

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Penelitian yang dilakukan dengan menggunakan metode LSTM sudah banyak dilakukan. Banyak diantaranya yang menghasilkan model LSTM dengan nilai error yang rendah dari data uji yang diberikan. Dari penelitian yang sebelumnya sudah dilakukan menunjukkan bahwa menggunakan model LSTM dapat memprediksi data runtun waktu atau *Time Series Forecasting* seperti harga saham, harga komoditas, serta data lainya yang termasuk dalam data *time series*.

Pertama, penelitian yang dilakukan oleh Gunawan Budiprasetyo, Mamluatul Hani'ah , Darin Zahira Aflah berjudul “Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM)”. Tujuan dari penelitian ini yaitu memprediksi harga saham syariah dengan mengimplementasikan metode *Long-short Term Memory*. Penelitian ini memanfaatkan data harga penutupan saham harian dari lima emiten syariah, yaitu PT Aneka Tambang Tbk (ANTM), Erajaya Swasembada Tbk (ERAA), Kalbe Farma Tbk (KLBF), Semen Indonesia (Persero) Tbk (SMGR), dan Wijaya Karya (Persero) Tbk (WIKA). Data ini diperoleh dari situs *Yahoo Finance* (<https://finance.yahoo.com>). Hasil penelitian menunjukkan bahwa model *Long Short Term Memory* yang dibuat memiliki performa yang sangat baik ditandai dengan nilai MAPE yang rendah [13].

Kedua, Penelitian yang dilakukan oleh Lailan Sahrina Hasibuan dan Yanda Novialdi berjudul “Prediksi Harga Minyak Goreng Curah dan Kemasan Menggunakan Algoritme LongShort-Term Memory (LSTM)” pada tahun 2022. Penelitian ini bertujuan untuk membuyat model LSTM yang dapat memprediksi harga minyak curah dan minyak kemasan. Hasil penelitian ini menunjukkan bahwa model LSTM yang dibangun memiliki performa yang baik dengan ditandai skor NRMSE (*Normalized Root Mean Squared Error*) yang kecil yaitu 0.019 untuk minyak goreng curah dan 0.039 untuk minyak goreng kemasan [14].

Ketiga, Penelitian yang dilakukan oleh Anisa Oktaviani dan Hustinawati berjudul “Prediksi Rata-Rata Zat Berbahaya Di DKI Jakarta Berdasarkan Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory” yang dilakukan pada tahun 2021. Penelitian ini bertujuan untuk membangun model LSTM yang dapat memprediksi Indeks Standar Pencemaran Udara (ISPU). Hasil penelitian ini model LSTM memiliki performa yang cukup baik dalam memprediksi ISPU dengan skor MAPE 12.28 persen yang termasuk kedalam kategori akurat [15].

Keempat, Penelitian yang berjudul “Prediksi Harga Komoditas Pangan Menggunakan Algoritma Long Short-Term Memory (LSTM)” dilakukan oleh Rizki Mugi Setia Adi dan Sudioanto pada tahun 2022. Penelitian ini memiliki permasalahan Praktik tengkulak yang memonopoli harga komoditas yang merugikan petani, karena mereka terpaksa menjual hasil panennya dengan harga rendah. Adapun tujuan penelitian ini adalah untuk membuat model LSTM yang dapat memprediksi harga komoditas pangan di Pasar Wage dan Pasar Manis, hasil penelitian ini memiliki hasil prediksi error terendah dengan skor RMSE 79.17% [16].

Kelima, Penelitian yang berjudul “Peramalan Harga Saham Pertambangan Pada Bursa Efek Indonesia (BEI) Menggunakan Long Short Term Memory (LSTM)” dilakukan oleh Roby Julian dan Muhammad Rizky Pribadi pada tahun 2021. Penelitian ini bertujuan untuk membuat model LSTM yang dapat memprediksi harga saham pertambangan di BEI. Hasil penelitian ini menunjukkan bahwa model LSTM dengan error terendah dengan penilaian metode RMSE adalah 31.71 [17].

Ke-enam, Penelitian yang berjudul “Perbandingan Gated Recurrent Unit (Gru) Dan Algoritma Long Short Term Memory (Lstm) Linear Refression Dalam Prediksi Harga Emas Menggunakan Model Time Series” dilakukan Anggi Putri Meriani, Alam Rahmatulloh pada tahun 2024. Penelitian tersebut membandingkan metode peramalan antara *Long -short Term Memory* dan *Gated Recurrent Unit*.

