

## BAB III

### METODE PENELITIAN

Pada bab ini dijelaskan mengenai subjek dan objek penelitian, alat dan bahan, dan rencana penelitian.

#### 3.1 Subjek dan Objek Penelitian

Subjek yang digunakan dalam penelitian ini adalah data jumlah kendaraan di Jalan Layang Mohammed Bin Zayed, Sedangkan objek penelitian ini adalah prediksi jumlah kendaraan di Jalan Layang Mohammed Bin Zayed.

#### 3.2 Alat dan Bahan

Dalam penelitian ini, penggunaan alat dan bahan menjadi faktor penting untuk keberhasilan penelitian, adapun alat dan bahan yang digunakan yaitu:

##### 3.2.1 Alat

Alat yang digunakan dalam penelitian ini memiliki dua jenis yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*) (lihat Tabel 3.1).

**Tabel 3.1** Spesifikasi Perangkat

No.	Komponen Perangkat Keras	Deskripsi
<b>A.</b>	<b>Perangkat Keras</b>	
1	<i>Processor</i>	Intel Core i7-9750h
2	<i>Memory</i>	16GB
3	Penyimpanan	512GB
<b>B.</b>	<b>Perangkat Lunak</b>	
1	Editor Program	<i>Jupyter Notebook</i>
2	Library	<i>Tensorflow</i>
3	Bahasa Pemrograman	Python

### 3.2.2 Bahan

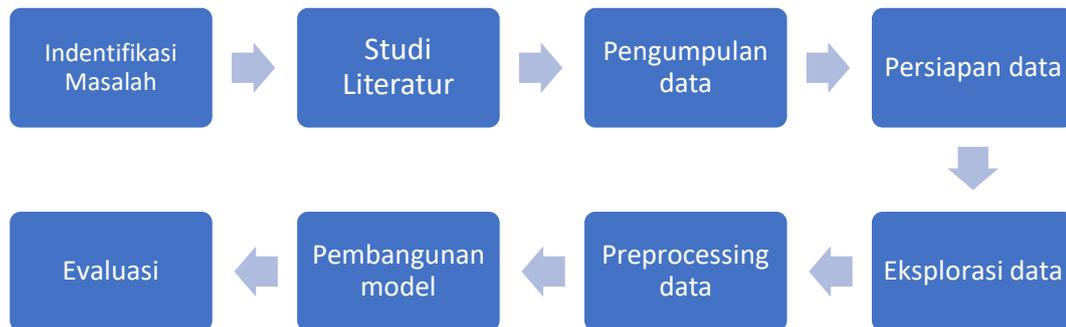
Bahan yang digunakan dalam penelitian ini adalah data jumlah kendaraan per-jam di Jalan Layang Mohammed Bin Zayed (MBZ) baik yang menuju Jakarta dan meninggalkan Jakarta dengan rentang tanggal 1 Januari 2022 jam 0 hingga tanggal 31 Juli 2023 jam 23 sebanyak 13848 baris, Data ini diperoleh dari salah satu Badan Usaha Milik Negara (BUMN) yaitu PT. Jasamarga Jalanlayang Cikampek (JJC). *Dataset* ini mencakup deret waktu jumlah kendaraan serta jam. Adapun atribut yang digunakan dalam penelitian dapat dilihat pada table 3.2.

**Tabel 3. 2** Atribut dataset

No	Atribut	Tipe Data	Keterangan
1	Tanggal	<i>Datetime</i>	Tanggal perubahan data jumlah kendaraan.
2	Jam	Numerik (Diskrit)	Jam perubahan data jumlah kendaraan
3	Jalur_a	Numerik	Jumlah kendaraan jalur a pada tanggal dan jam tertentu.
4	Jalur_b	Numerik	Jumlah kendaraan jalur b pada tanggal dan jam tertentu.

### 3.3 Alur Penelitian

Dalam menyusun laporan penelitian ini, sejumlah tahap telah dilaksanakan sebagai bagian dari proses penelitian. Berikut ini terdapat sebuah alur yang mengilustrasikan langkah-langkah penelitian yang diambil dalam penyusunan laporan ini:



**Gambar 3. 1** Alur Penelitian

### 3.3.1 Identifikasi Masalah

Penelitian dimulai dari identifikasi permasalahan yang dihadapi oleh PT. Jasamarga Jalanlayang Cikampek khususnya dalam peramalan jumlah kendaraan di Jalan Layang Mohammed Bin Zayed.

