

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam era globalisasi dan persaingan bisnis yang semakin kompetitif, perusahaan harus mampu mengantisipasi serta merespons perubahan permintaan pasar dengan cepat dan akurat. *Forecasting* atau peramalan penjualan adalah proses penting dalam pengambilan keputusan strategis yang memungkinkan perusahaan memprediksi kondisi masa depan berdasarkan data historis. Dalam dunia bisnis yang dinamis, khususnya industri dengan permintaan fluktuatif seperti otomotif, kemampuan untuk meramalkan secara akurat sangatlah penting untuk mengelola inventaris, perencanaan produksi, dan menghindari risiko *overstock* atau kekurangan stok.

Kompleksitas pasar dan perilaku konsumen yang terus berubah menuntut perusahaan otomotif untuk beradaptasi dengan cepat. Dengan penerapan metode peramalan yang lebih canggih, perusahaan dapat meningkatkan efisiensi operasional sekaligus memperkuat daya saingnya di pasar. Penelitian ini juga dapat menjadi referensi bagi perusahaan lain dalam mengoptimalkan strategi penjualannya, khususnya melalui pendekatan berbasis teknologi.

Metode SARIMA (*Seasonal Autoregressive Integrated Moving average*) telah lama digunakan dalam analisis deret waktu untuk memodelkan data dengan pola musiman atau tren. Namun, SARIMA memiliki keterbatasan dalam menangkap pola yang kompleks dan *non-linear* (Wahyudi, 2022). Sebagai pelengkap, LSTM (*Long Short-Term Memory*), sebuah jenis jaringan saraf, sangat efektif dalam mengenali pola dari data urutan panjang, termasuk data penjualan yang dipengaruhi oleh berbagai faktor eksternal. Dengan menggabungkan kedua pendekatan ini, model *Hybrid SARIMA-LSTM* mampu memberikan hasil peramalan yang lebih akurat dan tahan terhadap perubahan dibandingkan penggunaan metode tunggal (Erlanda, 2020).

Dari sudut pandang akademis, penelitian ini berkontribusi pada pengembangan ilmu pengetahuan dalam manajemen operasi dan statistik. Dengan memadukan teori peramalan tradisional dan teknologi pembelajaran mesin modern, penelitian ini membuka peluang eksplorasi lebih lanjut terkait penerapan teknik analisis data canggih di dunia bisnis. Penelitian ini juga memperkaya literatur terkait *forecasting*, sekaligus memberikan dasar yang kuat bagi riset lanjutan di bidang tersebut (Safitri, 2017).

Sebagian besar penelitian sebelumnya tentang model *Hybrid SARIMA-LSTM* berfokus pada konteks yang berbeda, seperti Pemodelan *Hybrid ARIMA-LSTM* dalam Meramalkan Harga Saham (Triardy, 2021.) atau *A Hybrid ARIMA-LSTM Model for Short Term Vehicle Speed Prediction* (Wei Wang dkk, 2021.), yang tidak secara langsung relevan dengan industri otomotif lokal. Oleh karena itu, penelitian ini bertujuan untuk mengisi kekosongan tersebut dengan mengaplikasikan model *Hybrid SARIMA-LSTM* pada konteks penjualan kendaraan di Indonesia sekaligus menilai akurasi penggunaan model *Hybrid SARIMA-LSTM* dalam peramalan *volume* penjualan kendaraan, khususnya di PT. Kalla Toyota. Dengan mengkaji bagaimana metode ini dapat digunakan untuk memprediksi volume penjualan kendaraan secara akurat, penelitian ini diharapkan memberikan kontribusi yang signifikan, tidak hanya dalam pengembangan metode *forecasting*, tetapi juga dalam mendukung pengambilan keputusan strategis di

sektor otomotif domestik. Hal ini menjadi langkah penting untuk menjawab kebutuhan spesifik industri otomotif di Indonesia, yang memiliki karakteristik pasar yang berbeda dibandingkan konteks global.

Berdasarkan uraian tersebut maka dilakukanlah penelitian ini dengan tujuan untuk menganalisis akurasi kinerja metode *Hybrid* SARIMA-LSTM dalam meramalkan volume penjualan kendaraan di PT. Kalla Toyota. Metode digunakan untuk menilai tingkat akurasi dalam menangani karakteristik data penjualan yang memiliki unsur musiman, tren, dan pola *non-linear*. Melalui penelitian ini, diharapkan diperoleh pemahaman yang lebih komprehensif mengenai metode peramalan yang paling efektif dan aplikatif untuk digunakan dalam konteks industri otomotif, khususnya dalam mendukung pengambilan keputusan operasional dan strategi penjualan berbasis data.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini difokuskan pada dua hal utama, yaitu:

1. Bagaimana akurasi model *Hybrid* SARIMA-LSTM dalam memprediksi volume penjualan kendaraan pada PT. Kalla Toyota?
2. Bagaimana hasil *Forecast* volume penjualan kendaraan PT. Kalla Toyota selama 30 hari menggunakan metode *Hybrid* SARIMA-LSTM?

