; Attention mechanism; Quality of Service; Network prediction; Model comparison*

Abstrak. Quality of Service (QoS) merupakan salah satu indikator penting dalam pengelolaan jaringan komputer karena memengaruhi kualitas layanan yang diterima pengguna. Kemampuan memprediksi kondisi QoS secara akurat dapat membantu administrator jaringan melakukan pengambilan keputusan secara proaktif. Penelitian ini bertujuan membandingkan performa empat arsitektur deep learning, yaitu Conv1D, LSTM, Conv1D+Attention, dan LSTM+Attention untuk prediksi QoS jaringan berbasis deret waktu. Dataset yang digunakan terdiri dari 95.611 observasi yang diperoleh melalui simulasi jaringan menggunakan GNS3 dengan tiga parameter utama, yaitu traffic_in_bps, traffic_out_bps, dan latency_ms. Sebelum pelatihan, data diproses menggunakan normalisasi Min-Max Scaling dan metode sliding window untuk membentuk data sekuensial. Evaluasi model dilakukan menggunakan Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan koefisien determinasi (R^2). Hasil eksperimen menunjukkan bahwa model LSTM+Attention memperoleh performa terbaik dengan nilai MAE sebesar 0,003224, RMSE sebesar 0,013578, dan R^2 sebesar 0,943872. Sementara itu, Conv1D menghasilkan performa yang kompetitif dengan nilai yang mendekati model terbaik. Analisis lebih lanjut menunjukkan bahwa penambahan mekanisme attention meningkatkan performa pada arsitektur LSTM, namun tidak memberikan peningkatan pada arsitektur Conv1D. Temuan ini menunjukkan bahwa efektivitas mekanisme attention dipengaruhi oleh karakteristik arsitektur dasar yang digunakan dalam prediksi QoS jaringan.
Kata Kunci: *Arsitektur deep learning; Mekanisme atensi; Quality of Service; Prediksi jaringan; Komparasi model.*

1. LATAR BELAKANG

Perkembangan layanan digital seperti cloud computing, Internet of Things (IoT), dan aplikasi real-time menyebabkan peningkatan kompleksitas trafik jaringan sehingga kualitas layanan menjadi aspek yang semakin penting. Dalam konteks ini, Quality of Service (QoS) digunakan sebagai indikator performa jaringan melalui parameter seperti throughput, latency, jitter, dan packet loss yang secara langsung memengaruhi pengalaman pengguna (Awaliyani & Aziz, n.d.; Tambunan, 2020).

Diterima: 12 Juni 2026; Direvisi: 28 Juli 2026; Disetujui: 15 September 2026; Tersedia Online: 02 Oktober 2026; Diterbitkan: 19 Oktober 2026.

Prediksi QoS menjadi kebutuhan penting karena memungkinkan administrator jaringan melakukan pengelolaan sumber daya secara proaktif. Namun, karakteristik data QoS yang bersifat dinamis, nonlinier, dan memiliki dependensi temporal menjadikan proses prediksi sebagai tantangan tersendiri. Oleh karena itu, berbagai pendekatan deep learning mulai banyak digunakan untuk meningkatkan akurasi prediksi dibandingkan metode konvensional (Hameed et al., 2022).

Di antara berbagai arsitektur deep learning, Long Short-Term Memory (LSTM) dikenal efektif dalam memodelkan hubungan temporal jangka panjang, sedangkan Convolutional Neural Network satu dimensi (Conv1D) mampu mengekstraksi pola lokal secara efisien pada data deret waktu (Bai et al., 2018; Na et al., 2023). Perkembangan selanjutnya menghadirkan mekanisme attention yang memungkinkan model memfokuskan perhatian pada informasi yang paling relevan, sehingga berhasil meningkatkan performa berbagai tugas prediksi deret waktu dan QoS jaringan (Abbasimehr & Paki, 2022; Hameed et al., 2022).

Meskipun berbagai penelitian telah menunjukkan keberhasilan penggunaan Conv1D, LSTM, maupun mekanisme attention secara individual, studi yang secara khusus membandingkan pengaruh attention terhadap arsitektur Conv1D dan LSTM pada prediksi Quality of Service (QoS) jaringan masih relatif terbatas. Sebagian besar penelitian berfokus pada pengembangan model tunggal atau arsitektur hibrida tanpa melakukan evaluasi komparatif pada kondisi eksperimen yang sama (Saha et al., 2024; Yuliana et al., 2026). Oleh karena itu, penelitian ini bertujuan membandingkan performa empat arsitektur deep learning, yaitu Conv1D, LSTM, Conv1D+Attention, dan LSTM+Attention untuk memprediksi QoS jaringan berdasarkan data deret waktu. Hasil penelitian diharapkan dapat memberikan pemahaman mengenai efektivitas mekanisme attention pada masing-masing arsitektur serta menjadi referensi dalam pengembangan sistem prediksi QoS yang lebih akurat.

2. KAJIAN TEORITIS

Quality of Service (QoS) merupakan ukuran performa jaringan yang umumnya dinilai berdasarkan parameter throughput, latency, jitter, dan packet loss. Informasi QoS sangat penting dalam menjaga kualitas layanan jaringan karena mencerminkan kondisi aktual yang diterima pengguna (Awaliyani & Aziz, n.d.; Tambunan, 2020).

Prediksi QoS termasuk permasalahan multivariate time series forecasting yang memerlukan kemampuan untuk menangkap pola temporal yang kompleks. Dalam konteks ini, deep learning banyak digunakan karena mampu mempelajari hubungan nonlinier dan

dependensi waktu secara otomatis tanpa memerlukan rekayasa fitur yang kompleks (Bai et al., 2018).

LSTM merupakan salah satu arsitektur yang dirancang untuk memodelkan data sekuensial melalui mekanisme memori yang mampu mempertahankan informasi penting dalam jangka panjang. Di sisi lain, Conv1D menggunakan operasi konvolusi untuk mengekstraksi pola lokal pada data deret waktu dengan kompleksitas komputasi yang lebih rendah (Bai et al., 2018; Na et al., 2023).

