**Jurnal Informatika dan Teknologi Interaktif**
Volume 2, Nomor 3, Desember 2025

DOI : 10.63547/jiite.v2i3.73
e-ISSN : 3048-1023



# Analisis Perbandingan XGBoost, Random Forest, dan LSTM untuk Prediksi Jumlah Penumpang Bus

**Rahmawati[1], Rias Estriana[2], Ratna Praptiwi[3], Brilliant Salsabila[4], Nazly Rafa Oktafian[5], Toik Zakiyudin[6]**
[1,2,3,4]Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Purwokerto, Indonesia
[5]Program Studi Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom Purwokerto, Purwokerto, Indonesia
[6]Program Studi Sistem Informasi, Fakultas Matematika dan Komputer, Universitas Nahdlatul Ulama Al Ghazali Cilacap, Cilacap, Indonesia
surel:[1]snozerhma@gmail.com,[2]estrianarias@gmail.com,[3]ratnapraptiwi1@gmail.com,[4]brillia246@gmail.com,
[5]rafaoktavian2005@gmail.com,[6]toikzaki5753@gmail.com

**Info Artikel**

***Sejarah artikel:***
Diterima 10-07-2025
Revisi 24-07-2025
Diterima 02-08-2025

***Kata kunci:***
Analisi Perbandingan
Extreme Gradient Boosting
Random Forest
Long Short Term Memory
Kasus Prediksi

**ABSTRAK**

Peningkatan kebutuhan transportasi publik di kawasan urban mendorong perlunya sistem prediksi jumlah penumpang yang akurat guna mendukung pengambilan keputusan dalam pengelolaan armada bus. Penelitian ini membandingkan tiga algoritma machine learning, yaitu XGBoost, Random Forest, dan Long Short-Term Memory (LSTM), dalam memprediksi jumlah penumpang bus bulanan berdasarkan dataset Los Angeles County Metropolitan Transportation Authority (LACMTA) periode 2009–2024. Pendekatan yang digunakan adalah time-series regression dengan teknik rekayasa fitur seperti lag features dan moving averages. Evaluasi model dilakukan menggunakan metrik MAE, RMSE, dan $R^2$ Score. Hasil penelitian menunjukkan bahwa XGBoost merupakan algoritma dengan performa terbaik dengan $R^2$ sebesar 0.9936 dan RMSE terendah sebesar 352.78, diikuti oleh Random Forest. Sementara itu, LSTM serta model statistik seperti Exponential Smoothing dan Auto-SARIMA menunjukkan performa yang buruk dan prediksi yang tidak stabil. Penelitian ini menegaskan bahwa model tree-based lebih unggul dalam menangani data deret waktu agregat bulanan dengan karakteristik ketimpangan dan fluktuasi yang tinggi.

***Penulis Korespondensi:***
Rahma Wati
Program Studi Informatika Fakultas Ilmu Komputer Universitas Amikom Purwokerto
Email: snozerhma@gmail.com

## 1. PENDAHULUAN

Perkembangan pesat urbanisasi dan pertumbuhan penduduk di kota-kota besar telah meningkatkan kebutuhan akan transportasi publik, terutama moda bus. Meskipun bus menjadi tulang punggung sistem transportasi, berbagai tantangan seperti ketidakpastian waktu tempuh, kemacetan, dan menurunnya kualitas layanan menyebabkan penurunan minat masyarakat untuk menggunakannya. Untuk menjaga kualitas layanan dan mendorong peningkatan pengguna, dibutuhkan pendekatan berbasis data melalui prediksi jumlah penumpang yang akurat. Prediksi yang akurat akan mendukung pengambilan keputusan dalam manajemen armada, penjadwalan, dan penyusunan rute layanan.