1.3 Tujuan Penelitian

Adapun Tujuan dari penelitian ini, yaitu :

1. Untuk melihat tingkat akurasi dari model *Hybrid* SARIMA-LSTM dalam memprediksi volume penjualan kendaraan pada PT. Kalla Toyota
2. Untuk menghasilkan *Forecast* volume penjualan kendaraan PT. Kalla Toyota selama 30 hari menggunakan metode *Hybrid* SARIMA-LSTM.

1.4 Landasan Teori

1.4.1 *Forecasting*

Forecasting atau peramalan adalah proses memprediksi kondisi atau kejadian di masa depan berdasarkan analisis data historis. Dalam konteks penjualan, *forecasting* digunakan untuk memperkirakan permintaan produk agar perusahaan dapat menyusun rencana anggaran penjualan yang lebih efektif dan efisien. Oleh karena itu *forecasting* harus disusun dengan teliti, cermat dan benar karena akan berpengaruh terhadap penyusunan anggaran penjualan (Sri Mulyati dkk, 2021)

1.4.2 Analisis Runtun Waktu

Analisis runtun waktu atau *time series analysis* adalah metode analisis data yang berfokus pada data yang dicatat dalam urutan waktu tertentu. Data runtun waktu digunakan untuk memahami pola masa lalu, mengenali tren atau musiman, dan memprediksi kondisi masa depan. Analisis runtun waktu digunakan secara luas dalam berbagai bidang, termasuk ekonomi, keuangan, dan bisnis, untuk membuat prediksi atau peramalan berdasarkan data historis (Heizer & Render, 2015).

Dalam analisis runtun waktu, terdapat beberapa komponen utama yang sering dianalisis:

1. Tren: Pergerakan data jangka panjang yang menunjukkan arah umum, seperti kenaikan atau penurunan penjualan kendaraan.
2. Musiman: Pola berulang dalam data yang terjadi secara periodik, misalnya peningkatan penjualan kendaraan di akhir tahun.
3. Siklus: Fluktuasi yang terjadi dalam periode yang lebih panjang, sering kali dipengaruhi oleh kondisi ekonomi.
4. Komponen Acak: Variasi data yang tidak dapat diprediksi atau dijelaskan oleh komponen lain.

1.4.3 Preprocessing Data

1.4.3.1 Missing value

Missing value adalah kekosongan nilai pada variabel tertentu dalam *dataset*. Tujuan utama penanganannya adalah menjaga kualitas dan akurasi model *machine learning*. Salah satu pendekatan umum dalam *cleaning data* untuk mengatasi *missing value* adalah menghapus entri data yang tidak lengkap atau mengisi nilai yang hilang dengan rata-rata (Ghufron dan Saepudin, 2023).

1.4.3.2 Normalisasi Data

Normalisasi bertujuan untuk memastikan bahwa kualitas data terjaga sebelum digunakan dalam model pembelajaran mesin. Proses ini penting karena normalisasi membantu menyelaraskan skala data. Nilai aktual diubah ke dalam rentang 0-1 sehingga nilai berada dalam rentang yang sama. Dengan cara ini, normalisasi dapat meminimalkan potensi bias yang mungkin muncul akibat perbedaan skala (Fitrinanda dan Djunaidy, 2022). Metode yang diterapkan dalam penelitian ini adalah *min-max normalization*, yang dapat dihitung menggunakan rumus berikut (Puteri, 2023):

$$x_{norm,i} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \quad i = 1, 2, 3, \dots, t \quad (1)$$

