



**PEMODELAN MENGGUNAKAN PENDEKATAN *RECURRENT NEURAL NETWORK LONG SHORT TERM MEMORY* (RNN-LSTM) PADA HARGA EMAS DUNIA**

**JURNAL ILMIAH**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Statistika**

Oleh

**SILVIE NOOR ALAWIYAH**

**B2A017025**

**SEMARANG**

**PROGRAM STUDI S1 STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS MUHAMMADIYAH SEMARANG  
2021**

## HALAMAN PENGESAHAN

Skripsi dengan judul “ **Pemodelan Menggunakan Pendekatan *Recurrent Neural Network Long Short Term Memory* (RNN-LSTM) pada Harga Emas Dunia** ” yang disusun oleh :

Nama : Silvie Noor Alawiyah

NIM : B2A017025

Program Studi : S-1 STATISTIKA

telah disetujui oleh dosen pembimbing pada 29 April 2021.

Pembimbing Utama

Pembimbing Pendamping



Dr. Rochdi Wasono, M.Si

Indah Manfaati Nur, S.Si, M.Si

NIK. 28.6.1026.119

NIK. 28.6.1026.221

Mengetahui,

Ketua Program Studi

Indah Manfaati Nur, S. Si, M. Si

NIK. 28.6.1026.221

**SURAT PERNYATAAN  
PUBLIKASI KARYA ILMIAH**

Yang bertandatangan di bawah ini, saya :

Nama : Silvie Noor Alawiyah  
NIM : B2A017025  
Fakultas/Jurusan : Matematika dan Ilmu Pengetahuan Alam/Statistika  
Jenis Penelitian : Skripsi  
Judul : Pemodelan Menggunakan Pendekatan *Recurrent Neural Network Long Short Term Memory (RNN-LSTM)* Pada Harga Emas Dunia  
Email : [silvie.nooralawiyah99@gmail.com](mailto:silvie.nooralawiyah99@gmail.com)

Dengan ini menyatakan bahwa saya menyetujui untuk :

1. Memberikan hak bebas royalti kepada Perpustakaan Unimus atas penulisan karya ilmiah saya, demi pengembangan ilmu pengetahuan.
2. Memberikan hak menyimpan, mengalih mediakan/ mengalih formatkan, mengelola dalam bentuk pangkalan data (*database*), mendistribusikannya, serta menampilkannya dalam bentuk *softcopy* untuk kepentingan akademis kepada Perpustakaan Unimus, tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta.
3. Bersedia dan menjamin untuk mengganggu secara pribadi tanpa melibatkan pihak perpustakaan Unimus, dari semua bentuk tuntutan hukum yang timbul atas pelanggaran hak cipta dalam karya ilmiah ini.

Demikian pernyataan ini saya buat dengan sesungguhnya dan semoga dapat digunakan sebagaimana mestinya.

Semarang, 29 April 2021  
Yang Menyatakan,



(Silvie Noor Alawiyah)  
NIM. B2A.017.025

# PEMODELAN MENGGUNAKAN PENDEKATAN *RECURRENT NEURAL NETWORK LONG SHORT TERM MEMORY (RNN-LSTM)* PADA HARGA EMAS DUNIA

Oleh: Silvie Noor Alawiyah  
Program Studi Statistika, Univeristas Muhammadiyah Semarang  
*e-mail*: silvie.nooralawiyah99@gmail.com

