

## ANALISIS PERBANDINGAN METODE TIME SERIES FORECASTING UNTUK PREDIKSI PENJUALAN OBAT DI APOTEK (STUDI KASUS: KIMIA FARMA APOTEK TAKOMA)

Amalia Kurniawati<sup>1</sup>, Muhammad Sabri Ahmad<sup>2</sup>, Muhammad Fhadli<sup>3</sup>, Salkin Lutfi<sup>4</sup>

<sup>1,2,3,4</sup> Program Studi Teknik Informatika, Fakultas Teknik, Universitas Khairun Jl. Jati Metro, Kota Ternate Selatan

Email: <sup>1</sup>amaliakurniawati100717@gmail.com, <sup>2</sup>sabriahmad@gmail.com, <sup>3</sup>mhadli@unkhair.ac.id, <sup>4</sup>salkin\_lutfi@yahoo.com

(Naskah masuk: 27-05-2023, diterima untuk diterbitkan: 01-06-2023)

### Abstrak

Keberadaan apotek sangat penting bagi masyarakat untuk memenuhi kebutuhan obat-obatan. Apotek Kimia Farma merupakan perusahaan yang bergerak dalam bidang farmasi atau *health care company* dan memiliki jaringan apotek terbesar di Indonesia. Prediksi penjualan obat pada apotek ialah salah satu kegiatan prioritas dalam penentuan penjualan kedepannya, hal ini bertujuan untuk pengendalian stok agar tidak terjadi kelebihan dan kekurangan stok serta mencegah ketidakterdediaan obat yang ingin dibeli konsumen. Pada penelitian ini dilakukan prediksi penjualan obat dengan melakukan perbandingan metode *machine learning* yaitu *Long Short Term Memory* (LSTM) dan metode statistik yaitu *Auto Regressive Moving Average* (ARIMA) dengan menggunakan 5 jenis data obat, kemudian hasil prediksi akan dibandingkan dengan metode evaluasi *Root Mean Square Error* (RMSE). Berdasarkan hasil pengujian yang dilakukan, didapatkan nilai RMSE metode LSTM lebih unggul dibandingkan nilai RMSE pada metode ARIMA dengan selisih perbandingan RMSE untuk model LSTM dan ARIMA untuk obat Acitral Suspensi yaitu 4.37, obat Paracetamol Sirup dengan nilai 38.93, obat Omepros dengan selisih perbandingan RMSE yaitu 13.60, untuk obat Calcium D Redoxon dengan selisih RMSE 1.25, dan obat Noza Tab@100 dengan selisih perbandingan RMSE yaitu 11.15. Walaupun model LSTM menghasilkan hasil RMSE yang rendah dibandingkan dengan RMSE pada model ARIMA, model LSTM yang telah dibuat tidak disarankan untuk digunakan karena mengalami *overfitting*, hal ini karena model LSTM tidak dapat memprediksi dengan akurat untuk data testing maupun terhadap jenis data baru.

**Kata kunci:** Prediksi Penjualan, Apotek, *Long Short Term Memory* (LSTM), *Auto Regressive Moving Average* (ARIMA)

## COMPARATIVE ANALYSIS OF METHOD TIME SERIES FORECASTING FOR PREDICTION OF DRUG SALES IN PHARMACY (CASE STUDY: CHEMICAL FARMA TAKOMA PHARMACY)

### Abstract

*The existence of a pharmacy is very important for the community to meet the needs of medicines. Kimia Farma Pharmacy is a company engaged in the pharmaceutical sector health care company and has the largest pharmacy network in Indonesia. Predicting drug sales at pharmacies is one of the priority activities in determining future sales, this aims to control stocks so that there are no excess and shortages of stock and prevent the unavailability of drugs that consumers want to buy. In this study, prediction of drug sales was carried out by comparing methods machine learning that is Long Short Term Memory (LSTM) and statistical methods ie Auto Regressive Moving Average (ARIMA) using 5 types of drug data, then the prediction results will be compared to with the evaluation method Root Mean Square Error (RMSE). Based on the results of the tests performed, the RMSE value of the LSTM method was superior to the RMSE value of the ARIMA method with a difference in the RMSE ratio for the LSTM and ARIMA models for the Acitral Suspension drug which was 4.37, for Paracetamol Syrup with a value of 38.93, for Omepros with a difference in RMSE ratio of 13.60, for Calcium D Redoxon with a RMSE difference of 1.25, and Noza Tab@100 with a RMSE difference of 11.15. Although the LSTM model produces lower RMSE results compared to the RMSE in the ARIMA model, the LSTM model that has been created is not recommended to be used because it experiences overfitting, this is because the LSTM model cannot predict accurately for data testing or for new data types..*

**Keywords:** Sales Prediction, Pharmacy, *Long Short Term Memory* (LSTM), *Auto Regressive Moving Average* (ARIMA)

## 1. PENDAHULUAN

Apotek merupakan tempat dilakukannya pelayanan kefarmasian serta tempat penyediaan obat-obatan yang dijual-belikan kepada masyarakat [1], oleh karena itu persediaan obat di apotek merupakan hal penting yang perlu diperhatikan dalam proses penjualan [2].

Dalam proses penjualan, tentunya mengalami kenaikan maupun penurunan penjualan, oleh karena itu pihak apotek perlu mengetahui informasi persediaan dan penjualan. Hal ini berpengaruh dalam mengatur jumlah persediaan untuk bulan-bulan berikutnya serta meminimalisir ketidaktersediaan barang yang dibutuhkan konsumen.

Prediksi merupakan proses meramalakan kondisi di masa depan dengan melakukan pengujian terhadap kondisi di masa lampau. Prediksi penjualan bertujuan untuk memperkirakan besaran penjualan serta dapat digunakan dalam pengambilan keputusan dan kebijakan berdasarkan hasil prediksi yang diperoleh [3].

Prediksi penjualan dapat digunakan untuk memperkirakan stok persediaan agar dapat memenuhi kebutuhan kosumen. Dengan memprediksi penjualan, dapat membantu pihak apotek untuk mencapai penjualan dan keuntungan yang diharapkan serta menghindari resiko yang tidak diinginkan [3].

