

Perancangan Pipeline ETL dan Peramalan Data NYC *Green Taxi Trip* Menggunakan *Long Short-Term Memory (LSTM)*

I Made Panji Adnyana Putra¹, I Wayan Sudiarsa², I Made Adi Sanjana Putra³,
I Gede Budhi Ari Saputra⁴, I Made Mandala Putra⁵

^{1,2,3,4,5}Institut Bisnis dan Teknologi Indonesia

E-mail: sudiarsa@instiki.ac.id

Article Info

Article history:

Received December 02, 2025

Revised December 16, 2025

Accepted December 27, 2025

Keywords:

Data Engineering, ETL, LSTM,
Time Series, Peramalan Taksi.

ABSTRACT

Taxi trip data is one of the important sources of time series data in the urban transportation system because it reflects people's mobility patterns. However, raw data on taxi trips generally still contains invalid values, anomalies, and unstructured formats that require a systematic processing process before being used for analysis and forecasting. This study aims to design an Extract, Transform, Load (ETL) pipeline and apply the Long Short-Term Memory (LSTM) algorithm in forecasting the number of NYC Green Taxi trips. The dataset used comes from Kaggle, which is NYC Green Taxi travel data for February 2016. The ETL process is carried out through the stages of data extraction, anomalous data cleaning, time format conversion, time-based feature engineering, and data aggregation into the number of trips per hour. The transformation data was then normalized using MinMaxScaler and formed into sequences using the sliding window method with a length of 24 hours before being used as an input model for the LSTM. The data is divided into 80% training data and 20% temporal test data. The LSTM model was built using two layers of LSTM with dropouts and one layer of Dense as the output, then trained for 50 epochs. Model performance evaluation was carried out using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) metrics. The test results showed that the LSTM model produced a MAE value of 318.12, an RMSE of 406.31, and a MAPE of 17.27% on the test data, which shows that the model has a good level of accuracy and generalization ability. The results of the study prove that the integration of the ETL pipeline and the LSTM model is effective in producing accurate and stable forecasting of the number of taxi trips. This approach has the potential to be used as a decision-making support system in fleet planning and data-driven urban transportation management.

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



Article Info

Article history:

Received December 02, 2025

Revised December 16, 2025

Accepted December 27, 2025

Keywords:

Data Engineering, ETL, LSTM,
Time Series, Peramalan Taksi.

ABSTRAK

Data perjalanan taksi merupakan salah satu sumber data deret waktu yang penting dalam sistem transportasi perkotaan karena mencerminkan pola mobilitas masyarakat. Namun, data mentah perjalanan taksi umumnya masih mengandung nilai tidak valid, anomali, serta format yang tidak terstruktur sehingga memerlukan proses pengolahan yang sistematis sebelum digunakan untuk analisis dan peramalan. Penelitian ini bertujuan untuk merancang pipeline Extract, Transform, Load (ETL) dan menerapkan algoritma Long Short-Term Memory (LSTM) dalam melakukan peramalan jumlah perjalanan NYC Green Taxi. Dataset yang digunakan berasal dari Kaggle, yaitu data perjalanan NYC Green Taxi bulan Februari 2016. Proses ETL dilakukan melalui tahap ekstraksi data, pembersihan data anomali, konversi format waktu, rekayasa fitur berbasis waktu, serta agregasi data menjadi jumlah perjalanan per jam. Data hasil

transformasi kemudian dinormalisasi menggunakan MinMaxScaler dan dibentuk menjadi sekuens menggunakan metode sliding window dengan panjang 24 jam sebelum digunakan sebagai input model LSTM. Data dibagi menjadi 80% data latih dan 20% data uji secara temporal. Model LSTM dibangun menggunakan dua lapisan LSTM dengan dropout dan satu lapisan Dense sebagai output, kemudian dilatih selama 50 epoch. Evaluasi kinerja model dilakukan menggunakan metrik Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan Mean Absolute Percentage Error (MAPE). Hasil pengujian menunjukkan bahwa model LSTM menghasilkan nilai MAE sebesar 318,12, RMSE sebesar 406,31, dan MAPE sebesar 17,27% pada data uji, yang menunjukkan bahwa model memiliki tingkat akurasi dan kemampuan generalisasi yang baik. Hasil penelitian membuktikan bahwa integrasi pipeline ETL dan model LSTM efektif dalam menghasilkan peramalan jumlah perjalanan taksi yang akurat dan stabil. Pendekatan ini berpotensi digunakan sebagai sistem pendukung pengambilan keputusan dalam perencanaan armada dan manajemen transportasi perkotaan berbasis data.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

I Wayan Sudiarsa

Institut Bisnis dan Teknologi Indonesia

Email: sudiarsa@instiki.ac.id

PENDAHULUAN

Perkembangan sistem transportasi perkotaan menghasilkan volume data yang sangat besar dan bersifat deret waktu (*time series*), salah satunya berasal dari data perjalanan taksi. Data perjalanan taksi tidak hanya mencerminkan pola mobilitas masyarakat, tetapi juga menjadi dasar penting dalam pengambilan keputusan operasional seperti perencanaan armada, manajemen lalu lintas, serta kebijakan transportasi berbasis data. Permasalahan ini menjadi semakin kompleks ketika volume data yang besar menimbulkan tantangan dalam pengolahan dan pemanfaatannya secara efektif.