Hasilnya adalah metode LSTM memiliki performa yang lebih baik daripada GRU dalam memprediksi harga emas dengan konfigurasi 100 *epoch* pelatihan [9].

Ke-tujuh, Penelitian yang berjudul “Perbandingan Model Lstm Dan Gru Untuk Memprediksi Harga Minyak Goreng DI Indonesia” yang dilakukan oleh Mochammad Agus Sholeh dan Rahmat Hidayat yang dilakukan pada tahun 2022. Penelitian ini bertujuan untuk membandingkan metode LSTM dan GRU untuk memprediksi harga minyak goreng di Indonesia dan hasilnya, LSTM memiliki performa yang lebih unggul dalam memprediksi harga minyak goreng di Indonesia [8].

Ke-delapan, Penelitian berjudul “Analisis Data Time Series Menggunakan LSTM (Long Short Term Memory) Dan ARIMA (Autocorrelation Integrated Moving Average) Dalam Bahasa Python.” yang dilakukan oleh Bayangkari Karno dan Adhitio Satyo pada tahun 2020 bertujuan untuk membandingkan performa antara ARIMA dan LSTM. Hasilnya metode LSTM memiliki kemampuan yang lebih dominan daripada ARIMA dalam memprediksi harga saham Telkom [7].

Ke-sembilan, Penelitian berjudul “Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory” yang dilakukan oleh Arfan, Adhib Lussiana ETP pada tahun 2019 membuat model LSTM untuk memprediksi harga saham di Indonesia dikarenakan harga saham cenderung fluktuatif sehingga investor tidak yakin untuk membeli saham. Hasil penelitian ini menunjukkan performa metode *Long-short Term Memory* memiliki performa yang sangat baik dalam memprediksi harga saham di Indonesia di tahun 2017 dan 2019. Dengan ditandai nilai MSE yang cukup rendah yaitu 0.0015 [18].

Ke-sepuluh, Penelitian yang berjudul ” Pendekatan Long Short-Term Memory untuk Memprediksi Kondisi Motor 10 kV pada PLTU Batubara” dilakukan oleh Muhammad Kamal Wisyaldin Gita Maya Luciana Henry Pariaman pada tahun 2020 bertujuan untuk memprediksi kondisi motor 10kV. Penelitian ini membandingkan beberapa metode peramalan yaitu RNN, ARIMA, LSTM dan GRU, dan hasilnya adalah LSTM memiliki performa yang paling dominan diantara metode lainnya [10].

Tabel 2. 1 Penelitian Sebelumnya

No	Judul	Penulis	Tahun	Studi Kasus	Metode	Hasil	Perbedaan
1	Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM).	Gunawan Budiprasetyo, Mamluatul Hani'ah, Darin Zahira Aflah	2022	Harga Saham Bank Syariah	LSTM	<p><i>MAPE:</i></p> <ol style="list-style-type: none"> 1. <i>PT. Aneka Tambang: 2,64</i> 2. <i>Erajaya Swasembada: 2,24</i> 3. <i>Kalbe Farma: 1,51</i> 4. <i>Semen Indonesia: 1,83</i> 5. <i>Wijaya Karya: 2,66</i> 	Data yang digunakan pada penelitian tersebut adalah harga saham lima perusahaan. Penelitian ini menggunakan data jumlah kendaraan Jalan Layang MBZ.
2	Prediksi Harga Minyak Goreng Curah dan Kemasan Menggunakan LSTM	Lailan Sahrina Hasibuan dan Yanda Novialdi	2022	Harga minyak curah dan minyak kemasan	LSTM	<p>skor NRMSE (<i>Normalized Root Mean Squared Error</i>) yang kecil yaitu 0.019 untuk minyak goreng curah dan 0.039 untuk minyak goreng kemasan</p>	<p>Penelitian tersebut menggunakan dataset harga minyak. Dan penilaian performa menggunakan NRMSE. Penelitian ini menggunakan data jumlah kendaraan Jalan Layang MBZ dan penilaian performa menggunakan MAPE</p>