Terdapat tiga kategori model peramalan secara umum, yaitu model kualitatif, kausal, dan *time series* [4]. Metode peramalan *time series* dibagi menjadi dua yaitu metode statistik dan *machine learning* (ML) [5]. Metode statistik beberapa diantaranya yakni *exponential smoothing*, *moving average*, regresi, dan yang paling sering digunakan adalah *Autoregressive moving average* (ARIMA) atau *Box Jenkins* [6]. *Deep Learning* (DL) adalah salah satu cabang ilmu *Machine Learning* [7] dan salah satu golongan dari *deep learning* yaitu *Recurrent Neural Network* (RNN) [8]. *Long Short-Term Memory* (LSTM) merupakan pengembangan dari RNN yang mampu mempelajari ketergantungan jangka panjang [9].

Metode ARIMA merupakan metode statistik yang digunakan untuk menganalisis dan meramalkan data *time series* yang telah banyak digunakan untuk melakukan peramalan penjualan [10]. LSTM merupakan metode *machine learning* yang juga mampu untuk mengelola data *times series* [11]

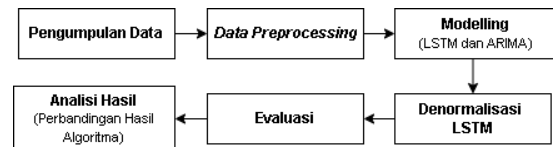
Beberapa penelitian yang menggunakan LSTM dan ARIMA untuk melakukan prediksi penjualan diantaranya yaitu penelitian yang dilakukan oleh [12]. Penelitian ini dilakukan untuk prediksi sediaan farmasi menggunakan data penjualan dan pembelian menggunakan metode LSTM dan metode statistik lainnya. Hasil menunjukkan metode LSTM memiliki *Mean Absolute Percentage Error* (MAPE) yang paling rendah jika dibanding dengan metode lainnya.

Penelitian lainnya oleh [13] untuk melakukan peramalan permintaan pada kasus pemodelan *perisahble product* Retail Sayur Tosaga dengan

menggunakan metode LSTM dan ARIMA. Hasil penelitian menunjukkan bahwa pada metode LSTM menghasilkan RMSE yang lebih rendah, namun *learning curve* yang dihasilkan termasuk dalam kategori *overfit*. Oleh karena itu metode yang lebih optimal digunakan ialah metode ARIMA.

## 2. METODE PENELITIAN

Dalam penelitian ini dilakukan sebuah eksperimen untuk memprediksi penjualan obat dengan menggunakan perbandingan 2 metode yaitu LSTM dan ARIMA menggunakan bahasa pemrograman Python. Tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

Tahapan penelitian dimulai dari pengumpulan data dan melakukan *preprocessing* yang terdiri dari *data cleaning* dan *data transformation* atau normalisasi data, dilanjutkan dengan pembuatan model dan dilakukan *training* menggunakan metode LSTM dan ARIMA. Pada tahapan pembuatan model LSTM dan ARIMA dilakukan dengan menggunakan beberapa skenario pengujian dan perubahan terhadap parameter LSTM maupun ARIMA. Selanjutnya hasil dari *training* model digunakan untuk melakukan prediksi. Setelah dilakukan proses pelatihan dan hasil prediksi telah didapatkan dari model yang telah dibangun, selanjutnya dilakukan *post-processing* atau denormalisasi data. Kemudian pada tahap akhir dilakukan evaluasi terhadap kinerja model dengan melihat nilai RMSE dari kedua metode dan melakukan analisis terhadap model yang ada.

### 2.1. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan berupa data-data hasil penjualan di Kimia Farma Apotek Takoma. Data yang didapatkan dari pihak apotek ialah sebanyak 5 jenis data obat dengan rentang penjualan yang berbeda-beda untuk tiap jenis obat. Tabel 1 adalah sampel data obat dan tabel 2 merupakan hasil penjualan obat di Apotek. Untuk mempermudah penyebutan untuk kelima jenis obat ini akan diinisialkan dengan obat 1, obat 2, obat 3, obat 4, dan obat 5.

Tabel 1 Struktur *Dataset* Penjualan Obat

No	Nama Obat
1	Acitral Suspensi
2	Paracetamol Sirup
3	Omepros
4	Calcium D Redoxon Fruit

5 Noza

Tabel 2 Struktur *Dataset* Penjualan Obat  
Contoh Nilai Atribut

Atribut	Contoh Nilai Atribut		
Tanggal	18/11/2019	19/11/2019	20/11/2019
Jumlah Masuk	28	120	-
Keterangan Masuk	Receive Order ...	Receive Order ...	-
Satuan	BT	BT	BT
Jumlah Keluar	0	0	1
Keterangan Keluar	-	-	Order T11120 ...
Jumlah <i>Stock</i>	28	148	147
No. <i>Batch</i>	S031I033	S031I033	S031I033
Exp.Date	1/7/2022	1/7/2022	1/7/2022

## 2.2. Data Preprocessing

Tahap *preprocessing* merupakan langkah awal sebelum data diolah. Pada tahap ini akan dilakukan analisis terhadap variabel pada dataset. Analisis data dilakukan untuk mengurangi kompleksitas data yang tidak penting, menghindari adanya *noise*, mendeteksi dan menghapus elemen yang tidak relevan. Pada tahap ini akan dilakukan data *cleaning* dan *data transformation*.

### 2.2.1. Data Cleaning

Pada tahapan ini, pembersihan data dilakukan untuk mengatasi *missing value* dan *noise* pada data.

### 2.2.2. Data Transformation

Pada tahap ini, data akan diubah menjadi bentuk yang sesuai dengan metode analisis. Data akan dinormalisasi untuk menskalakan nilai data dalam rentang tertentu (-2.0 hingga 1.0 atau 0.0 hingga 1.0). Proses transformasi data dilakukan dengan teknik *min max scaling* menggunakan *library python scikit learn MinMaxScaler*. Adapun perhitungan *min max scaling* dapat dilihat pada persamaan (1).

$$x' = \frac{(x_i - \min_x)}{\max_x - \min_x} \quad (1)$$

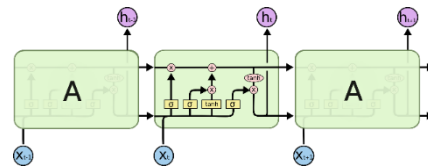
Dimana  $x'$  adalah data setelah ternormalisasi dan  $x_i$  merupakan data yang akan dinormalisasi.  $\min_x$  dan  $\max_x$  adalah data terkecil dan terbesar dari keseluruhan data.

## 2.3. Long Short Term Memory

Long Short-Term Memory (LSTM) merupakan pengembangan dari *Recurrent Neural Network* (RNN). LSTM mampu mempelajari

ketergantungan jangka panjang dan dapat mengingat informasi dalam jangka waktu yang lama [9].