Article history	Abstract
Submission :	Investment, which is essentially a commitment to a certain number of resources with the aim of obtaining future benefits. The economic recession that occurred resulted in a weakening of the value of money and made investors shift their financial assets to instruments that have a lower risk, such as gold. Gold price predictions are needed by investors to be taken into consideration. However, no one can predict with certainty about the future price of gold because there are many factors that cannot be controlled by human hands. In this research, modeling will be carried out using the Recurrent Neural Network Long Short Term Memory (RNN-LSTM) approach on the world gold price for 5 years with the period January 1, 2016 to February 26, 2021. LSTM is widely chosen by researchers for time-based predictions. series because it is known to be superior and reliable in making predictions over a long period of time compared to other algorithms. To improve accuracy in making prediction models, this study uses an Adaptive Moment Estimation (ADAM) optimization algorithm. The results of this study indicate that the best prediction model using RNN-LSTM with MAE error gets an error value of 14.5923 and a scenario of 30 neurons and 500 epochs. Absolute Percentage Error (MAPE), with a MAPE value of 1.09%, which indicates that the prediction results are very good.
Revised :	
Accepted :	
<b>Keyword:</b> <i>Adaptive Moment Estimation (ADAM)</i> , Harga Emas, Prediksi, RNN LSTM	

## PENDAHULUAN

Investasi pada hakikatnya adalah komitmen terhadap sejumlah sumber daya tertentu dengan tujuan untuk mendapatkan keuntungan dimasa depan (Abdul, 2005). Hal ini juga berarti bahwa investasi merupakan salah satu cara mengelola kelebihan dana agar bisa berkembang dari waktu ke waktu (Salim, 2010). Investasi ada yang bersifat *high risk, high gain* (risiko tinggi, keuntungan tinggi) dan *low risk, low gain* (risiko rendah, keuntungan rendah). Namun, emas tidak termasuk kedalam kedua sifat tersebut melainkan *middle risk* yang berarti investasi aman dengan hasil yang menguntungkan (Dipraja, 2011).

Kenaikan harga emas dunia banyak didorong oleh pelemahan dollar Amerika Serikat (AS). Kenaikan harga emas sangat terlihat dalam pertengahan tahun 2019 sampai akhir tahun 2020 ini dikarenakan kondisi politik dunia yang tidak stabil serta adanya pandemi virus Covid-19 yang belum bisa teratasi. Kenaikan harga emas berbanding terbalik dengan menguatnya Dollar AS. Jika dollar AS menguat, maka harga emas biasanya cenderung turun. Di Indonesia, harga emas berpatokan pada harga dunia, sehingga harga emas di dalam negeri mengalami kenaikan ketika rupiah melemah. Harga emas yang terlihat selalu mengalami kenaikan dan nyaris tidak pernah turun itu membuat minat para investor semakin besar terhadap emas.

Sebelum memasuki dunia investasi diperlukan pengetahuan keuntungan dan risiko yang akan didapatkan ketika mengambil langkah selanjutnya. Risiko investasi disini diartikan sebagai kemungkinan terjadinya perbedaan antara keuntungan yang aktual dengan keuntungan yang diharapkan. Risiko maupun harapan keuntungan dalam berinvestasi selalu ada dan berdampingan. Dalam berinvestasi disamping menghitung keuntungan yang diharapkan investor juga perlu memperhatikan risiko yang akan ditanggung (Abdul Halim, 2005).

Metode peramalan sangat banyak dan seringkali memerlukan asumsi-asumsi yang harus dipenuhi, namun terdapat juga model peramalan yang tidak memerlukan asumsi-asumsi salah satunya adalah *Neural Network* (NN). Selain metode peramalan ARIMA kini metode peramalan data dengan *time series* telah berkembang yaitu dengan pendekatan NN. RNN merupakan salah satu jenis dari *Artificial Neural Network* (ANN). RNN tidak membuang informasi pada data masa lalu dalam proses pembelajarannya, ini yang membedakan RNN dengan ANN. RNN biasanya digunakan untuk menyelesaikan masalah dengan data *time series*. Pada tahun 1997 Hochreiter dan Schmidhuber menemukan arsitektur baru untuk masalah data yang terlalu lama, yaitu dengan arsitektur *Long Short Term Memory* (LSTM). *Long Short Term Memory* (LSTM) sendiri merupakan pengembangan dari RNN untuk mengatasi ketidakkampuannya untuk mengolah informasi *sequential* dalam jangka panjang terutama pengolahan data berupa deret waktu (*time series*). LSTM menyaring informasi melalui struktur gerbang untuk mempertahankan dan memperbarui keadaan sel memori. Struktur pintunya mencakup gerbang *input*, *forget gate*, dan *output*. Beberapa penelitian terdahulu terkait prediksi harga emas telah dilakukan oleh beberapa ahli diantaranya, M. Azzam Fachruddin (2019) melakukan prediksi harga emas dalam bentuk mata uang dollar AS berdasarkan waktu AM (10:30) dan PM (15:00) dengan metode *Elman Recurrent Neural Network* (ERNN). Berdasarkan hasil pengujian, didapatkan nilai akurasi tertinggi sebesar 98,82% pada pembagian data *training* dan data *testing* 90%:10% dengan *learning rate* 0,1 dan nilai RMSE 0,0001588 pada pembagian data 90%:10% dengan *learning*

