BAB IV
HASIL DAN PEMBAHSAN

4.1 Hasil Penelitian

4.1.1 Data Penelitian

Penelitian ini menggunakan data kinerja jaringan telekomunikasi yang diperoleh dari Telkomsel. Data mencakup metrik kualitas layanan jaringan (QoS), yaitu *Packet Loss Ratio* (PLR), *Jitter* (Ms), *Delay* (Ms). Data dikumpulkan dari berbagai perangkat jaringan yang beroprasi di wilayah kota Bandar Lampung dengan granulasi per jam (*hourly*).

Setelah mempertimbangkan keterbatasan komputasi dan hasil yang optimal, data yang digunakan adalah data tahun 2024 dari bulan Maret hingga Desember. Periode ini mencakup 10 bulan, yaitu mulai dari 1 Maret 2024 pukul 00:00:00 hingga 1 Januari 2025 pukul 23:00:00, dengan total sebanyak 2.264.250 entri data.

Pemilihan periode tersebut didasarkan pada pertimbangan bahwa dalam rentang waktu tersebut terdapat variasi yang lebih signifikan pada metrik-metrik QoS, terutama PLR, sehingga diharapkan mampu memberikan informasi yang lebih kaya untuk proses pembelajaran model terhadap pola-pola kompleks dalam data.

Dalam penelitian ini, terdapat dua jenis variabel yang digunakan, yaitu variabel dependen dan variabel independen (fitur). Variabel dependen dalam penelitian ini adalah kejadian lonjakan PLR, yang dikonversi menjadi bentuk variabel biner. Proses konversi dilakukan berdasarkan nilai ambang (*threshold*) PLR ≥ 0.005, dengan ketentuan:

- Nilai 0 menunjukkan kondisi jaringan normal.
- Nilai 1 menunjukkan terjadinya lonjakan PLR.

Adapun variabel independen (fitur) yang digunakan meliputi:

- Metriks QoS utama: PLR, *Jitter*, dan *Delay*.
- Fitur hasil dari rekayasa fitur (*feature engineering)*:
    a. Nilai *lagged* dari masing masing QoS.
    b. Statistik bergerak (*rolling mean*/*rolling std*) untuk mengamati tren jangka pendek.
    c. Fitur temporal seperti jam (*HOUR*) dan hari dalam seminggu (*DayOfWeek*)

Penggunaan fitur-fitur tersebut bertujuan untuk meningkatkan akurasi prediksi model dengan menangkap dinamika temporal serta pola-pola historis yang relevan terhadap kejadian lonjakan PLR.

4.1.2 Hasil *Pre-Processing Data*

Tahap ini dilakukan untuk memastikan kualitas data dan kesiapannya untuk pemodelan LSTM.

4.1.2.1 Data Cleaning (Pembersihan Data)

Tahap ini dilakukan untuk memastikan kualitas dan kesiapan data sebelum digunakan dalam proses pemodelan dengan metode LSTM. Fokus utama pada tahap ini adalah identifikasi serta penanganan terhadap *missing values* dan *outlier* pada metrik kualitas layanan jaringan (QoS).

a. Penanganan *Missing Value*:

Ditemukan sejumlah missing values pada kolom Packet Loss Ratio (%), Delay (ms), dan Jitter (ms). Untuk mengatasi hal ini, dilakukan beberapa metode penanganan, yaitur:

- *Interpolasi Linier* untuk mengisi nilai yang hilang berdasarkan nilai sebelum dan sesudahnya.

- *Forward Fill dan Backward Fill* untuk mengisi nilai kosong yang berada di awal atau akhir deret waktu.

Setelah penerapan metode-metode tersebut, seluruh *missing values* berhasil diatasi. Visualisasi jumlah *missing values* sebelum dan sesudah penanganan ditampilkan pada Tabel 4.1.

Tabel 4. 1. Jumlah *Missing Values* sebelum dan sesudah penanganan.

Kolom	Sebelum Penanganan	Setelah Penanganan
Packet Loss Ratio (%)	18.961	0
Delay (Ms)	18.961	0
Jitter (Ms)	18.961	0
Target Device Name	0	0

b. Penanganan *Outlier*:

Sebelum dilakukan penanganan, proses deteksi outlier dilakukan menggunakan metode Interquartile Range (IQR). Jumlah outlier yang terdeteksi dapat dilihat pada Tabel 4.1.

Tabel 4. 2. Deteksi dan Penanganan *Outlier* pada Fitur Numerik.

No.	Fitur	Batas Bawah *Outlier*	Batas Atas *Outlier*	Jumlah *Outlier*	Metode Penanganan
1.	*PLR (%)*	-7.5704	15.2841	5.189	*Winsorization*
2.	*Delay (*Ms*)*	-0.4256	0.6927	45.508	*Winsorization*
3.	*Jitter* (Ms)	-0.0042	0.0069	396.087	*Winsorization*

Dari Tabel 4.2. terlihat bahwa batas outlier ditentukan menggunakan metode IQR. Penanganan dilakukan menggunakan teknik *Winsorization*, yaitu mengubah nilai ekstrem menjadi nilai batas bawah atau batas atas yang diperoleh dari distribusi data.

Untuk fitur Packet Loss Ratio (%), dilakukan penanganan tambahan menggunakan *Clipping*, agar nilai tetap berada dalam rentang valid (0 hingga 1), sesuai dengan domain data tersebu.

4.1.2.2 Analisis Data Eksplorasi (EDA)

Tahap ini dilakukan untuk memahami karakteristik, distribusi, pola, dan hubungan antar variabel dalam data. Analisis Data Explorasi (EDA) sangat membantu dalam memberikan wawasan awal sebelum dilakukan proses rekayasa fitur (*feature engineering*) dan pelatihan model.

a. Ringkasan Statistik

Berikut ini merupakan ringkasan statistik deskriptif dari data penelitian setelah melewati tahap *pre-processing*, namun sebelum dilakukan rekayasa fitur lebih lanjut. Angka-angka yang semula disajikan dalam notasi ilmiah (misalnya, e+06 berarti dikalikan 10^6, dan e-03 berarti dikalikan 10^{-3}) telah dikonversi ke dalam format desimal untuk mempermudah pemahaman dan interpretasi.

Berdasarkan Tabel 4.3, dapat diketahui bahwa dataset terdiri dari lebih dari 2,2 juta entri, menunjukkan volume data yang besar. Rata-rata nilai *Packet Loss Ratio* (PLR) sangat rendah, yaitu sekitar 0.0016%, yang mengindikasikan bahwa lonjakan PLR merupakan kejadian yang sangat jarang terjadi. Hal ini menjadi indikasi kuat bahwa permasalahan yang diangkat tergolong sebagai masalah klasifikasi data tidak seimbang (*imbalanced classification*)

Tabel 4. 3. Ringkasan Statistik Data Penelitian.

Metrik	*Packet Loss Ratio* (%)	*Delay* (Ms)	*Jitter* (Ms)
Count	2,264,250	2,264,250	2,264,250
Mean	0.001567	4.3610	0.1306
Std	0.002727	2.9246	0.1693
Min	0.0000	0.0000	0.0000
25%	0.0000	1.0000	0.0000
50%	0.0000	6.0273	0.0193
75%	0.002778	6.7136	0.2771
Max	0.00694	15.2841	0.6927

Sementara itu, metrik *Delay* (Ms) memiliki rata-rata sekitar 4.36 ms dengan standar deviasi (simpangan baku) sebesar 2.92 ms, yang menunjukkan adanya variasi waktu tunda yang cukup signifikan dalam jaringan. Selain itu, meskipun *Jitter* (Ms) memiliki nilai minimum 0.0 tetap menujukkan variasi yang signifikan dengan standar deviasi 0.169. Oleh karena itu, nilai *Jitter* (Ms) akan bermanfataan untuk pemodelan.

b. Analisis Distribusi *Packet Loss Ratio*

Setelah dilakukan penanganan missing values dan outlier, kolom *Packet Loss Ratio* (%) tidak lagi mengandung nilai NaN, dengan tipe data float64. Distribusi nilai *Packet Loss Ratio* ditampilkan pada Tabel 4.4.

Berdasarkan Tabel 4.4. terdapat sebanyak 2.949 nilai unik pada kolom *Packet Loss Ratio*. Dari jumlah tersebut, nilai 0.000000 menjadi nilai yang paling sering muncul, dengan frekuensi 1.659.344 entri data. Hal ini menunjukkan bahwa sebagian besar waktu jaringan berada dalam kondisi ideal, yaitu tidak terjadi kehilangan paket (*loss*).