### 3.3.2 Studi Literatur

Penelitian ini memerlukan referensi atau sumber literatur sebagai pedoman dalam melakukan penelitian, pengembangan model, dan konsep dari metode yang digunakan. Sumber tersebut berasal dari jurnal dan buku yang berusia maksimal 5 tahun terakhir tentang *preprocessing data* dan pemodelan *deep learning*.

### 3.3.3 Pengumpulan Data

Data jumlah kendaraan per-jam didapatkan secara langsung dari salah satu Badan Usaha Milik Negara (BUMN) yang mengelola Jalan Layang Mohammed Bin Zayed yaitu PT. Jalanlayang Jakarta Cikampek, Adapun tujuan diberikan data ini digunakan untuk pembangunan model yang dapat memprediksi jumlah kendaraan di masa yang akan datang berbasis Kecerdasan Buatan.

Data ini diambil dari alat penghitung jumlah kendaraan berbasis CCTV yang terpasang di Jalan Layang Mohammed Bin Zayed. Adapun rentang waktu untuk data yang diberikan dari Tanggal 1 Januari 2022 Jam 00:00 sampai dengan 31 Juli 2023 Jam 23:00.

Adapun atribut yang ada didalam dataset ini adalah tanggal, Jalur A (jalur yang mengarah ke Kota Jakarta), Jalur B (jalur yang meninggalkan Kota Jakarta) , jumlah kendaraan yang melintas pada jam tertentu di kedua jalur dan data ini memiliki 13848 baris.

Tanggal	Jalur	JAM						
		00.00 - 01.00	01.00 - 02.00	02.00 - 03.00	03.00 - 04.00	04.00 - 05.00	05.00 - 06.00	06.00 - 07.00
01-Jan-22	Jalur A	1292	1152	1028	952	1038	1982	2585
	Jalur B	709	611	649	523	451	415	528
02-Jan-22	Jalur A	832	680	515	571	636	1213	1727
	Jalur B	1568	1325	1237	1079	764	623	764
03-Jan-22	Jalur A	1026	741	648	775	1079	2141	2457
	Jalur B	2761	2845	2433	2351	2137	2180	2648
04-Jan-22	Jalur A	532	413	292	339	393	831	1911
	Jalur B	999	850	757	770	714	812	1058
05-Jan-22	Jalur A	157	117	227	341	344	882	1257
	Jalur B	805	702	780	760	665	759	929
06-Jan-22	Jalur A	572	424	389	485	605	1285	1875
	Jalur B	1522	1466	1323	1294	1172	1250	1545

**Gambar 3. 2** Visualisasi *Dataset* Jumlah Kendaraan

### 3.3.4 Persiapan Data

Langkah selanjutnya adalah transposisi dataset menggunakan bantuan *Microsoft Excel* dikarenakan bentuk dataset yang diberikan berbentuk horizontal atau memanjang kesamping. Proses ini dilakukan untuk keperluan pemodelan [41]. Setelah melewati tahap pertama persiapan data, proses ini menghasilkan data yang sudah bisa di eksplorasi lebih lanjut.

tanggal	jam	jalur_a	jalur_b
2022-01-01 00:00:00	0	1292	709
2022-01-01 00:00:00	1	1152	611
2022-01-01 00:00:00	2	1028	649
2022-01-01 00:00:00	3	952	523
2022-01-01 00:00:00	4	1038	451
2022-01-01 00:00:00	5	1982	415
2022-01-01 00:00:00	6	2585	528
2022-01-01 00:00:00	7	2532	686

**Gambar 3. 3** Visualisasi Dataset Pasca Transposisi

### 3.3.5 Eksplorasi Data

Statistik deskriptif digunakan untuk menyajikan informasi statistik dari dataset seperti rata-rata, median, min,max, standar deviasi dan jumlah data yang dimana ini membantu dalam pemahaman yang lebih baik tentang karakteristik keseluruhan dari kumpulan data yang diamati[41].

### 3.3.6 Preprocessing Data

*Preprocessing* data bertujuan untuk mempersiapkan data agar siap digunakan untuk kepentingan pemodelan yang meliputi penanganan *outlier*, normalisasi data, *sliding windows* dan *split dataset*.