Dalam beberapa tahun terakhir, machine learning menjadi pendekatan populer dalam prediksi transportasi karena kemampuannya menangani data dalam jumlah besar dan kompleks. Namun, tantangan besar muncul ketika

data memiliki distribusi yang timpang (imbalanced), di mana sebagian besar data menunjukkan nol aktivitas (tidak naik bus) dibandingkan dengan aktivitas nyata. Algoritma seperti XGBoost dan Random Forest banyak digunakan karena kemampuannya menangani data tabular dengan akurasi tinggi. [1]menunjukkan bahwa XGBoost memberikan hasil prediksi yang lebih akurat dibanding Random Forest pada sebagian besar jalur bus di Brisbane, khususnya dalam kondisi cuaca ekstrem. Di sisi lain, algoritma LSTM terbukti mampu menangkap pola jangka panjang pada data runtun waktu, dan sering digunakan dalam prediksi jumlah penumpang transportasi publik.

Penelitian sebelumnya, [2] mengusulkan penggunaan Deep-GAN untuk mengatasi ketimpangan data dalam prediksi jumlah penumpang per jam di kota Changsa, dan menunjukkan bahwa data sintetis dapat meningkatkan akurasi model. Meskipun model ini meningkatkan akurasi prediksi, penelitian tersebut lebih difokuskan pada level individu (micro-level) boarding behavior dan menggunakan data granular per jam. Sementara itu, penelitian [1] menyajikan perbandingan antara Random Forest dan XGBoost dalam memprediksi jumlah penumpang secara near-real-time di Brisbane, dengan menyimpulkan bahwa XGBoost cenderung memberikan hasil lebih akurat, terutama dalam kondisi cuaca ekstrem dibanding Random Forest. Penelitian [1] mengembangkan model LSTM untuk prediksi jumlah penumpang bus harian, yang mampu meningkatkan akurasi hingga 23% dibanding baseline model tradisional. Model ini sangat cocok diterapkan pada data runtun waktu seperti dataset LACMTA. Kebutuhan akan peramalan meningkat sejalan dengan usaha manajemen untuk mengurangi ketergantungannya pada hal-hal yang belum pasti. Salah satu metode untuk peramalan *(forecasting)* adalah Exponential Smoothing.[3] Berdasarkan uraian tersebut permasalahan dapat diselesaikan dengan menggunakan sebuah sistem prediksi yang dapat mengetahui jumlah penumpang bus.[4] jumlah penumpang pada tahun ke tahun mengalami peningkatan dan penurunan, dengan dilakukan simulasi untuk memprediksi jumlah pada 3 bulan selanjutnya[5].

Berbeda dengan penelitian sebelumnya yang lebih fokus pada: Prediksi boarding penumpang individu per jam (granular/mikro), Data dari sistem smart-card di wilayah lokal seperti Changsha, bukan transportasi publik metropolitan seperti di LACMTA, Level prediksi harian atau per jam *(hourly level)*, bukan bulanan agregat.

Penelitian ini mengambil pendekatan agregat bulanan pada moda bus dari dataset LACMTA (Los Angeles County) dengan cakupan waktu panjang (2009–2024), dan membandingkan secara langsung tiga algoritma: XGBoost, Random Forest, dan LSTM dalam konteks data dengan ketimpangan jumlah penumpang yang tinggi, namun tanpa missing value.