Dengan:

x_{norm} = Nilai normalisasi

x_{max} = Nilai maksimum data

x_{min} = Nilai minimum data

1.4.3.3 Denormalisasi Data

Setelah melakukan *testing* pada *test data*, langkah selanjutnya adalah melakukan proses denormalisasi. Proses ini bertujuan untuk mengembalikan *output* yang sebelumnya berada dalam rentang 0 hingga 1 menjadi nilai aktual dari data yang sesungguhnya. Denormalisasi sangat penting agar hasil prediksi dapat dipahami dan dibandingkan dengan nilai sebenarnya, sehingga memudahkan evaluasi kinerja model. Untuk menghitung hasil denormalisasi, kita dapat menggunakan persamaan berikut (Desril, 2024):

$$d = y \times (x_{max} - x_{min}) + x_{min} \quad (2)$$

Dengan:

d = Nilai denormalisasi

y = Nilai hasil prediksi

x_{max} = Nilai maksimum data

x_{min} = Nilai minimum data

1.4.3.3 Split Data *Train* dan Data *Test*

Pembagian data menjadi *train* dan *test* adalah langkah kritis dalam pengembangan model *machine learning*. Data *Train* berfungsi sebagai bahan pembelajaran, dan data *test* berfungsi untuk *testing* dan mengukur performa model yang didapatkan. Tujuannya adalah untuk mengevaluasi kemampuan kinerja model pada data yang belum pernah dilihat selama *training*. Rasio umum yang digunakan adalah 70:30 atau 80:20, tergantung pada ukuran dataset

1.4.4 SARIMA (*Seasonal Autoregressive Integrated Moving average*)

Model *Seasonal Autoregressive Integrated Moving average* (SARIMA) merupakan pengembangan dari model *Autoregressive Integrated Moving average* (ARIMA) yang dirancang khusus untuk menangani data deret waktu (*time series*) dengan pola musiman (*seasonality*). Model ini memadukan tiga komponen utama non-musiman, yaitu *Autoregressive* (AR), *Integrated* (I), dan *Moving Average* (MA), serta menambahkan komponen musiman untuk menghasilkan prediksi yang lebih akurat pada data dengan fluktuasi periodik. SARIMA menjadi dasar yang kuat untuk memahami pola data historis dan membuat perencanaan strategis yang lebih baik (Rachmad, 2020; Box et al., 2008).

Komponen Model Musiman SARIMA:

Autoregressive (AR): Bagian ini menunjukkan bahwa nilai data saat ini dipengaruhi oleh nilai data pada waktu sebelumnya. Orde model AR dilambangkan dengan simbol P.

$$Y_t = \Phi_1 Y_{t-s} + \Phi_2 Y_{t-2s} + \dots + \Phi_P Y_{t-PS} + a_t \quad (1)$$

Integrated (I): Komponen ini berkaitan dengan proses differencing, yaitu langkah untuk membuat data menjadi stasioner. Orde differencing dilambangkan dengan simbol D.

$$W_t = Y_t - Y_{t-s} \quad (2)$$

Moving Average (MA): Bagian ini menunjukkan bahwa nilai data saat ini juga dipengaruhi oleh kesalahan (*error*) dari model sebelumnya. Orde MA dilambangkan dengan simbol Q.

$$Y_t = \Theta_1 a_{t-s} + \Theta_2 a_{t-2s} + \dots + \Theta_Q a_{t-QS} + a_t \quad (3)$$

Model Musiman *ARIMA* secara keseluruhan dinyatakan dalam notasi *SARIMA*(p, d, q)(P, D, Q, s), di mana:

p = Orde dari proses *Autoregressive non-musiman*

d = Jumlah *differencing* yang dilakukan untuk mencapai stasioneritas non-musiman

q = Orde dari proses *Moving Average non-musiman*

P = Orde dari proses *Autoregressive musiman*

D = Jumlah *differencing* yang dilakukan untuk mencapai stasioneritas musiman

Q = Orde dari proses *Moving Average musiman*

s = Panjang siklus musiman

Rumus umum dari model musiman *SARIMA* dapat ditulis sebagai :

$$\Phi_p(B^s)\phi_p(B)(1-B)^d(1-B^s)^D Y_t = \theta_q(B^s)\theta_q(B)a_t \quad (4)$$