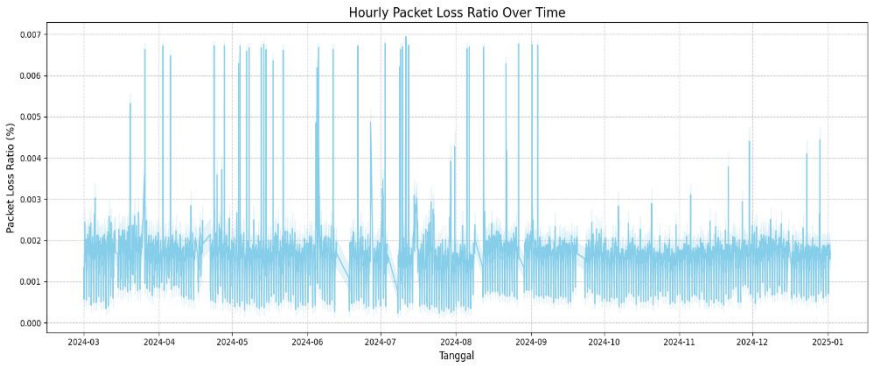
Tabel 4. 4. 5 Nilai Unik PLR dengan Frekuensi Tertinggi.

No.	Nilai *Packet Loss Ratio* (%)	Frekuensi
1.	0.000000	1.650.344
2.	0.006944	396.026
3.	0.002778	129.240
4.	0.005556	61.482
5.	0.002778	10.080

Namun, temuan ini juga kembali menegaskan bahwa data bersifat tidak seimbang, karena sebagian besar nilai berada pada kelas nol. Meski begitu, dengan nilai standar deviasi sebesar 0.0027, dapat disimpulkan bahwa terdapat variabilitas pada data, dan kejadian lonjakan *Packet Loss Ratio* memang terjadi, meskipun relatif jarang.

c. Visualisasi Data Deret Waktu dan Distribusi

1. Plot Deret Waktu *Packet Loss Ratio* per Jam.



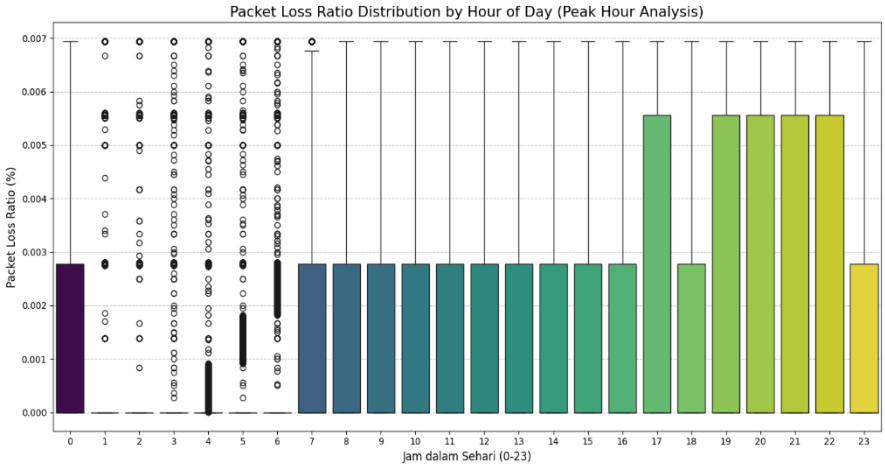
Gambar 4. 1 Plot Deret Waktu PLR per Jam Selama 10 Bulan

Gambar 4.1. menunjukkan visualisasi deret waktu *Packet Loss Ratio* (PLR) per jam selama periode 10 bulan. Visualisasi ini memperlihatkan bahwa sebagian besar waktu, nilai PLR berada

pada tingkat rendah atau nol. Namun, terdapat beberapa lonjakan (*spikes*) signifikan pada titik-titik waktu tertentu.

Pola ini secara visual mengonfirmasi bahwa data memiliki karakteristik tidak seimbang (*imbalanced*), di mana kondisi normal (PLR = 0) jauh lebih sering terjadi dibandingkan dengan lonjakan. Meskipun frekuensi lonjakan PLR rendah, keberadaannya penting untuk diidentifikasi karena dapat memberikan dampak signifikan terhadap kualitas layanan jaringan (QoS).

2. Distribusi *Packet Loss Ratio* Berdasarkan Jam dalam Sehari.



Gambar 4. 2 Distribusi PLR Berdasarkan Jam dalam Sehari

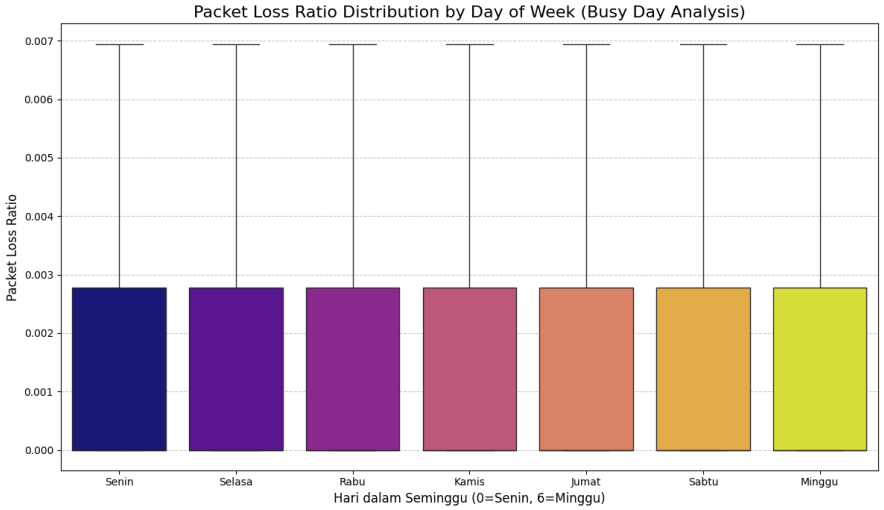
Gambar 4.2. menunjukkan boxplot distribusi *Packet Loss Ratio* (PLR) untuk setiap jam dalam sehari. Secara umum, median PLR cenderung rendah hampir di seluruh jam, menandakan stabilitas jaringan pada sebagian besar waktu.

Namun, terdapat peningkatan median dan rentang interkuartil (IQR) pada rentang waktu antara pukul 17:00 hingga 22:00, yang mengindikasikan adanya potensi jam sibuk (*peak hours*). Pada

periode ini, lonjakan PLR cenderung lebih sering terjadi dibandingkan jam lainnya.

Selain itu, tampak beberapa *outlier* (titik di luar *whiskers* pada boxplot) yang mencerminkan adanya kejadian lonjakan PLR yang ekstrem dan tidak mengikuti pola umum. Temuan ini penting untuk proses prediksi, karena menunjukkan waktu-waktu tertentu yang perlu mendapat perhatian lebih dalam manajemen jaringan.

3. Distribusi *Packet Loss Ratio* Berdasarkan Hari dalam Seminggu.

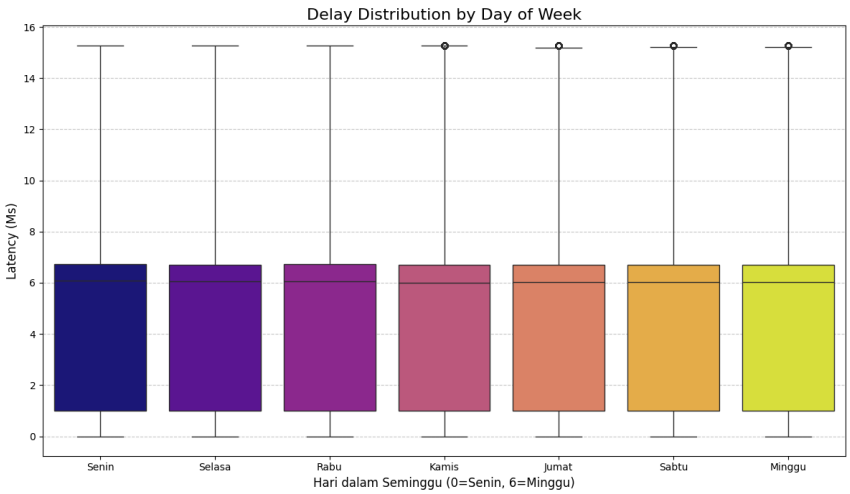


Gambar 4. 3 Boxplot Distribusi PLR per Hari dalam Seminggu.

Pada Gambar 4.3 menyajikan boxplot dari distribusi PLR berdasarkan hari dalam seminggu. Dari visualisasi tersebut, terlihat bahwa nilai median PLR relatif stabil di setiap hari, berkisar di sekitar 0.0028, dengan nilai maksimum mendekati 0.007 pada semua hari. Hal ini mengindikasikan bahwa tidak terdapat lonjakan signifikan pada hari tertentu yang secara konsisten menunjukkan peningkatan PLR.