Salah satu permasalahan utama dalam pemanfaatan data perjalanan taksi adalah kondisi data mentah yang belum siap digunakan untuk analisis maupun peramalan. Data mentah sering mengandung nilai tidak valid, anomali, serta format waktu yang tidak seragam sehingga memerlukan pengolahan data sebelum dianalisis. Dalam konteks data engineering, proses Extract, Transform, Load (ETL) merupakan pendekatan yang digunakan untuk mengekstraksi data dari sumbernya, membersihkan dan mentransformasikannya, serta memuat data ke dalam bentuk yang siap dianalisis untuk kebutuhan peramalan atau pelaporan. ETL menjadi komponen penting dalam pipeline data karena proses ini menyatukan data dari beragam sumber dan memastikan kualitas serta konsistensi data sebelum digunakan lebih lanjut dalam analitik atau pembelajaran mesin.

Selain tantangan pada tahap pengolahan data, aspek pemilihan metode peramalan yang mampu menangkap pola kompleks pada data deret waktu juga menjadi permasalahan penting. Data jumlah perjalanan taksi per jam menunjukkan adanya pola musiman, tren harian, serta fluktuasi nonlinier yang sulit dimodelkan oleh metode statistik konvensional. Model statistik klasik seperti Autoregressive Integrated Moving Average (ARIMA) sering kali kesulitan dalam

menghadapi pola nonlinier dan ketergantungan jangka panjang yang terdapat pada data berurutan .

Seiring dengan perkembangan *machine learning*, metode Long Short-Term Memory (LSTM) menjadi salah satu pendekatan yang banyak digunakan untuk peramalan deret waktu karena kemampuannya menangkap ketergantungan temporal jangka panjang dan struktur kompleks dalam data runtun waktu. LSTM adalah varian dari Recurrent Neural Network (RNN) yang dirancang untuk mengatasi permasalahan vanishing gradient pada RNN klasik sehingga dapat menyimpan informasi penting dalam jendela waktu panjang dan merepresentasikan hubungan temporal secara efektif .

Dalam domain peramalan permintaan transportasi perkotaan, terdapat penelitian yang menerapkan model LSTM untuk memprediksi permintaan taksi (*taxi demand forecasting*) dengan hasil yang lebih akurat dibandingkan metode statistik tradisional. Misalnya, studi tentang forecasting permintaan taksi di kota modern menunjukkan bahwa model LSTM dapat mempelajari pola musiman dan temporal data perjalanan kota dan memberikan prediksi yang lebih unggul dibandingkan model *baseline* lain seperti regresi atau metode statistik konvensional .

Berdasarkan permasalahan tersebut, penelitian ini memanfaatkan data perjalanan NYC Green Taxi yang diproses melalui tahapan ETL yang terstruktur menggunakan lingkungan Google Colab. Data mentah dibersihkan, dikonversi, dan diagregasi per jam sebelum digunakan sebagai input model LSTM untuk melakukan peramalan jumlah perjalanan taksi. Evaluasi kinerja model dilakukan menggunakan metrik kesalahan seperti Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan Mean Absolute Percentage Error (MAPE) untuk menilai tingkat akurasi dan kemampuan generalisasi model prediksi.

Dengan mengintegrasikan proses ETL yang sistematis dan metode peramalan berbasis LSTM, penelitian ini diharapkan mampu memberikan gambaran yang lebih akurat mengenai pola perjalanan taksi serta menghasilkan model prediksi yang dapat dijadikan sebagai dasar pendukung pengambilan keputusan dalam perencanaan transportasi berbasis data.

Kajian Teori

Penelitian ini berfokus pada peramalan jumlah perjalanan taksi berbasis data runtun waktu (*time series*), penerapan proses data engineering melalui tahapan Extract, Transform, Load (ETL), serta penggunaan algoritma Long Short-Term Memory (LSTM) sebagai metode prediksi. Jumlah perjalanan taksi per periode waktu tertentu merupakan indikator penting dalam sistem transportasi perkotaan karena merepresentasikan tingkat mobilitas masyarakat dan pola permintaan layanan transportasi. Data tersebut memiliki karakteristik deret waktu yang bersifat dinamis, tidak linier, serta dipengaruhi oleh ketergantungan temporal jangka pendek dan jangka panjang, sehingga memerlukan pendekatan analisis yang mampu menangkap pola kompleks secara efektif.

Dalam konteks transportasi perkotaan, data perjalanan taksi seperti NYC Green Taxi Trip tersusun dari jutaan catatan transaksi yang merekam waktu penjemputan, waktu pengantaran, jarak tempuh, dan jumlah penumpang. Data ini secara alami membentuk data *time series* ketika diagregasi berdasarkan interval waktu tertentu, misalnya per jam atau per hari. Peramalan jumlah perjalanan taksi bertujuan untuk memperkirakan permintaan di masa mendatang berdasarkan pola historis, sehingga dapat digunakan sebagai dasar pengambilan

keputusan dalam perencanaan armada, manajemen lalu lintas, dan optimalisasi layanan transportasi.

Data engineering memegang peranan penting dalam memastikan kualitas dan kesiapan data sebelum digunakan dalam proses pemodelan. Proses ETL merupakan pendekatan sistematis yang mencakup tahap ekstraksi data dari sumber mentah, transformasi data melalui pembersihan, normalisasi, rekayasa fitur, serta pemuatan data ke dalam format yang siap digunakan untuk analisis dan pemodelan. Pada data perjalanan taksi, proses transformasi mencakup konversi format waktu ke datetime, penghapusan data anomali seperti jarak perjalanan nol atau jumlah penumpang tidak valid, serta agregasi data menjadi jumlah perjalanan per interval waktu tertentu. Secara konseptual, pipeline ETL yang terstruktur mampu mengurangi noise, inkonsistensi, dan kesalahan data yang berpotensi menurunkan performa model prediksi, sekaligus menghasilkan data runtun waktu yang lebih stabil dan representatif.