Dimana,

Y_t = Nilai dari deret waktu pada waktu t

ϕ_p = Koefisien *Autoregressive (AR)* non-musiman

θ_q = Koefisien *Moving Average (MA)* non-musiman

a_t = *Error term* pada waktu t

B = Operator *Lag*

$(1-B)^d$ = Operator *Differencing* non-musiman

W_t = Data deret waktu setelah ditetapkan *differencing*

p = Orde dari proses *Autoregressive* non-musiman

d = Orde *differencing* non-musiman

q = Orde dari proses *Moving average* non-musiman

Φ_p = Koefisien *Autoregressive (AR)* musiman

θ_q = Koefisien *Moving Average (MA)* musiman

$(1-B^s)^D$ = Operator *Differencing* musiman

P = Orde dari proses *Autoregressive* musiman

D = Orde *differencing* musiman

Q = Orde dari proses *Moving Average* musiman

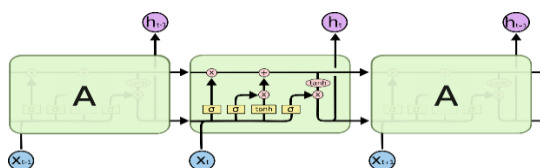
s = Periode Musiman

1.4.5 LSTM (*Long Short Term Memory*)

Long Short Term Memory atau yang biasa disebut dengan LSTM adalah metode canggih yang dikembangkan dari *Recurrent Neural Network* yang biasa disebut dengan RNN, yang pertama kali diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997. LSTM membawa inovasi penting dalam kemampuan jaringan saraf untuk menangani data berurutan dengan cara yang lebih efisien. Berbeda dengan RNN biasa, LSTM memiliki struktur *neuron* yang dilengkapi dengan beberapa gerbang (*gates*), yaitu *forget gate*, *input gate*, *new cell state candidate*, dan *output gate* yang memungkinkan setiap unit dalam jaringan untuk mengelola dan memanipulasi *memory* dengan cara yang cerdas. Gerbang-gerbang ini menentukan informasi mana yang harus disimpan dalam *memory* jangka panjang dan mana yang harus dilupakan, sehingga LSTM dapat

mengingat pola data yang relevan selama periode waktu yang lebih panjang dan lebih konsisten

Pada dasarnya, *RNN* menggunakan koneksi berulang yang menghubungkan *output* dari satu langkah waktu (*time step*) dengan *input* pada langkah waktu berikutnya. Dengan fitur ini, *RNN* dapat memodelkan data sekuensial dengan cukup baik. Namun, *RNN* menghadapi tantangan besar dalam hal pembelajaran pada urutan panjang, yang dikenal dengan masalah *vanishing gradient*. Ini terjadi karena saat *error* dihitung melalui banyak langkah waktu, gradien yang diperlukan untuk pembaruan bobot jaringan menjadi sangat kecil dan akhirnya menghilang, membuat model kesulitan untuk belajar dari informasi yang ada jauh di masa lalu. LSTM hadir sebagai solusi untuk masalah ini dengan memperkenalkan *cell gate* dan unit-unit gerbang yang mengontrol aliran informasi dalam jaringan. Dengan mekanisme ini, LSTM mampu mengatasi *vanishing gradient* dan menjaga *memory* jangka panjang lebih efisien, yang menjadikannya sangat efektif untuk memprediksi pola yang berjangka panjang atau dalam konteks data sekuensial yang kompleks (Fitrianda dkk., 2022).



Gambar 1 Arsitektur LSTM

Sumber: (Christoper, 2015)

Model LSTM dirancang untuk secara cerdas menyaring dan memanipulasi informasi melalui struktur gerbang yang sangat terorganisir, yang bertujuan untuk menjaga dan memperbarui *memory status* dalam *memory cell*. Dalam arsitekturnya, LSTM mengandalkan tiga jenis gerbang utama: *input gate*, *forget gate*, dan *output gate*. Masing-masing gerbang ini memiliki peran krusial dalam mengatur aliran informasi di dalam jaringan, memungkinkan model untuk memutuskan informasi mana yang perlu disimpan untuk jangka panjang, mana yang harus dilupakan, dan mana yang harus diteruskan sebagai *output*.

Pada setiap *memory cell*, terdapat tiga lapisan *sigmoid* yang bekerja untuk mengatur kontrol terhadap *memory* tersebut, serta satu lapisan *tanh* yang berfungsi untuk menghasilkan nilai *memory* yang baru atau diperbarui. Lapisan-lapisan *sigmoid* mengontrol keputusan-keputusan penting seperti seberapa besar pengaruh informasi baru yang harus ditambahkan ke *memory* (*input gate*), seberapa besar bagian dari *memory* yang harus dihapus (*forget gate*), dan seberapa banyak dari informasi yang ada yang akan diteruskan sebagai *output* (*output gate*). Sedangkan lapisan *tanh* bertanggung jawab untuk menghasilkan dan membatasi nilai baru yang akan ditambahkan ke *memory cell*, memastikan bahwa nilai yang disimpan berada dalam rentang yang sesuai untuk mempertahankan stabilitas numerik dan akurasi prediksi. Fungsi *Sigmoid* dapat dilihat dari persamaan berikut (Aulia, 2020):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

Dan fungsi $\tanh$ adalah sebagai berikut :