LSTM dirancang untuk meningkatkan algoritma RNN yang mampu memecahkan masalah gradien yang hilang dengan menambahkan *cell state* untuk mengingat atau melupakan data. *Cell state* memuat struktur yang disebut *cell gate* atau gerbang sel. Setiap gate terdiri dari empat bagian yaitu *input gate*, *forget gate*, *memory-cell state gate* dan *output gate* [14]. Dalam membuat prediksi, LSTM menggunakan data historis (masa lampau) dan data saat ini [15]. Arsitektur LSTM dapat dilihat pada gambar 2.



Gambar 2. Arsitektur LSTM

### 1. Forget Gate

*Gate* ini berfungsi untuk mengolah informasi setiap data inputan dan memilih data mana saja yang perlu disimpan atau dibuang pada *memory cells*. Langkah pertama yang dilakukan ialah memutuskan informasi mana yang akan dihapus dari *cell state*. *Sigmoid layer* yang bernama "*forget gate layer*" bertugas untuk membuat keputusan ini. *Forget gate layer* kemudian memproses  $h_{t-1}$  dan  $x_t$  sebagai *input*, dan *output* yang dihasilkan berupa angka 0 atau 1 pada *cell state*  $c_{t-1}$  [16]. Rumus untuk menghitung forget gate dapat dilihat pada persamaan 2.

$$f_t = \sigma(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + b_f) \quad (2)$$

dengan:

- $f_t$  = *Forget gate*.
- $\sigma$  = Fungsi aktivasi *sigmoid*.
- $x_t$  = Input pada waktu  $t$
- $W_f$  = Nilai bobot untuk *forget gate*.
- $h_{t-1}$  = Nilai *output* sebelumnya orde ke  $t$
- $b_f$  = Nilai *bias* pada *forget gate*.

### 2. Input Gate

Tahapan selanjutnya pada LSTM ialah memutuskan informasi baru apa yang akan di simpan di *cell state*. *Input gate* memiliki dua *gates* yang akan dilaksanakan. Pertama, lapisan *sigmoid* yang disebut "*input gate layer*" menentukan nilai mana yang akan diperbaharui. Selanjutnya, lapisan *tanh* membuat vektor nilai kandidat baru  $\tilde{C}_t$ , yang dapat ditambahkan pada *cell state* lalu keduanya digabungkan untuk pembaharuan ke *cell state* [16]. Rumus dari *input gate layer* ( $i_t$ ) dan nilai kandidat baru  $\tilde{C}_t$ . Rumus perhitungan *input gate* dan kandidat *cell state* dapat dilihat pada persamaan 3 dan 4.

$$i_t = \sigma((W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + b_i)) \quad (3)$$

$$\tilde{c}_t = \tanh((W_{cx} \cdot x_t + W_{ch} \cdot h_{t-1} + b_c)) \quad (4)$$

dengan:

- $i_t$  = Input gate.
- $\tilde{c}_t$  = Nilai baru yang dapat ditambahkan ke cell state
- $\tanh$  = Fungsi aktivasi  $\tanh$
- $W_i$  = Nilai bobot pada input gate.
- $W_c$  = Nilai bobot pada cell aktivasi.
- $b_i$  = Nilai bias pada input gate.
- $b_c$  = Nilai bias pada cell aktivasi.

3. Cell state  
 Pada tahapan ini, cell state lama ( $c_{t-1}$ ) akan diperbaharui menjadi cell state baru  $c_t$ . Dengan mengalikan  $c_{t-1}$  lama dengan  $f_t$ , untuk menghapus informasi yang telah ditentukan sebelumnya pada tahapan forget gate layer. Selanjutnya tambahkan dengan  $i_t * \tilde{c}_t$ . Ini adalah nilai kandidat baru dan akan digunakan untuk memperbaharui cell state [16]. Rumus dari  $c_t$  dapat dilihat pada persamaan 5.

$$c_t = (f_t * c_{t-1} + i_t * \tilde{c}_t) \quad (5)$$

- $c_t$  = Cell state.
- $c_{t-1}$  = Output hidden state sebelumnya atau state pada waktu t-1.

#### 4. Output Gate

Langkah terakhir dari LSTM bertujuan untuk memutuskan hasil keluaran. Output harus sesuai dengan cell state yang telah diproses sebelumnya. Pertama lapisan sigmoid dijalankan untuk menentukan bagian cell state mana yang menjadi output. Kemudian, menempatkan cell state melalui tanh (untuk mendorong nilai antara -1 dan 1) dan mengalikannya dengan output dari sigmoid gate, sehingga hanya mengeluarkan sesuai dengan apa yang diputuskan sebelumnya [16]. Rumus perhitungan output gate dan nilai output orde ke t dapat dilihat pada persamaan 3 dan 4.

$$o_t = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \quad (6)$$

$$h_t = o_t * \tanh(c_t) \quad (7)$$

dengan:

- $o_t$  = Output gate.
- $c_t$  = cell state.
- $W_o$  = Nilai bobot pada output gate.
- $b_o$  = Nilai bias pada output gate.
- $h_t$  = Nilai output orde t

#### 2.4. Autoregressive Integrated Moving Average

Autoregressive Intergrated Moving Average (ARIMA) merupakan salah satu metode time series yang sangat baik digunakan untuk melakukan peramalan jangka pendek, peramalan dilakukan dengan nilai masa lalu dan sekarang dari variabel dependen untuk menghasilkan peramalan yang akurat. Sementara untuk peramalan jangka panjang,

ketepatan peramalan yang dihasilkan kurang baik, cenderung flat (mendatar/konstan). Dalam membuat peramalan, model ARIMA mengabaikan variabel independen [17].

##### 1. Autoregressive

Bentuk umum dari model autoregressive dengan ordo p (AR(p)) atau model ARIMA(P,0,0). Model AR memprediksi variabel terikat ( $y_t$ ) dengan membentuk fungsi linear dari sejumlah ( $z_t$ ) sebelumnya ( $z_{t-1}, z_{t-2}$ , dst). Model umum AR dapat dilihat pada persamaan 8.

