

# Peramalan Jumlah Penumpang Kapal Laut Menggunakan Metode *Time Series Machine Learning* di PT Pelabuhan Indonesia (Persero) Regional 4 (Studi Kasus Pada Pelabuhan Soekarno Hatta Makassar)

Rismawati

Program Studi Matematika, Universitas Islam Negeri Alauddin Makassar, [rismawatia102@gmail.com](mailto:rismawatia102@gmail.com)

Wahidah Alwi\*

Program Studi Matematika, Universitas Islam Negeri Alauddin Makassar, [wahidah.alwi@uin-alauddin.ac.id](mailto:wahidah.alwi@uin-alauddin.ac.id)

Adnan Sauddin

Program Studi Matematika, Universitas Islam Negeri Alauddin Makassar, [adnan.sauddin@uin-alauddin.ac.id](mailto:adnan.sauddin@uin-alauddin.ac.id)

Adiatma

Program Studi Matematika, Universitas Islam Negeri Alauddin Makassar, [adiatma.rasyid@uin-alauddin.ac.id](mailto:adiatma.rasyid@uin-alauddin.ac.id)

\*Corresponding Author

---

**ABSTRAK.** Penelitian ini membahas tentang Peramalan Jumlah Penumpang Pelabuhan Soekarno Hatta Makassar pada PT Pelabuhan Indonesia (Persero) Regional 4. Jumlah penumpang Pelabuhan Soekarno Hatta Makassar sering mengalami lonjakan jumlah penumpang ketika musim libur, hari raya maupun akhir tahun. Metode yang digunakan dalam menyelesaikan masalah peramalan jumlah penumpang yaitu menggunakan *Long Short Term Memory*. Penelitian ini bertujuan untuk mengetahui model *Long Short Term Memory* yang kemudian digunakan untuk meramalkan jumlah penumpang keberangkatan pada kapal PELNI, serta mengetahui tingkat akurasi hasil prediksi menggunakan metode *Long Short Term Memory*. Berdasarkan hasil penelitian model LSTM yang diperoleh yaitu model sequential 3 layer terdiri dari dua layer LSTM dan satu layer dense dengan jumlah neuron hidden layer yaitu 10, jumlah batch size yaitu 1 dan jumlah epoch yaitu 100, serta menggunakan optimasi Adam dengan learning rate 0,01 di dapatkan Hasil peramalan dari Januari 2023 sebanyak 33.143 penumpang hingga Desember 2023 sebanyak 25.151 penumpang. Tingkat akurasi nilai MAPE yaitu sebesar 16,85702941%. Berdasarkan aturan range nilai MAPE yang didapatkan, dapat dikatakan bahwa akurasi peramalan yang dilakukan termasuk kedalam kategori baik karena nilai MAPE 10-20%.

---

**Kata Kunci:** Penumpang Kapal, Peramalan, Recurrent Neural Network, LSTM

---

## 1. PENDAHULUAN

Pelabuhan Makassar atau dikenal dengan Pelabuhan Soekarno Hatta Makassar adalah pusat lalu lintas kapal terbesar di kawasan Indonesia Timur. Pelabuhan ini memiliki lalu lintas penumpang tertinggi yang terletak di kota makassar Provinsi Sulawesi Selatan. Jumlah penumpang transportasi kapal laut di PT

Pelabuhan Indonesia (Persero) IV pada Pelabuhan Soekarno Hatta Makassar mengalami fluktuasi setiap tahunnya. Peningkatan jumlah kapal juga terjadi pada hari-hari tertentu misalnya pada musim libur, hari raya maupun akhir tahun. Oleh sebab itu peramalan merupakan hal yang penting dilakukan perusahaan, karena dengan mengetahui ramalan jumlah penumpang yang akan datang, perusahaan dapat mengantisipasi kenaikan jumlah penumpang dengan menyediakan fasilitas pelayanan dan juga armada kapal demi transportasi laut.

Peramalan pada jumlah penumpang kapal laut dipengaruhi oleh momen-momen tertentu. Di mana data-data yang dikumpulkan berdasarkan urutan waktu (*time series*) seperti jam, hari, minggu, bulan, kuartal, semester dan tahun dapat dianalisis menggunakan metode runtun waktu (*time series*). Data runtun waktu dapat dijadikan dasar dalam mengambil ketentuan untuk meramalkan kejadian yang terjadi di era yang akan datang [1].

Metode *time series* (runtun waktu) sering digunakan dalam peramalan. Salah satu metode *time series* adalah *deep learning*, yang memungkinkan pembelajaran dilakukan dengan lapisan yang lebih kompleks untuk mencapai akurasi yang tinggi, serta peramalan yang lebih efisien dan akurat dibandingkan dengan metode peramalan tradisional karena mampu memodelkan data yang linear dan data *nonlinear*.

Salah satu metode *deep learning* adalah *Long-Short Term Memory* (LSTM)[2].

LSTM merupakan salah satu pengembangan *neural network* yang dapat digunakan untuk pemodelan data *time series*. Tujuan utama dari LSTM pada kasus peramalan adalah membuat prediksi yang akurat terhadap suatu variabel. Adapun kelebihan metode LSTM yaitu merupakan modifikasi struktur dari *Recurrent Neural Network* (RNN) dengan sel memori yang ditambahkan di dalam lapisan tersembunyi sehingga dapat mengontrol arus informasi dalam suatu data *time series*[3].