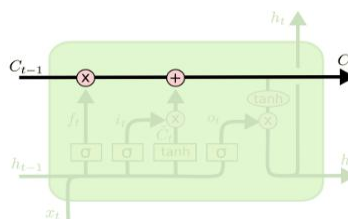
$$\tanh(x) = 2\sigma(2x) - 1 \quad (6)$$

Dimana:

σ = Fungsi aktivasi *sigmoid*

$\tanh$ = Fungsi aktivasi *hyperbolic tangent*

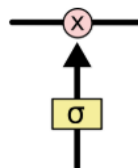
Konsep inti dari model *Long-short term memory* (LSTM) terletak pada jalur khusus yang menghubungkan *memory* dari sel sebelumnya (C_{t-1}) dengan *memory cell* yang baru (C_t). *Memory cell* ini memungkinkan informasi dari langkah waktu sebelumnya untuk dipertahankan dan ditransfer dengan sedikit modifikasi ke langkah waktu berikutnya (Desril, 2024).



Gambar 2 Memory Cell

Sumber: (Christoper, 2015)

LSTM merupakan jenis jaringan saraf yang memiliki kemampuan unik untuk mengelola informasi dalam *cell state* melalui mekanisme yang dikenal sebagai *gates*. *Gates* ini berfungsi untuk secara selektif menghapus atau menambahkan informasi, sehingga memungkinkan LSTM untuk mempertahankan informasi yang relevan dan mengabaikan yang tidak diperlukan. Setiap *gate* terdiri dari lapisan jaringan saraf *sigmoid* yang menghasilkan *output* dalam rentang 0 hingga 1. Nilai ini menentukan seberapa banyak informasi yang akan diteruskan, nilai 0 menunjukkan bahwa tidak ada informasi yang diperbolehkan untuk lewat, sedangkan nilai 1 berarti semua informasi dapat diterima.



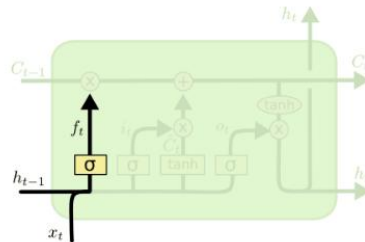
Gambar 3 Lapisan Sigmoid

Sumber: (Christoper, 2015)

Christoper, (2015) menjelaskan langkah-langkah yang akan dilalui dalam LSTM sebagai berikut:

Langkah awal dalam proses LSTM adalah menentukan informasi mana yang perlu dihapus dari *cell state*. Proses pengambilan keputusan ini dilakukan oleh lapisan *sigmoid* yang dikenal sebagai *forget gate layer*. Hal ini dapat dilihat pada h_{t-1}

dan x_t serta *output* angka antara 0 dan 1 untuk setiap nomor pada *cell state* (C_{t-1}). Angka 0 berarti "singkirkan sepenuhnya" dan angka 1 berarti "simpan sepenuhnya.". Dengan cara ini, *forget gate* berperan penting dalam menjaga relevansi data yang tersimpan dalam *cell state*, sehingga LSTM dapat berfungsi secara optimal dalam memproses urutan data yang kompleks.



Gambar 4 Alur Informasi pada *Forget gate*

Sumber: (Christoper, 2015)

Dimana f_t adalah:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (7)$$

Dengan:

f_t = *Forget gate*

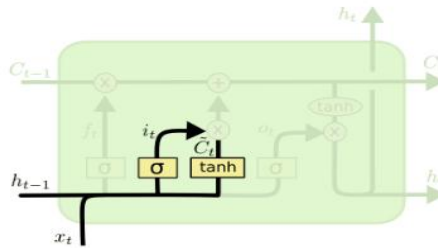
W_f = Nilai *weight* untuk *forget gate*

h_{t-1} = Nilai *output* sebelum orde ke- t

x_t = Nilai input pada orde ke- t

b_f = Nilai bias pada *forget gate*

Langkah berikutnya dalam proses LSTM adalah menentukan informasi mana yang harus disimpan dalam *cell state*. Proses ini terdiri dari dua bagian yang saling melengkapi. Pertama, lapisan *sigmoid* yang dikenal sebagai *input gate* berfungsi untuk memutuskan nilai-nilai mana yang perlu diperbarui dan dimasukkan ke dalam *cell state*. Ini memungkinkan model untuk secara selektif memilih informasi baru yang relevan untuk disimpan. Bagian kedua dari proses ini melibatkan lapisan fungsi aktivasi *tanh*, yang membuat vektor dari $\tilde{C}_t$ yang akan ditambahkan ke *cell state*. Vektor ini berfungsi sebagai representasi dari informasi baru yang akan disuntikkan ke dalam *cell state*. Pada tahap selanjutnya, kedua komponen ini *output* dari *input gate* dan vektor dari lapisan *tanh* akan dikombinasikan untuk memperbarui *cell state* secara keseluruhan, memastikan bahwa hanya informasi yang penting dan relevan yang dipertahankan.