Selain pembersihan data, proses ETL juga mencakup rekayasa fitur berbasis waktu, seperti jam dalam sehari (hour of day), hari dalam minggu (day of week), dan periode musiman lainnya. Fitur-fitur ini penting karena pola permintaan taksi sangat dipengaruhi oleh faktor waktu, misalnya perbedaan antara jam sibuk dan jam sepi atau antara hari kerja dan akhir pekan. Dengan demikian, proses ETL tidak hanya berfungsi sebagai tahap praproses data, tetapi juga sebagai fondasi konseptual yang menentukan kualitas masukan bagi model peramalan deret waktu.

Algoritma Long Short-Term Memory (LSTM) merupakan pengembangan dari Recurrent Neural Network (RNN) yang dirancang untuk mengatasi permasalahan vanishing gradient dalam pemodelan data sekuensial. LSTM memiliki mekanisme gate yang terdiri dari input gate, forget gate, dan output gate, yang memungkinkan model untuk mempertahankan informasi penting serta melupakan informasi yang tidak relevan. Mekanisme ini menjadikan LSTM mampu menangkap ketergantungan jangka panjang (long-term dependencies) pada data runtun waktu, yang sulit dimodelkan oleh metode statistik atau model konvensional.

Dalam penelitian peramalan deret waktu, LSTM banyak digunakan karena kemampuannya dalam mempelajari pola nonlinier dan fluktuasi kompleks pada data historis. Pada data jumlah perjalanan taksi, LSTM dapat memanfaatkan informasi dari beberapa jam sebelumnya untuk memprediksi jumlah perjalanan pada jam berikutnya, sehingga mampu menangkap pola musiman harian maupun tren jangka pendek. Sejumlah penelitian terdahulu menunjukkan bahwa LSTM memiliki performa yang lebih baik dibandingkan metode tradisional seperti regresi linier atau ARIMA dalam memodelkan data deret waktu yang memiliki dinamika kompleks dan ketergantungan temporal yang kuat.

Meskipun demikian, sebagian besar penelitian sebelumnya cenderung menitikberatkan pada evaluasi algoritma prediksi tanpa membahas secara mendalam peran data engineering dalam membentuk kualitas input model. Kajian teoritis ini mengarahkan penelitian pada asumsi bahwa kualitas data hasil proses ETL yang terstruktur akan berpengaruh signifikan terhadap kinerja model LSTM dalam melakukan peramalan jumlah perjalanan taksi. Oleh karena itu, integrasi antara pipeline ETL dan algoritma LSTM menjadi landasan konseptual utama penelitian ini untuk menghasilkan model prediksi jumlah perjalanan taksi yang lebih akurat, stabil, dan andal dalam konteks transportasi perkotaan berbasis data.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan desain eksperimental berbasis data runtun waktu (*time series forecasting*) untuk melakukan peramalan jumlah perjalanan NYC Green Taxi. Pendekatan ini dipilih karena data yang digunakan berupa data numerik berurutan yang memiliki pola temporal berupa tren, musiman, dan fluktuasi nonlinier. Penelitian mengintegrasikan proses data engineering melalui tahapan Extract, Transform, Load (ETL) dengan algoritma Long Short-Term Memory (LSTM) sebagai model peramalan. Seluruh proses dilakukan menggunakan bahasa pemrograman Python pada platform Google Colab dengan dukungan pustaka Pandas, NumPy, Scikit-Learn, TensorFlow/Keras, dan Matplotlib.

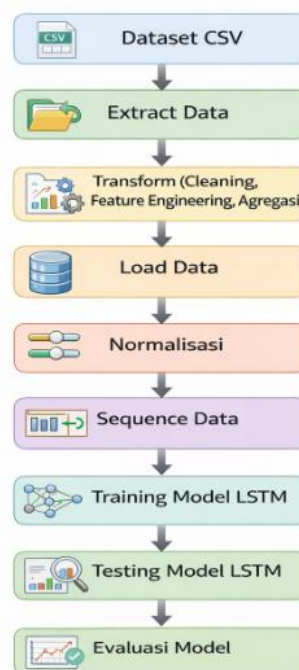
a. Sumber Data dan Lingkungan Penelitian

Data yang digunakan merupakan data sekunder yang diperoleh dari dataset *NYC Green Taxi Trip Record Data* bulan Februari 2016 dalam format CSV. Dataset ini berisi lebih dari satu juta catatan perjalanan yang mencakup informasi waktu penjemputan, waktu pengantaran, jumlah penumpang, jarak perjalanan, serta atribut pendukung lainnya. Seluruh proses penelitian dilakukan menggunakan lingkungan pemrograman Python pada platform Google Colab. Pemilihan Google Colab didasarkan pada kemudahan akses, dukungan komputasi berbasis cloud, serta ketersediaan pustaka analisis data dan deep learning yang lengkap. Pustaka utama yang digunakan meliputi: a) Pandas untuk manipulasi data; b) NumPy untuk komputasi numerik; c) Scikit-Learn untuk normalisasi dan evaluasi model; d) TensorFlow/Keras untuk pembangunan model LSTM; dan d) Matplotlib untuk visualisasi

b. Alur Metodologi Penelitian

Alur metodologi penelitian ditunjukkan pada **Gambar 1** yang menggambarkan tahapan penelitian secara berurutan mulai dari data mentah hingga evaluasi model.