Beberapa penelitian sebelumnya mengenai metode *Long Short-Term Memory* yang dilakukan oleh (Ardani Mela, 2022) dalam memprediksi jumlah penumpang pesawat domestik. Hasil penelitian menunjukkan bahwa akurasi metode LSTM dengan jumlah pola *time series* sebesar 3, jumlah *neuron hidden* sebesar 5, *batch size* sebesar 64, dan *max epoch* sebesar 50 kali mendapatkan hasil RMSE 910.08 dan MAPE sebesar 10,82%. Dari nilai MAPE yang didapatkan, dapat dikatakan bahwa akurasi prediksi yang dilakukan termasuk ke dalam kategori baik karena  $MAPE > 10\%$  [4].

Penelitian terdahulu yang dilakukan oleh (Kristina Sanjaya Putri dan Siana Halim, 2020) yang menggunakan metode *Long Short-Term Memory* dalam meramalkan pergerakan mata uang dengan menggunakan Metode Arima sebagai pembanding. Penelitian tersebut menghasilkan bahwa Metode LSTM memberikan hasil peramalan dengan tingkat akurasi yang lebih baik dibandingkan metode Arima. Nilai RMSE dengan Metode LSTM diperoleh sebesar 0,0051 dan Metode ARIMA diperoleh 0,0053 [5].

Penelitian lainnya dilakukan oleh (Don Ardhito, 2021) yang menggunakan perbandingan metode ARIMA, Exponential Smoothing, Linear Regression, SVM, dan LSTM untuk memprediksi kasus Covid-19 di dunia. Hasil perbandingan metode tersebut menunjukkan bahwa metode LSTM memiliki performa yang lebih baik dengan tingkat kesalahan terkecil dibandingkan dengan metode lainnya. Nilai RMSE dari metode LSTM diperoleh sebesar 1.719.164 [6].

Melihat dari berbagai penelitian sebelumnya metode *Long Short-Term Memory* ini lebih tepat digunakan untuk meramalkan jumlah penumpang kapal laut karena memiliki tingkat akurasi dan nilai optimum dengan kinerja yang lebih. Terlebih lagi LSTM memiliki kemampuan generalisasi yang kuat serta kemampuan *learning* yang baik sehingga cocok untuk himpunan data besar dan data kecil sekaligus memiliki kelebihan dalam memproses data *nonlinear*

## 2. TINJAUAN PUSTAKA

### Analisis Deret Waktu (*Time Series*)

Data *time series* (runtun waktu), yakni jenis data yang terdiri atas variabel-variabel yang dikumpulkan menurut urutan waktu dalam suatu rentang waktu tertentu untuk suatu kategori atau individu tertentu. Jika waktu dipandang bersifat diskrit (waktu dapat dipandang bersifat kontinu), frekuensi pengumpulan selalu sama (*equidistant*). Dalam kasus diskrit, frekuensi dapat berupa detik, menit, jam, hari, minggu, bulan atau tahun dan lain-lain [7]

Berikut ini ada empat jenis pola data, yaitu:

1. Pola data horizontal muncul saat data berfluktuasi di sekitar data rata-rata atau pada saat angka konstan. Format data ini dikenal sebagai data stasioner.
2. Pola data musiman adalah pola data yang memperlihatkan perubahan yang berulang secara berturut-turut dalam data. Jika data dipengaruhi oleh faktor musiman seperti triwulan, bulanan, atau mingguan, pola data musiman akan muncul di data.
3. Pola data siklis merupakan pola data yang memperlihatkan adanya pergerakan naik turun dalam jangka waktu yang lama dari suatu kurva *trend*.
4. Pola data *trend* terjadi ketika ada kecenderungan data bergerak ke satu arah dalam jangka waktu yang lama. *Trend* naik atau turun tersebut menunjukkan arah [8].

### Peramalan (*Forecasting*)

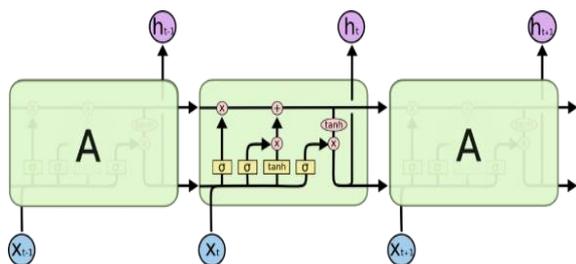
Analisis peramalan adalah proses memperkirakan sesuatu dimasa depan dengan menggunakan data dari masa lalu dan mengekstrapolasinya hingga saat ini menggunakan berbagai model matematika.

Terdapat tiga jenis peramalan berdasarkan rentang waktunya, yaitu:

1. Peramalan jangka panjang adalah proses membuat prediksi untuk 2 sampai 10 tahun ke depan.
2. Peramalan jangka menengah adalah proses membuat prediksi untuk 1 bulan sampai 2 tahun ke depan.
3. Peramalan jangka pendek adalah proses membuat prediksi untuk periode 1 sampai 5 minggu ke depan [9].

### Long Short Term-Memory (LSTM)

LSTM adalah teknik pembelajaran mendalam (*deep learning*) yang dikembangkan untuk mengatasi masalah pemrosesan RNN yaitu pemrosesan data urutan jangka panjang. Di mana RNN tidak dapat mempelajari cara menghubungkan informasi karena memori lama yang tersimpan akan semakin tidak berharga dan digantikan oleh memori baru [10].

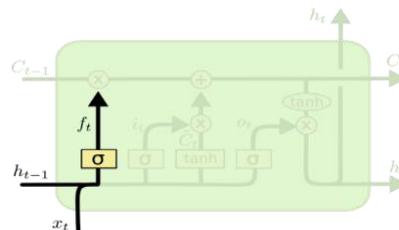


Gambar 2.1 Jaringan Long-Short Term Memory

Pada gambar 2.3, merupakan alur kerja dari *memory cells* pada setiap *neurons* LSTM. Dimana terdapat dua proses fungsi aktivasi yaitu fungsi *sigmoid* dan fungsi *tanh*. *Sigmoid layer* akan memberikan hasil menjadi bentuk *range* [0,1]. Yang di mana jika hasilnya 0 maka menunjukkan bahwa informasi akan dihentikan sedangkan jika hasilnya adalah 1 maka informasi akan diteruskan [11].