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + \alpha_t \quad (8)$$

Dengan:

- $Z_t$  : Besar data pada periode t
- $\phi_1, \phi_2, \dots, \phi_p$  : Koefisien orde p (autoregressive)
- $\alpha_t$  : Galat pada waktu t

##### 2. Moving Average

Bentuk umum dari model moving average dengan ordo q (MA(q)) atau model ARIMA (0,0,q). Model MA merupakan model untuk prediksi variabel terikat ( $y_t$ ) yang dipengaruhi oleh nilai residual periode sebelumnya. Model umum MA dapat dilihat persamaan 8.

$$Z_t = \alpha_t - \theta_1 \alpha_{t-1} + \theta_2 \alpha_{t-2} - \dots + \theta_q \alpha_{t-q} \quad (9)$$

dengan:

- $\theta_1, \theta_2, \dots, \theta_q$  : Koefisien orde q (moving average)
- $\alpha_1, \alpha_{t-1}, \dots, \alpha_{t-q}$  : Galat pada waktu t

##### 3. Autoregressive Moving Average

Pada Metode ARMA ordo p dan q (AR(p) dan MA(q)) adalah gabungan antara Autoregressive Model (AR) dan Moving Average (MA). Model MA merupakan model untuk prediksi variabel terikat ( $z_t$ ) yang digabungkan antara model AR dan MA. Model umum ARMA dapat dilihat pada persamaan 10.

$$Z_t = \theta_1 Z_{t-1} + \theta_2 Z_{t-2} \alpha_t + \dots + \theta_p Z_{t-p} + \alpha_t - \phi_1 \alpha_{t-1} - \phi_2 \alpha_{t-2} - \dots - \phi_q \alpha_{t-q} \quad (10)$$

#### 2.5. Denormalisasi

Setelah dilakukan proses pelatihan dan diperoleh hasil prediksi dari model, selanjutnya dilakukan post-processing, yaitu pengembalian data ke bentuk atau nilai aslinya [18]. Rumus perhitungannya dapat dilihat pada persamaan 11.

$$x = x'(max_x - min_x) + min_x \quad (11)$$

dengan:

- $x$  : Nilai hasil denormalisasi

#### 2.6. Evaluasi Model

Untuk melihat hasil peramalan dapat dilakukan dengan mengamati selisih nilai hasil peramalan dan nilai aktual [19]. Root Mean Square Error (RMSE), adalah jumlah dari kesalahan kuadrat atau selisih antara nilai sebenarnya dengan nilai prediksi yang

telah ditentukan [20]. Rumus perhitungan RMSE dapat dilihat pada persamaan 12.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_t - F_t)^2}{n}} \quad (12)$$

dengan:

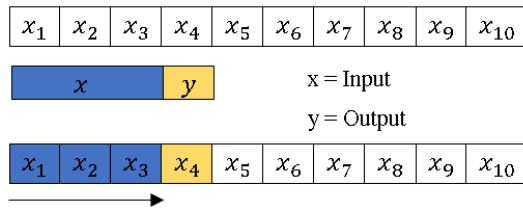
- $X_t$  : Nilai data aktual
- $F_t$  : Nilai hasil peramalan
- $n$  : Banyaknya data
- $\sum$  : *Summation* (Jumlahkan keseluruhan nilai)

### 2.7. Sliding Window

Metode *Sliding window* merupakan teknik pembentukan struktur pada data time series [21]. Metode *sliding window* dapat menghasilkan data dengan langkah waktu saat ini. Misalnya jika data Januari adalah nilai target yang akan diprediksi, maka data n-dimensi dapat dibuat dari data n bulan sebelumnya untuk membuat prediksi [22].

Misalkan *window size* = w, data ke t diprediksi dengan melihat w dari data sebelumnya. Sebagai contoh, dapat dilihat pada penjelasan berikut.

$$Sliding\ window = 3$$



Gambar 3. Ilustrasi *Sliding Window*

## 3. HASIL DAN PEMBAHASAN

### 3.1 Dataset

*Dataset* yang telah dilakukan *Preprocessing*, selanjutnya akan dibagi menjadi data *training* sebanyak 70% dan data *testing* sebanyak 30%. *Dataset* dapat dilihat pada tabel 3.

No	Obat	Data training (70%)	Data testing (30%)
1	Obat 1 (173 Data)	121	52
2	Obat 2 (173 Data)	121	52
3	Obat 3 (168 Data)	117	51
4	Obat 4 (156 Data)	109	47
5	Obat 5 (53 Data)	37	16

### 3.2 Implementasi Algoritma LSTM

Untuk tahapan implementasi LSTM dimulai dari mengimport *dataset* dan melakukan *preprocessing* data kemudian dilanjutkan dengan normalisasi untuk memperoleh data dengan rentang (0,1). Setelah itu dengan terapkan fungsi *sliding window* dan membentuk *shape array* menjadi 3 dimensi dengan memanfaatkan fungsi *reshape* pada *library numpy*.

### 3.2.1 Normalisasi

Dengan menggunakan rumus *min-max scaling* pada persamaan 1 dan untuk contoh perhitungannya akan digunakan pada data latih untuk obat 4 maka diperoleh hasil normalisasi seperti pada tabel 4.

Tanggal	Normalisasi
6/30/2019	0.090909
7/7/2019	0
7/14/2019	0.090909
⋮	⋮
6/5/2022	0
6/12/2022	0
6/19/2022	0

### 3.2.2 Perhitungan LSTM

Pada contoh perhitungan ini menggunakan contoh data pada tabel 4 dengan nilai *weight* dan *bias* yang sudah ditentukan secara acak oleh program. Untuk nilai *input*, *weight*, dan *bias* dapat dilihat pada tabel 5.

Tabel 5 Nilai *Input*, *Weight*, dan *Bias*

Variabel	$t = 0$
$x_t$	0.09090
$W_f$	-0.10777
$W_i$	-0.08255
$W_c$	0.12248
$W_o$	-0.10439
$b_f$	0.04181
$b_i$	-0.03696
$b_c$	-0.03114
$b_o$	0.02078