Alur Metodologi Penelitian



gambar 1

Bagan Alur Metodologi Penelitian

Gambar 1 memperlihatkan bahwa penelitian dimulai dari pengambilan dataset CSV. Data kemudian melalui tahapan Extract untuk dimuat ke dalam sistem analisis. Selanjutnya data diproses pada tahap Transform untuk dibersihkan, direkayasa fiturnya, serta diagregasi. Data hasil transformasi kemudian dimuat kembali pada tahap Load sebagai data siap olah. Tahapan berikutnya adalah normalisasi data, pembentukan sekuens, pelatihan model LSTM, pengujian model, dan evaluasi performa. Alur ini menunjukkan bahwa penelitian tidak hanya berfokus pada pemodelan, tetapi juga menekankan pentingnya kualitas data sebagai fondasi utama sistem peramalan.

c. Tahapan Extract, Transform, Load (ETL)

1. Extract

Tahap extract bertujuan untuk memindahkan data mentah dari sumber eksternal ke dalam sistem analisis. Dataset perjalanan NYC Green Taxi dimuat dari berkas CSV menggunakan fungsi `read_csv()` dari pustaka Pandas. Setelah data berhasil dimuat ke dalam DataFrame, dilakukan inspeksi awal menggunakan fungsi `head()`, `info()`, dan `describe()` untuk mengetahui struktur data, tipe variabel, jumlah baris, serta karakteristik statistik dasar. Proses ini bertujuan untuk memahami kondisi awal dataset dan mengidentifikasi potensi permasalahan seperti nilai kosong, tipe data yang tidak sesuai, serta

```
import pandas as pd

# Load the CSV file into a pandas DataFrame
df = pd.read_csv('green_tripdata_2016-02v2.csv')

print("First 5 rows of the DataFrame:")
# Display the first 5 rows of the DataFrame
print(df.head())

print("\nDataFrame Information:")
# Print a concise summary of the DataFrame
df.info()

print("\nDescriptive Statistics:")
# Generate descriptive statistics of the DataFrame's numerical columns
print(df.describe())
```

gambar 2

Proses Ekstraksi Data CSV ke DataFrame

ketidakkonsistenan format.

Gambar 2 menunjukkan proses pemuatan data ke dalam DataFrame menggunakan fungsi `read_csv()`. Selanjutnya dilakukan inspeksi awal menggunakan `head()`, `info()`, dan `describe()` untuk mengetahui struktur data, tipe data, jumlah baris, serta statistik deskriptif. Hasil inspeksi menunjukkan bahwa dataset terdiri dari lebih dari satu juta baris data dengan 15 atribut utama.

2. Transform

Tahap transform bertujuan untuk membersihkan, memperbaiki, dan membentuk ulang data agar siap digunakan sebagai input model peramalan. Pada tahap ini, kolom waktu

penjemputan dan pengantaran dikonversi ke format datetime agar dapat diproses sebagai data temporal. Selanjutnya dilakukan perhitungan durasi perjalanan dalam satuan menit dengan menghitung selisih antara waktu pengantaran dan waktu penjemputan. Data kemudian disaring untuk menghilangkan nilai anomali, yaitu perjalanan dengan jumlah penumpang nol atau negatif, jarak perjalanan nol atau negatif, serta durasi perjalanan yang tidak valid. Pembersihan ini dilakukan untuk menghindari distorsi pola data yang dapat memengaruhi proses pembelajaran model.

```
df['lpep_pickup_datetime'] = pd.to_datetime(df['lpep_pickup_datetime'])
df['lpep_dropoff_datetime'] = pd.to_datetime(df['lpep_dropoff_datetime'])

# Calculate trip duration in minutes
df['Trip_duration_minutes'] = (df['lpep_dropoff_datetime'] - df['lpep_pickup_datetime']).dt.total_seconds() / 60

# Filter out anomalies
df_cleaned = df[
    (df['Passenger_count'] > 0) &
    (df['Trip_distance'] > 0) &
    (df['Trip_duration_minutes'] > 0)
].copy()

print("Shape of DataFrame after cleaning:", df_cleaned.shape)
print("First 5 rows of cleaned DataFrame with new duration column:")
print(df_cleaned[['lpep_pickup_datetime', 'lpep_dropoff_datetime', 'Trip_duration_minutes', 'Passenger_count', 'Trip_distance']].head())
```

gambar 3

Proses Pembersihan Data dan Perhitungan Durasi Perjalanan

Gambar 3 menunjukkan bahwa setelah proses pembersihan, jumlah data berkurang namun memiliki kualitas yang lebih baik dan lebih representatif terhadap kondisi nyata. Setelah proses pembersihan, data transaksi diagregasi secara per jam menggunakan metode *resampling* untuk menghasilkan variabel target berupa jumlah perjalanan taksi per jam (*trip_count*). Selain itu dilakukan ekstraksi fitur waktu seperti jam, hari dalam minggu, tanggal, dan minggu dalam tahun untuk mempertahankan informasi temporal dalam dataset.

```
df_cleaned.set_index('lpep_pickup_datetime', inplace=True)
df_hourly = df_cleaned.resample('H').size().reset_index(name='trip_count')

# Reset the index and extract time-based features
df_hourly['hour_of_day'] = df_hourly['lpep_pickup_datetime'].dt.hour
df_hourly['day_of_week'] = df_hourly['lpep_pickup_datetime'].dt.dayofweek # Monday=0, Sunday=6
df_hourly['day_of_month'] = df_hourly['lpep_pickup_datetime'].dt.day
df_hourly['week_of_year'] = df_hourly['lpep_pickup_datetime'].dt.isocalendar().week.astype(int)

print("First 5 rows of hourly aggregated DataFrame with new features:")
print(df_hourly.head())
print("\nDataFrame Information for df_hourly:")
df_hourly.info()
```

gambar 4

Proses Agregasi Data Per Jam dan Rekrayasa Fitur Waktu

Selanjutnya data dibentuk menjadi agregasi per jam (hourly aggregation) untuk memperoleh variabel target *trip_count*. Selain itu dilakukan ekstraksi fitur waktu seperti jam, hari dalam minggu, tanggal, dan minggu dalam tahun. Tahap ini menghasilkan data runtun waktu yang bersih, konsisten, dan siap digunakan untuk proses pemodelan.