Pada arsitektur LSTM lapisan tersembunyi terdiri dari sel memori, satu sel memori tersusun atas tiga *gates*, yaitu *forget gate*, *input gate*, dan *output gate* [12]. Tahapan-tahapan pada proses penerapan LSTM adalah sebagai berikut:

1. Forget gate



Gambar 2.2 Forget Gate

*Forget gate* ( $f_t$ ), berperan untuk menentukan apakah informasi dari setiap data masukan diolah dan data mana yang dipilih untuk disimpan dalam *memory cell* atau dibuang. Data yang digunakan untuk menghitung *Forget gate* adalah data *output* sebelumnya atau  $h_t - 1$  dan data *input*  $x_t$  saat ini.

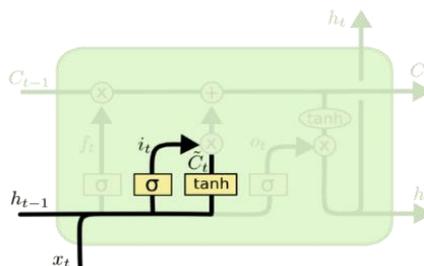
Persamaan *forget gate* diuraikan sebagai berikut.

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

Keterangan:

- $f_t$  = *forget gate*
- $\sigma$  = fungsi Aktivasi *sigmoid*
- $W_f$  = nilai *weights* 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*

2. Input Gate



Gambar 2.3 Input Gate

*Input gate* ( $i_t$ ), memiliki fungsi untuk mengambil nilai *output* sebelumnya ( $h_{t-1}$ ) dan nilai *input* baru ( $x_t$ ). Persamaan pada *gate* ini menggunakan dua fungsi aktivasi yaitu fungsi aktivasi *sigmoid* dan aktivasi *tanh* yang akan membuat vector nilai untuk disimpan pada *memory cell* sebagai nilai baru.

Persamaan *input gate* diuraikan sebagai berikut.

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

Di mana:

- $i_t$  = nilai *input gate*
- $\sigma$  = fungsi *sigmoid*

- $W_i$  = nilai *weights* pada *input gate*
- $h_{t-1}$  = nilai *output* sebelum orde ke t
- $x_t$  = nilai *input* sebelum orde ke t
- $b_i$  = nilai bias pada *input gate*

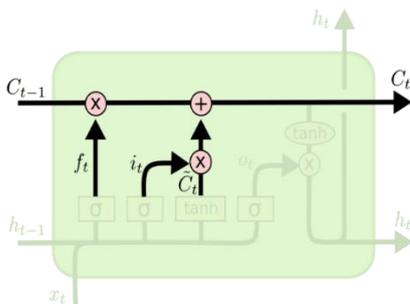
Persamaan untuk mencari nilai yang baru menggunakan fungsi aktivasi *tanh* adalah sebagai berikut.

$$\tilde{C}_t = \tanh(W_c \cdot [s_{t-1}, x_t] + b_c) \quad (2.3)$$

Di mana,

- $\tilde{C}_t$  = nilai baru yang ditambahkan ke dalam *cell state*
- tanh* = fungsi tanh (*tangen hiperbolik*)
- $W_c$  = nilai *weights* pada *cell state*
- $h_{t-1}$  = nilai *output* sebelum orde ke t
- $x_t$  = nilai *input* pada orde ke t
- $b_c$  = nilai bias pada *cell state*

Setelah itu *cell state* yang lama akan diperbarui menjadi *cell state* yang baru dengan mengalikan *state* lama dengan *forgot gate* ( $f_t$ ) untuk menghapus informasi yang telah ditentukan pada *layer forgot gate* kemudian ditambahkan dengan  $i_t \cdot \tilde{C}_t$  yang merupakan nilai baru untuk memperbarui *state*.



**Gambar 2.4** Cell State Baru

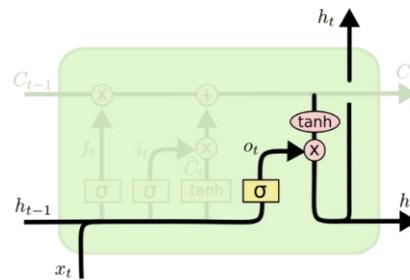
Persamaan *cell state* dinyatakan sebagai berikut.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (2.4)$$

Di mana,

- $C_t$  = nilai *cell state*
- $f_t$  = nilai *forget gate*
- $C_{t-1}$  = nilai *cell state* sebelum orde ke t
- $i_t$  = nilai *input gate*
- = nilai baru yang ditambahkan ke dalam *cell state*
- $\tilde{C}_t$

### 3. Output Gate



**Gambar 2.5** Jaringan Output Gate

*Output gate* ( $o_t$ ), ini berperan dalam menentukan nilai pada bagian *memory cell* mana yang akan dikeluarkan dengan menggunakan fungsi aktivasi *sigmoid*. Nilai pada *memory cell* ditempatkan dengan menggunakan fungsi aktivasi *tanh*. Terakhir, kedua *gates* tersebut dikalikan untuk mendapatkan nilai yang akan dikeluarkan. Persamaan *Output Gate* dinyatakan sebagai berikut.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

Di mana,

- $o_t$  = nilai *input gate*
- $\sigma$  = fungsi *sigmoid*
- $W_o$  = nilai *weights* untuk *output gate*
- $h_{t-1}$  = nilai *output* sebelum orde ke t
- $h_t$  = nilai *input* pada orde ke t
- $b_o$  = nilai bias pada *output gate*

Setelah menerima nilai dari *output gate* maka *cell state* ditempatkan melalui *tanh*. Kemudian, dikalikan dengan *output gate* dan *sigmoid* tanh. Adapun persamaan nilai *output* orde ke t yaitu sebagai berikut.

$$h_t = o_t \cdot \tanh(C_t) \quad (2.6)$$

Di mana:

- $h_t$  = nilai *output* orde t
- $o_t$  = nilai *output gate*
- tanh* = fungsi *tanh*
- $C_t$  = nilai *cell state* [13].

