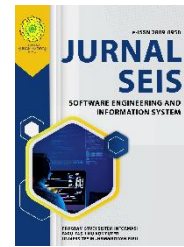




e-ISSN: 2809-0950



## **PREDIKSI CURAH HUJAN DI KOTA PEKANBARU MENGGUNAKAN LSTM (LONG SHORT TERM MEMORY)**

**Yos Hendra<sup>1</sup>, Harun Mukhtar<sup>2\*</sup>, Baidarus<sup>3</sup>, Rizka Hafsari<sup>4</sup>**

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

<sup>4</sup>Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

email: <sup>1</sup>[yoshendra33@gmail.com](mailto:yoshendra33@gmail.com), <sup>2</sup>[harunmukhtar@umri.ac.id](mailto:harunmukhtar@umri.ac.id), <sup>3</sup>[baidarus@umri.ac.id](mailto:baidarus@umri.ac.id),

<sup>4</sup>[rizkahafsari@umri.ac.id](mailto:rizkahafsari@umri.ac.id)

### **Abstract**

*Based on data obtained from BMKG Pekanbaru City in 2010-2020 there was an increase and decrease in the intensity of rainfall that occurred in Pekanbaru city. The increase in rainfall in the city of Pekanbaru will cause problems such as the occurrence of flooding of several roads and several areas in the city of Pekanbaru and the occurrence of other unexpected disasters that will cause problems and experience difficulties. In overcoming this problem, research was conducted in the form of Rainfall Prediction in Pekanbaru City Using LSTM (Long Short Term Memory) using 2 methods, namely in finding the accuracy of the error rate using RMSE (Root Mean Square Error) and MSE (Mean Square Error). The results showed that the predictions made were quite good. With the lowest error rate of 21,328 in the train and 454,901 in the test, the composition of the train data and the test data half gave the best results.*

**Keywords:** Prediction, Rainfall, Long Short Term Memory

### **Abstrak**

*Berdasarkan data yang diperoleh dari BMKG Kota Pekanbaru pada tahun 2010-2020 adanya peningkatan dan penurunan intensitas curah yang terjadi di kota Pekanbaru. Peningkatan curah hujan di kota Pekanbaru akan menimbulkan masalah seperti terjadinya banjir beberapa ruas jalan maupun beberapa daerah di kota Pekanbaru serta terjadinya bencana yang tidak terduga lainnya yang akan menimbulkan masalah dan mengalami kesulitan. Dalam mengatasi permasalahan ini maka dilakukan Penelitian berupa Prediksi Curah Hujan di Kota Pekanbaru Menggunakan LSTM (Long Short Term Memory) dengan menggunakan 2 metode yaitu dalam pencarian akurasi tingkat ke erroran dengan menggunakan RMSE (Root Mean Square Error) dan MSE (Mean Square Error). Dari Hasil penelitian menunjukkan bahwa prediksi yang dibuat cukup baik. Dengan tingkat error yang paling rendah 21.328 pada train dan 454.901 pada test, komposisi data train dan data test setengahnya memberikan hasil terbaik.*

**Keywords:** Prediksi, Curah Hujan, Long Short Term Memory

## PENDAHULUAN

Prediksi merupakan proses sistematis yang digunakan untuk memperkirakan sesuatu yang paling mungkin terjadi di masa depan dengan menggunakan informasi yang ada saat ini dan sebelumnya. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Perdana et al., 2015).

Menurut BMKG (Badan Meteorologi, Klimatologi, dan Geofisika) dalam kutipan dari Hujan adalah suatu bentuk presipitasi atau endapan cairan atau zat padat dari kondensasi yang jatuh dari awan ke permukaan bumi. Namun, sebagian besar air hujan tidak dapat mencapai permukaan bumi, karena sebagian menguap ketika jatuh melalui udara kering. Berdasarkan data yang diperoleh dari BMKG Kota Pekanbaru pada tahun 2010-2020 adanya peningkatan dan penurunan intensitas curah yang terjadi di kota Pekanbaru. Peningkatan curah hujan di kota Pekanbaru akan menimbulkan masalah seperti terjadinya banjir beberapa ruas jalan maupun beberapa daerah di kota Pekanbaru serta terjadinya bencana yang tidak terduga lainnya yang akan menimbulkan masalah dan mengalami kesulitan dalam memberikan pelayanan terbaik serta pengenalan pencegahan bencana sejak dini supaya ketika terjadi masyarakat khususnya di Kota Pekanbaru dapat mengatasinya dengan cepat.