No	Judul	Penulis	Tahun	Studi Kasus	Metode	Hasil	Perbedaan
3	Prediksi Rata-Rata Zat Berbahaya Di DKI Jakarta Berdasarkan Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory	Anisa Oktaviani dan Hustinawati	2021	Indeks Standar Pencemaran Cuaca DKI Jakarta	LSTM	Skor MAPE 12.28 persen	Penelitian tersebut menggunakan data ISPU DKI Jakarta. Penelitian ini menggunakan data jumlah kendaraan Jalan Layang MBZ
4	Prediksi Harga Komoditas Pangan Menggunakan Algoritma Long Short-Term Memory (LSTM)	Rizki Mugi Setiadi, Sudianto	2022	Harga Komoditas Pangan Pasar Wage dan Pasar Manis	LSTM	Skor RMSE 79.17 %	Penelitian tersebut menggunakan data harga komoditas pangan pasar wage dan pasar manis. Penilaian performa menggunakan RMSE. Penelitian ini menggunakan data jumlah kendaraan Jalan Layang MBZ serta penilaian performa dengan MAPE.
5	Peramalan Harga Saham Pertambangan Pada Bursa Efek Indonesia	Roby Julian dan Muhammad Rizky Pribadi	2021	Harga Saham Pertambangan di BEI.	LSTM	Skor RMSE 31.71	Penelitian tersebut menggunakan data harga saham pertambangan di BEI. Penilaian performa menggunakan RMSE.

No	Judul	Penulis	Tahun	Studi Kasus	Metode	Hasil	Perbedaan
	(BEI) Menggunakan Long-Short Term Memory (LSTM)						Penelitian ini menggunakan data jumlah kendaraan Jalan Layang MBZ serta penilaian performa dengan MAPE.
6	Perbandingan Gated Recurrent Unit (Gru) Dan Algoritma Long Short Term Memory (Lstm) Linear Refression Dalam Prediksi Harga Emas Menggunakan Model Time Series	Anggi Putri Meriani1,Alam Rahmatulloh	2024	Harga Emas	LSTM dan GRU	LSTM unggul dalam memprediksi harga emas.	Data yang digunakan adalah harga emas, sedangkan penelitian ini menggunakan data jumlah kendaraan.
7	Perbandingan Model Lstm Dan Gru Untuk Memprediksi Harga Minyak	Mochammad Agus Sholeh, Rahmat Hidayat	2022	Harga minyak goreng	LSTM dan GRU	LSTM lebih unggul dalam memprediksi harga minyak goreng.	Data yang digunakan adalah harga minyak goreng, sedangkan penelitian ini menggunakan data jumlah kendaraan.

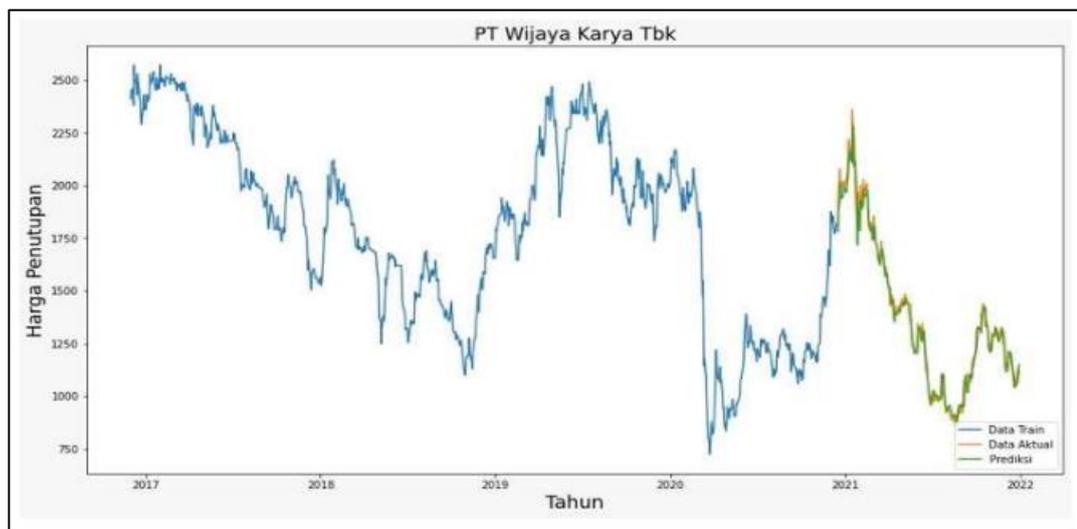
No	Judul	Penulis	Tahun	Studi Kasus	Metode	Hasil	Perbedaan
	Goreng DI Indonesia						
8	Analisis Data Time Series Menggunakan LSTM (Long Short Term Memory) Dan ARIMA (Autocorrelation Integrated Moving Average) Dalam Bahasa Python.	Bayangkari Karno, Adhitio Satyo	2020	Harga saham telkom	ARIMA dan LSTM	LSTM memiliki performa yang lebih baik dalam memprediksi data <i>time series</i>	Menggunakan data harga saham Telkom sedangkan penelitian ini menggunakan data jumlah kendaraan di jalan layang Mohammed Bin Zayed.
9	Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory	Arfan, Adhib Lussiana ETP	2019	Harga saham indonesia	LSTM	Performa LSTM dalam memprediksi harga saham cukup baik dengan ditandai nilai MSE yang minim yaitu 0.0015.	Menggunakan data harga saham Indonesia sedangkan penelitian ini menggunakan data jumlah kendaraan di jalan layang Mohammed Bin Zayed.
10	Pendekatan Long Short-Term Memory untuk Memprediksi	Muhammad Kamal Wisyaldin Gita Maya Luciana	2020	Motor 10Kv PLTU Batubara	RNN, ARIMA, GRU, LSTM	LSTM memiliki performa yang paling baik.	Menggunakan data kondisi motor, sedangkan penelitian ini menggunakan data jumlah kendaraan di jalan layang Mohammed Bin Zayed.