Perkembangan deep learning juga menghadirkan mekanisme attention yang memungkinkan model memberikan bobot lebih besar pada informasi yang dianggap relevan dalam proses prediksi. Mekanisme ini menjadi dasar berbagai model modern dan telah terbukti meningkatkan akurasi prediksi deret waktu, termasuk pada prediksi QoS dan trafik jaringan (Hameed et al., 2022; Vaswani et al., n.d.). Integrasi attention dengan LSTM maupun arsitektur hibrida lainnya juga dilaporkan mampu menghasilkan performa yang lebih baik dibandingkan model dasar pada berbagai studi sebelumnya (Abbasimehr & Paki, 2022; Yuliana et al., 2026).

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen komparatif kuantitatif yang bertujuan untuk mengevaluasi dan membandingkan performa berbagai arsitektur *deep learning* dalam memprediksi *Quality of Service* (QoS) jaringan. Alur penelitian dirancang secara terstruktur, mencakup tahapan simulasi lingkungan jaringan, pengumpulan data metrik QoS, prapemrosesan deret waktu, hingga pelatihan dan evaluasi model prediktif.

Untuk mengatasi keterbatasan ketersediaan *dataset* QoS publik yang spesifik, data yang digunakan dalam penelitian ini dibangkitkan secara mandiri melalui lingkungan simulasi jaringan. Simulasi dibangun secara virtual menggunakan perangkat *router* dan *host* yang merepresentasikan topologi *client-server* sederhana

Data metrik QoS yang terekam dari simulasi tersebut selanjutnya diekstraksi dan diproses menjadi format sekuensial untuk melatih empat varian arsitektur *deep learning*. Empat model yang akan diuji dan dibandingkan meliputi: arsitektur dasar *1-Dimensional Convolutional Neural Network* (Conv1D) , *Long Short-Term Memory* (LSTM) , serta varian hibrida berupa Conv1D yang diintegrasikan dengan mekanisme *Multi-Head Attention* dan LSTM dengan mekanisme *Multi-Head Attention*.

Fokus utama dari desain komparatif ini adalah untuk mengukur efektivitas komponen ekstraksi fitur temporal dasar (Conv1D dan LSTM) , sekaligus menganalisis secara empiris pengaruh penambahan lapisan *Attention* terhadap kemampuan model dalam menangkap

dependensi pola jangka panjang pada data QoS. Hasil pelatihan dari keempat model tersebut kemudian akan dievaluasi menggunakan metrik kesalahan standar untuk menentukan arsitektur yang memberikan performa terbaik pada lingkungan eksperimen yang digunakan dalam penelitian ini.

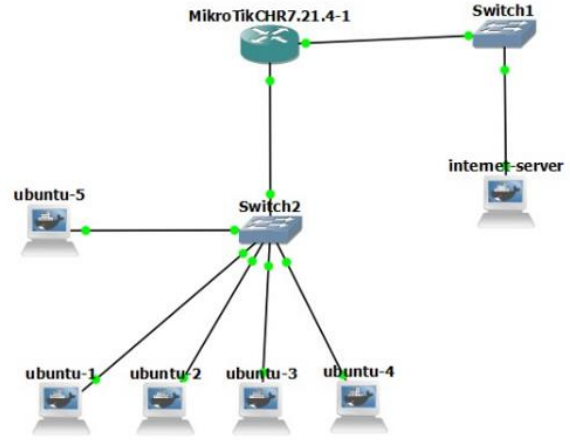
Lingkungan Eksperimen

Lingkungan eksperimen dalam penelitian ini dibangun menggunakan simulator jaringan GNS3 untuk merepresentasikan arsitektur jaringan *client-server*. Berdasarkan topologi yang diimplementasikan, infrastruktur jaringan ini menggunakan satu unit *router* MikroTik CHR (versi 7.21.4-1) sebagai pusat pengaturan rute trafik utama. *Router* tersebut terhubung ke dua buah *switch* virtual; *Switch1* dialokasikan khusus untuk menghubungkan *router* dengan *internet-server* (IP peladen: 200.200.200.8), sedangkan *Switch2* berfungsi sebagai konsentrator yang menghubungkan jalur komunikasi ke lima buah *node host* berbasis Ubuntu.

Pembagian peran dan fungsi dari setiap komponen di dalam lingkungan simulasi GNS3 ini dirancang secara terdekopel untuk menjaga akurasi pemantauan. Detail mengenai fungsionalitas masing-masing *node* di dalam jaringan dirangkum pada Tabel 1

Tabel 1. Daftar Perangkat dan Fungsi Node dalam Jaringan

Node	Fungsi
MikroTik CHR	Router utama yang mengelola rute trafik
Ubuntu 1–4	Generator trafik untuk mensimulasikan beban jaringan
Ubuntu 5	Pengumpul data (*collector*) metrik QoS jaringan
Internet Server	Peladen tujuan seluruh aktivitas trafik (*iperf3 server*)



Gambar 1. Topologi Jaringan

Proses pembangkitan trafik dilakukan secara terdistribusi pada *node* Ubuntu 1 hingga Ubuntu 4 menuju ke *Internet Server*. Sementara itu, Ubuntu 5 bertindak secara pasif sebagai pengumpul data (*collector*) yang merekam fluktuasi metrik kinerja jaringan tanpa mengganggu komunikasi utama antara *client* dan peladen.

Skenario Trafik

Untuk menghasilkan data deret waktu QoS yang memiliki variabilitas tinggi dan mendekati karakteristik dunia nyata, pembangkitan beban jaringan dilakukan melalui empat skenario aktivitas yang berbeda. Seluruh aktivitas trafik dibangkitkan menggunakan utilitas *iperf3* yang berjalan di latar belakang (*background mode*) pada *Internet Server* dengan membuka jajaran port dari 5201 hingga 5230.