Dengan demikian, beban jaringan cenderung merata sepanjang minggu tanpa hari dominan sebagai penyumbang utama PLR. Analisis ini penting untuk mendukung manajemen jaringan dalam menjadwalkan pemeliharaan atau peningkatan sistem tanpa mengganggu performa jaringan secara signifikan.

4. Distribusi *Delay* Berdasarkan Hari dalam Seminggu.



Gambar 4. 4 Boxplot Distribusi *Delay* Berdasarkan Hari dalam Seminggu

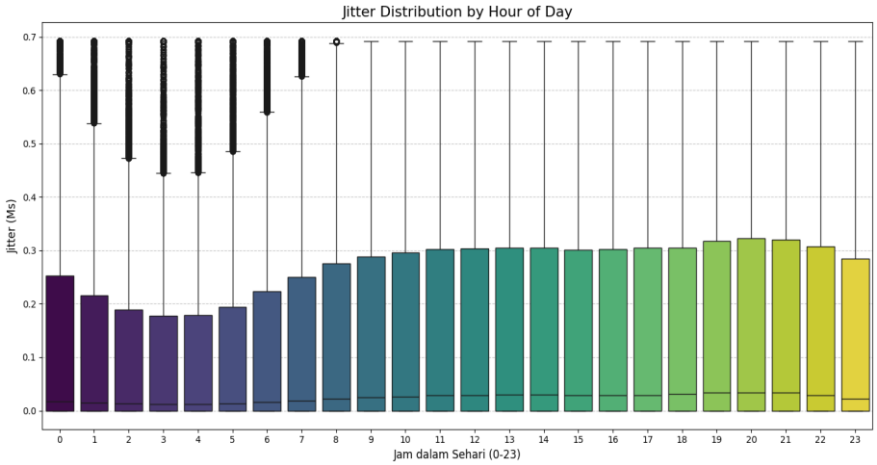
Berdasarkan Boxplot Gambar 4.5. menunjukkan distribusi *Delay* (Ms) berdasarkan hari dalam seminggu. Dari visualisasi boxplot tersebut, dapat diamati bahwa nilai median *Delay* berada di kisaran 6 ms untuk seluruh hari, mulai dari Senin hingga Minggu.

Rentang interkuartil (IQR) terlihat relatif konsisten, menunjukkan distribusi data yang stabil antarhari. Selain itu, beberapa outlier muncul di atas nilai 15 ms, tetapi tidak secara dominan pada hari tertentu. Hal ini menandakan bahwa lonjakan *Delay* bersifat sporadis dan tidak terpusat pada satu hari spesifik.

Secara keseluruhan, tidak terdapat indikasi adanya hari dengan beban jaringan signifikan yang menyebabkan peningkatan *Delay*. Dengan demikian, dapat disimpulkan bahwa performa jaringan dalam hal *Delay* cenderung merata sepanjang minggu, mendukung kesimpulan bahwa tidak terdapat *busy day* yang jelas terlihat berdasarkan metrik ini.

5. Distribusi *Jitter* Berdasarkan Jam dalam Sehari.

Gambar 4.5. menampilkan visualisasi boxplot distribusi jitter (dalam milidetik) berdasarkan jam dalam sehari (0–23). Terlihat bahwa nilai jitter cenderung lebih rendah dan stabil pada jam-jam malam hingga dini hari (sekitar pukul 0–6), dengan median jitter di bawah 0.2 ms. Namun, pada jam-jam tersebut juga ditemukan outlier yang cukup tinggi, menunjukkan adanya lonjakan jitter meskipun secara umum rendah.



Gambar 4. 5. Boxplot Distribusi *Jitter* Berdasarkan Jam dalam Sehari.

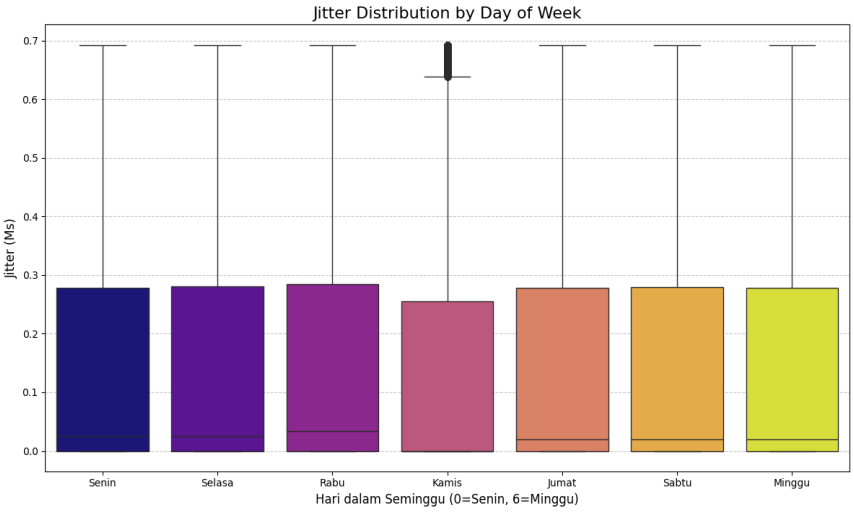
Mulai pukul 7 hingga 23, nilai median *Jitter* secara bertahap meningkat dan cenderung stabil pada kisaran yang lebih tinggi dibandingkan jam-jam sebelumnya. Hal ini mengindikasikan

bahwa *Jitter* cenderung lebih tinggi dan lebih stabil pada jam operasional atau saat aktivitas jaringan meningkat, yaitu dari pagi hingga malam hari, khususnya antara pukul 10:00 hingga 22:00.

Temuan ini memperkuat indikasi bahwa *Jitter* memiliki pola temporal harian yang signifikan dan dipengaruhi oleh aktivitas jaringan berdasarkan waktu dalam sehari. Pola ini dapat dimanfaatkan untuk pengembangan model prediktif dalam mengantisipasi performa jaringan pada jam-jam tertentu.

6. Distribusi *Jitter* Berdasarkan Jam dalam Seminggu.

Pada Gambar 4.6 menyajikan visualisasi boxplot distribusi *Jitter* (ms) berdasarkan hari dalam seminggu. Dari visualisasi tersebut terlihat bahwa distribusi *Jitter* cenderung stabil setiap harinya tanpa fluktuasi mencolok atau perbedaan signifikan antar hari.

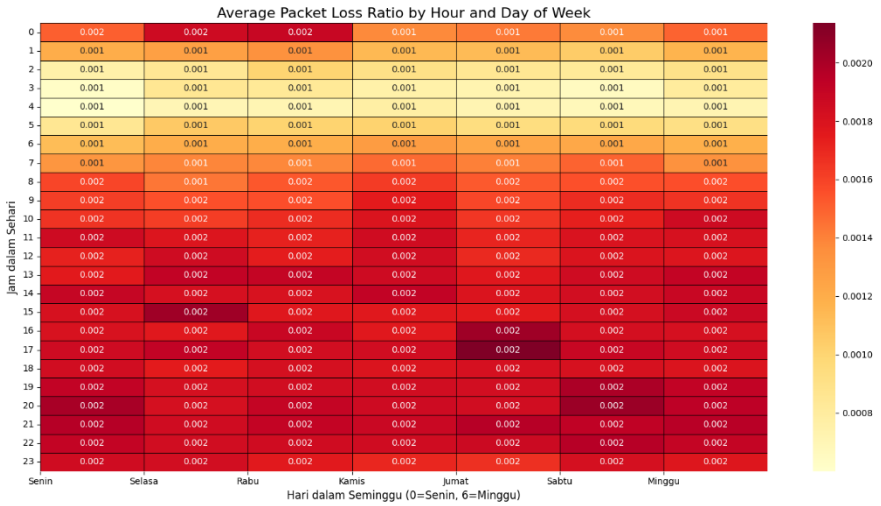


Gambar 4. 6. Boxplot Distribusi *Jitter* Berdasarkan Hari dalam Seminggu.

Temuan ini menunjukkan bahwa variabilitas *Jitter* tidak banyak dipengaruhi oleh faktor hari, melainkan lebih terkait

dengan pola waktu harian (*daily time pattern*). Dengan demikian, karakteristik temporal *Jitter* cenderung bersifat harian, yang dapat dimanfaatkan untuk meningkatkan akurasi dalam prediksi performa jaringan.

7. Rata – Rata *Packer Loss Ratio* Berdasarkan Jam dan Hari dalam Seminggu.



Gambar 4. 7. Heatmap Distribusi Rata-rata PLR Berdasarkan Jam dalam Seminggu.

Pada Gambar 4.7. menunjukkan visualisasi *heatmap* yang merepresentasikan nilai rata-rata *Packet Loss Ratio* (PLR), yang telah diagregasi berdasarkan kombinasi antara jam dalam sehari dan hari dalam seminggu. Pada visualisasi ini, semakin gelap warna yang ditampilkan (mendekati merah), maka semakin tinggi rata-rata nilai PLR. Sebaliknya, warna yang lebih terang mengindikasikan nilai PLR yang lebih rendah.