### Normalisasi

Normalisasi merupakan pra-pemrosesan data yang bertujuan untuk meminimalkan *error* pada *dataset* dengan cara mengubah data aktual menjadi nilai dengan bentuk *range* interval [0,1]. Adapun teknik normalisasi menggunakan rumus

*Min-Max Scalling* yang dinyatakan sebagai berikut.

$$X' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (2.7)$$

Di mana:

$X'$  = nilai hasil normalisasi

$x$  = data

$x_{min}$  = nilai *minimum* dari keseluruhan data aktual

$x_{max}$  = nilai *maksimum* dari keseluruhan data actual .

### Denormalisasi

Sebelum menentukan keakuratan hasil prediksi, harus melakukan tahap denormalisasi data terlebih dahulu yaitu data hasil prediksi diubah menjadi nilai asli karena data hasil peramalan tersebut masih dalam bentuk interval akibat dari proses normalisasi data sebelumnya. Rumus untuk denormalisasi adalah sebagai berikut.

$$d = d'(X_{max} - X_{min}) + X_{min} \quad (2.8)$$

Di mana,

$d$  = nilai hasil denormalisasi

$d'$  = nilai data normalisasi

$X_{max}$  = nilai maksimal dari data aktual

$X_{min}$  = nilai minimal dari data aktual [14]

### Mean Absolute Percentage Error (MAPE)

*Mean Absolute Percentage Error* (MAPE) merupakan nilai rata-rata kesalahan yang digunakan untuk menghitung persentase kesalahan antara nilai aktual dan nilai prediksi. Dimana nilai MAPE ini bertujuan untuk menentukan nilai akurasi peramalan. Rumus yang digunakan adalah sebagai berikut.

$$MAPE = 100 \times \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \tilde{y}_i}{y_i} \right| \quad (2.9)$$

Dengan:;

$\tilde{y}_i$  = nilai hasil peramalan

$y_i$  = nilai aktual

$n$  = jumlah data [15].

## 3. METODOLOGI

Data jumlah penumpang kapal laut yang diperoleh dari PT Pelabuhan Indonesia (Persero)

Ragional 4 tahun 2013-2022 menjadi objek dari penelitian ini. Jenis penelitian yang digunakan adalah penelitian terapan yang bertujuan untuk memecahkan masalah yang diidentifikasi, yang mulai diteliti dari bulan Mei hingga bulan Februari 2024. Penelitian ini menggunakan metode *long Short-Term Memory* (LSTM).

### Prosedur Analisis

Pada penelitian ini menggunakan metode *Long-Short Term Memory* (LSTM). Adapun sistematika analisis datanya adalah sebagai berikut.

1. Analisis data yang akan kita gunakan yaitu data jumlah penumpang keberangkatan Pelabuhan Soekarno Hatta Makassar
2. Membangun model *Long Short-Term Memory* (LSTM)

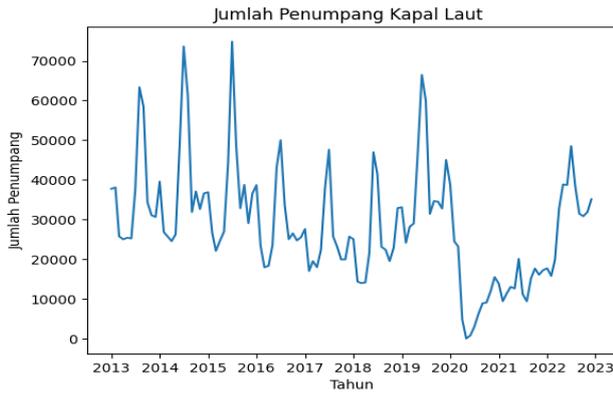
Adapun tahap pembuatan model data *Long Short-Term Memory* adalah sebagai berikut:

- a. Pra-Prapemrosesan Data
  - b. Melakukan split data yaitu membagi dataset kedalam data latih dan data uji
  - c. Melakukan proses *training* dengan metode *Long Short Term Memory*
  - d. Membangun model *Long Short Term Memory*
  - e. Menampilkan hasil peramalan yang didapatkan dari proses *testing* pada Model *Long Short Term Memory* menggunakan data yang telah di denormalisasi.
3. Menghitung nilai akurasi hasil peramalan menggunakan nilai *Root Mean Squared Error* (RMSE) pada persamaan 2.11 dan nilai *Mean Absolute Percentage Error* (MAPE) pada persamaan 2.12
  4. Menampilkan hasil peramalan 12 bulan kedepan menggunakan Model *Long Short-Term Memory*

## 4. PEMBAHASAN

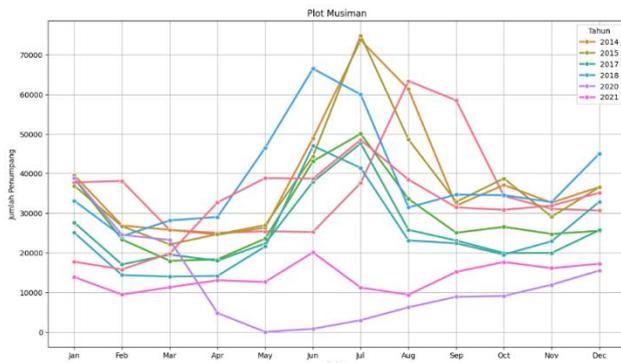
### Analisis Data

Data yang dianalisis diambil dari PT Pelabuhan Indonesia (Persero) Ragional 4 pada Pelabuhan Soekarno Hatta Makassar mengenai jumlah penumpang kapal laut sebagai berikut:



**Gambar 4.1** Grafik Jumlah Penumpang Kapal Laut

Berdasarkan Gambar 4.1 di atas menunjukkan bahwa pergerakan jumlah penumpang Kapal Laut pada Pelabuhan Soekarno Hatta Makassar cenderung fluktuasi setiap tahunnya yang terjadi pada musim-musim tertentu misanya musim libur, hari raya maupun akhir tahun. Jumlah penumpang tertinggi terjadi pada tahun 2015 dan mengalami penurunan jumlah penumpang pada saat datangnya pandemi covid19 yang terjadi pada Tahun 2020. Pada grafik diatas terlihat adanya pola berulang yang terjadi seiring berjalannya waktu. Berikut ini grafik pola data pada jumlah penumpang kapal laut, dapat dilihat pada Gambar 4.2. Berdasarkan tersebut terlihat pola bulanan jumlah penumpang kapal laut dengan warna garis yang berbeda dari Tahun 2013 hingga 2022. Jumlah penumpang tiap Bulan menunjukkan pola *time series* musiman yang yang memperlihatkan pola yang berulang secara berturut-turut yang terjadi disetiap tahunnya, kecuali pada Tahun 2013 untuk periode Januari hingga Maret dan Tahun 2020 periode April



**Gambar 4.2** Grafik Pola Musiman

hingga Agustus yang menunjukkan jumlah penumpang cenderung menurun secara secara

drastis. Pola musiman pada data jumlah penumpang menunjukkan bahwa data yang digunakan belum stasioner dan salah satu kelebihan metode LSTM dapat mempelajari pola kompleks pada data sehingga dapat mengolah data yang stasioner dan non-stasioner. Adapun gambaran umum tentang data jumlah penumpang kapal laut dapat dilihat pada Tabel 4.1 berikut ini:

**Tabel 4. 3** Statistika Deskriptif Data Jumlah Penumpang Kapal Laut

|                 |          |
|-----------------|----------|
| Min             | 4        |
| Kuartil Pertama | 19521,25 |
| Median          | 26022    |
| Mean            | 28762,74 |
| Kuartil Ketiga  | 36934    |
| Max             | 74824    |
| Standar Deviasi | 14367,08 |

**Membangun Model Long Short-Term Memory (LSTM)**

a) *Pra-Pemrosesan Data*

Tahap Pra-pemrosesan data dalam Analisis metode LSTM dilakukan dengan menormalisasikan data terlebih dahulu untuk mengurangi *error* yang terjadi. Adapun contoh perhitungan manual normalisasi data dapat dilihat pada Tabel 4.2.

**Tabel 4. 2** Perhitungan Manual Normalisasi Data

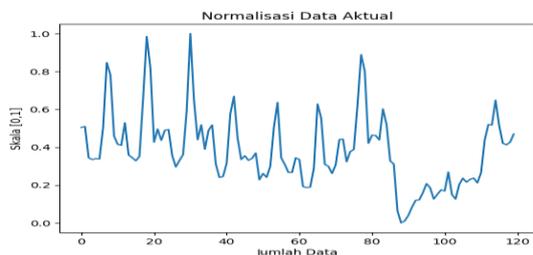
| Bulan         | $\frac{x_1 - x_{min}}{x_{max} - x_{min}}$ | X'         |
|---------------|---|------------|
| Januari 2013  | $\frac{37.727 - 4}{74.820 - 4}$           | 0,50423684 |
| Februari 2013 | $\frac{38.091 - 4}{74.820 - 4}$           | 0,50904838 |
| Maret 2013    | $\frac{25.740 - 4}{74.820 - 4}$           | 0,50904838 |

Berikut ini grafik hasil normalisasi data aktual sebagaimana pada Gambar 4.3.

b) *Data Split*

Membagi data menjadi dua bagian yaitu data latih dan data uji atau yang dikenal dengan sebutan data split. Dalam mencari pembagian data split yang akan kita gunakan dapat dilihat dari nilai MAPE terendah untuk

mengetahui persentase eror suatu model. Namun penggunaan model mesin learning perlu adanya jumlah neuron, optimizer, batch size dan epoch.



**Gambar 4. 4** Grafik Hasil Normalisasi Data Aktual

dan karena tidak adanya ketentuan pasti dalam menentukan penggunaan model maka dari itu penulis melakukan percobaan dengan menginput nilai kombinasi parameter pada model LSTM. Berikut ini kombinasi parameter model LSTM dapat dilihat pada Tabel 4.3 yaitu:

**Tabel 4. 3** Kombinasi Parameter

|               |      |
|---------------|------|
| Optimizer     | Adam |
| Learning Rate | 0,01 |
| Neuron        | 10   |
| Batch Size    | 1    |
| Epoch         | 1    |

Adapun nilai pada Tabel 4.3 digunakan untuk mencari nilai MAPE terendah pada data split. Nilai MAPE dapat dilihat pada Tabel 4.4 berikut ini:

**Tabel 4. 4** Perbandingan Data Split

| No | Perbandingan |     | MAPE         |
|----|--------------|-----|--------------|
|    | Latih        | Uji |              |
| 1  | 90           | 10  | 27,584016086 |
| 2  | 80           | 20  | 106,33227592 |
| 3  | 70           | 30  | 21649,166653 |
| 4  | 60           | 40  | 81,271688299 |