Guna menghindari pola trafik yang bersifat linear atau deterministik, setiap skenario menggunakan fungsi acak berbasis distribusi Gaussian (*Gaussian/Normal Distribution*) untuk mengatur parameter *bandwidth*, durasi pengiriman paket, serta jeda waktu antar-aktivitas (*think time*). Rincian konfigurasi dari keempat skenario trafik tersebut adalah sebagai berikut:

Web Browsing (Aktivitas Mahasiswa pada ubuntu-1): Skenario ini mensimulasikan aktivitas penelusuran halaman web ringan. Trafik dibangkitkan melalui 5 proses simultan (port 5201–5205) dengan karakteristik jeda baca (*think time*) rata-rata 10 detik dan durasi sesi rata-rata 30 detik. Alokasi *bandwidth* diatur dengan rata-rata unduh (*download*) sebesar 6 Mbps dan unggah (*upload*) sebesar 1 Mbps, di mana probabilitas aktivitas didominasi oleh unduhan sebesar 80%.

Social Media & Streaming Ringan (Aktivitas pada ubuntu-2): Skenario ini merepresentasikan perilaku pengguna saat mengakses media sosial atau memutar video berdurasi pendek. Pembangkitan menggunakan 5 proses simultan (port 5211–5215) dengan jeda antar-aksi yang lebih rapat (rata-rata 4 detik) dan durasi rata-rata 20 detik. Kebutuhan *bandwidth* diatur dengan rata-rata unduh sebesar 8 Mbps dan unggah sebesar 2 Mbps, dengan probabilitas unduhan sebesar 85%.

Download Skala Besar (Aktivitas pada ubuntu-3): Skenario ini mensimulasikan proses pemindahan data berkas berukuran besar yang membebani kapasitas jaringan secara intensif. Dijalankan melalui 3 proses (port 5221–5223), aktivitas ini murni melakukan pengunduhan data secara kontinu dengan alokasi *bandwidth* tinggi rata-rata 15 Mbps, durasi pengiriman yang lebih panjang (rata-rata 40 detik), dan jeda antar-sesi rata-rata 12 detik.

Data Backup / Upload (Aktivitas pada ubuntu-4): Skenario ini mensimulasikan aktivitas pencadangan data ke peladen luar secara periodik. Berjalan melalui 2 proses (port 5226–5227),

skenario ini fokus pada aktivitas unggah data tanpa mekanisme *reverse* unduhan, dengan parameter *bandwidth* unggah rata-rata 4 Mbps, durasi rata-rata 30 detik, dan jeda waktu istirahat antar-pencadangan rata-rata 8 detik.

Melalui kombinasi keempat skenario acak yang berjalan secara simultan ini, parameter performa jaringan pada *router* utama akan mengalami fluktuasi yang dinamis, yang kemudian diekstraksi sebagai parameter data latih model *deep learning*.

Pengumpulan Data

Proses pengumpulan metrik *Quality of Service* (QoS) dilakukan secara otomatis melalui skrip pemantauan berbasis Python yang diimplementasikan pada *node* khusus (Ubuntu 5). Pengumpulan data ini dilakukan secara proaktif dengan menginterogasi perangkat *router* MikroTik (sebagai titik kumpul trafik) melalui protokol *Secure Shell* (SSH).

Untuk memastikan proses otomasi berjalan secara stabil dan presisi, skrip dikembangkan dengan mengintegrasikan beberapa pustaka Python, antara lain: paramiko untuk mengelola autentikasi dan eksekusi perintah pemantauan jarak jauh ke MikroTik, pustaka ekspresi reguler (re) untuk mengekstraksi dan mengonversi nilai teks *byte* menjadi besaran bit per detik (bps), serta subprocess untuk melakukan injeksi paket pemantauan ICMP (*ping*) secara sistematis guna mengukur waktu tunda transmisi jaringan terhadap peladen internet.

Pengambilan sampel data (*sampling*) diatur untuk terekam pada interval setiap satu detik. Untuk setiap sampel, skrip mengekstraksi tiga variabel kinerja fungsional, yaitu volume data masuk kumulatif (traffic_in_bps), volume data keluar kumulatif (traffic_out_bps), serta hasil pengukuran waktu tunda jaringan (latency_ms). Sebagai bentuk penanganan galat (*fallback mechanism*), apabila terjadi kegagalan eksekusi *ping* yang menyebabkan putusnya pencatatan *latency*, sistem akan secara otomatis menduplikasi nilai metrik *latency* terakhir yang valid untuk menghindari timbulnya kekosongan data (*missing values*). Seluruh parameter metrik ini selanjutnya dicatat secara otomatis, beserta penanda waktu kronologisnya (timestamp), ke dalam format berkas *Comma-Separated Values* (CSV) untuk digunakan sebagai set data latih pada fase pemodelan *deep learning*.

Preprocessing

Sebelum digunakan untuk melatih model deep learning, data Quality of Service (QoS) yang telah dikumpulkan melalui proses monitoring jaringan terlebih dahulu melalui beberapa tahapan prapemrosesan. Tahapan tersebut bertujuan untuk memastikan data berada dalam format yang sesuai dengan kebutuhan model prediksi deret waktu. Ringkasan tahapan preprocessing ditunjukkan pada Tabel 3.

Tabel 2. Tahapan Preprocessing Data

Tahapan	Deskripsi
Pembersihan Data	Memastikan tidak terdapat nilai hilang (*missing values*) maupun anomali yang dapat mengganggu proses pelatihan model.
Normalisasi	Melakukan penskalaan fitur menggunakan metode Min-Max Scaling ke rentang [0,1] untuk menyamakan skala antar variabel QoS.
Sliding Window	Mengubah data deret waktu menjadi bentuk sekuens dengan memanfaatkan sejumlah observasi sebelumnya (*look-back window*) sebagai masukan model.
Train-Test Split	Membagi data menjadi himpunan data latih dan data uji untuk proses pelatihan dan evaluasi model.