Dari visualisasi tersebut, dapat diamati bahwa nilai PLR cenderung meningkat pada rentang waktu sore hingga malam hari, yaitu sekitar pukul 17:00 hingga 23:00, dan pola ini muncul hampir di semua hari. Temuan ini menguatkan hasil sebelumnya

yang diperoleh dari visualisasi boxplot PLR per jam, yang secara konsisten menunjukkan adanya *peak hour* atau jam sibuk.

Kondisi ini penting untuk diperhatikan dalam konteks prediksi dan manajemen performa jaringan, karena lonjakan PLR yang terjadi pada jam-jam tersebut dapat berdampak pada penurunan kualitas layanan (QoS).

d. Analisis Autokorelasi (ACF & PACF)

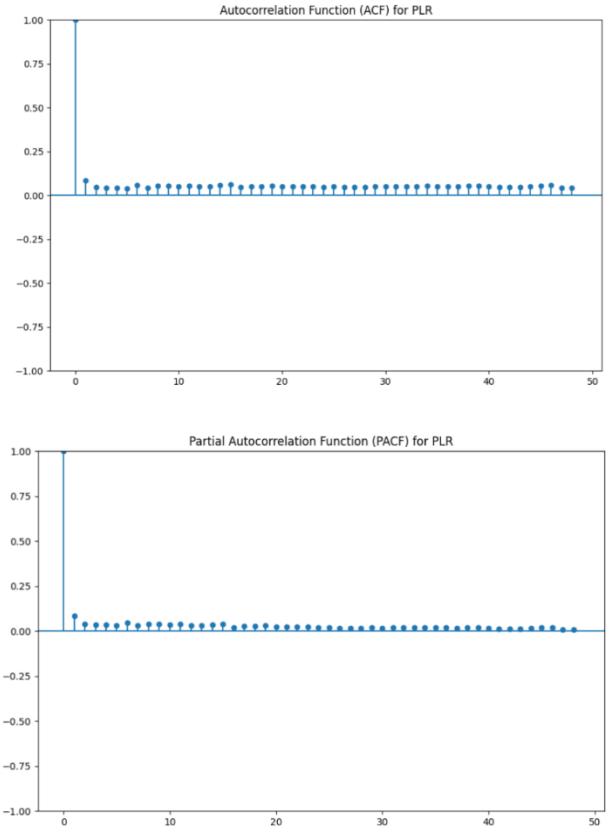
Analisis *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF) dilakukan terhadap data *Packet Loss Ratio* (PLR) untuk memahami pola ketergantungan temporal (*temporal dependency*) dalam data deret waktu. Analisis ini juga bertujuan untuk membantu dalam proses pemilihan fitur *lagged* yang relevan bagi pemodelan prediktif. Visualisasi ACF dan PACF ditampilkan pada Gambar 4.8.

Berdasarkan Gambar 4.8. plot ACF menunjukkan adanya korelasi yang signifikan pada beberapa *lag* awal. Nilai korelasi menurun secara bertahap namun tetap signifikan hingga lag yang relatif jauh, yang mengindikasikan adanya dependensi jangka panjang dalam deret waktu PLR.

Sementara itu, pada plot PACF tampak lonjakan (*spike*) yang signifikan pada lag ke-1, yang menunjukkan adanya korelasi kuat antara nilai PLR saat ini dengan nilai satu periode sebelumnya (PLR_lag1). Selain itu, terdapat lonjakan signifikan pada lag ke-24 dan lag ke-168, yang mengindikasikan adanya pola musiman harian (24 jam) dan mingguan (168 jam).

Setelah lag-lag tersebut, nilai PACF menurun drastis dan tidak menunjukkan signifikansi yang berarti. Temuan ini mendukung keberadaan komponen musiman serta tren jangka pendek maupun jangka

panjang dalam data PLR. Oleh karena itu, lag seperti PLR_lag1, PLR_lag24, dan PLR_lag168 relevan untuk digunakan dalam proses rekayasa fitur (*feature engineering*) pada model prediksi yang dikembangkan.

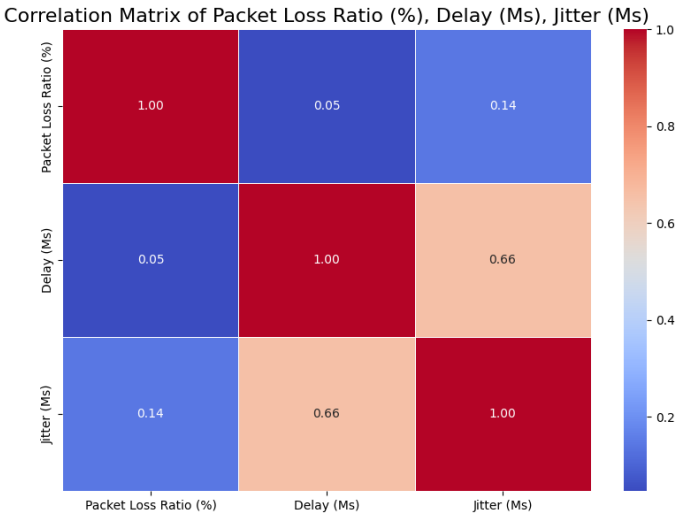


Gambar 4. 8. Visualisasi *Output* Fungsi Autokorelasi (ACF) dan Fungsi Autokorelasi Parsial (PAFC) untuk PLR.

e. Analisis Korelasi Antar Variabel

Korelasi ini dihitung untuk mengukur kekuatan dan arah hubungan linear antara masing-masing pasangan variabel. Gambar 4.11 menampilkan visualisasi heatmap korelasi antar metrik utama kualitas layanan (QoS), yaitu *Packet Loss Ratio* (PLR), *Delay*, dan *Jitter*.

Hasil korelasi menunjukkan bahwa hubungan linear antara PLR dan Delay memiliki nilai sebesar 0.05, yang tergolong sangat lemah. Selanjutnya, korelasi antara PLR dan *Jitter* juga rendah dengan nilai 0.14, menunjukkan bahwa keduanya tidak memiliki hubungan linear yang kuat. Namun, berbeda halnya dengan hubungan antara *Delay* dan *Jitter*, yang menunjukkan nilai korelasi sebesar 0.66 dan dapat dikategorikan sebagai korelasi linear yang cukup kuat.



Gambar 4. 9 Visualisasi *Output* Metrik Korelasi QoS.

Temuan ini menunjukkan bahwa meskipun PLR memiliki korelasi linear yang rendah dengan *Delay* maupun *Jitter*, hal tersebut tidak menutup kemungkinan adanya hubungan non-linear yang tidak dapat ditangkap oleh metrik korelasi sederhana. Oleh karena itu, untuk prediksi yang lebih akurat, fitur-fitur seperti nilai sebelumnya (*lagged values*) dan statistik bergulir (*rolling statistics*) dari *Delay* dan *Jitter* tetap digunakan dalam proses rekayasa fitur. Pendekatan ini sangat relevan untuk model deep learning seperti LSTM yang mampu menangkap pola kompleks dan hubungan non-linear dalam data deret waktu.

4.1.2.3 Rekayasa Fitur Deret Waktu

Pada tahap ini, dilakukan penambahan fitur-fitur baru untuk menangkap pola temporal serta karakteristik dinamis dari data. Fitur yang ditambahkan meliputi nilai lagged atau nilai historis dari tiga metrik utama kualitas layanan jaringan (QoS), yaitu *Packet Loss Ratio* (PLR), *Delay*, dan *Jitter*. Nilai *lagged* yang digunakan antara lain adalah *lag_1* (nilai satu jam sebelumnya), *lag_24* (nilai 24 jam sebelumnya), dan *lag_168* (nilai 168 jam sebelumnya).

Selain itu, ditambahkan fitur berupa statistik bergerak (rolling statistics) dalam bentuk rata-rata dan standar deviasi dari tiga jam terakhir untuk masing-masing metrik QoS. Fitur-fitur ini bertujuan untuk menangkap fluktuasi jangka pendek dan membantu model dalam mengenali perubahan dinamis dalam data.

Fitur temporal lainnya seperti *HOUR* dan *DayOfWeek* juga diekstraksi dari kolom Timestamp. Kedua fitur ini bersifat kategorikal numerik dan digunakan untuk merepresentasikan informasi waktu secara lebih detail. Penambahan fitur-fitur ini bertujuan untuk membantu model dalam memahami pola ketergantungan waktu, baik dalam skala harian maupun mingguan, sehingga dapat meningkatkan kemampuan prediksi terhadap nilai *Packet Loss Ratio* di masa mendatang.