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot x_t + W_f \cdot h_{t-1} + b_f) \\
 &= \sigma((-0.107) \cdot 0.090 + ((-0.107) \cdot 0) + 0.041) \\
 &= \sigma(0.032) = \frac{1}{1 + e^{-(0.032)}} = 0.508 \\
 i_t &= \sigma(W_i \cdot x_t + W_i \cdot h_{t-1} + b_i) \\
 &= \sigma((-0.082) \cdot 0.090 + ((-0.082) \cdot 0) + (-0.036)) \\
 &= \sigma(-0.044) \\
 &= \frac{1}{1 + e^{-(-0.044)}} = 0.489 \\
 \tilde{C}_t &= \tanh(W_c \cdot x_t + W_c \cdot h_{t-1} + b_c) \\
 &= \tanh((0.122 \cdot 0.090) + (0.122 \cdot 0) + (-0.031)) \\
 &= \tanh(-0.020) = \frac{e^{2 \times (-0.020)} - 1}{e^{2 \times (-0.020)} + 1} = 0.019
 \end{aligned}$$

$$\begin{aligned}
 o_t &= (W_o \cdot x_t + W_o \cdot h_{t-1} + b_o) \\
 &= \sigma((-0.104 \cdot 0.090) + ((-0.104) \cdot 0) + 0.020) \\
 &= \sigma(0.011) = \frac{1}{1 + e^{-(0.011)}} = 0.502 \\
 c_t &= f_t \times c_{t-1} + i_t \times \tilde{C}_t \\
 &= (0.508 \times 0) + (0.489 \times (-0.019)) \\
 &= -0.009
 \end{aligned}$$

$$\begin{aligned}
 h_t &= o_t * \tanh(c_t) \\
 &= 0.502 \times \tanh(-0.009) = -0.004
 \end{aligned}$$

Nilai  $h_t$  dan  $c_t$  akan digunakan sebagai *input* pada proses selanjutnya sesuai dengan jumlah *epoch* yang ditentukan.

### 3.2.3 Skenario Pengujian

Skenario pengujian digunakan untuk mengukur keberhasilan model dalam melakukan prediksi dan melihat nilai kesalahan hasil prediksi. Selain itu, akan dilakukan analisa dampak parameter terhadap akurasi yang didapatkan. Parameter yang diuji yaitu:

- Jumlah pola *time steps*: 4,8, dan 12
- Jumlah *neuron hidden*: 64,128 dan 256, untuk jumlah *neuron hidden* kedua dan *dense layer* akan disamakan untuk semua model yaitu 150 untuk *neuron hidden* kedua dan *dense layer* 100.
- *Max epoch*: 250, 300 dan 500

#### 1. Pengujian Obat 1

Parameter jumlah *time step* yang digunakan pada model LSTM untuk obat 1 yaitu 4,8, dan 12.

Tabel 5 Nilai Kombinasi Parameter Pola Time Steps

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.141	0.115
8	64	250	0.133	0.098
12	64	250	0.131	0.099

Tabel 5 menunjukkan kombinasi parameter yang digunakan dan akurasi yang diperoleh. Terlihat bahwa model dengan pola *time steps* 12 menghasilkan hasil yang lebih optimal dibandingkan pola *time steps* lainnya.

Untuk parameter jumlah neuron yang digunakan untuk model LSTM untuk obat 1 ialah 64,128, dan 256.

Tabel 6 Nilai Kombinasi Parameter Neuron Hidden

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.141	0.116
8	128	250	0.135	0.103
12	256	250	0.097	0.123

Tabel 6 menunjukkan kombinasi parameter yang digunakan dan akurasi yang diperoleh. Terlihat bahwa model dengan jumlah *neuron hidden* 256 menghasilkan nilai yang lebih optimal dibandingkan dengan model dengan *neuron hidden* 64 dan 128.

Untuk parameter jumlah max epoch yang digunakan pada model LSTM untuk obat 1 yaitu 250, 300 dan 500.

Tabel 7 Nilai Kombinasi Parameter Max Epoch

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
12	256	250	0.091	0.111
12	256	300	0.082	0.122
12	256	500	0.039	0.119

Tabel 7 menunjukkan kombinasi parameter yang digunakan dan akurasi yang diperoleh. Terlihat bahwa model dengan jumlah *max epoch* 500 lebih optimal dibandingkan dengan model dengan jumlah max epoch 250 dan 300.

Berdasarkan hasil dari eksperimen pengujian dengan menggunakan parameter yang telah ditentukan dan diujikan pada dataset obat 1, menunjukkan bahwa model yang paling optimal ialah model dengan jumlah timesteps 12, *neuron hidden* 256 dan menggunakan max epoch sebanyak 500.

#### 2. Pengujian Obat 2

Parameter jumlah *time step* yang digunakan pada model LSTM untuk obat 1 yaitu 4,8, dan 12. Tabel 8 menunjukkan nilai kombinasi parameter *time steps*.

Tabel 8 Nilai Kombinasi Parameter Pola Time Steps

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.114	0.033
8	64	250	0.072	0.026
12	64	250	0.071	0.021

Pada tabel 8 terlihat bahwa model dengan pola time 12 menghasilkan hasil yang lebih optimal dibandingkan pola time steps lainnya.

Untuk parameter jumlah neuron yang digunakan untuk model LSTM untuk obat 2 ialah 64,128, dan 256.

Tabel 9 Nilai Kombinasi Parameter Neuron Hidden

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.114	0.024
8	128	250	0.054	0.016
12	256	250	0.071	0.021

Tabel 9 menunjukkan kombinasi parameter yang digunakan dan akurasi yang diperoleh. Terlihat bahwa model dengan jumlah neuron hidden 128 menghasilkan nilai yang lebih optimal dibandingkan dengan model dengan neuron hidden 64 dan 256.

Untuk parameter jumlah max epoch yang digunakan pada model LSTM untuk obat 1 yaitu 250, 300 dan 500.

Tabel 10 Nilai Kombinasi Parameter Max Epoch

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
12	256	250	0.091	0.111
12	256	300	0.082	0.122
12	256	500	0.039	0.119

8	128	250	0.063	0.019
8	128	300	0.044	0.020
8	128	500	0.027	0.016

Tabel 10 menunjukkan kombinasi parameter yang digunakan dan akurasi yang diperoleh. Terlihat bahwa model dengan jumlah *max epoch* 500 lebih optimal dibandingkan dengan model dengan jumlah *max epoch* 250 dan 300.

Berdasarkan hasil dari eksperimen pengujian dengan menggunakan parameter yang telah ditentukan yang diujikan pada *dataset* obat 2, menunjukkan bahwa model yang paling optimal ialah model dengan jumlah *timesteps* 8, *neuron hidden* 128 dan menggunakan *max epoch* sebanyak 500.

### 3. Pengujian Obat 3

Parameter jumlah *time step* yang digunakan pada model LSTM untuk obat 2 yaitu 4,8, dan 12.

Tabel 11 Nilai Kombinasi Parameter Pola Time Steps

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.204	0.187
8	64	250	0.178	0.212
12	64	250	0.175	0.169

Berdasarkan tabel 11 menunjukan bahwa model dengan pola time 12 menghasilkan hasil yang lebih optimal dibandingkan pola *time steps* lainnya.