Lokasi Kota Pekanbaru adalah di  $101^{\circ}14'$  -  $101^{\circ}34'$  Bujur Timur dan  $0^{\circ}25'$  -  $0^{\circ}45'$  Lintang Utara. Dengan ketinggian 5 hingga 50 meter dari permukaan laut, permukaan bagian utara landai dan bergelombang dengan ketinggian 5 hingga 11 meter. Menurut Peraturan Pemerintah No. 19 Tahun 1987, yang ditetapkan pada 7 September 1987, wilayah Kota Pekanbaru diperluas dari  $\pm 62,96$  Km<sup>2</sup> menjadi  $\pm 446,50$  Km<sup>2</sup>, dengan 8 Kecamatan dan 45 Kelurahan/Desa. Berdasarkan pengukuran dan pematokan di lapangan oleh BPN Tk. I Riau, luas wilayah Kota Pekanbaru adalah 632,26 Km<sup>2</sup>. Iklim Kota Pekanbaru sebagian besar tropis, dengan suhu udara maksimum  $34,1-35,6^{\circ}\text{C}$  dan suhu minimum  $20,2-23,0^{\circ}\text{C}$ . Hujan tahunan berkisar antara 38,6-435,0 mm, dan musim hujan berlangsung dari Januari hingga April dan musim kemarau berlangsung dari Mei hingga Agustus. Kelembapan tertinggi adalah 96% hingga 100%, dan kelembapan terendah adalah

46% hingga 62%. (<https://pekanbaru.go.id/p/hal/wilayah-geografis>).

Berdasarkan dari kutipan berita ([www.riaugreen.com](http://www.riaugreen.com)) curah hujan di kota pekanbaru termasuk dalam kategori curah hujan yang sangat tinggi. Tidak heran jika masyarakat khususnya di kota Pekanbaru dihantui dengan rasa takut terjadinya banjir yang selalu terjadi ketika hujan deras terjadi. Kepala Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Riau Edwar Sanger, Senin (11/1/2021) menyatakan secara instansi, pihaknya sudah siap untuk menghadapi musibah banjir di Riau saat musim hujan.

Dalam memprediksi curah hujan di kota Pekanbaru ada beberapa metode/algorithm untuk memprediksi curah hujan seperti LSTM (*Long Short Term Memory*), SVM (*Support Vector Machines*), *fuzzy* dan lain-lain. Setiap metode memiliki kelebihan dan kekurangan, namun pada intinya semua metode atau algoritma memiliki prinsip yang sama, untuk mengolah sebuah data akan berjalan sesuai dengan prinsip dari algoritma tersebut yang memiliki kelebihan dan kekurangan di setiap metode atau algoritma. Adapun pada penelitian ini penulis memilih menggunakan metode atau algoritma LSTM (*Long Short Term Memory*).

Terdapat penelitian terdahulu dalam melakukan prediksi produksi roti coklat menggunakan *double exponential* smoothing oleh Asrianto & Anggraini (2021), klasifikasi curah hujan menggunakan SVM dan NBC (Luter Laia & Setyawan, 2020). Selain itu, memprediksi penjualan produk oleh Wiranda & Sadikin (2019) serta memprediksi kondisi motor 10 kV oleh (Wisyalidin et al., 2020).

Pemilihan menggunakan metode atau algoritma LSTM (*Long Short Term Memory*) merujuk kepada penelitian terdahulu yang telah melakukan perbandingan atau keakurasian dengan menggunakan metode LSTM (*Long Short Term Memory*) mampu menghasilkan akurasi diatas 85% dengan kasus yg berbeda, dengan demikian waktu prediksi curah hujan lebih cepat diketahui keakuratan dengan menggunakan metode LSTM (*Long Short Term Memory*) (Rizki et al., 2020).