Berdasarkan Tabel 4.4 diperoleh bahwa pembagian dataset 90:10 menghasilkan nilai MAPE terendah, sehingga pada penelitian ini digunakan data latih sebesar 90% dan data uji sebesar 10% dengan jumlah data latih sebesar 108 data dan data uji sebesar 12 data. Menggunakan 10 *timestep* pada data latih dan 1 data uji sebagai nilai target output.

c) Melakukan proses training dataset menggunakan metode Long Short-Term Memory

Melakukan training dataset pada data latih sebanyak 108 data dengan menggunakan 10 *timestep* untuk mendapatkan model LSTM yang cocok dalam meramalan jumlah penumpang kapal laut. Penentuan parameter model tidak berdasar pada ketentuan khusus, namun untuk mencari hasil yang optimal maka penelitian ini melakukan beberapa kali percobaan menggunakan aktivasi *sigmoid* dan *tanh* dengan menginput jumlah *neuron*, *optimizer*, *learning rate*, *batch size* dan *epoch* yang berbeda. Berikut ini Tabel 4.5 merupakan hasil training dari kombinasi parameter pada model LSTM menggunakan data latih 90% yaitu:

**Tabel 4. 5** Analisis Hasil Kombinasi Parameter Pada Model LSTM

| No | Optimizer   | Learning Rate | Neuron    | Batch Size | Epoch      | Loss MSE      |
|----|-------------|---------------|-----------|------------|------------|---------------|
| 1  | Adam        | 0,01          | 10        | 1          | 1          | 0,0372        |
| 2  | Adam        | 0,05          | 10        | 1          | 1          | 0,0401        |
| 3  | Adam        | 0,01          | 10        | 1          | 50         | 0,0060        |
| 4  | Adam        | 0,05          | 10        | 1          | 50         | 0,0060        |
| 5  | <b>Adam</b> | <b>0,01</b>   | <b>10</b> | <b>1</b>   | <b>100</b> | <b>0,0033</b> |
| 6  | Adam        | 0,05          | 10        | 1          | 100        | 0,0081        |
| 7  | Adam        | 0,01          | 10        | 2          | 50         | 0,0069        |
| 8  | Adam        | 0,05          | 10        | 2          | 50         | 0,0079        |
| 9  | Adam        | 0,01          | 10        | 2          | 100        | 0,0041        |
| 10 | Adam        | 0,05          | 10        | 2          | 100        | 0,0052        |
| 11 | Adam        | 0,01          | 20        | 1          | 50         | 0,0064        |
| 12 | Adam        | 0,05          | 20        | 1          | 50         | 0,0101        |
| 13 | Adam        | 0,01          | 20        | 1          | 100        | 0,0039        |
| 14 | Adam        | 0,05          | 20        | 1          | 100        | 0,0064        |
| 15 | Adam        | 0,01          | 20        | 2          | 50         | 0,0092        |
| 16 | Adam        | 0,05          | 20        | 2          | 50         | 0,0103        |
| 17 | Adam        | 0,01          | 20        | 2          | 100        | 0,0036        |
| 18 | Adam        | 0,05          | 20        | 2          | 100        | 0,0050        |
| 19 | Adam        | 0,01          | 50        | 1          | 50         | 0,0071        |
| 20 | Adam        | 0,05          | 50        | 1          | 50         | 0,0072        |
| 21 | Adam        | 0,01          | 50        | 2          | 100        | 0,0039        |
| 22 | Adam        | 0,05          | 50        | 2          | 100        | 0,0038        |
| 23 | Adam        | 0,01          | 100       | 1          | 50         | 0,0078        |
| 24 | Adam        | 0,05          | 100       | 1          | 50         | 0,0078        |
| 25 | Adam        | 0,01          | 100       | 2          | 50         | 0,0112        |

|    |      |      |     |   |     |        |
|----|------|------|-----|---|-----|--------|
| 26 | Adam | 0,05 | 100 | 2 | 50  | 0,0075 |
| 27 | Adam | 0,01 | 100 | 2 | 100 | 0,0043 |
| 28 | Adam | 0,05 | 100 | 2 | 100 | 0,0106 |

Berdasarkan Tabel 4.5 diperoleh model yang paling optimal pada data jumlah penumpang kapal laut yaitu menggunakan *optimizer* adam dengan *learning rate* 0,01, jumlah *neuron* pada *hidden layer* yaitu 10, jumlah *batch size* yaitu 1 dan jumlah *epoch* yaitu 100 menghasilkan nilai Loss MSE sebesar 0,0033.

d) Membangun model Long Short-Term Memory

Pada tahap ini digunakan model yang diperoleh dari hasil *training* pada data latih dengan menggunakan aktivasi *sigmoid* dan *tanh*. Adapun konfigurasi model yang paling optimal dapat dilihat pada Tabel 4.6 berikut ini:

**Tabel 4. 6** Analisis Hasil Kombinasi Parameter Pada Model LSTM

|               |      |
|---------------|------|
| Optimizer     | Adam |
| Learning Rate | 0,01 |
| Neuron        | 10   |
| Batch Size    | 32   |
| Epoch         | 50   |

Konfigurasi model pada Tabel 4.6 digunakan untuk melakukan proses *testing* pada dataset guna mendapatkan hasil prediksi jumlah penumpang kapal laut. Berikut ditampilkan hasil model *sequential LSTM* dengan menggunakan konfigurasi model yang paling optimal, sebagaimana pada Tabel 4.7 yaitu:

**Tabel 4. 7** Model Sequential LSTM

| Layer (type)             | Output Shape   | Param # |
|--------------------------|----------------|---------|
| lstm(LSTM)               | (None, 10, 10) | 480     |
| lstm_1(LSTM)             | (None, 10)     | 840     |
| dense (Dense)            | (None, 1)      | 11      |
| <hr/>                    |                |         |
| Total params: 40.901     |                |         |
| Trainable Params: 40.901 |                |         |
| Non-trainable params: 0  |                |         |

Berdasarkan Tabel 4.7 di atas menunjukkan bahwa model *sequential* yang digunakan untuk membangun arsitektur model secara berurutan memiliki dua *layer LSTM* dan satu *layer dense*. Setiap *layer* berfungsi sebagai komponen memori yang dapat mengontrol aliran informasi berdasarkan pada fungsi aktivasi *sigmoid* dan *tanh*. Adapun *Layer LSTM* pertama dengan *output shape* (None, 10, 10). Artinya, *layer* ini memiliki 10 *neuron* pada *hidden layer* dan menerima *input* dengan *timestep* 10. Total parameter pada *layer* ini adalah 480. *Layer LSTM* kedua dengan *output shape* (None, 10). Artinya, *layer* ini memiliki 10 *neuron* pada *hidden layer* dan menerima *input* dari *layer* sebelumnya dengan *timestep* 10. Total parameter pada *layer* ini adalah 840. *Layer dense* dengan *output* (None, 1), yang artinya memiliki 1 *neuron* dengan satu nilai *output* yaitu nilai target (y). Model yang terbentuk memiliki total keseluruhan parameter sebanyak 1.331 *weight* dan *biases*.

e) Hasil Peramalan

Berdasarkan Tabel 4.7 model *sequential LSTM* di atas digunakan untuk meramalkan jumlah penumpang kapal laut pada data *testing*. Kemudian melakukan proses denormalisasi pada hasil peramalan yang bertujuan mengembalikan nilai data yang diperoleh menjadi hasil peramalan dengan skala sebenarnya. Berikut Tabel 4.8 merupakan perbandingan data uji dan hasil peramalan pada data *testing* setelah data didenormalisasi yaitu:

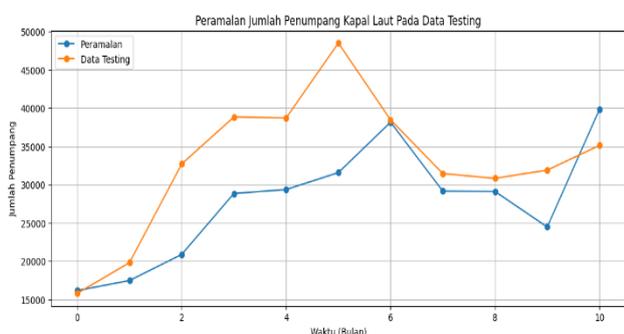
**Tabel 4. 8** Hasil Peramalan Data Testing

| Bulan         | Data Uji | Peramalan(y) |
|---------------|----------|--------------|
| Februari 2022 | 15794    | 16157        |
| Maret 2022    | 19806    | 17466        |
| April 2022    | 32696    | 20878        |
| Mei 2022      | 38841    | 28847        |
| Juni 2022     | 38711    | 29358        |
| Juli 2022     | 48499    | 31570        |
| Agustus 2022  | 38458    | 38156        |

| Bulan          | Data Uji | Peramalan(y) |
|----------------|----------|--------------|
| September 2022 | 31442    | 29145        |
| Oktober 2022   | 30821    | 29104        |
| November 2022  | 31893    | 24463        |
| Desember 2022  | 35120    | 39830        |

Adapun grafik perbandingan hasil peramalan data testing dengan data ujo pada model *Long Short-Term Memory* dapat dilihat pada Gambar 4.4 berikut ini:

**Gambar 4. 4** Grafik Perbandingan Data Peramalan dengan Data Testing



**Gambar 4. 5** Grafik Perbandingan Data Peramalan dengan Data *Testing*

Berdasarkan Gambar 4.4 diatas diperoleh bahwa pola data prediksi bersifat cenderung fluktuaktif mengikuti pola data *testing*. Di mana data peramalan direpresentasikan dengan warna Biru dan data *testing* berwarna jingga.

**Perhitungan nilai hasil Akurasi**

Untuk mengetahui presentase kesalahan antara nilai aktual dan nilai prediksi pada data *testing* sebanyak 11 data dapat dihitung menggunakan perhitungan nilai MAPE. Berikut ini perhitungan akurasi MAPE menggunakan rumus pada persamaan 2.11 sebagai berikut

$$MAPE = 100 \times \frac{1}{n} \sum_{i=1}^n \left| \frac{\tilde{y}_i - y_i}{y_i} \right|$$

$$MAPE = 100 \times \frac{1}{11} \left( \frac{16156,787 - 15794}{15794} + \dots + \frac{39830,438 - 35120}{35120} \right)$$

$$MAPE = 100 \times \frac{1}{11} (1,854273261)$$

$$MAPE = 16,85702941\%$$

**Menampilkan hasil peramalan periode berikutnya**

Peramalan jumlah penumpang kapal laut untuk 12 Bulan berikutnya diperoleh berdasarkan Model *Long Short Term Memory* yang paling optimal yaitu menggunakan *optimizer adam* dengan *learning rate* 0,01, jumlah *neuron* pada *hidden layer* yaitu 10, jumlah *batch size* yaitu 1 dan jumlah *epoch* yaitu 100 serta menggunakan dua fungsi aktivasi pada *layer LSTM* yaitu fungsi *sigmoid* dan *tanh*. Berikut ini Tabel 4.9 merupakan tabel hasil peramalan 12 bulan berikutnya yaitu pada Bulan Januari-Desember 2023:

**Tabel 4.9** Hasil Peramalan Periode Berikutnya

| Bulan          | Hasil Peramalan |
|----------------|-----------------|
| Januari 2023   | 33.143          |
| Februari 2023  | 33.531          |
| Maret 2023     | 22.189          |
| April 2023     | 25.760          |
| Mei 2023       | 29.450          |
| Juni 2023      | 27.810          |
| Juli 2023      | 40.444          |
| Agustus 2023   | 45.451          |
| September 2023 | 33.457          |
| Oktober 2023   | 27.614          |
| November 2023  | 23.235          |
| Desember 20223 | 25.151          |

Adapun grafik peramalan 12 bulan kedepan dapat dilihat pada Gambar 4.5 berikut ini:



**Gambar 4.5** Grafik Peramalan 12 bulan berikutnya

Berdasarkan Gambar 4.5 diatas diperoleh bahwa pola data peramalan 12 bulan kedepan bersifat cenderung mengikuti pola data sebenarnya. Di mana data sebenarnya direpresentasikan dengan warna Biru dan data peramalan berwarna jingga.

## 5. KESIMPULAN

Berdasarkan pengujian dan analisis yang dilakukan dalam penerapan metode LSTM untuk melakukan peramalan jumlah penumpang kapal laut, sehingga didapatkan kesimpulan:

1. Model *Long Short Term Memory* yang paling optimal untuk meramalkan jumlah penumpang kapal laut di PT Pelabuhan Indonesia (Persero) Ragonal 4 yaitu menggunakan parameter dengan jumlah *neuron* pada *hidden layer* yaitu 10, jumlah *batch size* yaitu 1 dan jumlah *epoch* yaitu 100, serta menggunakan optimasi *Adam* dengan *learning rate* 0,01. Sehingga menghasilkan *model sequential 2 layer LSTM* dengan fungsi aktivasi *sigmoid* dan *tanh* dan satu *Layer Dense* dengan total parameter sebanyak 1.331 *weight* dan *biases*.
2. Tingkat akurasi peramalan yang di peroleh menghasilkan nilai MAPE sebesar 16,85702941%. Berdasarkan aturan range nilai MAPE yang didapatkan, dapat dikatakan bahwa akurasi peramalan yang dilakukan termasuk kedalam kategori baik karena nilai MAPE 10-20%.
3. Hasil peramalan untuk 12 Bulan berikutnya menggunakan Model *Long Short Term Memory* yang paling optimal menghasilkan peramalan pada Bulan Januari 2023 sebanyak 33.143 penumpang hingga

Desember 2023 sebanyak 25.151 penumpang.

## 6. DAFTAR PUSTAKA

- [1] Nurjanah, I. S., Ruhiat, D., & Andiani, D. (2018). Implementasi Model Autoregressive Integrated Moving Average (ARIMA) untuk Peramalan Jumlah Penumpang Kereta Api di Pulau Sumatera. *Jurnal Teorema: Teori dan Riset MAtematika*, 3(2), 145-156.
- [2] Zhang, G. P. (2004). *Neural Network in Business Forecasting*. USA: Idea Group Publising.
- [3] Wardana, I. K., Jawas, N., & Aryanto, I. A. (2020). Prediksi Penggunaan Energi Listrik pada Rumah Hunian Menggunakan Long Short-Term Memory. *TIERS Informatika Technology Journal*, 1(1), 1-11
- [4] Ardani, Mela. (2022). Prediksi Jumlah Penumpang Pesawat Domestik dengan Menggunakan Long Short-Term Memory (Studi kKasus: Bandar Internasional SAMS Sepinggian Balikpapan). Diploma Thesis, Institut Teknologi Kalimantan.
- [5] Sanjaya, Putri Kristina & Halim Siana. 2020. Currency Movement Forecasting Using Time Series Analysys And Long Short-Term Memory. *Internasional Journal Of Industrial Optimization* Vol. 1, No.. 2, 71-80.
- [6] Ardhito, Don. (2020). Analisis Perbandingan Metode Machine Learning Untuk Memprediksi Kasus Covid-19 Di Dunia. Diploma Thesis Universitas Budi Luhur.
- [7] Rosadi, dedi. (2014). Analisis Runtun Waktu dan Aplikasinya dengan R. Yogyakarta : Gadjah Mada University Press.
- [8] Hanke, J. E., & Wichern, D. W. (2005). *Bussiness Forecasting* (Eight ed.). Pearson Prentice Hall: New Jersey.
- [9] Yudaruddin, R. (2019). *Forecasting: Peramalan Untuk Kegiatan Ekonomi Dan Bisnis*. Samarinda: RV Pustaka Horizon.
- [10] Arfan, A., & Lussiana, E. T. (2019). Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term

Memory. *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi*, 3(1), 225-230.

- [11] Kurniawan, D. (2021). *Pengenalan Machine Learning dengsn Python*. Bandung: Informatika Bandung.
- [12] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Long Short-Term Memory Based Operation Log Anomaly Detection. *International Conference Advance Computing Communication Informatics*, 236-242.
- [13] Aldi, M. W., Jondri, & Aditsania , A. (2018). Analisis dan Implementasi Long-Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *E-Proceeding Engineering*, 5(2), 3548-3555.
- [14] Ashar, N. M., Cholissodin, I., & Dewi, C. (2018). Penerapan Metode Extreme Learning Machine (ELM) Untuk Memprediksi Jumlah Produksi Pipa Yang Layak (Studi Kasus Pada PT .KHI Pipe Industries) . *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya*, 2(11), 4621-4628.
- [15] Riyadi, S. (2015). Aplikasi Peramalan Penjualan Obat Menggunakan Metode Pemulusan (Studi Kasus:Instalasi Farmasi Rsud Dr Murjani). *Open Journal System Universitas AMIKOM*(1), 1-6.