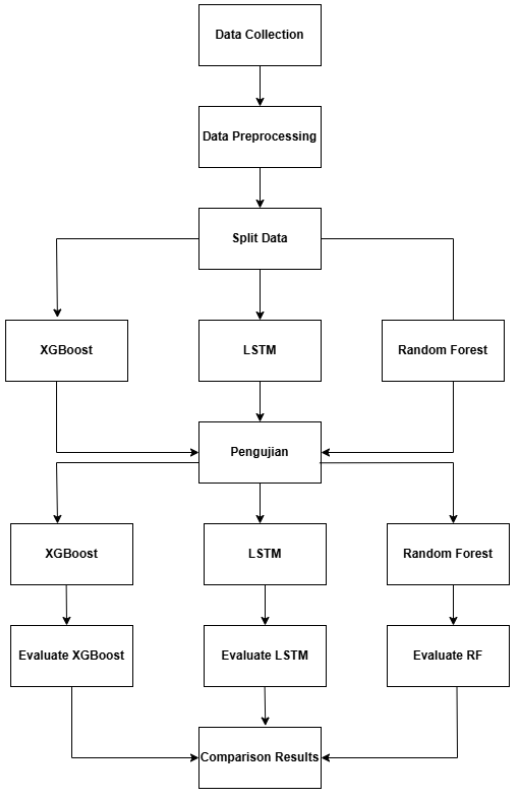
Berdasarkan beberapa penelitian diatas, tujuan utama penelitian ini adalah menganalisis dan membandingkan performa algoritma XGBoost, Random Forest, dan LSTM dalam memprediksi jumlah penumpang moda bus dari dataset LACMTA (2009–2024) dengan mengukur performa model berdasarkan metrik RMSE, MAPE dan R2, Selain itu, mengidentifikasi algoritma terbaik dalam menangani data time-series agregat bulanan dengan ketimpangan yang signifikan.

## 2. METODE

Jenis pendekatan dalam penelitian ini adalah pendekatan kuantitatif dengan metode prediktif menggunakan algoritma machine learning berbasis deret waktu. Pendekatan ini dipilih karena kemampuannya dalam menganalisis data berurutan dan menghasilkan prediksi yang akurat berdasarkan pola historis. Kerangka kerja penelitian diadaptasi dari jurnal "Hourly Bus Passenger Demand Prediction Through Machine Learning Algorithms" [2], yang berfokus pada data boarding per jam dan deep learning untuk memodelkan perilaku perjalanan.

Metode yang diusulkan penulis berbeda dengan metode yang ada pada jurnal "Hourly Bus Passenger Demand Prediction Through Machine Learning Algorithms" [2], karena penulis melakukan penyesuaian pada jenis data, yaitu data agregat bulanan dari *Los Angeles County Metropolitan Transportation Authority* (LACMTA) periode Januari 2009 hingga Maret 2024. Menunjukan kondisi data jumlah penumpang sangat bervariasi. Kolom *Riders* memiliki distribusi yang tidak merata, di mana sebagian besar nilai berada di bawah 10.000, namun terdapat outlier yang mencapai 171.162. Beberapa jalur tercatat hanya memiliki sedikit penumpang, sementara jalur lain sangat ramai, yang mengindikasikan adanya ketimpangan (data imbalance). Meski begitu, data ini tergolong sangat bersih karena tidak ditemukan *missing value*, dengan semua kolom terisi secara lengkap. Tiga algoritma digunakan untuk tujuan perbandingan performa, yaitu XGBoost, LSTM, dan Random Forest[6], dengan pengujian berbasis evaluasi metrik

regresi. Berikut visualisasi alur metodologi dan tahapan metodologi yang digunakan, seperti yang ditampilkan pada Gambar 1.




Gambar 1. Flowchart

### 2.1. Desain Penelitian

Penelitian ini bersifat eksperimental dengan desain longitudinal karena melibatkan data berurutan dari waktu ke waktu. Dataset yang digunakan adalah *monthly-ridership.csv* yang diambil dari Github LACMTA, berisi data jumlah penumpang bus berdasarkan waktu bulanan. Jumlah keseluruhan data sebelum difilter berdasarkan mode "bus" adalah 68.515 baris, dan setelah difilter menjadi 65.913 baris. Fitur-fitur penting yang digunakan antara lain*: Year, Month, DayType, Shakeup, Provider, Mode, Line, Days, dan Riders*. Target dari model adalah *Riders*, yang merupakan data numerik, sehingga masalah ini diklasifikasikan sebagai regresi *time series*[7].