Disamping itu, berdasarkan hasil penelitian Ningrum et al (2021), LSTM ini memiliki kinerja yang baik dalam melakukan pencarian

komposisi data. LSTM juga baik dalam mengingat informasi dengan waktu yang lama (Kumar et al., 2018).

Penelitian yang dilakukan oleh Kurniawan (2020) Evaluasi Hasil Pengukuran Curah Hujan di Stasiun Klimatologi Mlati Tahun 2018 Antara Hasil Pengukuran Permukaan (AWS, HELLMAN, OBS) dan Hasil Estimasi (Citra Satelit GSMaP). Penelitian ini menunjukkan bahwa penakar hujan di permukaan bumi (OBS, Hellman, dan AWS) dan hasil estimasi (GSMaP) di Stasiun Klimatologi Mlati pada 2018 memiliki perbedaan dalam nilai pengukuran curah hujan. Penakar hujan OBS memiliki nilai paling besar setiap bulan (Fitriyaningsih et al., 2018).

Adanya Prediksi Curah Hujan di Kota Pekanbaru Menggunakan LSTM (*Long Short Term Memory*) ini diharapkan dapat mengolah data yg menghasilkan hasil Akurasi dari hasil prediksi curah hujan di kota Pekanbaru.

## METODE PENELITIAN

Metode penelitian ini terbagi menjadi beberapa tahapan yaitu perencanaan penelitian, pengolahan data, pengujian, mengambil kesimpulan dan perangkat penelitian.

### Perencanaan Penelitian

Tahap pertama dari penelitian adalah perencanaan. Pada tahap ini, peneliti memulai dengan membaca literatur dan pustaka untuk menentukan masalah yang ingin diteliti, seperti bagaimana memprediksi curah hujan di Pekanbaru..

### Pengolahan Data

Setelah masalah diidentifikasi, proses berikutnya adalah pengolahan data. Ini dilakukan menggunakan metode LSTM. Tahapan Pengolahan data yang dilakukan sebagai berikut :

- a) Pengumpulan Data
- b) Normalisasi Data
- c) Pembagian Dataset
- d) Alur Kerja Sistem

### Pengujian

Pada tahap ini, data yang telah diperoleh sebelumnya melalui teknik-teknik pengumpulan

data yang akan dilakukan peneliti kemudian akan dilakukan Pengujian, berdasarkan metode yang di uji di dalam penelitian ini yaitu Metode *Long Short Term Memory*. Proses Pengujian pada penelitian ini, terdiri dari beberapa tahapan, dimana masing-masing proses terdiri dari perhitungan tersendiri yaitu *Training LSTM* dan *Menguji Data Testing*.

### Mengambil Kesimpulan

Dengan mempertimbangkan tujuan penelitian dan hasilnya, hasil akhir penelitian memberikan penjelasan tentang temuan dan rekomendasi yang akan membantu peneliti selanjutnya. Dalam hal ini, ide-ide yang dihasilkan dari pengolahan dan pengujian data menggunakan teori yang digunakan dalam penelitian diambil sebagai dasar dan dapat digunakan kembali dengan cara yang lebih baik.

### Perangkat Penelitian

Adapun berbagai perangkat yang digunakan pada penelitian prediksi curah hujan di Kota Pekanbaru menemukan LSTM.

### Software

Adapun *software* yang digunakan dalam penelitian ini adalah sebagai berikut:

**Tabel 1 Rincian Kebutuhan Software**

No	Software	Keterangan
1	Windows	Sistem Operasi Untuk Menjalankan Sistem Aplikasi
2	Microsoft Word	Untuk Pembuatan, Pengolahan dan Dokumentasi data yang didapat dari penelitian untuk menjadi Laporan.
3	Microsoft Excel	Untuk merancang dan mengolah data yang di dapat dari BMKG
4	Microsoft Powerpoint	Untuk mempresentasikan penelitian
5	Google Chrome	Aplikasi milik google untuk mengakses website
6	Python	Bahasa pemrograman level tinggi sebagai penerjemah kode
7	Google Colab	Pengolahan data dengan Bahasa pemrograman python

## Hardware

Adapun *Hardware* yang digunakan dalam penelitian ini adalah sebagai berikut:

**Tabel 2 Rincian Kebutuhan Hardware**

No	Hardware	Keterangan
1	Prosesor	AMD APU E1-1200 Dual Core 1,4GHz
2	Monitor	14" HD Acer cinecrystal LED LCD
3	Ram	2GB DDR3 Memory
4	Harddisk	500 GB SATA
5	Printer	EPSON L310

## HASIL DAN PEMBAHASAN

Berikut hasil dan pembahasan pada penelitian ini.