4.1.2.4 Pembentukan Target Biner

Nilai *Packet Loss Ratio* (PLR) yang semula bersifat kontinu kemudian ditransformasikan menjadi variabel target biner dengan nama *is_spike*, menggunakan nilai ambang batas sebesar 0,005. Artinya, jika nilai PLR melebihi 0,005 maka dikategorikan sebagai lonjakan (*spike*) dan diberi label 1. Sebaliknya, jika nilai PLR berada di bawah atau sama dengan ambang batas tersebut, maka diberi label 0.

Transformasi ini akan mengubah permasalahan regresi menjadi klasifikasi biner, yang bertujuan untuk deteksi dini terjadap kejadian ekstrem (*outlier*) pada jaringan. Dengan pendekatan ini, model dapat lebih fokus dalam mengenali kondisi yang cukup kritis bagi QoS.

4.1.2.5 Pembentukan Target Biner, Normalisasi, dan Sekuesial Data

Pada tahap transformasi target, nilai *Packet Loss Ratio* (PLR) yang semula bersifat kontinu dikonversi menjadi variabel target biner *is_spike* berdasarkan ambang batas 0.005. Artinya, jika ilai PLR melebihi ambang tersebut diklasifikasikan sebagai lonjakan (spike) dan diberi label 1, sedangkan nilai di bawahnya diberi label 0. Transformasi ini mengubah permasalahan awal menjadi tugas klasifikasi biner, yang lebih sesuai untuk mendeteksi kejadian ekstrem dalam jaringan.

Seluruh fitur numerik yang ada akan dinormalisasi menggunakan *MinMaxScaler* ker rentang [0,1], agar model LSTM dapat belajar secara stabil. Tahapan ini penting, karena LSTM cukup sensitive terhadap skala antar fitur. Normalisasi juga berfungsi untuk mencegah dominasi fitur dengan skala yang lebih besar terhadap fitur lainnya.

Setelah proses normalisasi, data diubah ke dalam bentuk sekuensial tiga dimensi (*samples*, *timesteps*, *features*) dengan panjang jendela (*window size*) selama 24 jam. Transformasi ini dilakukan untuk menyesuaikan format *input* dengan kebutuhan arsitektur model LSTM. Hal tersebut dilakukan karena LSTM dirancang untuk menangani data deret waktu dengan mempertimbangkan urutan dan ketergantungan temporal antar data.

4.1.2.6 Pembagian Data

Data dibagi secara temporal menjadi *training set* dan *testing set* untuk memastikan bahwa tidak terjadi kebocoran data (data *leakage*) dari masa

depan ke masa lalu dalam proses pelatihan model. Hasil pembagian data dapat dilihat pada Tabel 4.5.

Pada Tabel 4.5, angka 24 menunjukkan jumlah *timestep* atau panjang *window size*, yang berarti model memanfaatkan data dari 24 jam sebelumnya untuk memprediksi satu nilai target. Sementara itu, angka 13 menunjukkan jumlah fitur yang digunakan pada setiap *timestep*. Ukuran target hanya terdiri dari satu dimensi karena berisi label biner *is_spike* (0 atau 1) untuk setiap sekuens *input*.

Tabel 4. 5 Hasil Pembagian Data

Jenis Data	Ukuran Fitur (X)	Ukuran Target (y)	Keterangan
Trainin	[1.811.246, 24, 13]	[1.811.246]	Data untuk pelatihan model.
Testing	[452.812, 24, 13]	[452.812]	Data untuk evalusai, tidak akan dilibatkan dalam pelatihan.

4.1.2.7 Penanganan Ketidakseimbangan Kelas

Dataset yang digunakan memiliki ketidakseimbangan distribusi antara kelas mayoritas (normal) dan kelas minoritas (lonjakan). Ketidakseimbangan ini dapat menyebabkan model cenderung mengabaikan kelas minoritas dan menghasilkan kinerja prediksi yang buruk terhadap kejadian lonjakan (spike), yang justru merupakan fokus utama dalam penelitian ini.

Untuk mengatasi hal tersebut, digunakan strategi *class weighting* selama proses pelatihan model. Bobot untuk masing-masing kelas dihitung secara otomatis berdasarkan proporsi distribusi kelas dalam data pelatihan, yang menghasilkan bobot sebesar {0: 0.629, 1: 2.432}.

Nilai tersebut menunjukkan bahwa kesalahan prediksi pada kelas minoritas (label 1 – lonjakan) akan diberikan penalti yang lebih tinggi dibandingkan dengan kelas mayoritas (label 0 – normal). Dengan demikian, model didorong untuk lebih memperhatikan dan mengenali pola-pola yang mengarah pada kejadian lonjakan.

Pendekatan ini diharapkan dapat meningkatkan performa model terutama dari segi sensitivitas atau *Recall* terhadap kelas minoritas, yang penting dalam konteks deteksi dini gangguan jaringan.

4.1.3 Hasil Optimasi Model LSTM (*Hyperparameter Tuning*)

Proses optimasi *hyperparameter* model LSTM dilakukan menggunakan Proses optimasi hyperparameter pada model LSTM dilakukan menggunakan Keras Tuner dengan algoritma *Hyperband*, yang dikenal efisien dalam melakukan pencarian pada ruang parameter yang luas dengan keterbatasan sumber daya komputasi. Proses tuning dilakukan menggunakan data pelatihan, dengan menerapkan *class weighting* untuk menangani ketidakseimbangan kelas, sebagaimana dijelaskan pada subbab sebelumnya.

Tujuan utama dari proses optimasi ini adalah untuk memaksimalkan nilai *recall* pada data validasi (*val_recall*), khususnya untuk kelas minoritas (label 1 – lonjakan), karena metrik ini merupakan indikator utama dalam keberhasilan sistem peringatan dini terhadap gangguan jaringan.

Berdasarkan hasil eksplorasi ruang parameter menggunakan Hyperband, diperoleh konfigurasi seperti yang ditampilkan pada Tabel 4.6 berikut:

Tabel 4. 6 Parameter Optimal Model LSTM Berdasarkan *Hyperband Tuner*.

Parameter	Nilai
Jumlah *Units* LSTM *Layer* 1	64
Dropout Rate Layer 1	0.3

Jumlah *Units* LSTM *Layer* 2	64
Dropout Rate Layer 2	0.4
Learning Rate (*Adam Optimizer*)	0.01

Kombinasi *hyperparameter* di atas merupakan hasil terbaik yang diperoleh dari proses tuning, dengan mempertimbangkan performa model pada data validasi. Konfigurasi ini menunjukkan keseimbangan antara kemampuan generalisasi model (menghindari *overfitting*) dan sensitivitas terhadap kelas minoritas, sehingga dapat digunakan sebagai konfigurasi akhir dalam pelatihan model LSTM untuk prediksi lonjakan *Packet Loss Ratio* (PLR).

4.1.4 Hasil Pelatihan Model Akhir

Setelah diperoleh konfigurasi hyperparameter terbaik, model LSTM akhir dibangun dan dilatih menggunakan seluruh data pelatihan (*x_train, y_train*) dengan parameter optimal yang diperoleh pada tahap sebelumnya. Proses pelatihan dilakukan dengan batas maksimum 40 epoch.

Untuk mencegah overfitting dan meningkatkan efisiensi pelatihan, digunakan *Callback EarlyStopping* dengan nilai patience sebesar 15. Artinya, proses pelatihan akan dihentikan secara otomatis apabila nilai *validation loss* tidak menunjukkan perbaikan selama 15 epoch berturut-turut. Strategi ini bertujuan untuk menghindari pelatihan yang berlebihan, yang dapat menurunkan kemampuan generalisasi model terhadap data yang tidak terlihat.

Selain itu, diterapkan juga *Callback ModelCheckpoint* dengan tujuan menyimpan bobot model terbaik sepanjang proses pelatihan. Kriteria penyimpanan bobot model ini didasarkan pada nilai *validation recall* tertinggi, sehingga model akhir yang digunakan merupakan versi terbaik dalam mendeteksi kelas minoritas (lonjakan) secara akurat.

Dengan penerapan kedua *callback* tersebut, proses pelatihan menjadi lebih adaptif dan berorientasi pada metrik kunci sistem, yakni kemampuan dalam mendeteksi potensi lonjakan *Packet Loss Ratio* (PLR).

4.1.5 *Classification Report*

Evaluasi performa model dilakukan menggunakan *Classification Report* tersebut disajikan pada Tabel 4.7.

Tabel 4. 7. Hasil Evaluasi Klasifikasi.