Setelah proses normalisasi selesai dilakukan, data ditransformasikan menjadi tensor tiga dimensi menggunakan metode sliding window sehingga sesuai dengan format masukan yang dibutuhkan oleh arsitektur Conv1D, LSTM, maupun model berbasis Attention. Data yang telah ditransformasikan kemudian dipisahkan menjadi data latih dan data uji untuk mengevaluasi kemampuan generalisasi model terhadap data yang belum pernah diamati sebelumnya.

Arsitektur Model

Conv1D

Arsitektur Conv1D digunakan sebagai model baseline berbasis konvolusi untuk menangkap pola lokal pada data deret waktu QoS. Model ini terdiri atas dua lapisan Conv1D yang berfungsi mengekstraksi fitur temporal dari data masukan, diikuti oleh lapisan pooling untuk mereduksi dimensi representasi fitur. Selanjutnya, representasi fitur diratakan (Flatten) dan diteruskan ke lapisan Dense untuk menghasilkan prediksi nilai QoS pada horizon waktu berikutnya.

Layer (type)	Output Shape	Param #
input_layer_25 (InputLayer)	(None, 120, 3)	0
conv1d_19 (Conv1D)	(None, 120, 64)	640
conv1d_20 (Conv1D)	(None, 120, 64)	12,352
global_average_pooling1d_12 (GlobalAveragePooling1D)	(None, 64)	0
dense_42 (Dense)	(None, 64)	4,160
dense_43 (Dense)	(None, 3)	195
reshape_24 (Reshape)	(None, 1, 3)	0

Gambar 2. Arsitektur Baseline Conv1D

LSTM

Arsitektur LSTM digunakan sebagai model baseline berbasis jaringan rekuren yang dirancang untuk menangkap dependensi temporal jangka panjang. Model memanfaatkan lapisan LSTM sebagai komponen utama untuk memproses urutan data QoS secara sekuensial. Representasi temporal yang dihasilkan kemudian diteruskan ke lapisan Dense guna menghasilkan nilai prediksi pada langkah waktu berikutnya.

Layer (type)	Output Shape	Param #
input_layer_26 (InputLayer)	(None, 120, 3)	0
lstm_18 (LSTM)	(None, 128)	67,584
dense_44 (Dense)	(None, 64)	8,256
dense_45 (Dense)	(None, 3)	195
reshape_25 (Reshape)	(None, 1, 3)	0

Gambar 3. Arsitektur Baseline LSTM

Conv1D + Attention

Model Conv1D + Attention dibangun dengan menambahkan mekanisme Multi-Head Attention pada arsitektur Conv1D. Lapisan Conv1D bertugas mengekstraksi pola lokal dari data QoS, sedangkan mekanisme attention digunakan untuk memberikan bobot yang berbeda terhadap bagian sekuens yang dianggap paling relevan. Integrasi kedua komponen ini diharapkan mampu meningkatkan kemampuan model dalam memahami hubungan temporal yang lebih kompleks dibandingkan Conv1D konvensional.

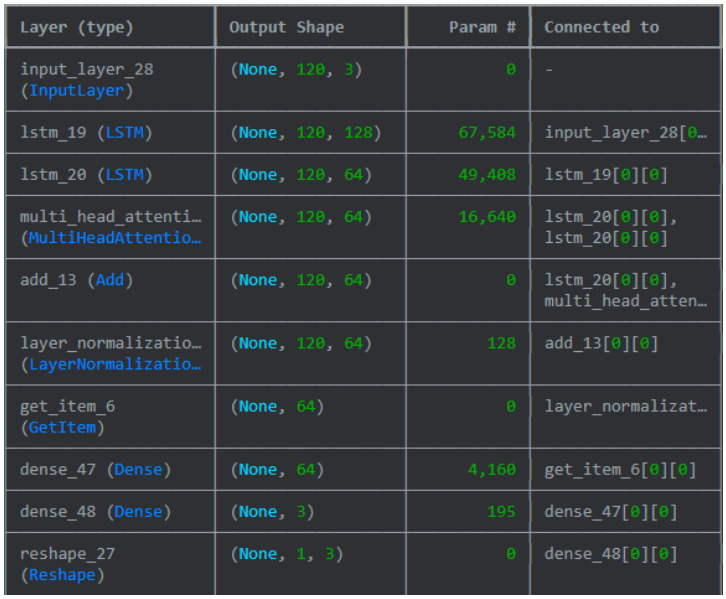
Layer (type)	Output Shape	Param #	Connected to
input_layer_27 (InputLayer)	(None, 120, 3)	0	-
conv1d_21 (Conv1D)	(None, 120, 64)	640	input_layer_27[0…
multi_head_attenti… (MultiHeadAttentio…	(None, 120, 64)	16,640	conv1d_21[0][0], conv1d_21[0][0]
add_12 (Add)	(None, 120, 64)	0	conv1d_21[0][0], multi_head_atten…
layer_normalizatio… (LayerNormalizatio…	(None, 120, 64)	128	add_12[0][0]
global_average_poo… (GlobalAveragePool…	(None, 64)	0	layer_normalizat…
dense_46 (Dense)	(None, 3)	195	global_average_p…
reshape_26 (Reshape)	(None, 1, 3)	0	dense_46[0][0]

Gambar 4. Arsitektur Attention-Conv1D

LSTM + Attention

Model LSTM + Attention menggabungkan kemampuan LSTM dalam memodelkan dependensi temporal dengan mekanisme Multi-Head Attention untuk meningkatkan fokus

model terhadap informasi historis yang penting. Keluaran dari lapisan LSTM diproses lebih lanjut oleh modul attention sebelum diteruskan ke lapisan Dense. Arsitektur ini dirancang untuk mengevaluasi apakah penambahan attention dapat meningkatkan performa prediksi dibandingkan LSTM standar.