Untuk parameter jumlah neuron yang digunakan untuk model LSTM untuk obat 2 ialah 64,128, dan 256.

Tabel 12 Nilai Kombinasi Parameter Neuron Hidden

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.195	0.193
8	128	250	0.170	0.218
12	256	250	0.170	0.187

Berdasarkan tabel 12 menunjukkan bahwa model dengan jumlah *neuron hidden* 256 menghasilkan nilai yang lebih optimal dibandingkan dengan model dengan *neuron hidden* 64 dan 128.

Untuk parameter jumlah *max epoch* yang digunakan pada model LSTM untuk obat 2 yaitu 250, 300 dan 500.

Tabel 13 Nilai Kombinasi Parameter Max Epoch

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
12	256	250	0.159	0.209
12	256	300	0.166	0.270
12	256	500	0.092	0.323

Berdasarkan tabel 13 menunjukkan model dengan jumlah *max epoch* 500 lebih optimal dibandingkan dengan model dengan jumlah *max epoch* 250 dan 300.

Berdasarkan hasil dari eksperimen pengujian dengan menggunakan parameter yang telah ditentukan yang diujikan pada *dataset* obat 3,

menunjukkan bahwa model yang paling optimal ialah model dengan jumlah *timesteps* 12, *neuron hidden* 256 dan menggunakan *max epoch* sebanyak 500.

### 4. Pengujian Obat 4

Parameter jumlah *time step* yang digunakan pada model LSTM untuk obat 1 yaitu 4,8, dan 12.

Tabel 14 Nilai Kombinasi Parameter Pola Time Steps

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.100	0.062
8	64	250	0.092	0.037
12	64	250	0.062	0.046

Berdasarkan tabel 14 menunjukan bahwa model dengan pola time 12 menghasilkan hasil yang lebih optimal dibandingkan pola *time steps* lainnya.

Untuk parameter jumlah neuron yang digunakan untuk model LSTM untuk obat 1n ialah 64,128, dan 256.

Tabel 15 Nilai Kombinasi Parameter Neuron Hidden

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.099	0.059
8	128	250	0.090	0.048
12	256	250	0.061	0.044

Berdasarkan tabel 15 menunjukkan bahwa model dengan jumlah *neuron hidden* 256 menghasilkan nilai yang lebih optimal dibandingkan dengan model dengan *neuron hidden* 64 dan 128.

Untuk parameter jumlah *max epoch* yang digunakan pada model LSTM untuk obat 1 yaitu 250, 300 dan 500.

Tabel 16 Nilai Kombinasi Parameter Max Epoch

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
12	256	250	0.064	0.049
12	256	300	0.058	0.039
12	256	500	0.060	0.039

Berdasarkan tabel 16 menunjukkan model dengan jumlah *max epoch* 500 lebih optimal dibandingkan dengan model dengan jumlah *max epoch* 250 dan 300.

Berdasarkan hasil dari eksperimen pengujian dengan menggunakan parameter yang telah ditentukan yang diujikan pada *dataset* obat 4, menunjukkan bahwa model yang paling optimal ialah model dengan jumlah *timesteps* 12, *neuron hidden* 256 dan menggunakan *max epoch* sebanyak 500. Pengujian Obat 4.

### 5. Pengujian Obat 5

Parameter jumlah *time step* yang digunakan pada model LSTM untuk obat 3 yaitu 4,8, dan 12.

Tabel 17 Nilai Kombinasi Parameter Pola Time Steps

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
-----------------	---------------	-----------	---------------	--------------

4	64	250	0.010	0.109
8	64	250	0.011	0.091
12	64	250	0.024	0.645

Berdasarkan tabel 17 menunjukkan bahwa model dengan pola time 12 menghasilkan hasil yang lebih optimal dibandingkan pola *time steps* lainnya.

Untuk parameter jumlah neuron yang digunakan untuk model LSTM untuk obat 1n ialah 64,128, dan 256.

Tabel 18 Nilai Kombinasi Parameter Neuron Hidden

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
4	64	250	0.012	0.107
8	128	250	0.010	0.100
12	256	250	0.016	0.093

Berdasarkan tabel 18 menunjukkan bahwa model dengan jumlah *neuron hidden* 128 menghasilkan nilai yang lebih optimal dibandingkan dengan model dengan *neuron hidden* 64 dan 256.

Untuk parameter jumlah *max epoch* yang digunakan pada model LSTM untuk obat 3 yaitu 250, 300 dan 500

Tabel 19 Nilai Kombinasi Parameter Max Epoch

Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
8	128	250	0.011	0.099
8	128	300	0.013	0.103
8	128	500	0.009	0.105

Berdasarkan tabel 19 menunjukkan model dengan jumlah max epoch 250 lebih optimal dibandingkan dengan model dengan jumlah max epoch 300 dan 500.

Berdasarkan hasil dari eksperimen pengujian dengan menggunakan parameter yang telah ditentukan yang diujikan pada dataset obat 5, menunjukkan bahwa model yang paling optimal ialah model dengan jumlah *timesteps* 8, *neuron hidden* 128 dan menggunakan *max epoch* sebanyak 250.

Pada tabel 20 menyajikan model LSTM yang terbaik yang telah diperoleh dari masing-masing obat.

Tabel 20 Nilai Kombinasi Parameter Untuk Seluruh Obat

Obat	Pola Time Steps	Neuron Hidden	Max Epoch	RMSE Training	RMSE Testing
Obat 1	12	256	500	0.039	0.119
Obat 2	8	128	500	0.016	0.019
Obat 3	12	256	500	0.092	0.323
Obat 4	12	256	300	0.059	0.039
Obat 5	8	128	250	0.011	0.099

### 3.3 Implementasi Algoritma ARIMA

Langkah-langkah yang akan dilakukan ialah melakukan identifikasi model, penentuan estimasi parameter, evaluasi model dan melakukan peramalan.

#### 3.3.1 Identifikasi Model

Identifikasi dilakukan eksplorasi terhadap pola data dari kelima *dataset* obat.

##### 1. Plot Data Timeseries

Kita dapat mengidentifikasi jenis data time series berdasarkan pola grafik. Data juga dapat divisualisasikan menggunakan metode yang disebut dekomposisi deret waktu yang memungkinkan untuk menguraikan deret waktu menjadi tiga komponen berbeda yaitu tren, musiman, dan *noise*.