No	Judul	Penulis	Tahun	Studi Kasus	Metode	Hasil	Perbedaan
	Kondisi Motor 10 kV pada PLTU Batubara	Henry Pariaman					

2.2 Landasan Teori

2.2.1 Time Series Forecasting

Runtun waktu atau *Time Series* adalah data yang dikumpulkan dalam rentang waktu tertentu seperti harian, mingguan, bulanan dan tahunan yang disusun berdasarkan waktu[19]. Data runtun waktu biasanya digunakan untuk analisis prediktif yang dapat menghasilkan estimasi, pola data pada masa depan [20].

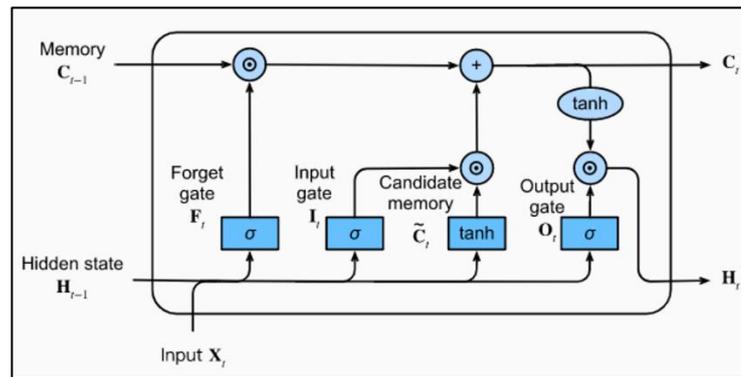


Gambar 2. 1 *Time Series Forecasting* harga saham Wijaya Karya [13]

Peramalan *Time Series* adalah suatu metode yang dapat digunakan untuk memprediksi nilai masa depan berdasarkan data historis [21]. Teknik peramalan *Time Series* mencoba mengidentifikasi pola tren, musiman dengan mempelajari data historis sehingga dapat memperkirakan nilai yang akan muncul di masa depan [22] seperti yang di tampilkan pada Gambar 2.1.

2.2.2 Long-Short Term Memory

Long Short-Term Memory adalah pengembangan dari arsitektur RNN yang diperkenalkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997 [11]. LSTM dapat mengatasi *Vanishing Gradient* dikarenakan menggunakan sistem gerbang (*gate*) yang rumit[7]. LSTM memiliki tiga gerbang utama yaitu *Forget Gate*, *Input Gate*, *Output Gate* dan komponen lainnya bernama *cell state* dan *hidden state* [11] Komponen tersebut dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Komponen LSTM [23]

Namun dalam praktiknya LSTM dapat mengerjakan beberapa tugas tergantung tugas yang diberikan, seperti penerjemahan bahasa, pembuatan teks, klasifikasi, pemberian *caption* pada gambar dan salah satunya adalah peramalan nilai di masa depan dikarenakan LSTM memiliki struktur berulang yang dapat mengingat informasi lampau yang relevan serta melupakan informasi yang sudah tidak relevan [24]. Arsitektur LSTM untuk tugas *time series forecasting* umumnya memiliki lapisan utama yaitu LSTM dan *Fully connected* yang saling berhubungan dan setiap lapisan tersebut memiliki fungsi masing-masing, lapisan tersebut bisa dirubah sesuai dengan tugas yang akan dikerjakan [13] .