Layer (type)	Output Shape	Param #	Connected to
input_layer_28 (InputLayer)	(None, 120, 3)	0	-
lstm_19 (LSTM)	(None, 120, 128)	67,584	input_layer_28[0…
lstm_20 (LSTM)	(None, 120, 64)	49,408	lstm_19[0][0]
multi_head_attenti… (MultiHeadAttentio…	(None, 120, 64)	16,640	lstm_20[0][0], lstm_20[0][0]
add_13 (Add)	(None, 120, 64)	0	lstm_20[0][0], multi_head_atten…
layer_normalizatio… (LayerNormalizatio…	(None, 120, 64)	128	add_13[0][0]
get_item_6 (GetItem)	(None, 64)	0	layer_normalizat…
dense_47 (Dense)	(None, 64)	4,160	get_item_6[0][0]
dense_48 (Dense)	(None, 3)	195	dense_47[0][0]
reshape_27 (Reshape)	(None, 1, 3)	0	dense_48[0][0]

Gambar 5. Arsitektur Attention-LSTM

Pelatihan Model

Seluruh model dilatih menggunakan konfigurasi pelatihan yang sama untuk memastikan proses komparasi dilakukan secara konsisten. Konfigurasi pelatihan yang digunakan ditunjukkan pada Tabel 2.

Tabel 2. Konfigurasi Pelatihan Model

Parameter	**Nilai**
Optimizer	Adam
Learning Rate Awal	0,0005
Loss Function	Mean Absolute Error (MAE)
Maksimum Epoch	40
Batch Size	64
EarlyStopping	Patience = 5, Restore Best Weights = True
ReduceLROnPlateau	Factor = 0,5, Patience = 3
ModelCheckpoint	Menyimpan model dengan validation loss terbaik

Untuk meningkatkan stabilitas pelatihan, digunakan callback EarlyStopping, ReduceLROnPlateau, dan ModelCheckpoint. EarlyStopping digunakan untuk menghentikan proses pelatihan secara otomatis ketika performa validasi tidak lagi mengalami peningkatan. ReduceLROnPlateau digunakan untuk menyesuaikan learning rate secara adaptif ketika proses

pembelajaran mengalami stagnasi, sedangkan ModelCheckpoint digunakan untuk menyimpan bobot model terbaik berdasarkan nilai validation loss terendah yang diperoleh selama pelatihan.

Evaluasi

Evaluasi performa model dilakukan menggunakan tiga metrik, yaitu Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan Coefficient of Determination (R^2). Ketiga metrik tersebut digunakan untuk mengevaluasi tingkat kesalahan prediksi serta kemampuan model dalam merepresentasikan pola data Quality of Service (QoS) jaringan.

Mean Absolute Error (MAE) digunakan untuk mengukur rata-rata selisih absolut antara nilai aktual dan nilai prediksi. MAE dihitung menggunakan Persamaan (1).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Root Mean Squared Error (RMSE) digunakan untuk mengukur besarnya kesalahan prediksi dengan memberikan penalti yang lebih besar terhadap kesalahan bernilai tinggi. RMSE dihitung menggunakan Persamaan ini.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Coefficient of Determination (R^2) digunakan untuk mengukur seberapa baik model mampu menjelaskan variasi data aktual. Nilai R^2 dihitung menggunakan Persamaan.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}i)^2}{\sum i = 1^n\,(y_i - \bar{y})^2}$$

Pada persamaan di atas, (y_i) menyatakan nilai aktual, $(\hat{y}_i)$menyatakan nilai hasil prediksi model, $(\bar{y})$menyatakan rata-rata nilai aktual, dan (n) menyatakan jumlah sampel pengujian. Nilai MAE dan RMSE yang lebih kecil menunjukkan tingkat kesalahan prediksi yang lebih rendah, sedangkan nilai R^2 yang semakin mendekati 1 menunjukkan kemampuan model yang semakin baik dalam merepresentasikan pola data QoS jaringan.

4. HASIL DAN PEMBAHASAN

Karakteristik Dataset QoS

Data *Quality of Service* (QoS) yang digunakan dalam penelitian ini merupakan hasil ekstraksi dari simulasi jaringan *client-server* menggunakan GNS3. Proses pengambilan sampel dilakukan dengan interval satu detik selama 26, 5 jam, menghasilkan total observasi sebanyak 95611 sampel deret waktu. Data ini merekam dinamika trafik jaringan yang dibangkitkan dari

empat skenario secara acak, yaitu *web browsing*, *streaming*, *download* skala besar, dan *backup* data.

Karakteristik statistik deskriptif dari ketiga metrik QoS yang direkam (*Traffic In*, *Traffic Out*, dan *Latency*) dirangkum pada Tabel 3.

Tabel 3. Statistik Deskriptif Dataset QoS

Metrik	Satuan	Minimum	Maksimum	Rata-rata	Standar Deviasi
Traffic In	bps	0	3.198.259.000.000	66.002.050	109.224.400.000
Traffic Out	bps	0	2.336.424.000.000	48.253.370	79.807.870.000
Latency	ms	14,60	684,00	95,92	35,27

Berdasarkan Tabel 3, parameter traffic_in_bps memiliki nilai rata-rata sebesar 66.002.050 bps dengan nilai maksimum mencapai 3.198.259.000.000 bps. Sementara itu, traffic_out_bps memiliki rata-rata sebesar 48.253.370 bps dan nilai maksimum sebesar 2.336.424.000.000 bps. Rentang nilai yang sangat lebar pada kedua parameter tersebut menunjukkan bahwa beban trafik yang dihasilkan selama simulasi mengalami fluktuasi yang cukup tinggi, sesuai dengan karakteristik skenario jaringan yang melibatkan berbagai jenis aktivitas pengguna.