3. Load

Tahap load dilakukan dengan menyimpan data hasil transformasi ke dalam struktur DataFrame agregasi per jam. Data ini menjadi dasar utama untuk tahap prapemrosesan lanjutan dan pemodelan LSTM. Tahap load memastikan bahwa hasil

ETL berada dalam format yang konsisten, stabil, dan siap diproses oleh algoritma pembelajaran mesin.

```
df_hourly = df_cleaned.resample('h').size().reset_index(name='trip_count')

# Reset the index and extract time-based features
df_hourly['hour_of_day'] = df_hourly['lpep_pickup_datetime'].dt.hour
df_hourly['day_of_week'] = df_hourly['lpep_pickup_datetime'].dt.dayofweek # Monday=0, Sunday=6
df_hourly['day_of_month'] = df_hourly['lpep_pickup_datetime'].dt.day
df_hourly['week_of_year'] = df_hourly['lpep_pickup_datetime'].dt.isocalendar().week.astype(int)

print("First 5 rows of hourly aggregated DataFrame with new features:")
print(df_hourly.head())
print("\nDataFrame Information for df_hourly:")
df_hourly.info()

First 5 rows of hourly aggregated DataFrame with new features:
   lpep_pickup_datetime  trip_count  hour_of_day  day_of_week \
0 2016-02-01 00:00:00+00:00      1266         0         0
1 2016-02-01 01:00:00+00:00       882         1         0
2 2016-02-01 02:00:00+00:00       569         2         0
3 2016-02-01 03:00:00+00:00       413         3         0
4 2016-02-01 04:00:00+00:00       453         4         0

   day_of_month  week_of_year
0             1             5
1             1             5
2             1             5
3             1             5
4             1             5

DataFrame Information for df_hourly:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 475 entries, 0 to 474
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  --
0   lpep_pickup_datetime    475 non-null    datetime64[ns, UTC]
1   trip_count              475 non-null    int64
2   hour_of_day             475 non-null    int32
3   day_of_week             475 non-null    int32
4   day_of_month            475 non-null    int32
5   week_of_year            475 non-null    int64
dtypes: datetime64[ns, UTC](1), int32(3), int64(2)
memory usage: 16.8 KB
```

gambar 5

Data hasil ETL siap digunakan untuk pemodelan

d. Prapemrosesan Data untuk LSTM

1) Normalisasi Data

```
from sklearn.preprocessing import MinMaxScaler

# 1. Select the 'trip_count' column as the target variable
data = df_hourly['trip_count'].values.reshape(-1, 1)

# 2. Initialize a MinMaxScaler and fit it to the 'trip_count' data, then transform it
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data)

print("Shape of original data:", data.shape)
print("Shape of scaled data:", scaled_data.shape)
print("First 5 scaled trip counts:", scaled_data[:5].flatten())
```

gambar 6

Proses Normalisasi Data Menggunakan MinMaxScaler

Data hasil ETL kemudian dinormalisasi menggunakan metode MinMaxScaler dengan rentang nilai antara nol hingga satu. Normalisasi dilakukan untuk menyamakan skala data sehingga tidak ada perbedaan dominasi nilai besar terhadap nilai kecil. Proses ini sangat penting karena LSTM sensitif terhadap perbedaan skala data dan dapat mengalami kesulitan konvergensi jika data tidak dinormalisasi.

2) Pembentukan Sequence Data

```
def create_sequences(data, look_back):
    X, y = [], []
    for i in range(len(data) - look_back):
        X.append(data[i:(i + look_back), 0])
        y.append(data[i + look_back, 0])
    return np.array(X), np.array(y)

# Set look_back window size
look_back = 24

# Create sequences for LSTM
import numpy as np
X, y = create_sequences(scaled_data, look_back)

# Split data into training and testing sets (80% train, 20% test)
train_size = int(len(X) * 0.8)
X_train, X_test = X[0:train_size], X[train_size:len(X)]
y_train, y_test = y[0:train_size], y[train_size:len(y)]

# Reshape input to be [samples, timesteps, features] for LSTM
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
```

gambar 7

Proses Pembentukan Sequence Data dengan Sliding Window

Data yang telah dinormalisasi kemudian dibentuk menjadi sekuens menggunakan metode sliding window dengan panjang *look-back* sebesar dua puluh empat. Setiap sekuens berisi dua puluh empat data historis sebagai input dan satu data berikutnya sebagai target. Proses ini mengubah data runtun waktu menjadi format supervised learning sehingga dapat dipelajari oleh model LSTM.