1. *Forget Gate*

Forget Gate dalam arsitektur LSTM bertujuan untuk menghapus atau melupakan informasi yang tidak relevan dari sel memori yang dimana akan membantu LSTM mengelola informasi jangka panjang, gerbang ini menghasilkan ‘persentase’ informasi akan disimpan atau dilupakan [10]. *Forget Gate* di formulasikan secara matematis sebagai berikut:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

Notasi:

f_t = *Forget Gate* ke t

σ = Fungsi aktivasi sigmoid

w_f = bobot yang menghubungkan output sebelumnya

h_{t-1} = *Hidden state* sebelumnya.

x_t = *Input*

b_f = bias forget gate

Gabungan antara *hidden state* sebelumnya dan *input* dikalikan dengan matriks bobot *forget gate* ukuran $(n_{h-1}, n_{h-1} + n_{xt})$, kemudian ditambahkan dengan bias *forget gate* setelah itu akan dihitung dengan fungsi aktivasi *sigmoid* yang akan menghasilkan nilai dengan rentang 0 hingga 1, jika 0 maka informasi dari *cell state* sebelumnya akan dilupakan jika mendekati 1 maka informasi dari *cell state* sebelumnya akan tetap dijaga.

2. *Input Gate*

Input Gate dalam LSTM bertujuan untuk memutuskan menyimpan informasi baru yang akan masuk kedalam sel memori berdasarkan masukan saat ini dan *cell state* sebelumnya [25]. *Input Gate* diformulasikan secara matematis sebagai berikut:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

Notasi:

i_t = *Input Gate* ke t

σ = Fungsi aktivasi sigmoid

W_i = bobot yang menghubungkan input

h_{t-1} = *Hidden state* sebelumnya

X_i = *Input*

b_i = bias input

Setelah penggabungan antara *hidden state* sebelumnya dan *input* selanjutnya dikalikan dengan matriks bobot *input gate* ukuran $(n_{h-1}, n_{h-1} + n_{xt})$, kemudian ditambahkan dengan bias *forget gate*, setelah itu akan dihitung dengan fungsi aktivasi *sigmoid* yang akan menghasilkan nilai dengan rentang 0 hingga 1 yang nantinya akan dikalikan dengan kandidat *cell* untuk pembaharuan *cell state* pada *timesetep* saat ini.

3. *Output Gate*

Output Gate dalam LSTM bertujuan untuk menentukan keluaran pada waktu tertentu yang dapat dijadikan *output* LSTM serta mengatur informasi mana yang akan disimpan atau diabaikan oleh sel LSTM pada saat ini [26]. *Output Gate* diformulasikan secara matematis sebagai berikut:

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.3)$$

Notasi:

o_t = *Output Gate* ke t

σ = Fungsi aktivasi sigmoid

w_o = bobot yang menghubungkan *output*

b_o = *Bias output gate*.

h_{t-1} = *Hidden state* sebelumnya

x_t = Input

Sama seperti perhitungan gerbang lainnya, *hidden state* pada *timestep* sebelumnya dan *input* akan digabungkan kemudian dikalikan dengan matriks bobot ukuran $(n_{h-1}, n_{h-1} + n_{xt})$ dan ditambahkan dengan matriks bias setelah itu dilakukan perhitungan dengan fungsi aktivasi sigmoid sehingga menghasilkan keluaran matriks dengan rentang 0 hingga 1. Keluaran dari *output gate* akan dilakukan perkalian titik dengan *cell state* pada saat proses penentuan *hidden state* di *timestep* saat ini.

4. *Cell State*

Cell State merupakan aliran utama yang berfungsi menjaga informasi jangka panjang sehingga memungkinkan model LSTM mempelajari data pada masa lampau dengan baik, *Cell State* memiliki dua perhitungan yaitu kandidat dan update *cell state*[27]. Untuk rumus pengkandidatan *cell state* dan update *cell state* di formulasikan sebagai berikut:

$$\tilde{c} = \text{Tanh}(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

Notasi:

\tilde{c} = Kandidat *Cell State*

Tanh = fungsi aktivasi *Tangent Hyperbolic*

ht-1 = Hidden state sebelumnya

X_t = *Input*

w_c = bobot yang menghubungkan *cell state*.

b_c = *Bias cell state*.