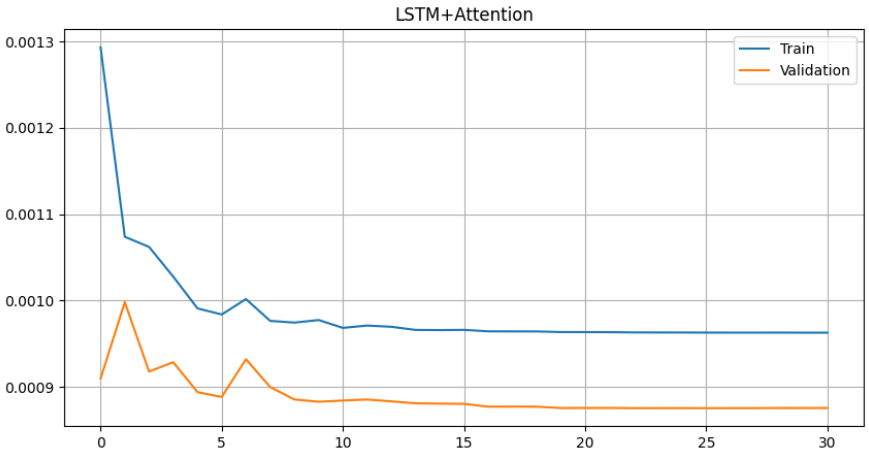
Parameter latency_ms memiliki nilai rata-rata sebesar 95,92 ms dengan standar deviasi sebesar 35,27 ms. Nilai minimum latency tercatat sebesar 14,60 ms, sedangkan nilai maksimum mencapai 684 ms. Variasi nilai latency ini menunjukkan adanya perubahan kondisi jaringan selama proses simulasi berlangsung, terutama ketika terjadi peningkatan intensitas trafik pada waktu-waktu tertentu.

Karakteristik dataset yang dinamis dan mengandung variasi temporal ini menjadikan data QoS yang digunakan relevan untuk dievaluasi menggunakan pendekatan deep learning berbasis deret waktu, karena model perlu mempelajari pola perubahan trafik dan latency yang terjadi secara berkelanjutan.

Dataset yang digunakan dalam penelitian ini tersedia secara publik melalui Kaggle (Tarigan, 2026).

Hasil Pelatihan Model

Pelatihan model dilakukan menggunakan optimizer Adam dengan learning rate awal sebesar 0,0005 dan fungsi loss Mean Absolute Error (MAE). Untuk menjaga stabilitas proses pembelajaran, diterapkan mekanisme EarlyStopping, ReduceLROnPlateau, dan ModelCheckpoint sebagaimana dijelaskan pada bagian metodologi.



Gambar 6. *Learning Curve* model Attention-LSTM

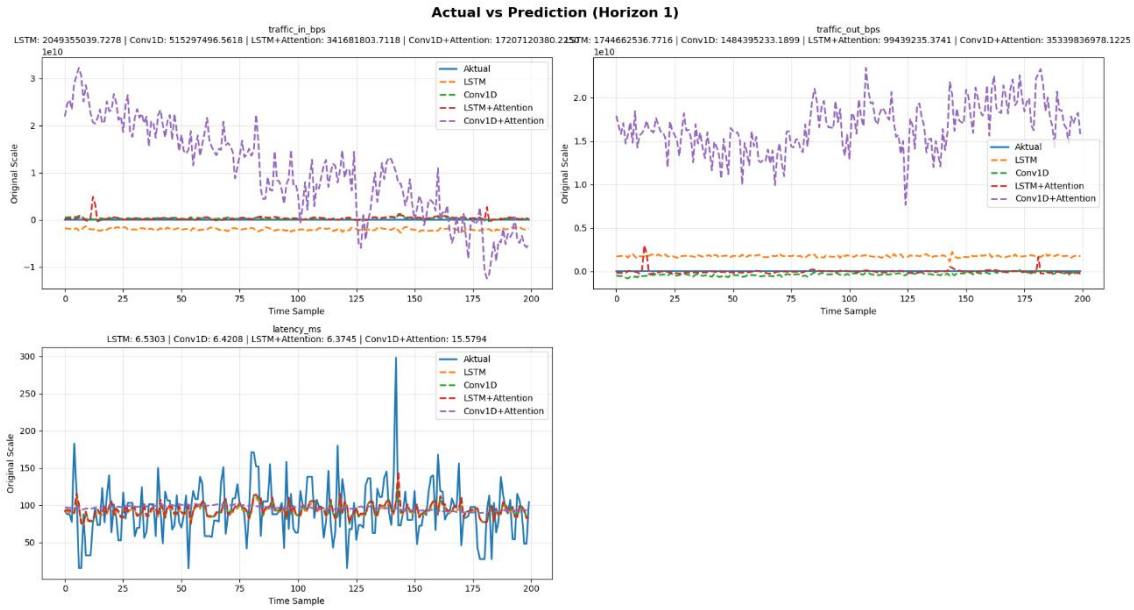
Gambar 6 menunjukkan kurva training loss dan validation loss dari model LSTM+Attention yang memperoleh performa terbaik pada tahap evaluasi. Terlihat bahwa nilai training loss mengalami penurunan yang cukup signifikan pada beberapa epoch awal sebelum berangsur stabil pada epoch berikutnya. Pola yang serupa juga ditunjukkan oleh validation loss yang terus menurun hingga mencapai kondisi konvergen.

Selain menunjukkan tren penurunan yang konsisten, jarak antara kurva training loss dan validation loss relatif kecil selama proses pelatihan berlangsung. Hal ini mengindikasikan bahwa model mampu mempelajari pola data pelatihan dengan baik tanpa menunjukkan perbedaan performa yang mencolok pada data validasi. Dengan demikian, proses pelatihan dapat dikatakan berlangsung secara stabil pada lingkungan eksperimen yang digunakan.

Berdasarkan hasil tersebut, model LSTM+Attention berhasil mencapai konvergensi dengan baik dan selanjutnya digunakan sebagai salah satu kandidat utama pada tahap evaluasi performa prediksi QoS jaringan.

Hasil Prediksi

Performa prediksi dari keempat model dievaluasi secara visual dengan membandingkan nilai aktual dan nilai hasil prediksi pada data pengujian. Gambar 7 menampilkan hasil prediksi untuk tiga parameter Quality of Service (QoS), yaitu *traffic_in_bps*, *traffic_out_bps*, dan *latency_ms*.