*rate* 0,1. Sedangkan Penelitian sebelumnya yang menggunakan metode *Long Short-Term Memory* telah banyak dilakukan diantaranya adalah Prediksi Harga Ethereum Berdasarkan Informasi *Blockchain* Menggunakan Metode *Long Short-Term Memory* yang ditulis oleh Nilda Aulia (2020) yang menjelaskan bahwa metode LSTM memerlukan beberapa parameter yang tepat guna menghasilkan prediksi yang akurat. Penelitian menganalisis beberapa parameter seperti jumlah neuron pada *hidden layer* dan *max epoch* yang paling tepat digunakan. Hasil analisis menunjukkan bahwa dengan menggunakan neuron 50 dan *max epoch* 500 mampu melakukan prediksi harga ethereum menggunakan informasi *blockchain* dengan baik, dilihat dari *error* yang sangat kecil yaitu MAPE sebesar 1.69 %.

Untuk meningkatkan akurasi prediksi dalam penelitian ini menggunakan algoritma optimasi *Adaptive moment estimation* (ADAM). Dalam beberapa penelitian terdahulu terbukti jika algoritma optimasi ADAM dipilih karena menghasilkan tingkat keakuratan yang tinggi dan terbaik diantara algoritma optimasi yang lain.

Berdasarkan permasalahan dan penjelasan yang telah diuraikan, peneliti tertarik untuk melakukan penelitian tentang prediksi harga emas menggunakan pendekatan *Recurrent Neural Network Long Short Term Memory* (RNN-LSTM), karena mengingat pentingnya prediksi harga emas bagi investor.

## LANDASAN TEORI

### Investasi Emas

Pada umumnya orang memilih berinvestasi dalam bentuk emas untuk memperoleh keuntungan (Maya Apriyanti, 2012: 37). Emas memiliki *supply* yang terbatas dan tidak mudah didapat, sementara permintaan terhadap emas tidak pernah berkurang, akibatnya harga emas cenderung mengalami kenaikan dari tahun ke tahun.

Faktor-faktor yang mempengaruhi harga emas sebagai berikut (Maya Apriyanti, 2012: 58):

1. Inflasi

Inflasi terjadi jika nilai mata uang mengalami penurunan sehingga harga barang di pasaran mengalami kenaikan. Kondisi inflasi seperti inilah yang mendorong naiknya harga emas.

2. Krisis Finansial  
Saat terjadi krisis finansial, orang lebih jeli dalam berinvestasi. Orang akan mencari keamanan dalam berinvestasi, sehingga mereka akan memilih investasi yang memberikan keuntungan.
3. Naiknya Permintaan Emas di Pasaran  
Harga emas dapat naik karena permintaan akan emas dipasaran yang mengalami peningkatan.
4. Kurs Dollar  
Karena harga emas dihitung berdasarkan kurs dollar, maka jika dollar mengalami kenaikan, harga emas akan ikut terdorong naik.
5. Harga Minyak  
Pada korelasi antara emas dan minyak, mereka memiliki hubungan berbanding lurus. Jika harga minyak melonjak, hal ini menyebabkan produksi emas akan menurun dikarenakan biaya produksi semakin mahal. Penambangan emas sangatlah bergantung pada minyak untuk operasionalnya.
6. Situasi Politik  
Kenaikan harga emas pada akhir tahun 2002 dan awal tahun 2003 terjadi karena sekutu yang dikomando AS akan melakukan serangan ke Iraq. Pelaku pasar beralih investasi dari pasar uang dan pasar saham ke investasi emas sehingga permintaan emas melonjak tajam.