### 2.2. Prosedur Penelitian

Alur penelitian secara umum meliputi beberapa tahap penting sebagai berikut:

#### 2.2.1. Pengumpulan Data

Dataset yang digunakan dari *Los Angeles County Metropolitan Transportation Authority* (LACMTA) yaitu monthly-ridership.csv, yang berisi data jumlah penumpang angkutan umum tiap bulan. Data ini kemudian disaring agar hanya mencakup moda transportasi "Bus". Ini dilakukan karena fokus penelitian adalah memprediksi jumlah penumpang bus, bukan moda lain seperti kereta atau trem[4].

#### 2.2.2. Preprocessing Data

Tahapan ini sangat krusial untuk membentuk data yang bersih dan siap digunakan:

a. Kolom waktu Year dan Month dikombinasikan menjadi kolom Date.
b. Data kemudian diurutkan berdasarkan tanggal agar urutan kronologisnya sesuai.

c. Nilai kosong *(missing values)* pada kolom numerik diisi dengan median, sementara data kategorikal yang kosong diisi dengan nilai 'Unknown'.
d. Encoding dilakukan untuk fitur kategorikal seperti DayType dan Provider dengan one-hot encoding.
e. Fitur tambahan *(feature engineering)* dibuat untuk memperkaya model:
   - Fitur waktu seperti month, year, dan month_since_start.
   - Lag features, yaitu jumlah penumpang pada 1 hingga 12 bulan sebelumnya.
   - Moving averages, yaitu rata-rata jumlah penumpang dalam 3 atau 6 bulan terakhir.
f. Terakhir, semua baris yang mengandung nilai kosong setelah penambahan lag dibuang agar model tidak terganggu oleh data tidak lengkap.

### 2.2.3. Menentukan Fitur dan Target

Setelah data siap, fitur-fitur input (X) dan target output (y) dipisahkan. Fitur meliputi semua hasil dari rekayasa fitur, sedangkan targetnya adalah jumlah penumpang (Riders).

### 2.2.4. Pembagian Data

Data dibagi menjadi data latih dan data uji dengan rasio sekitar 80:20. Karena ini adalah data deret waktu *(time series)*, pembagiannya dilakukan tanpa pengacakan (shuffle=False) agar pola urutan waktu tetap terjaga dan tidak rusak.

### 2.2.5. Cross-Validation untuk Time Series

Untuk menghindari overfitting dan memastikan hasil yang stabil, digunakan metode *TimeSeriesSplit*, yaitu *cross-validation* khusus yang mempertahankan urutan waktu saat membagi data ke dalam beberapa lipatan *(folds)*.

### 2.2.6. Pelatihan Model

a. Algoritma XGBoost

Model XGBoost dilatih dengan bantuan GridSearchCV, yaitu pencarian parameter terbaik menggunakan validasi silang. Ini dilakukan agar model mendapatkan konfigurasi yang optimal[8].

b. Algoritma Random Forest

Proses pelatihan model Random Forest dilakukan dengan cara yang sama seperti XGBoost, yaitu melalui GridSearchCV dan TimeSeriesSplit. Model ini cocok untuk data tabular dan cukup kuat terhadap noise[9].

c. Algoritma LSTM
   - Persiapan Data untuk LSTM

     Karena LSTM (Long Short-Term Memory) membutuhkan data dalam format sekuensial dan bernilai skala seragam, maka dilakukan normalisasi data numerik dengan MinMaxScaler. Kemudian data diubah ke format 3 dimensi: [samples, time_steps, features], di mana setiap sampel mewakili 12 bulan data sebagai input untuk memprediksi bulan ke-13[10].
   - Pelatihan LSTM

     Model LSTM yang digunakan memiliki dua lapisan LSTM dengan dropout untuk mencegah overfitting, serta satu layer Dense untuk menghasilkan output prediksi. Pencarian kombinasi parameter terbaik dilakukan menggunakan RandomizedSearchCV, yang lebih cepat dibanding grid search untuk model neural network[11].

d. Holt-Winters dan Auto-SARIMA

Sebagai pembanding, dua model statistik juga dilatih:
   - Holt-Winters (Exponential Smoothing), yang mempertimbangkan trend dan seasonality.
   - Auto-SARIMA, yang mencari parameter ARIMA terbaik secara otomatis dengan mempertimbangkan pola musiman dan keacakan dalam data.