	Precision	*Recall*	*F1-Score*	*Support*
0	0.87	0.47	0.61	370223
1	0.22	0.67	0.33	82589
Accuracy			0.51	452812
Macro AVG	0.54	0.57	0.47	452812
Weighted AVG	0.75	0.51	0.56	452812

Berdasarkan hasil pada Tabel 4.7. model menunjukkan pola kinerja yang mencerminkan *trade-off* khas pada dataset yang tidak seimbang, khususnya dalam konteks deteksi lonjakan *Packet Loss Ratio* (PLR). Rincian kinerja untuk masing-masing kelas adalah sebagai berikut:

1. Untuk kelas kelas 1 (lonjakan), model memiliki:
    - *Precision* memiliki nilai 0.22 (22%), menunjukkan bahwa dari seluruh prediksi "lonjakan" yang dihasilkan oleh model, hanya 22% yang benar-benar merupakan lonjakan aktual.
    - *Recall* memiliki nilai cukup tinggi yaitu 0.67 (67%), mengindikasikan bahwa sebagian besar kejadian lonjakan aktual berhasil dikenali oleh model.

- *F1-Score* sebesar 0.33 (33%), yang mencerminkan keseimbangan antara precision yang rendah dan recall yang cukup tinggi.

2. Untuk kelas kelas 0 (normal), model memiliki:
    - *Precision* sebesar 0.87 (87%), mengindikasikan bahwa sebagian besar prediksi "normal" yang dihasilkan oleh modelmodel adalah benar.
    - *Recall* memiliki nilai sebesar 0.47 (47%), menunjukkan bahwa hanya sekitar setengah dari semua kejadian "normal" yang berhasil dideteksi oleh model.
    - *F1-Score* memiliki nilai sebesar 0.61 (61%), performa yang relatif lebih seimbang pada kelas ini dibandingkan kelas lonjakan.

Secara keseluruhan, model mencapai akurasi sebesar 51%, yang secara kasat mata mungkin tampak rendah. Namun, dalam konteks klasifikasi dengan data yang tidak seimbang, akurasi bukan merupakan metrik utama. Fokus utama dalam studi ini adalah kemampuan model dalam mendeteksi lonjakan, sehingga nilai *recall* pada kelas 1 yang mencapai 67% menjadi indikator keberhasilan model dalam mengenali kejadian penting.

Meskipun demikian, nilai *precision* yang rendah pada kelas lonjakan menunjukkan adanya jumlah *False Positives* yang cukup tinggi, yang berarti model masih sering salah mengklasifikasikan kondisi normal sebagai lonjakan. Hal ini berpotensi meningkatkan jumlah peringatan palsu (*false alarm*), yang dapat mengganggu sistem manajemen jaringan.

Adapun metrik rata-rata lainnya memberikan gambaran umum sebagai berikut:

- *Macro average*: *Precision* 0,54, *Recall* 0,57, dan *F1-Score* 0,47. Metrik ini menghitung rata-rata tanpa mempertimbangkan proporsi kelas, sehingga mencerminkan performa yang seimbang antar kelas.

- *Weighted average*: *Precision* 0,75, *Recall* 0,51, dan *F1-Score* 0,56. Metrik ini mempertimbangkan jumlah sampel pada setiap kelas, sehingga lebih mencerminkan performa keseluruhan model.

Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa model memiliki kecenderungan lebih kuat dalam mendeteksi kelas minoritas (lonjakan), meskipun dengan konsekuensi peningkatan *false positive*. Strategi ini umum digunakan dalam sistem yang memprioritaskan deteksi terhadap kejadian kritis atau abnormal seperti lonjakan *Packet Loss Ratio*.

4.1.6 Hasil Evaluasi Model Menggunakan Ambang Batas

Model LSTM yang telah dilatih sepenuhnya kemudian dievaluasi menggunakan data pengujian (*x_test, y_test*) yang sebelumnya tidak pernah dilihat oleh model. Prediksi yang dihasilkan berupa probabilitas dikonversi menjadi kelas biner (0 atau 1) menggunakan ambang batas (*threshold*) sebesar 0.5.

Tabel 4. 8. Metrik Evaluasi Model LSTM.

Metrik	Nilai
Accuracy	0.4179
Precision (Lonjakan)	0.2135
Recall (Lonjakan)	0.8166
F1-Score (Lonjakan)	0.3385

Evaluasi ini bertujuan untuk mengukur performa model dalam mengklasifikasikan kejadian lonjakan (*spike*) pada nilai *Packet Loss Ratio* (PLR) menggunakan metrik-metrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Hasil evaluasi model disajikan pada Tabel 4.8 berikut

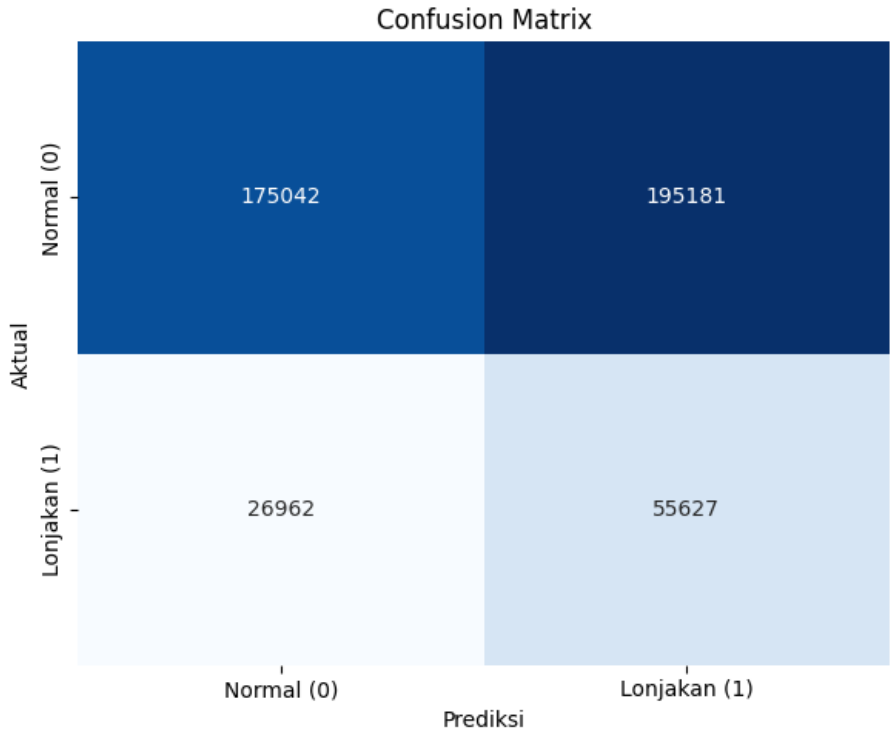
Berdasarkan hasil evaluasi pada Tabel 4.8, dapat disimpulkan hal-hal berikut:

1. *Accuracy* model secara keseluruhan adalah sebesar 41.79%. Meskipun tampak rendah, nilai ini bukan merupakan indikator utama dalam kasus klasifikasi dengan distribusi kelas yang tidak seimbang. Dalam konteks ini, metrik seperti *recall* dan *F1-score* untuk kelas minoritas lebih relevan untuk menilai performa model.
2. *Precision* untuk kelas *lonjakan* sebesar 21.35%, yang menunjukkan bahwa dari seluruh prediksi lonjakan yang dihasilkan model, hanya sekitar 21% yang benar-benar merupakan lonjakan aktual. Nilai ini mengindikasikan masih adanya cukup banyak prediksi positif palsu (*false positives*), yang merupakan konsekuensi dari strategi peningkatan *recall*.
3. *Recall* untuk kelas "Lonjakan" (kelas 1) mencapai 81.66%, yang berarti model berhasil mengidentifikasi lebih dari 80% kejadian lonjakan PLR yang sebenarnya terjadi. Ini merupakan pencapaian penting, karena tujuan utama model ini adalah sebagai sistem peringatan dini terhadap penurunan kualitas jaringan
4. *F1-Score* sebesar 33.85% menunjukkan keseimbangan antara *precision* dan *recall*. Nilai ini menggambarkan sejauh mana model mampu mempertahankan performa dalam mendeteksi kelas minoritas secara efektif tanpa terlalu banyak menghasilkan prediksi palsu..

Secara keseluruhan, model menunjukkan kemampuan yang kuat dalam mengenali kejadian lonjakan PLR, meskipun masih menghadapi tantangan dalam mengurangi jumlah *false positive*. Namun, dalam konteks sistem deteksi dini di jaringan, kemampuan tinggi dalam mengenali potensi gangguan (tingginya *recall*) merupakan aspek yang lebih diutamakan.

4.1.7 *Confusion Matrix*

Confusion Matrix memberikan gambaran visual atas kinerja model klasifikasi dalam membedakan antara dua kelas: *normal* dan *lonjakan*. Visualisasi ini memperlihatkan jumlah prediksi benar maupun salah yang dilakukan oleh model terhadap masing-masing kelas. Gambar 4.11 menunjukkan hasil visualisasi heatmap dari confusion matrix model LSTM yang telah dilatih dan dioptimasi.



Gambar 4. 10. Heatmap Distribusi *Confusion Matrik*.