Gambar 7. Perbandingan nilai aktual dan hasil prediksi keempat model

Pada parameter *traffic_in_bps*, model Conv1D, LSTM, dan LSTM+Attention mampu mengikuti pola aktual dengan cukup baik. Sebaliknya, model Conv1D+Attention menunjukkan deviasi prediksi yang jauh lebih besar dibandingkan model lainnya, terutama pada beberapa rentang waktu tertentu. Kondisi ini menunjukkan bahwa penambahan mekanisme attention pada arsitektur Conv1D tidak memberikan peningkatan kemampuan prediksi pada dataset yang digunakan.

Pola serupa juga terlihat pada parameter *traffic_out_bps*. Model Conv1D dan LSTM+Attention menghasilkan prediksi yang relatif dekat dengan data aktual, sedangkan model Conv1D+Attention menghasilkan penyimpangan yang lebih besar. Hal ini mengindikasikan bahwa kombinasi Conv1D dan attention pada penelitian ini belum mampu merepresentasikan pola trafik keluar secara optimal.

Pada parameter *latency_ms*, seluruh model mampu menangkap kecenderungan nilai rata-rata latency yang berada pada rentang sekitar 80–100 ms. Namun demikian, data aktual menunjukkan fluktuasi yang cukup tinggi dengan beberapa lonjakan (spike) yang muncul secara tiba-tiba. Lonjakan-lonjakan tersebut belum dapat direpresentasikan secara akurat oleh seluruh model yang diuji.

Secara keseluruhan, hasil visualisasi menunjukkan bahwa model LSTM+Attention dan Conv1D mampu menghasilkan prediksi yang paling mendekati data aktual pada sebagian besar pengamatan. Sementara itu, model Conv1D+Attention memperlihatkan deviasi yang lebih besar, khususnya pada parameter trafik jaringan. Temuan ini mengindikasikan bahwa

penambahan mekanisme attention tidak selalu menghasilkan peningkatan performa prediksi dan efektivitasnya sangat bergantung pada arsitektur dasar yang digunakan.

Komparasi Performa Model

Performa keempat model dievaluasi menggunakan tiga metrik, yaitu Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan koefisien determinasi (R^2). Hasil evaluasi disajikan pada Tabel 4.

Tabel 4. Hasil Evaluasi ke-4 Model

Model	MAE	RMSE	R^2
LSTM+Attention	0.003224	0.013578	0.943872
Conv1D	0.003463	0.013594	0.94374
LSTM	0.003714	0.013645	0.943322
Conv1D+Attention	0.014593	0.019362	0.885869

Berdasarkan Tabel 4, model LSTM+Attention memperoleh performa terbaik dengan nilai MAE sebesar 0,003224 dan RMSE sebesar 0,013578. Selain itu, model ini juga menghasilkan nilai R^2 tertinggi, yaitu 0,943872. Hasil tersebut menunjukkan bahwa kombinasi LSTM dan mekanisme Multi-Head Attention mampu menghasilkan tingkat kesalahan prediksi yang paling rendah pada dataset QoS yang digunakan.

Model Conv1D menempati posisi kedua dengan nilai MAE sebesar 0,003463, RMSE sebesar 0,013594, dan R^2 sebesar 0,943740. Meskipun menggunakan arsitektur yang lebih sederhana dibandingkan model berbasis attention, Conv1D mampu menghasilkan performa yang sangat kompetitif dengan selisih yang relatif kecil terhadap model LSTM+Attention.

Model LSTM memperoleh nilai MAE sebesar 0,003714 dan RMSE sebesar 0,013645 dengan nilai R^2 sebesar 0,943322. Hasil ini menunjukkan bahwa kemampuan LSTM dalam memodelkan dependensi temporal tetap memberikan performa yang baik pada data QoS, meskipun sedikit berada di bawah Conv1D dan LSTM+Attention.

Sebaliknya, model Conv1D+Attention menghasilkan nilai MAE dan RMSE tertinggi dibandingkan model lainnya. Temuan ini konsisten dengan hasil visualisasi pada Gambar 9 yang menunjukkan deviasi prediksi lebih besar pada beberapa parameter QoS. Oleh karena itu, diperlukan analisis lebih lanjut untuk memahami pengaruh mekanisme attention terhadap masing-masing arsitektur yang digunakan.

Analisis Pengaruh Attention

Untuk menganalisis pengaruh mekanisme attention terhadap performa prediksi, dilakukan perbandingan antara model dasar dan model yang telah ditambahkan lapisan Multi-Head Attention. Analisis difokuskan pada perubahan nilai Mean Absolute Error (MAE) karena

metrik ini secara langsung merepresentasikan rata-rata kesalahan prediksi yang dihasilkan oleh model.

Berdasarkan Tabel 4, penambahan mekanisme attention memberikan dampak yang berbeda pada masing-masing arsitektur. Pada model LSTM, nilai MAE menurun dari 0,003714 menjadi 0,003224. Penurunan tersebut setara dengan peningkatan performa sebesar 13,2%. Hasil ini menunjukkan bahwa mekanisme attention mampu membantu LSTM dalam memfokuskan perhatian pada informasi historis yang lebih relevan sehingga menghasilkan prediksi yang lebih akurat.

Sebaliknya, pada arsitektur Conv1D, penambahan attention justru meningkatkan nilai MAE dari 0,003463 menjadi 0,014593. Kenaikan tersebut menunjukkan penurunan performa yang cukup signifikan dibandingkan model Conv1D dasar. Temuan ini mengindikasikan bahwa mekanisme attention tidak selalu memberikan manfaat ketika dikombinasikan dengan arsitektur berbasis konvolusi pada dataset QoS yang digunakan.