### Hasil Pengolahan Data

Pengumpulan data dilakukan untuk mendapatkan dataset yang akan digunakan dalam penelitian ini. Dataset yang didapat dari BMKG dan digunakan adalah dataset public yang diperoleh berjumlah 2.129 data curah hujan dan terdiri 5 label dalam bentuk format file (.xlsx). Label yang digunakan dari dataset yang didapat adalah "laporan iklim harian". Dataset yang digunakan akan diinputkan ke dalam *Google collab* untuk tahapan proses dalam pengolahan data yang akan diprediksi.

Langkah awal yang dilakukan adalah mengimport Modul atau *library* yang akan digunakan

```
! pip install pandas
```

Pandas adalah *library* untuk data lobular

```
# import module / library
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
```

Setelah Modul atau *library* yang akan digunakan diimport, kemudian dataset diload untuk digunakan.

```
# import dataset via pandas
dataset = pd.read_csv("/content/drive/MyDrive/skripsi 2021/DATA SET 10 TAHUN revisi.csv", header=0, sep=";")
print(dataset)
```

Contoh dataset atau *library* yang di akan digunakan dan sebelum melewati Proses seperti gambar di bawah ini.

	Tanggal	Tavg	RR	ss	ff_avg	ddd_car
0	1012010	26.1	6.9	0.2	3	W
1	2012010	27.0	2.0	0.2	2	NW
2	3012010	28.2	0.0	0.2	3	S
3	4012010	26.8	12.0	0.1	2	NE
4	5012010	28.6	0.0	0.3	3	N
...	...	...	...	...	...	...
2124	24122020	27.5	1.0	4.9	2	N
2125	25122020	25.9	11.5	8.7	0	N
2126	26122020	27.7	0.2	0.0	1	N
2127	29122020	25.4	1.8	5.1	1	N
2128	31122020	25.2	3.1	0.4	0	N

[2129 rows x 6 columns]

**Gambar 5 Dataset**

Keterangan:

8888: data tidak terukur

9999: Tidak Ada Data (tidak dilakukan pengukuran) Tavg: Temperatur rata-rata (°C)

RR: Curah hujan (mm)

ss: Lamanya penyinaran matahari (jam) ff\_avg:

Kecepatan angin rata-rata (m/s) ddd\_car: Arah angin terbanyak (°)

Dataset yang digunakan akan di bagi menjadi dua bagian yaitu data testing dan data training dengan jumlah data 1.065 untuk data testing dan 1.064 untuk data training.

```
# dataset.drop(columns="Tanggal", inplace=True)
dataset.Tanggal.replace(dataset.Tanggal.values, 1/dataset.Tanggal.values, inplace=True)

print(dataset)
dataset.RR.replace(8888, np.nan, inplace=True)
dataset.ff_avg.replace(np.nan, 0, inplace=True) dataset.ff_avg.astype(int)
dataset.dropna(inplace=True)
```

*Drop* dilakukan untuk menghilangkan kolom yang tidak digunakan sebagai dataset dalam pemrosesan. Dilakukan proses drop ini supaya tidak mempengaruhi hasil yang akan diprediksi.

Replace dilakukan untuk mengganti nilai pada dataset yang digunakan dalam pemrosesan.