3) Pembagian Data

Data sekuens kemudian dibagi menjadi data latih sebesar delapan puluh persen dan data uji sebesar dua puluh persen secara temporal. Pembagian ini bertujuan menjaga urutan kronologis data serta mencegah kebocoran informasi masa depan ke dalam proses pelatihan model.

```
print(f"Original X shape: {X.shape}")
print(f"X_train shape: {X_train.shape}")
print(f"y_train shape: {y_train.shape}")
print(f"X_test shape: {X_test.shape}")
print(f"y_test shape: {y_test.shape}")

Original X shape: (451, 24)
X_train shape: (360, 24, 1)
y_train shape: (360,)
X_test shape: (91, 24, 1)
y_test shape: (91,)
```

gambar 8

Pembagian data latih dan data uji

4) Arsitektur dan Pelatihan Model LSTM

```
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout

# 1. Define the LSTM model architecture
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(look_back, 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(units=1))

# 2. Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Print model summary to verify architecture
print("\nModel Summary:")
model.summary()

# 3. Train the model
print("\nStarting model training...")
history = model.fit(X_train, y_train, epochs=50, batch_size=32, verbose=1)
print("Model training finished.")
```

gambar 9
Arsitektur Model LSTM

Arsitektur model LSTM yang digunakan dalam penelitian. Model terdiri dari dua lapisan LSTM dengan masing-masing 50 unit neuron, dua lapisan dropout dengan nilai 0,2 untuk mengurangi risiko overfitting, serta satu lapisan Dense sebagai output. Total parameter yang dilatih berjumlah 30.651 parameter. Arsitektur ini dirancang untuk menangkap pola temporal jangka pendek dan jangka panjang pada data runtun waktu jumlah perjalanan taksi. Struktur model yang bertingkat memungkinkan proses pembelajaran yang lebih representatif terhadap dinamika data historis.

```
Starting model training...
Epoch 1/50 12/12 4s 24ms/step - loss: 0.1838
Epoch 2/50 12/12 0s 25ms/step - loss: 0.0549
Epoch 3/50 12/12 0s 23ms/step - loss: 0.0558
Epoch 4/50 12/12 0s 23ms/step - loss: 0.0509
Epoch 5/50 12/12 0s 25ms/step - loss: 0.0478
Epoch 6/50 12/12 0s 23ms/step - loss: 0.0419
Epoch 7/50 12/12 0s 23ms/step - loss: 0.0355
Epoch 8/50 12/12 0s 24ms/step - loss: 0.0297
Epoch 9/50 12/12 0s 25ms/step - loss: 0.0280
Epoch 10/50 12/12 0s 23ms/step - loss: 0.0214
Epoch 11/50 12/12 0s 23ms/step - loss: 0.0205
Epoch 12/50 12/12 0s 25ms/step - loss: 0.0164
Epoch 13/50 12/12 0s 23ms/step - loss: 0.0177
Epoch 14/50 12/12 0s 23ms/step - loss: 0.0163
Epoch 15/50 12/12 0s 23ms/step - loss: 0.0148
Epoch 16/50 12/12 0s 26ms/step - loss: 0.0186
Epoch 17/50 12/12 0s 23ms/step - loss: 0.0150
Epoch 18/50 12/12 0s 23ms/step - loss: 0.0145
Epoch 19/50 12/12 0s 25ms/step - loss: 0.0150
Epoch 20/50 12/12 0s 24ms/step - loss: 0.0134
Epoch 21/50 12/12 0s 25ms/step - loss: 0.0173
Epoch 22/50 12/12 0s 25ms/step - loss: 0.0142
Epoch 23/50 12/12 1s 23ms/step - loss: 0.0148
Epoch 24/50 12/12 1s 40ms/step - loss: 0.0134
```

gambar 10
Proses Pelatihan Model LSTM

Model LSTM dibangun menggunakan dua lapisan LSTM dengan masing-masing lima puluh unit neuron, dropout sebesar dua puluh persen, serta satu lapisan Dense sebagai output. Model dikompilasi menggunakan optimizer Adam dan fungsi loss Mean Squared Error. Pelatihan dilakukan selama lima puluh epoch dengan batch size tiga puluh dua hingga model mencapai kondisi konvergensi.

5) Evaluasi Model

```
import matplotlib.pyplot as plt

# Plot training history
plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Training loss')
plt.title('Model Training Loss')
plt.xlabel('Epoch')
plt.ylabel('loss')
plt.legend()
plt.show()

# Make predictions
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)

# Inverse transform predictions and actual values to original scale
train_inverse = scaler.inverse_transform(train_predict)
y_train_inverse = scaler.inverse_transform(y_train.reshape(-1, 1))
test_inverse = scaler.inverse_transform(test_predict)
y_test_inverse = scaler.inverse_transform(y_test.reshape(-1, 1))

# Create empty arrays for plotting, to shift test predictions
# Explicitly set dtype to float to allow NaN values
train_plot = np.empty_like(data, dtype=float)
train_plot[:, :] = np.nan
train_plot[look_back:len(train_predict) + look_back, :] = train_predict

test_plot = np.empty_like(data, dtype=float)
test_plot[:, :] = np.nan
test_plot[look_back:len(test_predict) + look_back, :] = test_predict

# Plot actual vs. predicted values
plt.figure(figsize=(15, 7))
plt.plot(df_hourly[look_pickup_datetime:], scaler.inverse_transform(scaled_data), label='Actual Trip Count')
plt.plot(df_hourly[look_pickup_datetime:], train_plot, label='Train Predict')
plt.plot(df_hourly[look_pickup_datetime:], test_plot, label='Test Predict')
plt.title('Actual vs. Predicted Trip Counts')
plt.xlabel('Date')
plt.ylabel('Trip Count')
plt.legend()
plt.show()

print("Model evaluation and visualization completed.")
```

gambar 11

Proses Evaluasi dan Visualisasi Hasil Prediksi

Evaluasi kinerja model dilakukan dengan membandingkan nilai prediksi dan nilai aktual pada data uji menggunakan beberapa metrik evaluasi, yaitu Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan Mean Absolute Percentage Error (MAPE). Selain itu, dilakukan visualisasi perbandingan antara data aktual dan hasil prediksi untuk menganalisis kemampuan model LSTM dalam menangkap pola musiman dan tren jumlah perjalanan taksi.