#### 1. Handling Outlier

Outlier adalah data yang memiliki ketidaknormalan yang berbeda dengan data lainnya, dengan adanya data yang menyimpang ini akibatnya akan menyulitkan model untuk mempelajari pola dari dataset[42]. Untuk mendeteksi adanya *outlier* didalam sebuah dataset dapat menggunakan metode visualisasi grafik *boxplot* seperti dibawah ini:



**Gambar 3. 4** Visualisasi *Boxplot* Jalur A

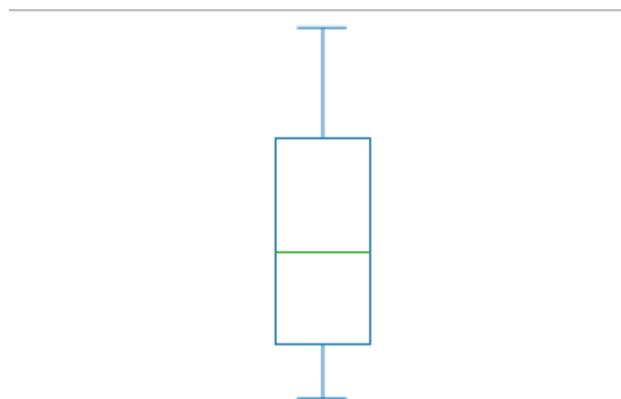


**Gambar 3. 5** Visualisasi *Boxplot* Jalur B

Dapat dilihat grafik *boxplot* pada Gambar 3.4 dan 3.5, terdapat outlier yang ditandai dengan tanda bulat yang melewati batas atas, untuk mengatasi data *outlier* dapat menggunakan metode *quantile based capping and flooring* yang dimana metode ini mengganti nilai yang melewati batas atas dengan nilai *percentile* ke 90 dan *percentile* ke 10 untuk data yang melewati batas bawah[30]. Hasil dari penerapan metode *quantile based capping and flooring* dapat dilihat pada visualisasi *boxplot* berikut:



**Gambar 3. 6** Visualisasi *Boxplot* Jalur A Pasca Penanganan



**Gambar 3. 7** Visualisasi *Boxplot* Jalur B Pasca Penanganan

## 2. Normalisasi Data

Tahap normalisasi bertujuan untuk merubah nilai dengan rentang yang jauh menjadi antara 0 hingga 1 yang dimana metode ini membantu model untuk meningkatkan efisiensi dalam pelatihan dikarenakan rentang data yang lebih sempit[31].

### 3. *Sliding Windows*

*Sliding Windows* digunakan untuk merubah struktur data runtun waktu dengan cara mengelompokkan data pada rentang waktu tertentu dalam penelitian ini adalah jam yang dimana teknik ini membantu model LSTM untuk melakukan observasi dengan cara melihat data pada *timestep* sebelumnya sebelum melakukan prediksi[29]. Dalam penentuan *windows size* tidak ada aturan yang pasti dan bersifat eksperimental[43] maka dari itu penelitian ini menggunakan eksperimen dengan konfigurasi *window size* = 8 yang berarti untuk memprediksi satu langkah kedepan perlu menggunakan *input* 8 langkah sebelumnya Contoh penerapan *sliding windows* di gambarkan sebagai berikut dengan konfigurasi *window size* = 8.

jam	jalur_a	jalur_b
0	1292	709
1	1152	611
2	1028	649
3	952	523
4	1038	451
5	1982	415
6	2585	528
7	2532	686
8	2488	911

**Gambar 3. 8** Data Sebelum diterapkan *Sliding Windows*

x1	x2	x3	x4	x5	x6	x7	x8	y
[0,1292,709]	[1,1152,611]	[2,1028,649]	[3,952,523]	[4,1038,451]	[5,1982,415]	[6,2585,528]	[7, 2532, 686]	[2488,911]

**Gambar 3. 9** Visualisasi data setelah penerapan *Sliding Windows*

Dapat dilihat pada Gambar 3.9, untuk memprediksi nilai pada jam 8 digunakan *input* berupa 8 data sebelumnya yaitu data jam, jalur a, jalur b pada jam 0 hingga jam 7 dikarenakan konfigurasi *windows size* = 8, Hal ini dilakukan hingga data pada indeks terakhir.