Gambar 4 Visualisasi data penjualan dan dekomposisi data

Pada gambar 4 menampilkan visualisasi untuk obat 1 dan pada grafik dekomposisi data, Deret waktu tidak menunjukkan trend, ataupun musiman. Pada gambar menunjukkan pola musiman tidak terlihat bagus, plot musiman dapat diamati dengan pola siklus frekuensi tetap yang bagus Namun kita akan mencoba mempertimbangkan peramalan dengan musiman yaitu SARIMA.

##### 2. Melakukan Uji Stasioneritas

Uji stasioner dapat dilakukan dengan menggunakan pengujian *Augmented Dickey-Fuller* atau ADF untuk melihat apakah data tersebut stasioner atau tidak. Untuk melihat kestasioneran data, dilihat dengan nilai *p-value* < 0.05. Data stasioner jika nilai *p-value* di bawah 0.05.

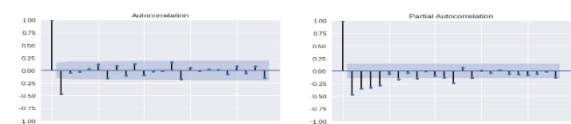
ADF Statistic: -6.671486  
p-value: 4.575951239318953e-09

Gambar 6 Adfuller Test

Gambar 6 menunjukkan nilai *p-value* < 0.05 artinya data stasioner. Jika data belum stasioner maka perlu dilakukan *differencing*. Setelah melakukan *differencing* satu kali dan data sudah stasioner maka ini menunjukkan bahwa dapat menggunakan nilai 1 pada parameter d.

#### 3.3.2 Estimasi Parameter Model

Jika data stasioner maka nilai orde d = 0, dan untuk data yang dilakukan *differencing* satu kali, nilai orde d = 1. Begitu juga untuk SARIMA. Selanjutnya, perlu dilihat korelogram *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF) pada data yang telah stasioner. Setelah melihat diagram ACF maupun PACF maka kita dapat menentukan nilai p dan q dan menentukan beberapa estimasi parameter.





Gambar 5 Plot ACF dan PACF

Berdasarkan gambar 5 terlihat bahwa plot ACF untuk data obat Acitral Suspensi mengalami *cut off* (menurun drastis pada lag tertentu) pada lag 1. Pada plot PACF mengalami *dying down*. Dapat disimpulkan bahwa adanya pola *autoregressive* (MA). Untuk menentukan model yang terbaik yang akan digunakan untuk peramalan, dipilih model ARIMA yang memenuhi semua asumsi yaitu parameter signifikan, residualnya memenuhi asumsi *white noise* dan data terdistribusi secara normal, dan mempunyai nilai *Akaike info criterion* (AIC) yang terkecil. Model dapat diestimasi dengan melihat pola ACF dan PACF.

### 3.3.3 Estimasi Parameter Model

Model yang parameternya telah memenuhi uji signifikan, selanjutnya akan dilakukan beberapa pemeriksaan diagnostik terhadap residual dengan mengevaluasi Prob(Q) dan Prob(JB) masing-masing menerapkan Tes *Ljung-Box* dan *Jarque-Bera*. Setelah itu akan dipilih model dengan nilai AIC terkecil. Tabel 9 menunjukkan model yang dipilih untuk melakukan peramalan.

Tes *Ljung-Box* : Hipotesis nol = Tidak ada korelasi dalam residual.

Tes *Jarque-Bera* : Hipotesis nol = Residu terdistribusi secara normal.

Tabel 21 Nilai Kombinasi Parameter Untuk Seluruh Obat

Variabel Data	Model
	ARIMA(0,1,1)
Obat 1	SARIMA(0,1,1)(0,1,1,12)
Obat 2	SARIMA(0,0,0)(1,1,1,12)
Obat 3	SARIMA(0,0,0)(1,1,0,12)
Obat 4	SARIMA(0,0,0)(1,1,0,12)
Obat 5	ARIMA(1,0,0)

### 3.3.4 Melakukan peramalan (*forecasting*)

Setelah didapatkan model terbaik dari seluruh data obat, selanjutnya dilakukan peramalan.

Tabel 22 Nilai Kombinasi Parameter Untuk Seluruh Obat

Variabel Data	RMSE
Obat 1	4.37082
Obat 2	38.93587
Obat 3	13.60198
Obat 4	1.49092
Obat 5	10.41667

Pada tabel 22 menampilkan hasil perhitungan kesalahan peramalan dengan RMSE terhadap 5 jenis data obat yang diperoleh dari model ARIMA.

### 3.4 Analisis Perbandingan Hasil LSTM dan ARIMA

Dari percobaan yang dilakukan, diperoleh hasil perhitungan RMSE yang dapat dilihat pada tabel 23.

Tabel 22 Perbandingan RMSE LSTM dan ARIMA

Obat	RMSE Testing	RMSE Testing	Selisih
	LSTM	ARIMA	
Obat 1	0.119	4.370	4.251
Obat 2	0.016	38.935	38.919
Obat 3	0.323	13.601	13.278
Obat 4	0.039	1.490	1.451
Obat 5	0.099	10.416	10.316

## 4. KESIMPULAN

Berdasarkan pengujian evaluasi yang telah dilakukan terhadap model LSTM dan ARIMA, maka dapat diambil beberapa kesimpulan sebagai berikut.

1. Berdasarkan hasil perhitungan selisih antara RMSE dari pemodelan LSTM dan ARIMA menunjukkan nilai akurasi LSTM lebih rendah dari akurasi pemodelan ARIMA. Hal ini menunjukkan bahwa LSTM lebih baik dalam melakukan prediksi terhadap data penjualan obat di Apotek dengan selisih rata-rata nilai RMSE sebesar 13.643.

2. Walaupun model LSTM memiliki hasil RMSE yang lebih rendah dibandingkan mode ARIMA, model LSTM mengalami *overfit* dimana model bisa bekerja sangat baik pada data *training*, tapi buruk dalam menghadapi data *testing* serta dalam melakukan prediksi untuk 3 bulan kedepan, menunjukkan adanya *overfit* dimana nilai prediksi yang dihasilkan terdapat minus (-). Dalam *machine learning*, selain melihat akurasi, perlu dilakukan evaluasi berdasarkan generalisasinya. Akurasi dilihat untuk mengukur seberapa baik model memprediksi nilai target dengan tepat, dan generalisasi berarti melihat seberapa baik model berperilaku pada data yang terlihat dan data yang tidak terlihat.