6) Model Konseptual Penelitian

Secara konseptual, penelitian ini menunjukkan bahwa kualitas data hasil proses ETL menjadi fondasi utama yang memengaruhi stabilitas dan akurasi model LSTM dalam melakukan peramalan jumlah perjalanan taksi. Integrasi antara data engineering dan deep learning membentuk kerangka kerja peramalan berbasis data yang sistematis dan terstruktur.

HASIL DAN PEMBAHASAN

a. Proses Pembuatan Model LSTM

Data hasil ETL dibentuk menjadi sekuens menggunakan pendekatan sliding window dengan panjang 24 jam sebagai input untuk memprediksi jumlah perjalanan pada jam berikutnya. Data kemudian dibagi menjadi 80% data latih dan 20% data uji. Model LSTM dibangun menggunakan dua lapisan LSTM dengan masing-masing 50 unit neuron, dropout sebesar 0,2, serta satu lapisan Dense sebagai output. Model dikompilasi menggunakan optimizer Adam dan fungsi loss Mean Squared Error.

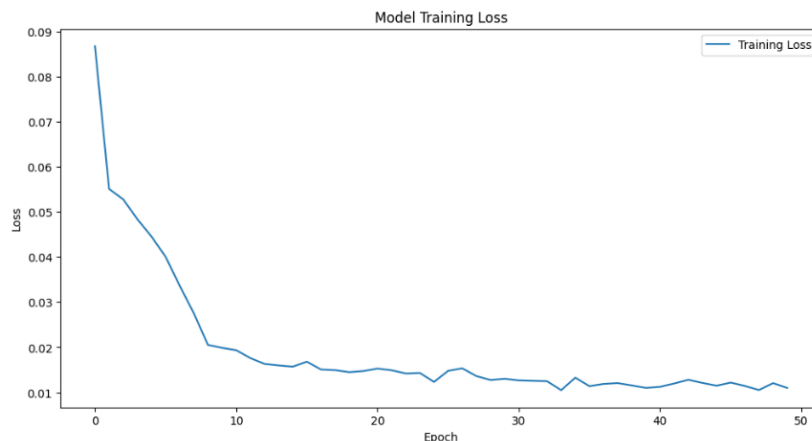
b. Kinerja Model LSTM

Hasil evaluasi menunjukkan metrik sebagai berikut:

Training	Testing
MAE = 346,29	MAE = 318,12

MSE = 209530,74	MSE = 165088,14
RMSE = 457,75	RMSE = 406,31
MAPE = 19,26%	MAPE = 17,27%

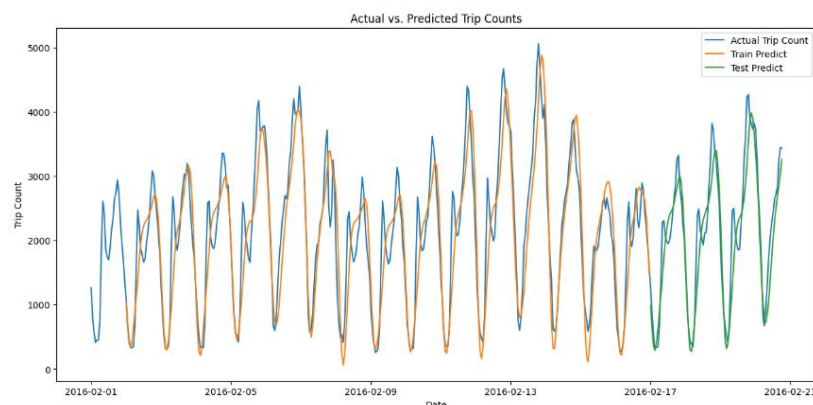
Nilai kesalahan pada data uji lebih rendah dibandingkan data latih, yang menunjukkan bahwa model memiliki kemampuan generalisasi yang baik.



Gambar 10

Gambar 10 menunjukkan grafik nilai loss selama proses pelatihan model LSTM. Terlihat bahwa nilai loss menurun signifikan pada epoch awal dan kemudian mencapai kondisi stabil mendekati epoch ke-50. Pola ini menunjukkan bahwa model berhasil mencapai konvergensi dan tidak mengalami overfitting yang signifikan.

c. Pola Prediksi dan Data Aktual



Gambar 11

Perbandingan Data Aktual dan Hasil Prediksi Jumlah Perjalanan Taksi

Gambar 11 memperlihatkan perbandingan antara data aktual dan hasil prediksi model LSTM. Kurva prediksi mengikuti pola musiman dan tren harian jumlah perjalanan taksi dengan baik. Meskipun terdapat perbedaan pada beberapa titik fluktuasi ekstrem, secara umum model mampu menangkap dinamika perubahan jumlah perjalanan secara konsisten. Garis prediksi

yang lebih halus dibandingkan data aktual menunjukkan bahwa model melakukan generalisasi untuk mereduksi noise jangka pendek sehingga menghasilkan prediksi yang lebih stabil.

d. Peran Proses ETL

Stabilitas hasil prediksi menunjukkan bahwa pipeline ETL berkontribusi besar terhadap kualitas input model. Data yang telah dibersihkan, dinormalisasi, dan diagregasi dengan baik membantu model LSTM mencapai konvergensi lebih cepat serta menghasilkan kesalahan prediksi yang lebih rendah.

Implikasi Hasil Penelitian

1) Implikasi Teoretis

Penelitian ini membuktikan bahwa kualitas data hasil proses ETL memiliki pengaruh langsung terhadap performa model deep learning pada data runtun waktu.

2) Implikasi Terapan

Model hasil penelitian dapat digunakan sebagai sistem pendukung keputusan dalam perencanaan armada, manajemen lalu lintas, dan optimalisasi layanan transportasi perkotaan berbasis data.

KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan pipeline Extract, Transform, Load (ETL) serta model Long Short-Term Memory (LSTM) untuk peramalan jumlah perjalanan taksi hijau NYC berbasis data runtun waktu. Proses ETL mampu menghasilkan data yang bersih, konsisten, dan terstruktur melalui tahapan pembersihan data, rekayasa fitur berbasis waktu, serta agregasi data per jam sehingga layak digunakan sebagai input model peramalan.

Hasil pengujian menunjukkan bahwa model LSTM memiliki kinerja yang baik dengan nilai MAE sebesar 318,12, RMSE sebesar 406,31, dan MAPE sebesar 17,27% pada data pengujian. Nilai tersebut mengindikasikan bahwa rata-rata kesalahan prediksi berada pada tingkat yang relatif rendah dan model mampu melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya. Visualisasi perbandingan antara data aktual dan hasil prediksi juga menunjukkan bahwa model mampu mengikuti pola musiman serta tren harian jumlah perjalanan taksi secara konsisten. Berdasarkan hasil tersebut, dapat disimpulkan bahwa integrasi pipeline ETL dan model LSTM efektif digunakan untuk peramalan jumlah perjalanan taksi hijau NYC dan berpotensi mendukung pengambilan keputusan pada sistem transportasi perkotaan berbasis data.

Sebagai saran, penelitian selanjutnya disarankan untuk menambahkan variabel eksternal seperti kondisi cuaca, hari libur, dan aktivitas khusus kota untuk meningkatkan akurasi prediksi. Selain itu, optimasi hiperparameter dan perbandingan dengan metode lain seperti ARIMA, Prophet, atau GRU dapat dilakukan guna memperoleh performa model yang lebih optimal.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan, bimbingan, serta kontribusi selama proses penelitian ini berlangsung. Ucapan terima

kasih secara khusus disampaikan kepada dosen pembimbing yang telah memberikan arahan, masukan, serta evaluasi yang sangat berharga sehingga penelitian ini dapat diselesaikan dengan baik dan sistematis. Penulis juga mengucapkan terima kasih kepada Institut Teknologi dan Bisnis Indonesia (INSTIKI) yang telah menyediakan fasilitas akademik, lingkungan pembelajaran, serta dukungan sarana dan prasarana yang menunjang kegiatan penelitian. Selain itu, penulis menyampaikan apresiasi kepada Kaggle sebagai penyedia platform dataset terbuka serta kepada Owen Ouyang sebagai pemilik dan penyedia dataset NYC Green Taxi Trip Record Data yang digunakan dalam penelitian ini. Ucapan terima kasih juga disampaikan kepada keluarga yang telah memberikan doa, dukungan moral, dan motivasi secara berkelanjutan, serta kepada rekan-rekan yang telah memberikan semangat, diskusi, dan bantuan selama proses penyusunan penelitian. Seluruh dukungan tersebut memiliki peran penting dalam kelancaran dan penyelesaian penelitian ini. Akhir kata, penulis berharap penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang data engineering dan analisis deret waktu, serta dapat menjadi referensi bagi penelitian selanjutnya.

DAFTAR PUSTAKA

- Paul, C. (2022). ETL in the Era of Big Data: Challenges and Solutions. ResearchGate. https://www.researchgate.net/publication/387534198_ETL_in_the_Era_of_Big_Data_Challenges_and_Solutions
- Souibgui, M., Atigui, F., Zammali, S., Cherfi, S., & Ben Yahia, S. (2019). Data quality in ETL process: A preliminary study. *Procedia Computer Science.* <https://www.sciencedirect.com/science/article/pii/S1877050919314097>
- Chanda, D. (2024). Automated ETL Pipelines for Modern Data Warehousing. *East & South Institute Journal.* <https://esj.eastasouth-institute.com/index.php/esiscs/article/view/523>
- Foidl, H. (2024). Data pipeline quality: Influencing factors, root causes of issues. *ScienceDirect.* <https://www.sciencedirect.com/science/article/pii/S0164121223002509>
- Extract, transform, load. (n.d.). In Wikipedia. https://en.wikipedia.org/wiki/Extract%2C_transform%2C_load
- Couyang. 2016 Feb NYC Green Cab Trip Record Data. Kaggle Datasets. Retrieved from <https://www.kaggle.com/datasets/couyang/2016-feb-nyc-green-cab-trip-record-data>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Retrieved from <https://arxiv.org/abs/1506.00019>
- Time Series Forecasting Using LSTM Networks: A Survey and Case Studies. (2017). Retrieved from <https://arxiv.org/abs/1703.03071>
- Comparing ARIMA and LSTM for Time Series Forecasting. (2017). Retrieved from <https://arxiv.org/abs/1709/05206>
- Qumsieh, S. (2013). Challenges of ETL in Big Data Environments. Retrieved from <http://davis.wpi.edu/~matt/courses/cs546/papers/qumsieh13.pdf>

- Souibgui, M., et al. (2019). Data quality in ETL process: A preliminary study. ScienceDirect. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1877050919314097>
- Zhou, et al. (2024). Taxi Demand Forecasting in Urban Transportation. Retrieved from <https://www.scitepress.org/Papers/2024/132054/132054.pdf>
- Time Series Forecasting using LSTM and ARIMA. (2020). Retrieved from https://thesai.org/Downloads/Volume14No1/Paper_33-Time_Series_Forecasting_using_LSTM_and_ARIMA.pdf
- Logistics Journal: Transport Analytics. (2023). Retrieved from <https://www.mdpi.com/2305-6290/7/4/86>