#### 2.2.7. Evaluasi Model

Semua model dievaluasi menggunakan metrik regresi standar, yaitu:

a. MAE (Mean Absolute Error)
b. RMSE (Root Mean Squared Error)
c. $R^2$ Score

Evaluasi dilakukan baik pada data uji maupun menggunakan k-fold cross-validation untuk memastikan bahwa hasil tidak hanya berlaku pada satu subset data[12].

#### 2.2.8. Visualisasi dan Hasil Evaluasi Model

Hasil evaluasi dari semua model disimpan ke dalam file CSV dan divisualisasikan dalam bentuk grafik perbandingan. Hal ini membantu dalam menilai model mana yang memiliki performa terbaik dalam memprediksi jumlah penumpang bus.

### 2.3. Spesifikasi Perangkat

Tabel 1. Spesifikasi Perangkat

| Komponen | Spesifikasi |
|---|---|
| Prosesor | Intel core i5 |
| RAM | 16GB |
| Sistem Operasi | Windows 11 |
| Platform Analisis | Spyder |
| Library Python | Pandas, NumPy, Scikit-learn, XGBoost, Keras, Matplotlib, Seaborn |

## 3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan serangkaian output yang mencerminkan efektivitas masing-masing tahapan dalam alur kerja sistem prediksi jumlah penumpang bus bulanan. Setiap fase dievaluasi berdasarkan keberhasilan transformasi data, pemilihan fitur, performa model, dan validitas hasil prediksi.

### 3.1. Pengumpulan dan Penyaringan Data

Data awal diperoleh dari github yaitu Dataset *Los Angeles County Metropolitan Transportation Authority* (LACMTA) yang berisi catatan bulanan jumlah penumpang berdasarkan moda transportasi. Untuk menjaga fokus dan konsistensi, data disaring agar hanya mencakup baris dengan Mode = 'Bus'. Hasil dari langkah ini untuk memastikan bahwa pola dan tren yang dipelajari benar-benar merepresentasikan sektor transportasi bus, bukan moda lainnya seperti kereta atau trem.

Data ini terbukti cukup kaya akan informasi, mencakup kombinasi waktu (Year, Month), kategori operasional (DayType, Provider), dan atribut target (Riders). Setelah penyaringan, dataset layak untuk tahap pemrosesan.

### 3.2. Preprosesing dan Feature Engineering

Hasil Tahap ini melibatkan transformasi data mentah menjadi bentuk yang dapat dipahami oleh mesin. Proses yang dilakukan meliputi:

a. Konversi waktu: Year dan Month digabung menjadi kolom Date.
b. Pembersihan data: Nilai kosong diisi menggunakan median (untuk numerik) atau label “Unknown” (untuk kategorikal).
c. Encoding kategorikal: Kolom DayType dan Provider diubah menjadi representasi numerik menggunakan one-hot encoding.
d. Rekayasa fitur: Ditambahkan fitur-fitur waktu seperti month, month_since_start, dan lag features (lag_1 hingga lag_12). Fitur ini memungkinkan model memahami pola masa lalu seperti efek musiman atau tren jangka panjang.

Data berhasil dibentuk menjadi format yang kaya dan informatif. Informasi waktu dapat ditelusuri secara runtut dan pola historis penumpang mulai terlihat jelas. Hal ini membuat model dapat "belajar" dengan baik dari data yang telah dirapikan dan disusun.

### 3.3. Pembagian Data dan Validasi

Data dalam penelitian ini dibagi secara proporsional, yaitu 80% digunakan untuk pelatihan (training set) dan 20% sisanya digunakan untuk pengujian (test set). Pembagian ini dilakukan untuk memastikan bahwa model memiliki cukup data untuk belajar sekaligus dapat diuji performanya secara adil pada data yang belum pernah dilihat sebelumnya.