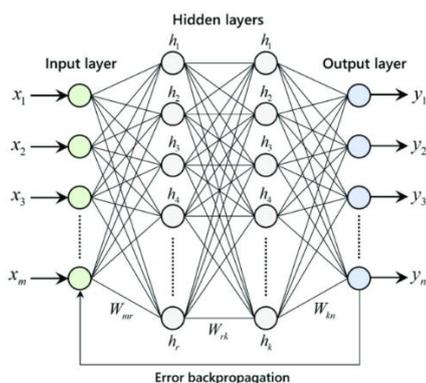
### **Artificial Neural Networks (ANN)**

*Artificial Neural Networks* (ANN) atau juga disebut Jaringan Saraf Tiruan (JST) adalah teknik pembelajaran mesin yang terinspirasi oleh jaringan saraf biologis di otak manusia. ANN terdiri dari sejumlah prosesor sangat sederhana dan saling berhubungan yang disebut *neuron* (Kurniawansyah, 2018). Penentuan jumlah lapisan dan jumlah *neuron* di lapisan tersembunyi dan koneksi di antara

*neuron* tersebut adalah hal yang penting (Saritas & Yasar, 2019). *Neuron* mempunyai karakteristik yang sama dalam ANN, terdiri dalam kelompok-kelompok yang disebut *layer*. *Neuron-neuron* dalam satu *layer* terhubung dalam *layer-layer* lainnya yang berdekatan. Kekuatan hubungan antar *neuron* yang berdekatan direpresentasikan dalam kekuatan hubungan atau bobot (Dharma, Putera, & Ardana, 2011).

Struktur ANN terdiri dari sejumlah lapisan *input layer* dan *output layer*, yang terkoneksi dan pada setiap koneksinya terdapat *weight* yang bisa diubah-ubah dengan tujuan untuk mendapatkan hasil prediksi sesuai dengan yang diinginkan, serta (dalam kebanyakan kasus) *hidden layer* yang terdiri dari unit yang mengubah input menjadi sesuatu yang dapat digunakan oleh lapisan output. Berikut merupakan lapisan yang ada dalam ANN:

1. *Input layer* (Lapisan masukan): Lapisan terluar yang menghubungkan sumber data ke dalam jaringan pemrosesan. Setiap masukan akan merepresentasikan variabel-variabel bebas yang nantinya akan berpengaruh terhadap keluaran.
2. *Hidden layer* (Lapisan tersembunyi): Lapisan perambat variable-variable input untuk mendapatkan hasil output yang sesuai dengan keinginan. Pada ANN multi layer dapat memiliki satu atau lebih *hidden layer*.
3. *Output layer* (Lapisan keluaran): Lapisan yang merupakan hasil dari pemrosesan ANN. Output yang dihasilkan dipengaruhi oleh bobot, jumlah lapisan tersembunyi (*hidden layer*), dan fungsi aktivasi yang diterapkan.



Gambar 1. Struktur Jaringan ANN

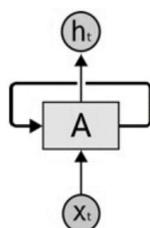
dikatakan RNN di desain khusus untuk *handle* data berurutan. *Input*  $x_t$  menghasilkan *output*  $h_t$ .

*Recurrent Neural Network* (RNN) terdiri dari unit *input*, unit *output*, dan unit tersembunyi (*hidden*). Model RNN pada dasarnya memiliki aliran informasi satu arah dari unit input ke unit tersembunyi, dan sintesis aliran informasi satu arah dari unit tersembunyi sementara sebelum ke unit tersembunyi waktu saat ini. Unit tersembunyi dapat dilihat sebagai penyimpanan seluruh jaringan, yang mengingat informasi dari ujung ke ujung (Yin, 2017).