	Tanggal	Tavg	RR	ss	ff_avg	ddd_car
0	9.881325e-07	26.1	6.9	0.2	3	W
1	4.970154e-07	27.0	2.0	0.2	2	NW
2	3.320042e-07	28.2	0.0	0.2	3	S
3	2.492516e-07	26.8	12.0	0.1	2	NE
4	1.995208e-07	28.6	0.0	0.3	3	N
...	...	...	...	...	...	...
2124	4.145590e-08	27.5	1.0	4.9	2	N
2125	3.980572e-08	25.9	11.5	8.7	0	N
2126	3.828188e-08	27.7	0.2	0.0	1	N
2127	3.433828e-08	25.4	1.8	5.1	1	N
2128	3.213159e-08	25.2	3.1	0.4	0	N

[2129 rows x 6 columns]

Gambar 6 Proses drop dan replace

```
# merubah mata angin menjadi angka
dataset["ddd_car"].replace(["N", "NE", "E", "SE", "S", "SW", "W", "NW"], [1, 2, 3, 4, 5, 6, 7, 8], inplace=True)
```

ddd_car	ddd_car
W	7
NW	8
S	5
NE	2
N	1
...	...
N	1
N	1
N	1
N	1
N	1

Gambar 7 Replace Tanggal

**Process**

Proses persiapan dataset dilakukan dengan membuat fungsi yang memprediksi dataset yang akan digunakan proses.

```
# membagi label untuk dipelajari dan diprediksi
sampel = dataset.drop(columns="RR").astype("float")
target = dataset["RR"]

# memisahkan 7 data untuk pengujian dari luar
# sampel_uji = sampel[-7:]
# target_uji = target[-7:]
# sampel = sampel[:-7]
# target = target[:-7]
```

Setelah melakukan proses inputan maka akan ditentukan tabel atau kolom yang akan diproses untuk prediksi yaitu RR. Dimana RR ini adalah data curah hujan yang akan diolah sebagai target dalam akurasi yang dilakukan.

	Tanggal	Tavg	ss	ff_avg	ddd_car
0	9.881325e-07	26.1	0.2	3.0	7.0
1	4.970154e-07	27.0	0.2	2.0	8.0
2	3.320042e-07	28.2	0.2	3.0	5.0
3	2.492516e-07	26.8	0.1	2.0	2.0
4	1.995208e-07	28.6	0.3	3.0	1.0
...	...	...	...	...	...
2124	4.145590e-08	27.5	4.9	2.0	1.0
2125	3.980572e-08	25.9	8.7	0.0	1.0
2126	3.828188e-08	27.7	0.0	1.0	1.0
2127	3.433828e-08	25.4	5.1	1.0	1.0
2128	3.213159e-08	25.2	0.4	0.0	1.0

[2053 rows x 5 columns]

0 6.9  
1 2.0  
2 0.0  
3 12.0  
4 0.0  
...  
2124 1.0  
2125 11.5  
2126 0.2  
2127 1.8  
2128 3.1  
Name: RR, Length: 2053, dtype: float64

Gambar 8 Pemisahan Target

Setelah melakukan proses pemisahan atau penentuan target maka dilakukan langkah selanjutnya yaitu pembagian dataset.

**Dataset**

Untuk melakukan teknik ilmu data mining, kumpulan data adalah kumpulan data yang dibuat dari informasi masa lalu dan diproses menjadi informasi baru.

**Data Training**

Data Training adalah data yang diperoleh dari dataset dan berfungsi untuk membuat prediksi atau menjalankan fungsi dari algoritma LSTM dimana untuk data training ini sebanyak 50%.

**Data Testing**

Data Testing adalah data yang diperoleh dari dataset dan berfungsi untuk pengolahan data. Dimana untuk data testing ini sebanyak 50%.

```
# membagi dataset untuk train dan test
test = int(.5 * len(sampel))
sampel_test = sampel[:test]
target_test = target[:test]
sampel_train = sampel[test:]
target_train = target[test:]
```

Setelah melakukan pembagian data *training* dan data *testing* maka dilakukanlah pengujiannya untuk melihat akurasi dan performa dari data *training* dan data *testing*.

```
Epoch 1/1000
1/1 - 2s - loss: 660.2300 - val_loss: 576.2531
Epoch 2/1000
1/1 - 0s - loss: 654.9675 - val_loss: 572.7848
Epoch 3/1000
1/1 - 0s - loss: 646.8431 - val_loss: 567.7458
Epoch 4/1000
1/1 - 0s - loss: 633.4152 - val_loss: 559.5461
Epoch 5/1000
1/1 - 0s - loss: 609.2794 - val_loss: 546.6390
Epoch 6/1000
1/1 - 0s - loss: 572.5860 - val_loss: 533.6290
Epoch 7/1000
1/1 - 0s - loss: 546.7941 - val_loss: 527.2751
Epoch 8/1000
1/1 - 0s - loss: 538.5458 - val_loss: 523.5916
Epoch 9/1000
1/1 - 0s - loss: 534.3214 - val_loss: 520.6836
Epoch 10/1000
1/1 - 0s - loss: 531.4907 - val_loss: 518.0084
```