Perbedaan hasil tersebut kemungkinan disebabkan oleh karakteristik masing-masing arsitektur. LSTM dirancang untuk memodelkan hubungan temporal secara sekuensial sehingga informasi yang dihasilkan oleh lapisan LSTM dapat dimanfaatkan secara lebih efektif oleh mekanisme attention. Sebaliknya, Conv1D telah mengekstraksi pola lokal melalui operasi konvolusi sehingga penambahan attention tidak selalu menghasilkan representasi fitur yang lebih baik dan pada kondisi tertentu dapat menambah kompleksitas model tanpa memberikan peningkatan akurasi yang sepadan.

Hasil analisis ini menunjukkan bahwa efektivitas mekanisme attention sangat dipengaruhi oleh arsitektur dasar yang digunakan. Pada lingkungan eksperimen penelitian ini, attention terbukti memberikan manfaat pada model LSTM, namun tidak menunjukkan peningkatan performa ketika diterapkan pada model Conv1D.

Hasil penelitian ini menunjukkan bahwa penambahan mekanisme attention dapat meningkatkan performa pada arsitektur LSTM. Temuan ini konsisten dengan beberapa penelitian sebelumnya yang melaporkan bahwa attention membantu model dalam memfokuskan informasi temporal yang relevan pada tugas prediksi deret waktu. Namun demikian, hasil penelitian ini juga menunjukkan bahwa attention tidak selalu menghasilkan peningkatan performa pada seluruh arsitektur yang diuji.

5. KESIMPULAN DAN SARAN

Penelitian ini telah melakukan komparasi terhadap empat arsitektur deep learning, yaitu Conv1D, LSTM, Conv1D+Attention, dan LSTM+Attention untuk prediksi Quality of Service

(QoS) jaringan menggunakan data deret waktu yang diperoleh dari simulasi jaringan berbasis GNS3. Evaluasi dilakukan menggunakan metrik Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan koefisien determinasi (R^2).

Berdasarkan hasil pengujian, model LSTM+Attention memperoleh performa terbaik dengan nilai MAE sebesar 0,003224, RMSE sebesar 0,013578, dan R^2 sebesar 0,943872. Model Conv1D berada pada posisi kedua dengan performa yang sangat dekat terhadap LSTM+Attention, diikuti oleh model LSTM. Sementara itu, model Conv1D+Attention menghasilkan performa terendah pada lingkungan eksperimen yang digunakan.

Hasil penelitian juga menunjukkan bahwa pengaruh mekanisme attention berbeda pada setiap arsitektur. Pada model LSTM, penambahan Multi-Head Attention berhasil meningkatkan performa prediksi dengan menurunkan nilai MAE dari 0,003714 menjadi 0,003224. Sebaliknya, pada model Conv1D, penambahan attention tidak menghasilkan peningkatan performa dan justru meningkatkan nilai MAE menjadi 0,014593. Temuan ini menunjukkan bahwa efektivitas mekanisme attention tidak hanya dipengaruhi oleh karakteristik data, tetapi juga oleh arsitektur dasar yang digunakan.

Selain itu, hasil visualisasi prediksi menunjukkan bahwa seluruh model masih mengalami kesulitan dalam merepresentasikan lonjakan (*spike*) latency yang terjadi secara tiba-tiba. Kondisi ini mengindikasikan bahwa prediksi terhadap perubahan QoS yang bersifat ekstrem masih menjadi tantangan pada dataset yang digunakan dalam penelitian ini. E

Untuk penelitian selanjutnya, pengujian dapat diperluas menggunakan dataset yang diperoleh dari lingkungan jaringan nyata maupun skenario trafik yang lebih beragam. Selain itu, penelitian lanjutan dapat mengeksplorasi arsitektur lain seperti Transformer murni, Temporal Convolutional Network (TCN), atau model hibrida yang menggabungkan beberapa mekanisme pemodelan temporal untuk meningkatkan kemampuan prediksi pada kondisi jaringan yang lebih kompleks.

DAFTAR REFERENSI

Abbasimehr, H., & Paki, R. (2022). Improving time series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing*, *13*(1), 673–691. https://doi.org/10.1007/s12652-020-02761-x

Awaliyani, I., & Aziz, R. A. (n.d.). *Analisis Quality Of Service (QoS) Jaringan Internet Pelanggan Pada ISP Jalurdata.Net Menggunakan Metode Hierarchical Token Bucket (HTB) dan Per Connection Queue (PCQ)*. https://doi.org/10.33364/algoritma/v.21-2.1792

Bai, S., Kolter, J. Z., & Koltun, V. (2018). *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. http://arxiv.org/abs/1803.01271

Hameed, A., Violos, J., Leivadeas, A., Santi, N., Grunblatt, R., & Mitton, N. (2022). Toward QoS Prediction Based on Temporal Transformers for IoT Applications. *IEEE Transactions on Network and Service Management*, *19*(4), 4010–4027. https://doi.org/10.1109/TNSM.2022.3217170

Na, H., Shin, Y., Lee, D., & Lee, J. (2023). LSTM-based throughput prediction for LTE networks. *ICT Express*, *9*(2), 247–252. https://doi.org/10.1016/j.icte.2021.12.001

Saha, S., Das, S., & Carvalho, G. H. S. (2024). ConvLSTMTransNet: A Hybrid Deep Learning Approach for Internet Traffic Telemetry. *2024 IEEE Virtual Conference on Communications (VCC)*, 1–6. https://doi.org/10.1109/VCC63113.2024.10914473

Tambunan, A. A. (2020). Analisis Perbandingan Quality Of Service (Qos) Pada Performa Bandwidth Jaringan Dengan Metode Hierarchical Token Bucket (Htb) Dan Per Connection Queque (Pcq). *Jurnal Teknologi Informasi*, *3*.

Tarigan, Y. H. (2026). *Network Quality Dataset*. https://doi.org/https://doi.org/10.34740/KAGGLE/DSV/16739775

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (n.d.). *Attention Is All You Need*.

Yuliana, H., Iskandar, & Hendrawan. (2026). A Hybrid LSTM-Transformer Approach for Accurate Time Series Traffic Prediction in Cellular Networks. *IEEE Access*, *14*, 35047–35073. https://doi.org/10.1109/ACCESS.2026.3669215