### Recurrent Neural Network (RNN)

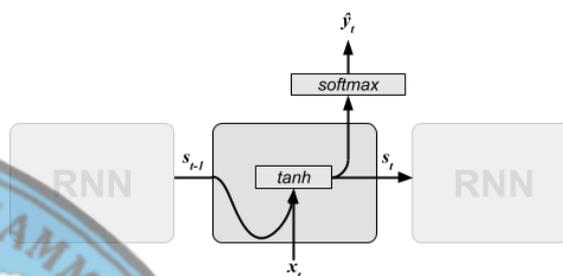
RNN adalah sejenis jaringan saraf tiruan yang memiliki kemampuan yang untuk melihat korelasi tersembunyi yang terjadi pada data dalam aplikasi untuk pengenalan suara, pemrosesan bahasa alami, dan prediksi deret waktu. RNN sangat baik untuk masalah pemodelan urutan dengan beroperasi pada informasi input serta jejak informasi yang diperoleh sebelumnya karena koneksi berulang (Tian & dkk, 2018).

*Recurrent Neural Network* (RNN) pertama kali dikembangkan oleh Jeff Elman pada tahun 1990. RNN merupakan variasi dari *Artificial Neural Network* (ANN) yang di rancang khusus untuk memproses data yang bersambung atau berurutan. RNN biasa digunakan untuk menyelesaikan masalah dengan data *time series*.



Gambar 2. Proses perulangan informasi pada RNN

Pada gambar 2. di atas dapat dijelaskan, bahwa struktur dari RNN akan lebar tengahnya selebar panjang pola data yang ingin di pelajari oleh RNN karena itu



Gambar 3. layer tanh pada RNN

$$S_t = \tanh(U \cdot x_t + W \cdot S_{t-1}) \quad (1)$$

$$\hat{y}_t = \text{softmax}(V \cdot S_t) \quad (2)$$

Pada model RNN sinyal dapat mengalir secara *forward* dan *backward* secara berulang. Untuk bisa melakukan hal tersebut, maka ditambahkan sebuah *layer* baru yang disebut dengan *context layer*. Selain melewati *input* antar *layer*, *output* dari setiap *layer* juga menuju ke *context layer* untuk digunakan sebagai inputan pada timestep berikutnya. RNN menyimpan informasi di *context layer*, yang membuatnya dapat mempelajari urutan data dan menghasilkan output atau urutan lain.

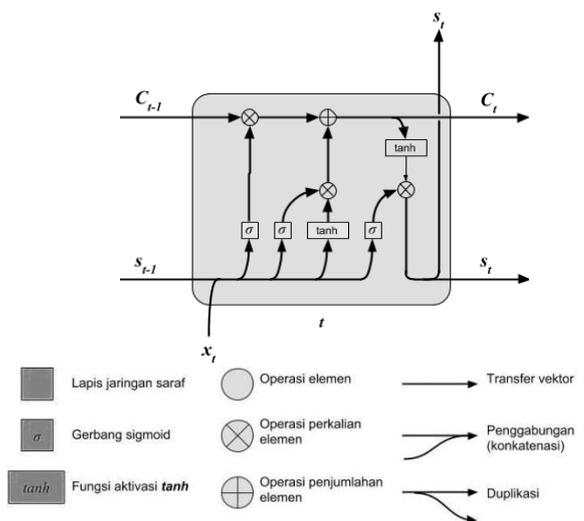
Berbeda dengan koneksi bobot yang dibangun antara lapisan dalam jaringan saraf dasar, RNN dapat menggunakan status *internal* (memori) untuk memproses urutan *input*. Status tersembunyi menangkap informasi pada titik waktu sebelumnya, dan *output* berasal dari waktu saat ini dan memori sebelumnya. RNN berkinerja baik ketika output dekat dengan input terkait informasi dari node sebelumnya diteruskan ke node berikutnya. Secara teori,

RNN juga mampu menangani ketergantungan jangka panjang. Namun, dalam aplikasi praktis, RNN tidak dapat menyimpan informasi sebelumnya dengan baik ketika interval waktu lama karena masalah gradien menghilang. Untuk mengatasi kelemahan ini dan meningkatkan kinerja RNN, sebuah tipe khusus arsitektur RNN yang disebut LSTM diusulkan oleh Hochreiter & Schmidhuber (1997) (Tian & dkk, 2018).