Gambar 5 Alur Informasi yang Melewati *Input gate*
 Sumber: (Christoper, 2015)

Dimana i_t dan $\tilde{C}_t$ adalah:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (8)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (9)$$

Dengan:

i_t = *Input gate*

W_i = Nilai *weight* untuk *input gate*

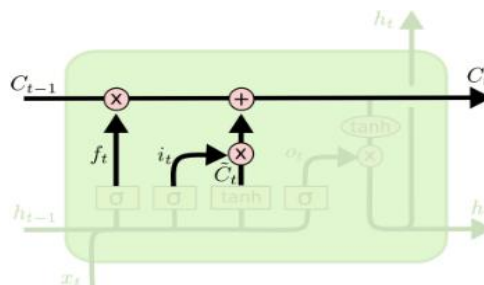
b_i = Nilai bias pada input

$\tilde{C}_t$ = Nilai baru yang dapat ditambahkan ke *cell state*

W_c = Nilai *weight* untuk *cell state*

b_c = Nilai bias pada *cell state*

Proses selanjutnya adalah memperbarui *cell state* yang lama, (C_{t-1}), ke dalam *cell state* yang baru, (C_t). Proses ini dilakukan dengan mengalikan *cell state* lama dengan *forget gate* (f_t), yang berfungsi untuk menghapus informasi yang telah ditentukan untuk dihilangkan. Dengan demikian, informasi yang tidak relevan akan disingkirkan dari *memory*. Setelah itu, nilai baru dihasilkan oleh *input gate* i_t dan vektor kandidat $\tilde{C}_t$ yang telah dibuat sebelumnya. Nilai ini kemudian ditambahkan ke dalam *cell state*, menghasilkan nilai baru yang diperbarui.



Gambar 6 Memperbarui *Cell State*
 Sumber: (Christoper, 2015)

Dimana C_t adalah

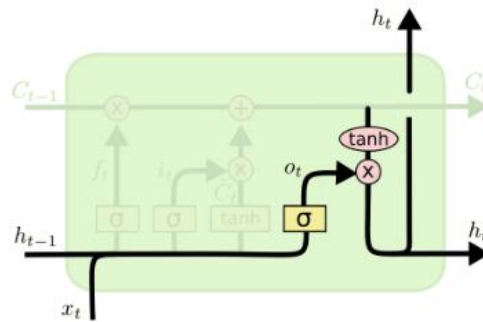
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (10)$$

Dengan:

C_t = *Cell state* order ke-t

C_{t-1} = *Cell state* satu periode sebelum orde ke-t

Langkah terakhir adalah menentukan apa yang akan menjadi *output* dari model. *Output* ini dihasilkan berdasarkan *cell state* yang telah disaring sebelumnya. Proses dimulai dengan menjalankan lapisan *sigmoid*, yang berfungsi untuk menentukan bagian mana dari *cell state* yang akan digunakan sebagai *output*. Dengan kata lain, lapisan ini melakukan seleksi terhadap informasi yang relevan untuk dikeluarkan. Setelah itu, *cell state* yang telah terpilih akan melewati fungsi aktivasi *tanh*. Fungsi ini bertujuan untuk menyesuaikan nilai *cell state* agar berada dalam rentang antara -1 dan 1, sehingga *output* dapat dinormalisasi. Terakhir, nilai hasil dari fungsi *tanh* ini akan dikalikan dengan *output* dari *sigmoid gate*, menghasilkan keluaran akhir yang siap digunakan dalam langkah-langkah berikutnya dalam model.