#### 4. *Split Dataset*

Setelah melakukan perubahan struktur data *dataset* dengan *sliding windows*, langkah selanjutnya adalah *Split dataset* proses ini bertujuan untuk memisahkan data *training* dan data *testing* . Data *training* digunakan untuk melatih model LSTM dan data *testing* digunakan untuk menilai performa dari model yang telah dilatih[44]. Skenario yang digunakan dalam pemisahan dataset ini adalah 80 % untuk data latih dan 20% untuk data *testing* atas dasar penelitian sebelumnya yang menggunakan rasio 80% data latih dan 20% data uji menunjukkan hasil yang optimal[45].

#### 3.3.7 Pembangunan Model

Tahap utama dari penelitian ini adalah membangun model LSTM untuk memprediksi jumlah kendaraan pada kedua jalur, untuk membangun model LSTM penelitian ini menggunakan Bahasa pemrogramman *python* dan bantuan library *tensorflow* dikarenakan *tensorflow* menyediakan modul untuk pembangunan arsitektur *deep learning* seperti LSTM.

#### **Algoritma 3.1:** Model LSTM

```

1. Inisialisasi bobot dan bias (input_gate, forget gate, cell,
   output gate)
2. Inisialisasi state LSTM:
   a. Hidden_state = 0
   b. Cell_state = 0
3. For each input_t in input_sequences:
   a. Gabungkan input ke-t dengan hidden_state
      Gabung = concat(input_t, hidden_state)
   b. Hitung Forget Gate
      Input_gate = sigmoid(W_f * Gabung + b_f)
   c. Hitung input_gate
      Input_gate = sigmoid(W_i * Gabung + b_f)
   d. Hitung kandidat cell
      Kandidat = tanh(W_c * gabung + b_c)
   e. Update cell state
      Cell_state = forget_gate * cell_state + input_gate *
      kandidat
   f. Hitung output_gate
      Output_gate = sigmoid(W_o * gabung + b_o)
   g. Update Hidden_state

```

```

hidden_state = output_gate * tanh(cell_state)
4. Output = hidden_state

```

Dalam pembangunan model LSTM perlu dilakukan perencanaan skenario konfigurasi lapisan pada model LSTM. Pada penelitian ini menggunakan empat skenario yang kemudian akan dipilih salah satu dari ke-empat skenario berdasarkan evaluasi model.

Skenario yang digunakan menggunakan tiga skenario arsitektur LSTM yang berasal dari penelitian sebelumnya dengan tugas yang sama yaitu *time series forecasting* dan satu skenario yang di ajukan oleh penelitian ini. Ke empat skenario tersebut akan dibandingkan dengan cara mencari nilai kerugian yang paling rendah pada saat pengujian performa dalam memprediksi data uji yang sudah disiapkan sebelumnya, arsitektur yang diajukan didalam penelitian ini sebagai berikut:

**Tabel 3. 3** Skenario Arsitektur

No	Skenario	Lapisan
1	Skenario 1	LSTM(128) ( <i>Input</i> ) LSTM(256) LSTM(512) DENSE(100) DENSE(100) DENSE(2) ( <i>Output</i> )

2	Skenario 2	<p>LSTM(96, DROPOUT=0.2) (<i>Input</i>)</p> <p>LSTM(96, DROPOUT=0.2)</p> <p>DENSE(2) (<i>Output</i>)</p>
3	Skenario 3	<p>LSTM(64, DROPOUT = 0.2) (<i>Input</i>)</p> <p>LSTM(64, DROPOUT = 0.2)</p> <p>LSTM(64, DROPOUT = 0.2)</p> <p>LSTM(64, DROPOUT = 0.2)</p> <p>DENSE(2) (<i>Output</i>)</p>
4	Skenario 4	<p>LSTM(50, DROPOUT = 0.2) (<i>Input</i>)</p> <p>LSTM(50, DROPOUT = 0.2)</p> <p>LSTM(50, DROPOUT = 0.2)</p> <p>LSTM(50, DROPOUT = 0.2)</p> <p>DENSE(2) (<i>Output</i>)</p>

Arsitektur yang diajukan pada skenario 1 memiliki 3 lapisan LSTM, LSTM 1 sebagai *input layer* yang menerima data. Kemudian LSTM 2 menerima informasi

dari LSTM 1 yang masih berbentuk *sequence* kemudian dilanjutkan ke lapisan LSTM 3. Pada lapisan LSTM 3 dengan jumlah unit sebesar 512 tidak perlu mengembalikan *sequence* karena informasi yang telah diproses lapisan LSTM perlu dilanjutkan ke lapisan *fully connected layer* dan lapisan LSTM 3 menghasilkan *shape* data 1 dimensi . Pada lapisan *Dense 1* dan lapisan *Dense 2 neuron* akan mempelajari informasi dari lapisan LSTM dengan fungsi aktivasi *linear*, pada *output layer* menggunakan lapisan *Dense* dengan jumlah unit 2 dikarenakan hasil dari lapisan ini digunakan untuk menghasilkan prediksi sesuai dengan jumlah jalur yaitu Jalur A (jalur Mengarah ke Kota Jakarta) dan Jalur B (jalur meninggalkan Kota Jakarta).