**Gambar 5 Proses Validasi**

## Metode LSTM

LSTM merupakan salah satu jenis dari *Recurrent Neural Network* (RNN) dimana dilakukan modifikasi pada RNN dengan menambahkan memory cell yang dapat menyimpan informasi untuk jangka waktu yang lama (Pontoh et al., 2022). LSTM disarankan untuk mengatasi *vanishing gradient* pada RNN saat memproses data *sequential* yang panjang.

Adapun parameter model yang disesuaikan adalah:

1. Model menggunakan model *sequential*
2. *Embedding* adalah proses di mana satu klausa dimasukkan atau ditanamkan ke dalam klausa lain. Menurut kesimpulan kami, yang berasal dari situs Thoughtco, ini juga disebut sebagai nesting atau bersarang.
3. Input model LSTM 256
4. *Dense* adalah menggunakan berbagai pendekatan terpisah untuk melindungi sistem dari serangan khusus.

Setelah melakukan parameter model maka dilakukanlah pembagian model LSTM

```
# membuat model
lstm.compile(loss="mean_squared_error",
             optimizer="RMSprop")

# menampilkan dimensi
model.summary()
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 5, 300)	12000
lstm (LSTM)	(None, 5, 256)	570368
dense (Dense)	(None, 5, 1)	257

Total params: 582,625  
Trainable params: 570,625  
Non-trainable params: 12,000

**Gambar 6 LSTM**

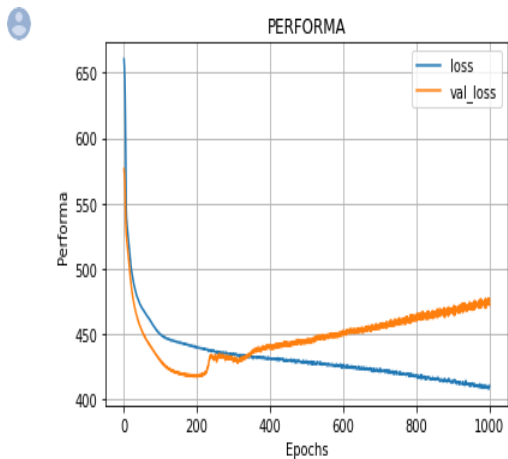
Tujuan dari pelatihan model berikutnya adalah untuk melihat hasil dari kinerja dari hasil kerja metode LSTM. Dengan membaca sejarah yang sudah dilakukan sebelumnya, kita dapat melihat hasil dari grafik atau alur kinerja dari kinerja metode LSTM.

```
# melatih model
history = lstm.fit(training_sampel, training_target, epochs=1000, batch_size=len(training_target), verbose=2, validation_data=(test_sampel, test_target))
```

```
Epoch 1/1000
1/1 - 2s - loss: 660.2300 - val_loss: 576.2531
Epoch 2/1000
1/1 - 0s - loss: 654.9675 - val_loss: 572.7848
Epoch 3/1000
1/1 - 0s - loss: 646.8431 - val_loss: 567.7458
Epoch 4/1000
1/1 - 0s - loss: 633.4152 - val_loss: 559.5461
Epoch 5/1000
1/1 - 0s - loss: 609.2794 - val_loss: 546.6390
Epoch 6/1000
1/1 - 0s - loss: 572.5860 - val_loss: 533.6290
Epoch 7/1000
1/1 - 0s - loss: 546.7941 - val_loss: 527.2751
Epoch 8/1000
1/1 - 0s - loss: 538.5458 - val_loss: 523.5916
Epoch 9/1000
1/1 - 0s - loss: 534.3214 - val_loss: 520.6836
Epoch 10/1000
1/1 - 0s - loss: 531.4907 - val_loss: 518.0084
```

**Gambar 7 Melatih Model**