Gambar 7 Alur Informasi Melewati *Output gate*
Sumber: (Christoper, 2015)

Dimana o_t dan h_t adalah:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (12)$$

Dengan :

o_t = *Output gate*

W_o = Nilai *weight* untuk *output gate*

b_o = Nilai bias pada *output gate*

h_t = Nilai *output* orde ke-t

tanh = Fungsi aktivasi *hyperbolic tangent*

1.4.6 Hybrid SARIMA-LSTM

Dalam peramalan deret waktu, karakteristik model *linear* dan *non-linear* memainkan peran penting dalam menentukan kemampuan model untuk mengenali pola pada data. Model *linear*, seperti SARIMA, hanya efektif dalam menangkap pola *linear*, sementara model *non-linear*, seperti LSTM, memiliki keunggulan dalam menangani hubungan yang lebih kompleks dan *non-linear*. Meskipun beberapa penelitian (Delbert, 2024; Amara 2023; Rachmad, 2020,) telah menunjukkan bahwa penggunaan model tunggal dapat memberikan hasil yang baik untuk menangani data dengan pola tertentu, namun pendekatan ini memiliki batasan ketika diterapkan pada masalah yang lebih kompleks, seperti deret waktu dengan komponen campuran. Sebagai contoh, dalam analisis penjualan kendaraan, ada tren musiman yang bersifat *linear* serta fluktuasi mendadak dalam permintaan yang bersifat *non-linear*. Penggunaan hanya model SARIMA mungkin akan gagal menangkap fluktuasi *non-linear*, sementara LSTM mungkin mengalami *overfitting* karena terlalu mengandalkan data *training* (Zidny, 2023).

Untuk mengatasi keterbatasan ini, pendekatan *Hybrid* yang menggabungkan kedua jenis model dapat memberikan solusi yang lebih baik. Sebagai contoh, pendekatan SARIMA-LSTM mengkombinasikan model SARIMA untuk menangkap pola *linear* terlebih dahulu, lalu residual dari prediksi SARIMA digunakan sebagai input untuk model LSTM guna menangkap pola *non-linear*. Pendekatan ini terbukti lebih efektif dalam menangani data dengan karakteristik campuran dan meningkatkan akurasi prediksi. Sejumlah studi, seperti yang dilakukan oleh (Dehe Xu dkk., 2022), juga menunjukkan bahwa penggabungan model *linear* dan *non-linear* dalam peramalan deret waktu dapat menghasilkan prediksi yang lebih akurat dibandingkan dengan penggunaan model tunggal. Penggabungan model-model ini, yang dikenal dengan model *Hybrid*, muncul sebagai solusi untuk permasalahan deret waktu yang sering mengandung komponen *linear* dan *non-linear*. Dengan menggabungkan kekuatan masing-masing model, metode *Hybrid* dapat menangkap pola data yang lebih kompleks, menghasilkan akurasi prediksi yang lebih tinggi, dan mengatasi keterbatasan masing-masing model ketika diterapkan secara terpisah. Secara matematis, sebuah deret waktu Z_t dapat diekspresikan sebagai penjumlahan dari komponen *linear* L_t dan komponen *non-linear* N_t sebagaimana diformulasikan oleh Zhang (2003):

$$Z_t = L_t + N_t \quad (13)$$

Dengan:

Z_t = Data periode ke- t

L_t = Komponen *linear* periode ke- t

N_t = Komponen *non-linear* periode ke- t

t = Waktu

1.4.7 Evaluasi Model

Setelah model diterapkan, tahap evaluasi dilakukan untuk mengukur akurasi prediksi dan menilai kinerja model yang dibangun. Dalam penelitian ini, evaluasi dilakukan

menggunakan dua metrik utama, yaitu *Root Mean Square Error (RMSE)* dan *Mean Absolute Percentage Error (MAPE)*. RMSE digunakan untuk mengukur seberapa besar penyimpangan antara nilai prediksi dan nilai aktual, sementara MAPE menilai akurasi dalam bentuk persentase kesalahan relatif.

1.4.7.1 Root Mean Squared Error (RMSE)

Root Mean Square Error (RMSE) adalah salah satu metrik evaluasi yang digunakan untuk mengukur tingkat akurasi model prediksi. RMSE dihitung sebagai akar kuadrat dari rata-rata kuadrat selisih antara nilai prediksi dan nilai sebenarnya. Semakin rendah nilai RMSE, semakin rendah tingkat kesalahan prediksi. Kelebihan utama RMSE adalah memberikan penalti lebih besar pada kesalahan yang besar, sehingga sangat sensitif terhadap *outlier* atau prediksi yang jauh meleset. Selain itu, satuannya sama dengan satuan data asli, menjadikannya mudah diinterpretasikan (Suci Amalia, 2022).

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2} \quad (19)$$

Di mana:

Y_t = Nilai yang sebenarnya pada waktu t ,

$\hat{Y}_t$ = Nilai yang diprediksi oleh model pada waktu t ,

n = Jumlah data yang digunakan.