Penentuan jumlah unit disetiap lapisannya tidak ada aturan dan sesuai dengan kebutuhannya namun berdasarkan penelitian sebelumnya [4] menggunakan unit yang lebih besar akan menghasilkan performa yang baik namun akan memakan sumber daya yang besar.

Arsitektur kedua berasal dari studi literatur yang berjudul “Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM)” dengan nilai kerugian terendah MAPE yaitu 2.64% [13]. Arsitektur ini menggunakan lapisan LSTM sebanyak 8 *layer* dengan setiap lapisannya memiliki 96 unit dan *dropout* sebesar 0.2 setelah lapisan LSTM akan langsung masuk kedalam lapisan *output layer Dense* dengan jumlah unit 2 dikarenakan model ini memiliki keluaran dua dimensi yang artinya dapat memprediksi dua jalur sekaligus.

Arsitektur ketiga berasal dari penelitian yang berjudul “Prediksi Rata-Rata Zat Berbahaya Di DKI Jakarta Berdasarkan Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory” dengan hasil nilai kerugian MAPE sebesar 12.28 [15]. Arsitektur ini menggunakan empat lapisan LSTM dan *Dropout* dengan masing masing unit tiap lapisannya 64 dan *dropout* sebesar 0.2 kemudian langsung masuk kedalam *output layer Dense* dengan dengan jumlah 2 unit.

Arsitektur ke-empat berasal dari penelitian yang berjudul “Prediksi Data Time Series Saham Bank BRI Dengan Mesin Belajar LSTM (Long ShortTerm

Memory)” dengan nilai RMSE(*Root Mean Squared Error*) sebesar 227,47[5]. Arsitektur ini menggunakan 4 lapisan LSTM dan *Dropout* yang tiap lapisannya memiliki 50 unit dan *dropout* sebesar 0.2. Setelah lapisan LSTM selanjutnya adalah *output layer dense* dengan jumlah unit 2.

**Tabel 3. 4** Konfigurasi Pelatihan

No	Parameter	Nilai
1	<i>Epoch</i>	100
2	<i>Batch size</i>	256
3	<i>Optimizer</i>	Adam
4	<i>Loss</i>	<i>MSE</i>

Penentuan nilai pada setiap skenario konfigurasi pelatihan seperti *epoch* dan *batch size* dan *optimizer* berdasarkan penelitian sebelumnya yang menggunakan konfigurasi pelatihan dengan memiliki beberapa skenario yang berbeda menunjukkan bahwa tidak ada aturan dalam pemilihan jumlah *epoch* dan *batch size*[17] Namun pada penelitian tersebut menunjukkan bahwa semakin besar *epochs* akan meningkatkan performa model namun akan memakan waktu yang lebih lama dalam proses pelatihan. Penggunaan *optimizer adam* berdasarkan penelitian sebelumnya menunjukkan performa yang baik dalam melakukan tugas *time series forecasting*[46]. Dengan alasan tersebut maka dari itu konfigurasi pelatihan yang digunakan untuk penelitian ini menggunakan parameter yang sama untuk semua arsitektur agar adil dan penelitian yang dilakukan lebih cepat.

### 3.3.8 Evaluasi Model

Evaluasi model yang telah dibangun dilakukan dengan cara membandingkan data asli dengan data hasil prediksi, data hasil prediksi didapatkan dengan cara memprediksi data *testing* yang sudah disiapkan pada tahap *split dataset*. Kemudian dilakukan penilaian performa dengan metode *Mean Absolute Percentage Error* dengan acuan penilaian pada table berikut:

**Tabel 3. 5** Acuan penilaian MAPE

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

Selain menggunakan MAPE sebagai metrik evaluasi, MSE atau *Mean Squared Error* digunakan untuk mengevaluasi sejauh mana kesalahan hasil prediksi dibandingkan dengan data asli, semakin kecil nilai MSE maka semakin baik juga model prediktif yang dibuat.