```
# membuat grafik pelatihan model
plt.plot(np.arange(1, 1000 + 1, 1), history.history["loss"])
plt.plot(np.arange(1, 1000 + 1, 1), history.history["val_loss"])
plt.title("PERFORMA")
plt.xlabel("Epochs")
plt.ylabel("Performa")
plt.legend(["loss", "val_loss"])
plt.grid()
plt.show()
plt.close()
```



Gambar 8 Performa

**Hasil**

Untuk mendapatkan kesimpulan dari semua model maka dilaksanakan proses prediksi dan evaluasi menggunakan RMSE dan MSE dimana dalam menentukan hasil peneliti melakukan 5 kali percobaan seperti tabel di bawah ini

**Tabel 3 Hasil**

No	Training / Test	LSTM	Epoch	RMSE	MSE
1	10% / 90%	256	1.000	22.349	499.482
2	20% / 80%	256	1.000	21.499	462.219
3	30% / 70%	256	1.000	21.461	460.609
4	40% / 60%	256	1.000	21.738	472.570
5	50% / 50%	256	1.000	21.328	454.901

**RMSE (Root Mean Square Error)**

Root Mean Square Error (RMSE) adalah hasil dari akar kuadrat kesalahan kuadrat. Nilai RMSE yang kecil menunjukkan bahwa metode estimasi kesalahan pengukuran lebih akurat daripada metode estimasi dengan RMSE lebih besar.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (A_t - F_t)^2}{n}} \dots\dots\dots(1)$$

**MSE (Mean Square Error)**

MSE adalah nilai error kuadrat rata-rata antara gambar asli dan gambar manipulasi (dalam steganografi). MSE adalah nilai error kuadrat rata-rata antara gambar cover-image dan gambar hasil penyisipan.

$$MSE = \frac{1}{MN} \sum_{X=1}^M \sum_{Y=1}^N (S_{xy} - C_{xy})^2 \dots\dots\dots(2)$$

```
# memberikan data untuk diprediksi
testing = np.array(sampel_test)
prediksi = lstm.predict(testing).reshape(len(target_test), 5)
prediksi = pd.DataFrame(prediksi)
print(prediksi)
prediksi = np.array(tf.keras.layers.Average()([prediksi[0], prediksi[1], prediksi[2], prediksi[3], prediksi[4]]))
print(prediksi)
```

	0	1	2	3	4
0	1.377386	3.028002	2.641892	1.629100	1.874833
1	16.118505	19.657652	20.515497	12.889806	7.682824
2	17.408491	16.590794	11.981902	8.156916	10.221013
3	9.881715	18.807533	25.881172	22.971001	14.514989
4	14.579188	9.176243	6.691815	4.893757	8.268287
...	...	...	...	...	...
1021	5.090649	13.777480	19.847815	14.279666	9.170596
1022	9.881715	8.095790	2.365064	1.435275	1.315512
1023	14.579188	9.176242	9.537902	8.501563	9.221045
1024	9.881715	8.095790	2.365064	1.435275	1.315511
1025	16.118507	11.653884	9.263734	0.971154	0.473576

[1026 rows x 5 columns]  
[ 2.1102426 15.372858 12.871823 ... 10.203188 4.618671 7.696171 ]

Gambar 9 Hasil Prediksi