### Long Short Term Memory (LSTM)

*Long Short Term Memory (LSTM)* merupakan pengembangan dari *Recurrent Neural Network (RNN)* dengan mengatasi salah satu kekurangan RNN yaitu kemampuan pengelolaan informasi dalam periode yang lama. Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, LSTM banyak dipilih untuk prediksi berbasis waktu atau time-series karena dikenal lebih unggul dan handal dalam melakukan prediksi dalam waktu lama dibanding algoritma lain (Zahara, Sugianto, & Ilmiddafiq, 2019).

LSTM baik dalam mengingat informasi untuk waktu yang lama. Pada RNN perulangan jaringan hanya menggunakan satu layer sederhana, yaitu layer tanh. (Habibie, 2018). Karena lebih banyak informasi sebelumnya dapat mempengaruhi akurasi model, LSTM menjadi pilihan penggunaan yang wajar. Komponen dasar LSTM adalah status sel, sebuah baris yang berjalan dari *memory cell* sebelumnya ( $S_{t-1}$ ) ke *memory cell* saat ini ( $S_t$ ). *Memory cell* adalah garis horizontal yang menghubungkan semua *output layer* pada LSTM. Dengan adanya jalur tersebut, suatu nilai *memory cell* yang lama akan dengan mudah diteruskan ke *memory cell* yang baru dengan sedikit sekali modifikasi. Ini memungkinkan informasi mengalir lurus ke bawah. Jaringan dapat menentukan jumlah informasi sebelumnya mengalir. Itu dikendalikan melalui lapisan pertama ( $\sigma_1$ ) (Kumar & dkk, 2018).



Gambar 4. Ilustrasi LSTM

Pada gambar ilustrasi LSTM, setiap garis membawa seluruh vektor, dari output satu simpul (*node*) ke input yang lain. Lingkaran merah muda mewakili operasi elemen, seperti penambahan atau perkalian elemen vektor, sedangkan kotak kuning adalah lapis jaringan saraf (mengandung parameter dan bias) yang bisa belajar. Dua garis yang bergabung menandakan penggabungan dua matriks/vektor, sementara garis berpisah menandakan kontennya disalin dan salinannya pergi ke simpul yang berbeda. Model LSTM menyaring informasi melalui struktur gerbang untuk mempertahankan dan memperbarui keadaan sel memori. Struktur pintunya mencakup *input gate*, *forget gate*, dan *output gate*.

Dari gambar diatas didapatkan beberapa rumus persamaan sebagai berikut:

1. *Forget gate*

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

Dimana:

$f_t$  = forget gate

$\sigma$  = fungsi sigmoid

$W_f$  = nilai weight 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

$$I_t = \sigma(W_I \cdot [h_{t-1}, x_t] + b_I) \quad (4)$$

Dimana:

$I_t$  = input gate

$\sigma$  = fungsi sigmoid

$W_I$  = nilai weight untuk input gate

$h_{t-1}$  = nilai output sebelum orde ke t

$x_t$  = nilai input pada orde ke t

$b_I$  = nilai bias pada input gate

## 3. Memory update

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (5)$$

$$C_t = f_t \times C_{t-1} + I_t \times \tilde{C}_t \quad (6)$$

Dimana:

$\tilde{C}_t$  = nilai baru yang dapat ditambahkan ke cell state

$\tanh$  = fungsi tanh

$W_C$  = nilai weight untuk 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 gate

$C_t$  = cell state

$f_t$  = forget gate

$C_{t-1}$  = cell state sebelum orde ke t

$I_t$  = input gate

## 4. Output gate

$$O_t = \sigma(W_O [h_{t-1}, x_t] + b_O) \quad (7)$$

$$h_t = O_t * \tanh(C_t) \quad (8)$$

Dimana:

$O_t$  = output gate

$\sigma$  = fungsi sigmoid

$W_O$  = nilai weight untuk output gate

$h_{t-1}$  = nilai output sebelum orde ke t

$x_t$  = nilai input pada orde ke t

$b_O$  = nilai bias pada output gate

$h_t$  = nilai output orde t

$O_t$  = output gate

$\tanh$  = fungsi tanh

$C_t$  = cell state

## Algoritma Optimasi Adaptive Moment Estimation (ADAM)

Optimasi berbasis gradien stokastik adalah kepentingan praktis inti dalam banyak bidang sains dan teknik. Banyak masalah dalam bidang ini dapat dilemparkan sebagai optimalisasi beberapa fungsi tujuan berstandarisasi scalar yang membutuhkan maksimalisasi atau minimalisasi berkenaan dengan parameternya. *Adaptive Moment Estimation* (ADAM) adalah metode optimasi yang menghitung tingkat pembelajaran secara adaptif untuk setiap parameter. Adam merupakan algoritma optimasi yang dikembangkan dengan memanfaatkan kelebihan dari algoritma *Adaptive Gradient* (AdaGrad) yang bekerja dengan baik dengan gradien jarang dan *Root Mean Square Propagation* (RMSProp) yang bekerja dengan baik secara *online* dan non Pengaturan Stasioner. Adam tidak hanya mengadaptasi tingkat pembelajaran parameter berdasarkan rata-rata pertama (*mean*) seperti dalam RMSProp, namun ADAM juga menggunakan rata-rata kedua dari gradien (*varians uncentered*). Algoritma menghitung rata-rata pergerakan eksponensial dari gradien dan gradient kuadratnya, dan parameter  $\beta_1$  dan  $\beta_2$  mengontrol tingkat peluruhan rata-rata pergerakan (Aldi, Jondri, & Aditsania, 2018).

Beberapa keuntungan ADAM adalah bahwa besarnya pembaruan parameter tidak sama dengan penskalaan gradien, ukurannya kira-kira dibatasi oleh *hyperparameter stepsize*, tidak memerlukan alat tulis stasioner. *Adaptive Moment Estimation* (ADAM) adalah metode yang menghitung *adaptive learning rate* untuk setiap parameter. Nilai parameter yang direkomendasikan adalah  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , dan  $\epsilon = 10^{-8}$  dengan  $\beta_1 = \beta_2 =$  tingkat penurunan eksponensial dan  $\epsilon =$  nilai epsilon untuk update parameter (Kingma & Ba, 2015).

## Parameter Evaluasi

Dalam mengevaluasi akurasi dan peramalan kinerja model berbeda, beberapa indeks evaluasi yang sering digunakan salah satu diantaranya yaitu *Mean Absolute Percentage Error* (MAPE). *Mean Absolute Percentage Error* adalah nilai *absolute* dari persentase *error* data teradap *mean*. Sehingga didapatkan persamaan rumus sebagai berikut:

$$MAPE = \frac{\sum_{i=1}^n \frac{|\hat{Y}_i - Y_i|}{Y_i}}{n} * 100\% \quad (9)$$

Dimana  $n$  merupakan jumlah data dalam penelitian. Sedangkan  $Y_i$  adalah nilai data yang sebenarnya (aktual) dan  $\hat{Y}_i$  adalah nilai data prediksi.

Berikut merupakan nilai parameter evaluasi MAPE (Julio Warmansyah & Dida Hilpiah, 2019):

Tabel 1. Rentang nilai MAPE

Nilai MAPE	Akurasi Peramalan MAPE
$MAPE \leq 10\%$	Tinggi
$10\% < MAPE \leq 20\%$	Baik
$20\% < MAPE \leq 50\%$	Reasonable
$MAPE \geq 50\%$	Rendah

## METODE PENELITIAN

### Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data ini merupakan data historis harga emas yang *real time* dan diperoleh dari *website www.investing.com*. Dalam penelitian ini mengambil data harian harga emas, indeks dollar dan harga minyak dunia dengan menggunakan harga harian dalam periode waktu 1 Januari 2016 sampai dengan 26 Februari 2021. Sehingga total data yang digunakan adalah sebanyak 1330 data penelitian.