## 5. DAFTAR PUSTAKA

- [1] Rofi'ah, I., & Hantoro, K. 2022. Perancangan Sistem Informasi Penjualan Obat-Obatan Berbasis Web Pada Apotek Diana Menggunakan Algoritma Horspool. *Jurnal Penelitian Mahasiswa Ilmu Komputer*, 3(2) 195-2016. <https://doi.org/10.31599/jsrsc.v3i2.1404>.
- [2] Dewanti, FP, Setiyowati, S., & Harjanto, S. 2022. Prediksi Persediaan Obat Untuk Proses

- Penjualan Menggunakan Metode Decision Tree Pada Apotek. *Jurnal Teknologi Informasi Dan Komunikasi (TIKOMSiN)*, 10 (1), 25-33. <https://doi.org/10.30646/tikomsin.v10i1.604>
- [3] Wibowo, D. A. (2018). Prediksi Penjualan Obat Herbal Hp Pro Menggunakan Algoritma Neural Network. *Technologia Jurnal Ilmiah*, 9(1), <https://doi.org/10.31602/tji.v9i1.1100>
- [4] Render, B., Ralph, M. S, Jr., & Michael, E.H. 2018. Quantitative Analysis for Management Thirteenth Edition.
- [5] Makridakis, S., Evangelos, S., Vassilios, A. 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 12(3), 1-26. <https://doi.org/10.1371/journal.pone.0194889>
- [6] Gunaryati, A., Fauziah, F., & Andryana, S. 2018. Perbandingan Metode-metode Peramalan Statistika untuk Data Indeks Harga Pangan. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 2 (3), 241-248. <https://doi.org/10.30998/string.v2i3.2200>.
- [7] Arifin, I., Haidi, RF, & Dzalhaqi, M. (2021). Penerapan Computer Vision Menggunakan Metode Deep Learning pada Perspektif Generasi Ulul Albab. *Jurnal Teknologi Terpadu*, 7 (2), 98-107. <https://doi.org/10.54914/jtt.v7i2.436>
- [8] Karno, A.S.B. (2020). Analisis Data Time series Menggunakan LSTM (Long Short Term Memory) dan ARIMA (Autocorrelation Integrated Moving Average) dalam Bahasa Python. *ULTIMA InfoSys*, 9(1), 1-7. <https://doi.org/10.31937/si.v9i1.1223>.
- [9] Le, Xuan Hien., Hung, Viet Ho., Giha, Lee., & Sungho, Jung. (2019). Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water*, 2-19. <https://doi.org/10.3390/w11071387>.
- [10] Vavliakis, K. N., Siailis, A., & Symeonidis, A. L. (2021). Optimizing Sales Forecasting in e-Commerce with ARIMA and LSTM Models. In *WEBIST* (pp. 299-306). <https://doi.org/10.5220/0010659500003058>
- [11] Karno, A. S. B. (2020). Analisis Data Time Series Menggunakan LSTM (Long Short Term Memory) Dan ARIMA (Autocorrelation Integrated Moving Average) Dalam Bahasa Python. *Ultima InfoSys: Jurnal Ilmu Sistem Informasi*, 11(1), 1-7. <https://doi.org/10.31937/si.v9i1.1223>
- [12] Anshory, Maulana Ichwan., Yusuf, Priyandari., & Yuniaristanto. (2020). Peramalan Penjualan Sediaan Farmasi Menggunakan Long Short-term Memory: Studi Kasus pada Apotik Suganda. *Performa: Media Ilmiah Teknik Industri*, 19(2), 159-174. <https://doi.org/10.20961/performa.19.2.45962>
- [13] Salsabila, S.E. (2020). Model Prediksi Penjualan Multi-Item Time Series Berbasis Machine Learning Menggunakan Metode ARIMA dan LSTM pada produk perishable (studi kasus: retail sayur tosa). Universitas Islam Indonesia Yogyakarta.
- [14] Poomka, Pumrapee., Wattana Pongsena., Nittaya Kerdprasop., & Kittisak Kerdprasop. 2019). SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit, *International Journal of Future Computer and Communication*,8(1), 11-15. <https://doi.org/10.18178/ijfcc.2019.8.1.532>.
- [15] Silvanie, A., & Rino, S. (2022). Aplikasi Chatbot Untuk FAQ Akademik Di Ibi-K57 Dengan Lstm Dan Penyematan Kata, *JIKO*, 5(1), 19-27. DOI:10.33387/jiko.
- [16] Aju, C.N. (2020). Pemodelan Long Short Term Memory (LSTM) untuk Prakiraan Penjualan Berdasarkan Basis Data Penjualan Retail pada Kontrol Persediaan (Disertasi doktoral, IPB University).
- [17] Buchori, M & Tedjo, S. (2018). Peramalan Produksi Menggunakan Metode Autoregressive Integrated Moving Average (ARIMA) Di PT. XYZ, *Prozima*, 2(1), 27-33. <http://doi.org/10.21070/prozima.v2i1.1290>.
- [18] Putri, E.S & Mujiono, S. (2021). Prediksi Penjualan Produk Untuk Mengestimasi Kebutuhan Bahan Baku Menggunakan Perbandingan LSTM dan ARIMA, 10(2), 162-171. <https://doi.org/10.22441/format.2021.v10.i2.007>.
- [19] Jaya, J.D. (2019). Peramalan Jumlah Populasi Sapi Potong di Kalimantan Selatan Menggunakan Metode Moving Average, Exponential Smoothing dan Trend Analysis. 6(1), 41-50. <https://doi.org/10.34128/jtai.v6i1.88>.
- [20] Sofi, K., Sunge, A. S., Riady, S. R., & Kamalia, A. Z. (2021). Perbandingan algoritma linear regression, LSTM, dan GRU dalam memprediksi harga saham dengan model time series. *PROSIDING SEMINASTIKA*, 3(1), 39-46. <https://doi.org/10.47002/seminastika.v3i1.275>
- [21] Arsi, P., Tri, A., Desty, R., Pungkas, S. (2022). Implementasi Sliding Window Algorithm pada Prediksi Kurs berbasis Neural Network, *Journal of Computer and Information Technology*, 6(1), 51-59. <http://doi.org/10.25273/doubleclick.v6i1.13496>.
- [22] Dong, L., Desheng, F., Xi, W., Wei, W., Robertas, D., Rafał, S., Marcin, W. (2020). Prediction of Streamflow Based on Dynamic

Sliding Window LSTM, Water, 1-11.  
doi:10.3390/w12113032