Karena dataset bersifat time series, pembagian dilakukan tanpa pengacakan (no shuffle). Untuk model berbasis machine learning (XGBoost dan Random Forest), digunakan metode TimeSeriesSplit agar validasi tetap mempertahankan urutan waktu. Hal ini penting untuk menghindari "data leakage" dan menjaga akurasi prediksi jangka waktu mendatang[13].

Hasil pengujian model terhadap data baru menunjukkan bahwa model mampu mengenali pola umum yang tidak hanya berlaku di masa lalu, tetapi juga tetap relevan untuk memprediksi data bulan berikutnya.

### 3.4. Evaluasi Model Berdasarkan Metrik

Hasil evalusi perbandingan pendekatan terbaik dalam prediksi jumlah penumpang bulanan menunjukan bahwa hasil performa model dapat dilihat pada Tabel 1.

Tabel 1. Hasil Evaluasi Model

| Model | MAE | RMSE | $R^2$ Score |
|---|---|---|---|
| XGBoost | 273.82 | 352.78 | 0.9936 |
| Random Forest | 326.43 | 434.86 | 0.9903 |
| LSTM | 3397.45 | 4558.56 | -0.0708 |
| Exponential Smoothing | 12907.95 | 15288.03 | -110.454 |
| Auto-SARIMA | 3337.98 | 4492.09 | -0.0400 |

#### 3.4.1. XGBoost dan Random Forest

Kedua model ini menunjukkan performa yang sangat tinggi. Dengan nilai $R^2$ mendekati 1, XGBoost terbukti mampu memahami kompleksitas data yang kaya fitur seperti lag dan moving average sekaligus menjadi model terbaik dalam eksperimen ini dengan nilai error yang sangat rendah dan tingkat akurasi mendekati sempurna. Random Forest juga memberikan hasil akurat, meski sedikit di bawah XGBoost[14]. Sebagaimana hasik evaluasi ditunjukan pada Gambar 2.


Gambar 2. Hasil Evaluasi XGBoost dan Random Forest

#### 3.4.2. LSTM

Sebagai model deep learning, LSTM diharapkan mampu memahami urutan waktu dengan baik. Namun hasil menunjukkan sebaliknya. Nilai $R^2$ yang negatif menunjukkan bahwa model justru gagal memprediksi data secara akurat. Hal ini diduga karena: Ukuran dataset tidak mencukupi untuk LSTM, Parameter model belum optimal, Fitur yang digunakan kurang efektif dalam format sekuensial[15]. Sebagaimana terlihat pada gambar 3.


Gambar 3. Hasil Evaluasi LSTM

### 3.4.3. Holt-Winters dan Auto-SARIMA

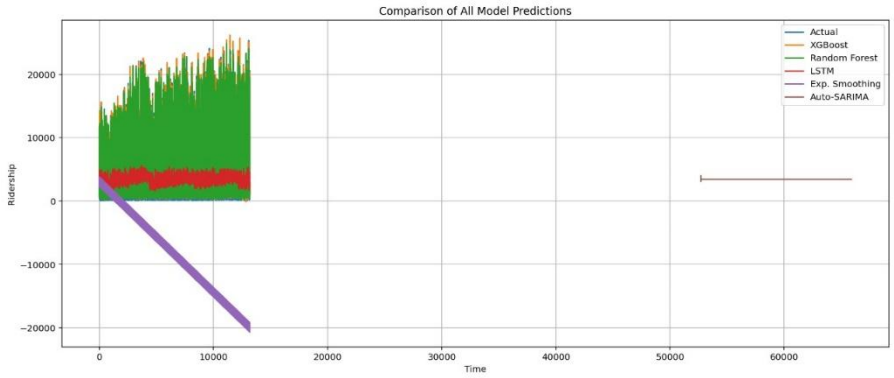
Kedua model ini menunjukkan performa yang jauh di bawah model machine learning. Exponential Smoothing bahkan memunculkan nilai prediksi negatif yang secara logika tidak mungkin (jumlah penumpang tidak bisa negatif). Ini menunjukkan bahwa model statistik klasik kurang mampu menangkap kompleksitas dan variasi pola penumpang pada dataset ini. Sebagaimana ditunjukan pada Gambar 4 dan Gambar 5.