$$c = f_t \odot c_{t-1} + i_t \odot \tilde{c} \quad (2.5)$$

Notasi:

c = *Cell State*

F_t = Forget Gate

\odot = Perkalian titik

C_{t-1} = *Cell State* pada *timestep* sebelumnya

I_t = *Input Gate*

Pada saat menghitung kandidat *cell state*, keluaran dari persamaan (2.4) akan dinormalisasi dengan fungsi aktivasi *tangent hyperbolic* sehingga menjadi nilai dengan rentang -1 hingga 1. Persamaan (2.5) berfungsi untuk memperbaharui *cell state* dengan cara melakukan perkalian titik terhadap *forget gate* pada *timestep* saat ini dengan *cell state* pada *timestep* sebelumnya, setelah itu dilakukan perkalian titik antara matriks *input gate* dengan kandidat *cell state* yang pada akhirnya dijumlahkan sehingga menghasilkan *cell state* untuk *timestep* saat ini.

5. *Hidden State*

Hidden State mengandung informasi yang telah diproses dari input saat ini dan informasi pada time step saat ini atau sebelumnya yang berfungsi untuk mempengaruhi output dan dalam mempelajari data[28]. *Hidden state* diformulasikan secara matematis sebagai berikut:

$$h_t = o_t \odot \text{Tanh}(c_t) \quad (2.6)$$

Notasi:

$h_t = \text{Hidden State}$

$O_t = \text{Output Gate}$

$\text{Tanh} = \text{Fungsi aktivasi } \tan \text{ hyperbolic}$

$C_t = \text{Cell State}$

Hidden state akan menghasilkan keluaran sekaligus menetapkan nilai *hidden state* pada timestep saat ini, matriks *output gate* akan dilakukan perkalian titik dengan matriks *cell state* pada *timestep* saat ini yang dinormalisasikan dengan fungsi aktivasi *tangent hyperbolic* untuk setiap elemennya sehingga menghasilkan matriks *hidden state* untuk *timestep* saat ini.

2.2.3 Sliding Windows

Sliding Windows digunakan untuk merubah struktur data runtun waktu dengan mengelompokan data pada rentang waktu tertentu yang dapat membantu model LSTM melakukan observasi sebelum melakukan prediksi kedepan[29]. *Sliding Windows* digambarkan sebagai berikut:

Diberikan 10 data runtun waktu dan kemudian data tersebut di rubah struktur datanya menggunakan *sliding windows* dengan konfigurasi *window size* = 2.

100	150	700	200	900	300	700	600	500	200
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Gambar 2. 3 Contoh data

Sehingga struktur data tersebut berubah, dikarenakan *window size* berukuran 2 maka untuk memprediksi nilai 700 pada indeks ke tiga, *inputnya* adalah 2 data sebelum index ketiga yaitu data indeks pertama (100) dan indeks kedua (150) dan langkah ini dilakukan sampai dengan indeks terakhir.

2.2.4 Quantile Based Flooring-Capping

Quantile Based Flooring-Capping digunakan untuk menangani data outlier dengan cara merubah nilai yang melewati batas bawah (*flooring*) dan melewati batas atas (*capping*) menjadi nilai batas bawah dan batas atas yang sesuai berdasarkan quantile tertentu. Teknik ini membantu memastikan bahwa nilai-nilai

ekstrem tidak mengganggu analisis statistik dan kinerja model prediktif, sehingga meningkatkan stabilitas hasil analisis. Dengan menentukan batas bawah dan batas atas, misalnya melalui persentil ke-10 dan persentil ke-90 dari data dengan menerapkan metode ini dapat mengurangi dampak dari *outlier* yang berpotensi merusak akurasi model [30].

2.2.5 *MinMax Scaler*

MinMax Scaler digunakan untuk merubah rentang nilai menjadi 0 hingga 1 yang bertujuan untuk meningkatkan efisiensi system selama pelatihan model dikarenakan rentang nya hanya dari 0 hingga 1 [31]. *MinMax Scaler* diformulasikan secara matematis sebagai berikut:

$$X_{new} = \frac{X_i - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)} \quad (2.7)$$

Notasi:

X_{new} = Data pasca normalisasi

X_i = Data X ke-i

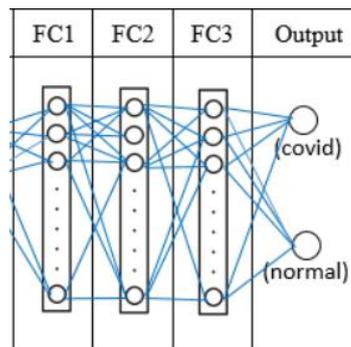
$\text{Min}(X)$ = Data paling minimum.