Berdasarkan Gambar 4.11, diperoleh rincian nilai sebagai:berikut:

1. True Negative (TN): 175.042
   Jumlah data dengan kondisi normal yang berhasil diklasifikasikan dengan benar sebagai normal. Nilai ini mencerminkan kemampuan

model dalam mengidentifikasi sebagian besar kondisi normal secara akurat.

2. False Positive (FP): 195.181
   Jumlah data *normal* yang secara keliru diprediksi sebagai *lonjakan*. Nilai FP yang tinggi berkontribusi langsung pada penurunan precision pada kelas *lonjakan*, karena banyak peringatan yang diberikan meskipun lonjakan sebenarnya tidak terjadi (*false alarm*).
3. False Negative (FN): 269.62
   Jumlah kasus *lonjakan* yang tidak berhasil dideteksi oleh model dan diklasifikasikan sebagai *normal*. FN yang tinggi dapat menurunkan nilai recall dan membahayakan sistem jika peristiwa penting tidak terdeteksi.
4. True Negative (TN): 55.627
   Jumlah data dengan kondisi *lonjakan* yang berhasil diprediksi dengan benar sebagai *lonjakan*. Nilai TP yang tinggi sangat diharapkan dalam konteks sistem peringatan dini, karena menunjukkan efektivitas model dalam mengenali kejadian yang berisiko.

Secara keseluruhan, model ini mengadopsi pendekatan yang cenderung agresif dalam mendeteksi lonjakan, yaitu dengan meningkatkan jumlah *True Positives* (TP) meskipun disertai dengan peningkatan *False Positives* (FP). Strategi ini sejalan dengan tujuan utama penelitian, yaitu memaksimalkan recall pada kelas lonjakan, guna memastikan bahwa kejadian penting tidak terlewatkan.

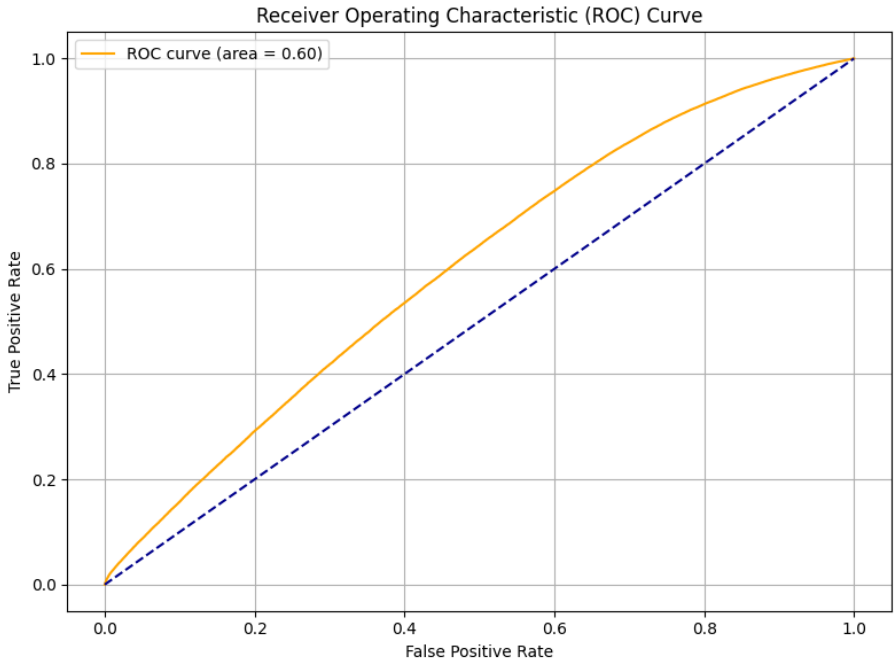
Namun, konsekuensi dari pendekatan ini adalah meningkatnya jumlah *false alarm*, yang dapat mengurangi efisiensi operasional sistem jika tidak ditangani dengan strategi mitigasi yang tepat.

4.1.8 ROC *Curve* dan AUC *Score*

Gambar 4.12 memperlihatkan kurva *Receiver Operating Characteristic* (ROC) yang digunakan untuk mengevaluasi kemampuan model dalam

membedakan dua kelas, yaitu lonjakan (kelas positif) dan normal (kelas negatif). Kurva ROC menunjukkan hubungan antara *True Positive* (TP) dan *False Positive* (FP) pada berbagai ambang batas klasifikasi.

Dalam grafik tersebut, kurva berwarna oranye merepresentasikan performa model, sementara garis biru putus-putus merupakan baseline dengan nilai AUC sebesar 0,5 yang menggambarkan performa model acak. Berdasarkan kurva yang dihasilkan, diperoleh nilai *Area Under the Curve* (AUC) sebesar 0,60.

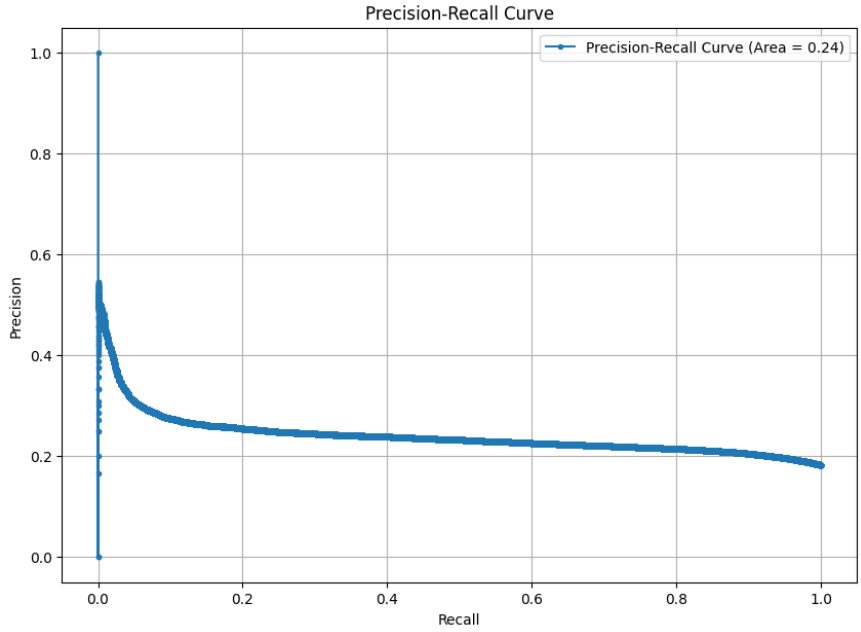


Gambar 4. 11. *Kurva Receiver Operating Characteristik* (ROC) *Model LSTM.*

Nilai ini mengindikasikan bahwa kemampuan diskriminatif model dalam membedakan antara kelas positif dan negatif masih tergolong rendah, meskipun sedikit lebih baik daripada tebakan acak. Kurva yang cenderung mendekati garis diagonal baseline menunjukkan bahwa model belum optimal dalam mengklasifikasikan data secara akurat.

Kondisi ini dapat disebabkan oleh sejumlah faktor, seperti keterbatasan fitur yang digunakan, kurang optimalnya pemilihan parameter, atau ketidakseimbangan kelas pada data latih. Nilai AUC sebesar 060 juga menandakan adanya ruang untuk peningkatan performa model melalui penambahan fitur yang lebih informatif, seperti tuning *hyperparameter*, serta teknik penyeimbangan data untuk meningkatkan kemampuan model dalam mengenali pola lonjakan secara lebih akurat.

4.1.9 *Precision-Recall Curve* dan *Optimal Threshold*



Gambar 4. 12. Kurva *Precision-Recall* Model LSTM.

Pada Gambar 4.13. menunjukkan kurva *Precision-Recall* (PR *Curve*) yang digunakan untuk mengevaluasi kinerja model dalam mengidentifikasi kejadian lonjakan *Packet Loss Ratio* (PLR). Evaluasi ini menjadi sangat penting mengingat distribusi kelas pada dataset bersifat tidak seimbang, di mana jumlah data pada kelas "lonjakan" (positif) jauh lebih sedikit dibandingkan kelas "normal" (negatif). Kurva ini menggambarkan hubungan antara nilai

precision (ketepatan) dan *recall* (cakupan) pada berbagai nilai ambang batas klasifikasi (*threshold*).

Berdasarkan hasil visualisasi, diperoleh nilai *Area Under the Precision-Recall Curve* (PR AUC) sebesar 0.24, yang menunjukkan bahwa model belum mampu menghasilkan kombinasi *precision* dan *recall* yang optimal dalam mendeteksi kejadian lonjakan PLR. Nilai ini tergolong rendah, yang mengindikasikan bahwa model cenderung menghasilkan banyak *false positive* ketika mencoba meningkatkan *recall*. Hal ini terlihat dari pola kurva yang menurun tajam, di mana *precision* menurun drastis saat *recall* meningkat.