Gambar 4. Hasil Evaluasi Holt-Winters(Exponential Smoothing)


Gambar 5. Hasil Evaluasi Auto-SARIMA

### 3.5. Visualisasi Hasil Perbandingan

Visualisasi prediksi menunjukkan bahwa kurva XGBoost dan Random Forest sangat dekat dengan data aktual, mampu mengikuti pola naik-turun jumlah penumpang dari waktu ke waktu secara konsisten. Sebaliknya, kurva prediksi dari model LSTM dan Auto-SARIMA terlihat tidak stabil dan menyimpang dari tren aktual. Sementara itu, kurva model Exponential Smoothing menunjukkan penurunan ekstrem hingga menghasilkan nilai negatif, yang menandakan adanya kesalahan besar dalam proses prediksi. Gambar hasil visualisasi yang ditampilkan pada Gambar 6 memperkuat hasil metrik evaluasi dan menunjukkan secara intuitif model mana yang layak diterapkan dalam prediksi nyata.




Gambar 6. Visualisasi Hasil Perbandingan

## 4. KESIMPULAN

Penelitian ini bertujuan untuk membandingkan kinerja tiga algoritma prediksi, yaitu XGBoost, Random Forest, dan LSTM dalam memprediksi jumlah penumpang bus bulanan menggunakan data LACMTA. Sistem yang diusulkan dibangun melalui tahapan pengumpulan data, preprocessing, pembentukan fitur historis (lag dan moving average), pelatihan model, serta evaluasi berbasis metrik regresi.

Hasil evaluasi menunjukkan bahwa model XGBoost memberikan performa terbaik dengan nilai RMSE terendah sebesar 352.78 dan $R^2$ tertinggi sebesar 0.9936, diikuti oleh Random Forest yang juga cukup akurat. Sebaliknya, model LSTM dan dua model statistik konvensional (Exponential Smoothing dan Auto-SARIMA) menunjukkan performa yang jauh lebih rendah, dengan hasil prediksi yang tidak stabil dan kurang dapat diandalkan.

Secara keseluruhan, sistem prediksi berbasis machine learning, khususnya model XGBoost, terbukti efektif dalam menangani data time-series bulanan dengan pola musiman dan fluktuasi yang kompleks. Model ini layak untuk digunakan sebagai dasar dalam pengambilan keputusan manajemen transportasi, seperti penjadwalan armada dan perencanaan rute layanan bus.

Berdasarkan hasil penelitian, saran yang dapat diberikan adalah agar penelitian selanjutnya menggunakan data yang lebih rinci, seperti data harian atau per jam, agar model seperti LSTM dapat bekerja lebih optimal. Selain itu, penambahan fitur eksternal seperti cuaca atau hari libur juga disarankan untuk meningkatkan akurasi prediksi. Mengingat performa XGBoost dan Random Forest sangat baik, kedua model ini layak dikembangkan lebih lanjut ke dalam sistem prediksi jumlah penumpang secara real-time untuk membantu perencanaan transportasi yang lebih efisien.

## REFERENSI


[1] F. Rowe, M. Mahony, and S. Tao, "Assessing Machine Learning Algorithms for Near-Real Time Bus Ridership Prediction During Extreme Weather," pp. 1–35, 2022.

[2] S. V. Ramana, "HOURLY BUS PASSENGER DEMAND PREDICTION".

[3] I. N. Rahma and T. Setiadi, "Penerapan Data Mining Untuk Memprediksi Jumlah Penumpang Bus Trans Jogja Menggunakan Time Series Data," *J. Sarj. Tek. Inform.*, vol. 2, no. 3, pp. 161–171, 2014.