$\text{Max}(X)$ = Data paling maksimum.

Nilai-nilai yang ada di dalam data akan diubah sedemikian rupa sehingga nilai minimum dari fitur tersebut akan menjadi 0 dan nilai maksimumnya akan menjadi 1. Proses ini mempertahankan proporsi relatif antara nilai-nilai fitur, namun mengubah skala absolutnya sehingga rentangnya sesuai dengan yang diinginkan.

2.2.6 *Fully Connected Layer*

Lapisan *fully connected* berperan sebagai penentu nilai keluaran dari dalam hal *time series forecasting* nilai tersebut adalah hasil prediksi nilai masa depan, dalam penentuan jumlah kepadatan atau *Dense* perlu memperhatikan keseimbangan kompleksitas model dan kinerjanya agar menghasilkan keluaran yang akurat [32]. Lapisan *fully connected* biasanya ditempatkan di akhir lapisan setelah lapisan lainnya dan dapat berfungsi sebagai lapisan keluaran yang menghasilkan nilai akhir untuk memprediksi ataupun menentukan kelas [33].



Gambar 2. 4 *Fully Connected Layer*

2.2.7 Fungsi Aktivasi

Fungsi aktivasi adalah elemen dalam struktur jaringan saraf tiruan yang memperkenalkan kemampuan non-linearitas ke dalam model. Ketika sinyal-sinyal masukan dikalikan dengan bobot-bobot, fungsi aktivasi memutuskan apakah neuron-neuron tersebut harus "menyalakan" atau "mematikan" aktivitasnya berdasarkan pada nilai-nilai yang dihasilkan [34]. Dengan demikian, fungsi aktivasi bertindak sebagai filter kritis yang menentukan bagaimana informasi yang masuk akan diproses dan ditransformasikan melalui jaringan.

1. *Tangent Hyperbolic* (Tanh)

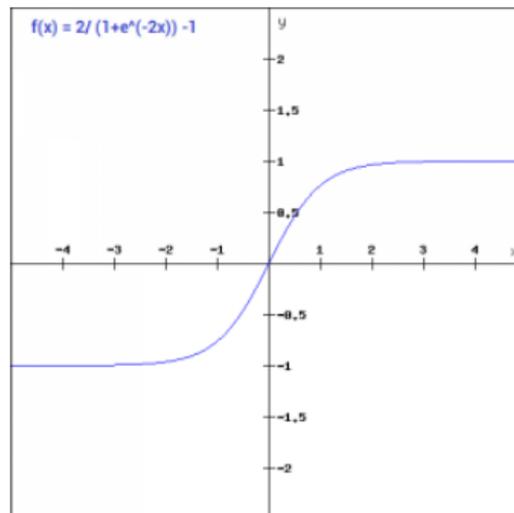
Fungsi aktivasi Tanh (*Tangent Hyperbolic*) bertujuan untuk merubah rentang nilai menjadi -1 sampai dengan 1 sehingga memiliki rentang yang lebih luas daripada fungsi aktivasi sigmoid [35] Fungsi Tanh sering digunakan di dalam lapisan-lapisan tersembunyi jaringan saraf, terutama dalam model yang memerlukan pengambilan keputusan yang lebih kompleks dan non-linear. Tanh diformulasikan pada persamaan 2.8:

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.8)$$

Notasi:

$\text{Tanh}(x)$ = *Input* fungsi aktivasi tanh

e = Konstanta *euler*



Gambar 2. 5 Grafik fungsi aktivasi *Tanh*[36]

2. *Sigmoid*

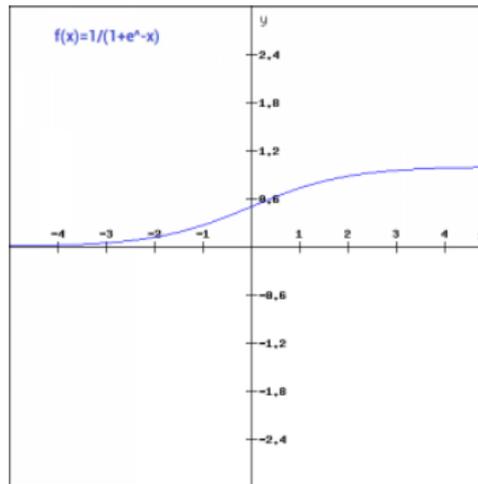
Sigmoid adalah fungsi aktivasi yang memiliki keluaran dengan rentang 0 hingga 1 [37]. Dalam konteks LSTM penggunaan fungsi aktivasi sigmoid digunakan pada seluruh gerbang yang ada di komponen LSTM seperti *forget gate*, *input gate*, *output gate* dimana semakin dekat 1 pada *output* yang dihasilkan maka keputusannya adalah informasi tersebut disimpan kedalam memori [38]. *Sigmoid* diformulasikan sebagai berikut:

$$\sigma(x) = \frac{1}{1+e^x} \quad (2.9)$$

Notasi:

$\sigma(x)$ = input untuk sigmoid

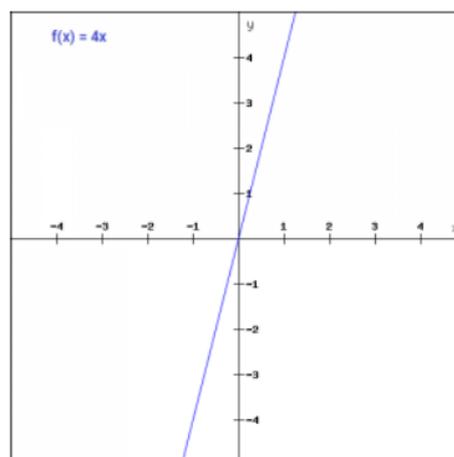
e^x = konstanta euler yang dipangkatkan dengan x



Gambar 2. 6 Grafik fungsi aktivasi *Sigmoid*[36]

3. *Linear*

Penggunaan fungsi aktivasi linear lebih cocok dalam tugas-tugas regresi yang memprediksi nilai kontinu. Fungsi aktivasi linear memungkinkan jaringan saraf untuk secara langsung mempelajari hubungan linier antara fitur-fitur masukan dan nilai-nilai target yang ingin diprediksi. Dengan demikian, model dapat dengan mudah menyesuaikan bobot-bobotnya untuk mencocokkan data pelatihan dan menghasilkan prediksi yang akurat[36]. Maka dari fungsi aktivasi linear kurang sesuai untuk tugas-tugas lainnya seperti klasifikasi yang perlu mempelajari hubungan yang non-linear antara fitur-fitur masukan dan kelas-kelas target.



Gambar 2. 7 Grafik fungsi Aktivasi Linear [36]

2.2.8 Mean Squared Error

Mean Squared Error digunakan sebagai metrik evaluasi dengan cara menghitung rata-rata kesalahan antara nilai sebenarnya dan nilai hasil diprediksi dari model prediktif yang Dimana semakin kecil nilainya semakin bagus performa model nya[39]. *MSE* dirumuskan secara matematis pada persamaan 2.10:

$$MSE = \frac{1}{n} \sum (Y_i - \hat{Y}_i)^2 \quad (2.10)$$

Notasi:

Y_i = Nilai aktual ke-i

\hat{Y}_i = Nilai prediksi ke-i

Tujuan dari mengkuadratkan pada persamaan MSE adalah untuk menonjolkan kesalahan yang besar sehingga MSE sensitif terhadap kesalahan yang besar maka dari itu MSE sering digunakan untuk pemodelan [39].

2.2.9 Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) digunakan untuk evaluasi model LSTM dalam memprediksi jumlah kendaraan dengan menghitung persentase rata-rata selisih absolut antara nilai hasil prediksi dengan nilai yang sebenarnya, MAPE dinyatakan dalam bentuk persentase, yang membuatnya mudah untuk diinterpretasikan. MAPE sering digunakan dalam konteks *time series forecasting* dan berbagai aplikasi lain di mana evaluasi keakuratan model prediksi diperlukan. Nilai MAPE yang lebih rendah menunjukkan model prediksi yang lebih akurat [40].

$$MAPE = \frac{1}{n} \sum \left| \frac{A_t - F_t}{A_t} \right| \cdot 100\%$$

Notasi:

A_t = Nilai aktual

F_t = Nilai Hasil Prediksi

Dalam menilai performa model dari perhitungan MAPE, berikut tabel acuan untuk penilaian performa model.

Tabel 2. 2 Acuan Penilaian MAPE [40]

No	Range MAPE	Keterangan
1	<10%	Performa Model Sangat Baik
2	10 – 20%	Performa Model Baik
3	20 – 50%	Layak
4	>50%	Tidak Akurat