```
# evaluasi dengan RMSE
from sklearn.metrics import mean_squared_error
RMSE = mean_squared_error(target_test, prediksi, squared=False)
MSE = mean_squared_error(target_test, prediksi, squared=True)
print(RMSE, " | " ,MSE)
```

Setelah melakukan evaluasi dengan RMSE dan MSE maka dapat dilihat tingkat error dari data set yang terbagi menjadi dua yaitu data training dan data testing maka tingkat ke errorannya adalah sebagai berikut

21.328427588277982 | 454.90182338841737

Gambar 10 Evaluasi RMSE dan MSE

**SIMPULAN DAN SARAN**

Setelah melakukan pengumpulan data, analisis, pengolahan, dan prediksi menggunakan algoritma Long Short Term Memory (LSTM) untuk prediksi curah hujan kota Pekanbaru, menunjukkan bahwa hasil prediksi yang diperoleh tergolong cukup baik. Sehingga dapat disimpulkan bahwa pada penelitian ini dapat

bekerja cukup optimal. Berikut pembahasan terkait pada penelitian ini sebagai berikut:

1. Tingkat error yang paling rendah ditemukan pada komposisi 50% data train dan 50% data test, masing-masing 21.328 dan 454.901.
2. Penelitian ini menggunakan 2 metode dalam pencarian akurasi tingkat ke erroran dengan menggunakan RMSE (*Root Mean Square Error*) dan MSE (*Mean Square Error*)

Berdasarkan penelitian yang telah dilakukan, model masih bisa dikembangkan untuk mendapatkan performa yang lebih baik lagi. Model yang digunakan bisa disesuaikan dengan parameter model, usahakan dalam penelitian berikutnya menggunakan algoritma yang berbeda dengan penelitian saat ini supaya hasilnya lebih akurat dan lebih baik.

## DAFTAR PUSTAKA

- Asrianto, R., & Anggraini, Y. (2021). Prediksi Produksi Roti Coklat Berdasarkan Data Trends Google Menggunakan Metode Double Exponential Smoothing. *Journal of Software Engineering and Information Systems*, 1(1), 43–57. <https://doi.org/10.37859/seis.v1i1.2821>
- Fitriyaningsih, I., Basani, Y., & Lit Malem, G. (2018). Aplikasi Prediksi Curah Hujan, Debit Air, dan Kejadian Banjir Berbasis Web dengan Machine Learning di Deli Serdang. *Jurnal Penelitian Komunikasi Dan Opini Publik*, 22(2), 150–160.
- Kumar, J., Goomer, R., & Singh, A. K. (2018). Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters. *Procedia Computer Science*, 125, 676–682. <https://doi.org/10.1016/j.procs.2017.12.087>
- Kurniawan, A. (2020). Verifikasi data pengukuran curah hujan antara Vaisala Hydromet-MAWS201 Menggunakan Sensor Hujan Rain Gauge Qmr101 dengan penakar hujan observasi (OBS) di SPAG Bukit Kototabang pada Januari-Juni 2010. *Megasains*, 4(1), 1–7.
- Luter Laia, M., & Setyawan, Y. (2020). Perbandingan Hasil Klasifikasi Curah Hujan Menggunakan Metode Svmdan Nbc. *Jurnal Statistika Industri Dan Komputasi*, 05(2), 51–61.
- Ningrum, A. A., Syarif, I., Gunawan, A. I., Satriyanto, E., & Muchtar, R. (2021). Algoritma Deep Learning-Lstm Untuk Memprediksi Umur Deep Learning-Lstm Algorithm for Power Transformer Lifetime. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(3), 539–548. <https://doi.org/10.25126/jtiik.202184587>
- Perdana, D. A., Zakaria, A., & Sumiharni. (2015). Studi Pemodelan Sintetik Curah Hujan Harian Pada Beberapa Stasiun Hujan di Kabupaten Pringsewu. *Jurnal Rekayasa Sipil Dan Desain*, 3(1), 45–56.
- Pontoh, R. S., Toharudin, T., Ruchjana, B. N., Gumelar, F., Putri, F. A., Agisya, M. N., & Caraka, R. E. (2022). Jakarta Pandemic to Endemic Transition: Forecasting COVID-19 Using NNAR and LSTM. *Applied Sciences (Switzerland)*, 12(12). <https://doi.org/10.3390/app12125771>
- Rizki, M., Basuki, S., & Azhar, Y. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory (LSTM) Untuk Prediksi Curah Hujan Kota Malang. *Jurnal Repositor*, 2(3), 331–338. <https://doi.org/10.22219/repositor.v2i3.470>
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 8(3), 184–196.
- Wisyaladin, M. K., Luciana, G. M., & Pariaman, H. (2020). Pendekatan Long Short-Term Memory untuk Memprediksi Kondisi Motor 10 kV pada PLTU Batubara. *Kilat*, 9(2), 311–318. <https://doi.org/10.33322/kilat.v9i2.997> <https://pekanbaru.go.id/p/hal/wilayah-geografis>