[4] A. Yaqin, A. S. Budi, and P. H. Susilo, "Sistem Prediksi Jumlah Penumpang Di Bandar Udara Juanda Surabaya Dengan Metode Double Exponential Smooting," *Joutica*, vol. 7, no. 1, p. 546, 2022, doi: 10.30736/jti.v7i1.801.

[5] K. Alfikrizal, S. Defit, and Y. Yunus, "Simulasi Monte Carlo dalam Prediksi Jumlah Penumpang Angkutan Massal Bus Rapid Transit Kota Padang," *J. Inform. Ekon. Bisnis*, vol. 3, pp. 78–83, 2020, doi: 10.37034/infeb.v3i2.72.

[6] J. Siswanto, D. Manongga, I. Sembiring, and S. Wijono, "Deep Learning Based LSTM Model for Predicting the Number of Passengers for Public Transport Bus Operators," *J. Online Inform.*, vol. 9, no. 1, pp. 18–28, 2024, doi: 10.15575/join.v9i1.1245.

[7] E. Makmur and D. A. L. Sari, "Public Transportation Line Passenger Prediction Using Mamdani Method of Fuzzy Logic to Foresee Holiday Passenger Surge," *Log. J. Ranc. Bangun dan Teknol.*, vol. 23, no. 3, pp. 188–193, 2023, doi: 10.31940/logic.v23i3.188-193.

[8] M. Alkaff, A. Baskara, and A. Ainiyyah, "Penerapan Metode XGBoost Untuk Memprediksi Jumlah Kejadian Kecelakaan Lalu Lintas di Kota Banjarmasin," *Gener. J.*, vol. 7, no. 1, pp. 61–69, 2023, doi: 10.29407/gj.v7i1.19807.

[9] Y. M. Nurak *et al.*, "METODE RANDOM FOREST DAN XGBOOST," vol. 9, no. 4, pp. 5796–5802, 2025.

[10] J. Siswanto *et al.*, "Predicting the Number of Passengers in Public Transportation Areas Using the Deep Learning Model," vol. 15, no. 3, pp. 173–185, 2024.

[11] N. Maulidah, "Prediksi Peningkatan Jumlah Nasabah Deposito Berjangka Menggunakan Algoritma KNN, Decision Tree, Random Forest Dan Xgboost," *InComTech J. Telekomun. dan Komput.*, vol. 13, no. 2, p. 90, 2023, doi: 10.22441/incomtech.v13i2.16921.

[12] T. Nurholipah, R. Kurniawan, and Y. A. Wijaya, "Evaluasi Performa Model Regresi Linear Dengan Rmse Pada Jumlah Penumpang Bus Transjakarta," *JIKA (Jurnal Inform.*, vol. 8, no. 2, p. 180, 2024, doi: 10.31000/jika.v8i2.10405.

[13] A. Lisanthoni, F. I. Sari, E. L. Gunawan, and C. A. Adhigiadany, "Model Prediksi Kepadatan Lalu Lintas: Perbandingan Algoritma Random Forest dan XGBoost," *Pros. Semin. Nas. Sains Data*, vol. 3, no. 1, pp. 296–303, 2023, doi: 10.33005/senada.v3i1.126.

[14] M. Garma, A. Rianto, A. L. Prasasti, and A. Novianty, "Implementasi Model XGBoost untuk Prediksi Jumlah Transaksi dan Total Pendapatan di Jaringan Restoran CV Balibul," vol. 2, no. 2, pp. 43–46, 2024.

[15] G. Y. Gustiegan and S. Wulandari, "JURNAL LOCUS : Penelitian & Pengabdian Optimasi Algoritma Long Short-Term Memory dengan Adaptive Moment Estimation untuk Estimasi Jumlah Penumpang Bus Transjakarta," vol. 4, no. 4, pp. 1433–1444, 2025, doi: 10.58344/locus.v4i4.3997.