Situasi ini mencerminkan bahwa ketika model berusaha mendeteksi lebih banyak kejadian lonjakan (meningkatkan *recall*), model justru menghasilkan banyak prediksi yang salah (menurunnya *precision*). Hal tersebut dapat berimplikasi pada sistem peringatan dini yang terlalu sensitif, sehingga menghasilkan banyak peringatan palsu.

Dalam konteks penelitian ini, pemilihan ambang batas (*threshold*) klasifikasi yang optimal menjadi sangat krusial. Terdapat pertukaran (trade-off) yang perlu dipertimbangkan:

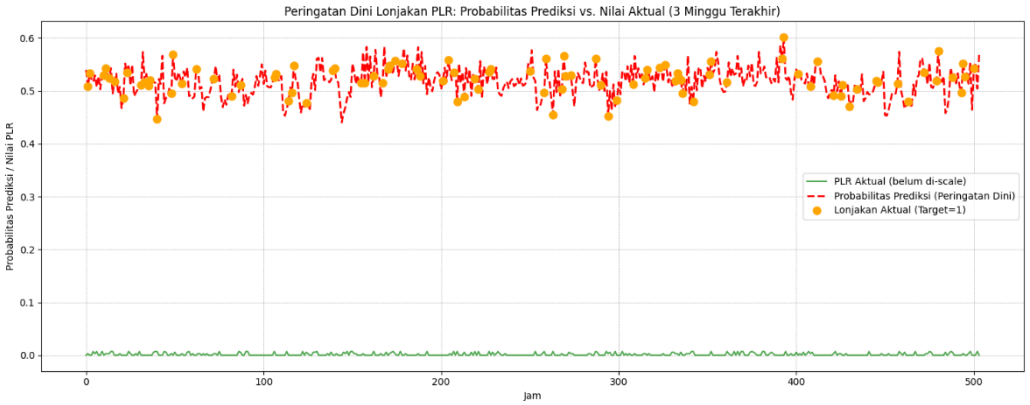
- Ambang batas terlalu rendah → *Recall* tinggi, namun *Precision* menurun drastis akibat tingginya *false positive.*
- Ambang batas terlalu tinggi → *Precision* mungkin meningkat, namun banyak kejadian lonjakan yang tidak terdeteksi (*recall* rendah).

Untuk itu, dalam penelitian ini dipilih ambang batas optimal sebesar 0.4567. Pemilihan nilai ini bertujuan untuk meningkatkan kemampuan model dalam mendeteksi kejadian lonjakan (memaksimalkan *recall*), meskipun dengan konsekuensi penurunan *precision*. Strategi ini sesuai dengan fokus penelitian, yaitu mengembangkan sistem prediktif sebagai peringatan dini terhadap potensi lonjakan PLR dalam jaringan. Dengan demikian, meskipun prediksi

tidak sepenuhnya akurat, model tetap memberikan informasi penting yang dapat digunakan untuk langkah mitigasi secara proaktif.

4.1.10 Analisis Prediksi Lonjakan PLR Berdasarkan Periode Waktu

Selain evaluasi model secara keseluruhan, dilakukan pula analisis kinerja model berdasarkan periode waktu tertentu untuk memberikan gambaran yang lebih rinci mengenai kemampuan model dalam mendeteksi lonjakan *Packet Loss Ratio* (PLR). Pada penelitian ini, evaluasi dilakukan dengan membandingkan hasil prediksi setiap tiga minggu terakhir dari data uji.



Gambar 4. 13 Visualisasi Prediksi Model Per Jam Dibandingkan dengan Kejadian Lonjakan PLR Aktual (Data Uji 3 Minggu Terakhir).

Gambar 4.13 memperlihatkan perbandingan antara nilai probabilitas prediksi dengan nilai aktual PLR pada periode tiga minggu terakhir. Garis merah putus-putus merepresentasikan probabilitas prediksi model (peringatan dini), sedangkan titik berwarna oranye menunjukkan kejadian aktual lonjakan PLR (target = 1). Adapun garis hijau tipis menggambarkan nilai PLR aktual sebelum dilakukan proses *scaling*.

Hasil analisis per periode ini memberikan gambaran tambahan bahwa model tidak hanya bekerja secara rata-rata, tetapi juga dapat mempertahankan performanya pada rentang waktu tertentu.

Dari grafik tersebut terlihat bahwa:

1. Konsistensi Prediksi

   Probabilitas prediksi model cenderung berada di sekitar nilai ambang batas (threshold = 0,5), yang menandakan sensitivitas model dalam mendeteksi potensi lonjakan.
2. Deteksi Lonjakan

   Titik-titik lonjakan aktual (oranye) sebagian besar berhasil diikuti oleh peningkatan probabilitas prediksi, sehingga model mampu memberikan indikasi peringatan dini sebelum terjadi lonjakan PLR.
3. Performa Periode Panjang

   Dengan pembagian data ke dalam periode waktu (tiga minggu), terlihat bahwa pola prediksi model relatif stabil dan tidak menunjukkan fluktuasi ekstrem. Hal ini menunjukkan bahwa model memiliki generalisasi yang cukup baik pada data deret waktu jaringan.

Dengan demikian, evaluasi berbasis periode waktu dapat dijadikan dasar untuk implementasi sistem peringatan dini yang lebih praktis dalam pemantauan jaringan secara *real-time*.

4.2 Analisis Dan Pembahasan

Hasil evaluasi menunjukkan bahwa model LSTM yang dikembangkan memiliki performa yang baik dalam mendeteksi lonjakan *Packet Loss Ratio* (PLR), khususnya saat dilatih menggunakan data selama 10 bulan. Dengan nilai *Recall* sebesar 81,66%, model mampu mengidentifikasi lebih dari 80% kejadian lonjakan PLR yang sebenarnya. Capaian ini menunjukkan bahwa model berpotensi digunakan sebagai komponen dalam sistem peringatan dini (*early warning system*).

Dalam konteks operasional jaringan, kemampuan untuk mendeteksi sebagian besar lonjakan adalah prioritas utama, meskipun harus disertai dengan konsekuensi munculnya peringatan palsu. Hal ini tercermin dari nilai *Precision* yang relatif rendah, yaitu sebesar 21%. Kompromi antara *Recall* yang tinggi dan *Precision* yang rendah merupakan fenomena umum pada permasalahan klasifikasi dengan distribusi data yang tidak seimbang, di mana kelas minoritas (lonjakan) jauh lebih sedikit dibandingkan dengan kelas mayoritas (normal).

Model yang dikembangkan pada penelitian ini dirancang dengan sensitivitas tinggi untuk meminimalisasi terjadinya kesalahan tipe II (*False Negative*), yaitu kondisi ketika lonjakan *Packet Loss Ratio* (PLR) tidak berhasil terdeteksi. Namun, strategi ini berdampak pada meningkatnya kesalahan tipe I (*False Positive*), sehingga nilai *Precision* cenderung menurun. Kondisi tersebut masih dapat ditoleransi dalam implementasi nyata, karena potensi terjadinya *alarm fatigue* pada operator jaringan dapat diminimalkan melalui langkah-langkah tambahan, seperti penerapan *post-processing*, penyesuaian nilai ambang batas (*threshold*), ataupun integrasi dengan sistem pemantauan jaringan lain pada tahap produksi.

Selain itu, dilakukan pula pengujian pada periode waktu yang lebih pendek, yaitu tiga minggu, untuk melihat konsistensi model terhadap variasi jangka waktu data. Hasil menunjukkan bahwa kinerja model cenderung menurun pada skala waktu yang lebih singkat. Hal ini dapat dijelaskan karena data tiga minggu tidak cukup merepresentasikan pola musiman maupun variasi jangka panjang yang memengaruhi kualitas jaringan. Dengan data yang terbatas, model memiliki kecenderungan *overfitting* terhadap pola jangka pendek, sehingga kemampuan generalisasi menjadi berkurang.

Penggunaan data dalam rentang waktu 10 bulan memberikan keuntungan signifikan dalam mengenali pola temporal yang lebih kompleks dan stabil. Volume data yang lebih besar dan bervariasi, termasuk fluktuasi nilai *Jitter* yang tidak konstan, memungkinkan model LSTM mempelajari pola musiman dan siklus

jangka panjang yang berpotensi memengaruhi performa jaringan. Hal ini secara langsung mendukung peningkatan nilai *Recall* yang diperoleh.

Secara keseluruhan, model LSTM yang dioptimalkan dalam penelitian ini menunjukkan potensi besar sebagai solusi prediktif untuk meningkatkan efisiensi pemantauan kualitas jaringan telekomunikasi. Kemampuannya untuk secara proaktif mengidentifikasi lonjakan PLR dapat membantu operator dalam mengambil tindakan mitigasi yang cepat, sehingga meminimalkan dampak negatif pada pengalaman pengguna.