1.4.7.2 Mean Absolute Percentage Error (MAPE)

MAPE mengukur kesalahan dalam bentuk persentase, yang berguna untuk memahami akurasi relatif model prediksi terhadap nilai sebenarnya. Metrik ini memberikan gambaran jelas tentang seberapa besar kesalahan relatif prediksi model dalam persentase. Semakin kecil nilai MAPE, semakin akurat model prediksi (Suci Amalia, 2022).

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times 100\% \quad (20)$$

Di mana:

Y_t = Nilai yang sebenarnya pada waktu t ,

$\hat{Y}_t$ = Nilai yang diprediksi oleh model pada waktu t ,

n = Jumlah data yang digunakan.

Penggunaan MAPE dalam evaluasi hasil prediksi dapat membantu menghindari bias pengukuran akurasi yang terlalu tergantung pada besarnya nilai aktual, sehingga memberikan perspektif akurasi yang lebih netral. Kriteria untuk menilai hasil MAPE dapat dilihat pada tabel berikut:

Tabel 1. Kriteria Nilai MAPE

Nilai(%)	Kriteria
$x < 10\%$	Sangat Baik
$10\% \leq x < 20\%$	Baik
$20\% \leq x < 50\%$	Cukup
$x \geq 50\%$	Buruk

BAB II

METODOLOGI PENELITIAN

2.1 Pendekatan dan Jenis Penelitian

Penelitian ini menggunakan data kuantitatif berupa volume penjualan kendaraan PT. Kalla Toyota pada periode 2021-2024. Data tersebut termasuk data primer yang diperoleh langsung dari pihak PT. Kalla Toyota, khususnya melalui laporan penjualan resmi perusahaan yang digunakan oleh perusahaan.

2.2 Waktu dan Tempat Penelitian

Waktu dilaksanakannya penelitian ini mulai dari Maret 2025. Tempat penelitian yaitu di Laboratorium Big Data Prodi Ilmu Aktuaria Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

2.3 Objek Penelitian

Adapun objek dari penelitian ini adalah Volume Penjualan Kendaraan PT. Kalla Toyota

2.4 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data kuantitatif, dengan data yang diambil berupa data Volume Penjualan kendaraan PT. Kalla Toyota selama 4 tahun, periode 01 Januari 2021 sampai dengan 30 November 2024. Data tersebut termasuk data primer yang diperoleh langsung dari pihak PT. Kalla Toyota, khususnya melalui laporan penjualan resmi perusahaan yang digunakan oleh perusahaan. Alat bantu penelitian yang digunakan adalah Google Colaboratory dan Kaggle.

2.5 Metode Pengumpulan Data

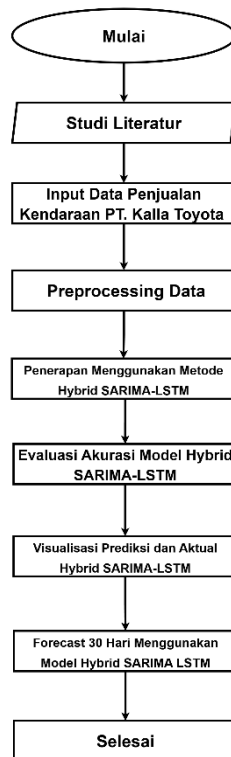
Metode pengumpulan data dalam penelitian ini dilakukan melalui studi literatur, yaitu dengan menghimpun berbagai referensi dari sumber-sumber ilmiah seperti buku, skripsi, jurnal, dan artikel penelitian terdahulu yang relevan. Literatur yang dikaji secara khusus berkaitan dengan penerapan dan pengembangan model prediktif berbasis algoritma *Hybrid* SARIMA-LSTM dalam konteks peramalan deret waktu.

2.6 Metode Analisis Data

Adapun tahapan analisis data yang terdapat dalam penelitian ini adalah sebagai berikut:

1. Mengidentifikasi masalah.
2. Melakukan Studi Literatur.
3. Mengumpulkan Data penjualan kendaraan PT. Kalla Toyota periode Januari 2021-Desember 2024.
4. Melakukan *preprocessing* data.
5. Menerapkan metode *Hybrid* SARIMA-LSTM terhadap data Volume Penjualan Kendaraan
6. Mengevaluasi kinerja metode *Hybrid* SARIMA-LSTM.
7. Menampilkan visualisasi hasil prediksi dan aktual model *Hybrid* SARIMA LSTM.
8. Melakukan *Forecasting* 30 Hari menggunakan metode terbaik.
9. Selesai.

2.2 Alur Kerja



Gambar 8 Alur Kerja