### Variabel dan Struktur Data

Tabel 2. Variabel Penelitian

No.	Variabel	Keterangan
1	Harga Emas	Harga emas yang

(Y)		dikonversi kedalam United States Dollar (USD)
2	Harga Emas ( $X_1$ ) atau ( $Y_{t-1}$ )	Lag 1 dari harga emas yang dikonversi kedalam United States Dollar (USD)
3	Harga Minyak Dunia ( $X_2$ )	Harga minyak dunia yang dikonversi kedalam United States Dollar (USD)
4	Indeks Dollar ( $X_3$ )	Indeks United States Dollar (USD) terhadap mata uang asing lainnya

Tabel 3. Struktur Data Penelitian

Y	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>
Y <sub>1</sub>	X <sub>1,1</sub>	X <sub>2,1</sub>	X <sub>3,1</sub>
Y <sub>2</sub>	X <sub>1,2</sub>	X <sub>2,2</sub>	X <sub>3,2</sub>
⋮	⋮	⋮	⋮
Y <sub>1330</sub>	X <sub>1,1330</sub>	X <sub>2,1330</sub>	X <sub>3,1330</sub>

### Langkah Penelitian

Dalam penelitian ini menggunakan *software Python* dan langkah-langkah dalam penelitian ini adalah sebagai berikut:

1. *Preprocessing* data, *preprocessing* mengubah data yang belum terstruktur menjadi data yang terstruktur dengan kebutuhan prosesnya, maka akan dibagi menjadi ekstraksi fitur dan *ascending*. Ekstraksi fitur baru yaitu dengan menambahkan rata-rata pada data untuk diolah. *Ascending* dilakukan untuk mengurutkan data harga emas, *ascending* yang dibutuhkan untuk proses ini yaitu mengurutkan data emas dari yang terdahulu hingga yang terbaru.
2. Analisis deskriptif data harga emas
3. Membentuk model RNN LSTM
4. Melakukan analisis RNN LSTM dengan tahapan sebagai berikut:
  - Melakukan normalisasi dataset dengan mengubah data aktual menjadi nilai range interval

[0,1] menggunakan *min-max scaling* seperti pada rumus

$$x' = \frac{x - \min_x}{\max_x - \min_x} \quad (10)$$

- Membagi data menjadi data *training* 80% dan data *testing* 20%
  - Menentukan jumlah *neuron* pada *hidden layer* untuk percobaan masing-masing adalah 10, 20, 30, 40, dan 50.
  - Menentukan jumlah *epoch* untuk percobaan masing-masing yang akan digunakan sebesar 100, 500, dan 1000.
  - Melakukan optimasi model dengan algoritma optimasi *Adaptive Moment Estimation (ADAM)*
  - Melakukan prediksi pada data.
5. Denormalisasi data prediksi, proses pengembalian data aktual untuk mengetahui prediksi yang sebenarnya. Adapun rumus yang digunakan dalam denormalisasi seperti rumus sebagai berikut:
- $$dn_i = ((n_i)(x_{max} - x_{min})) + (x_{min}) \quad (11)$$
6. Melakukan evaluasi dengan MAPE dari hasil prediksi.

## HASIL PENELITIAN dan PEMBAHASAN

### Preprocessing Data

#### 1. Ekstraksi Fitur

Ekstraksi fitur untuk menambahkan fitur baru yaitu fitur rata-rata. Nilai rata-rata didapatkan dari:

$$rata - rata = \frac{Buka + Tertinggi + Terendah + Tutup}{4} \quad (12)$$

#### 2. Ascending

*Ascending* dilakukan untuk mengurutkan data harga emas, *ascending* yang dibutuhkan untuk proses ini yaitu mengurutkan data emas dari yang terdahulu hingga yang terbaru.

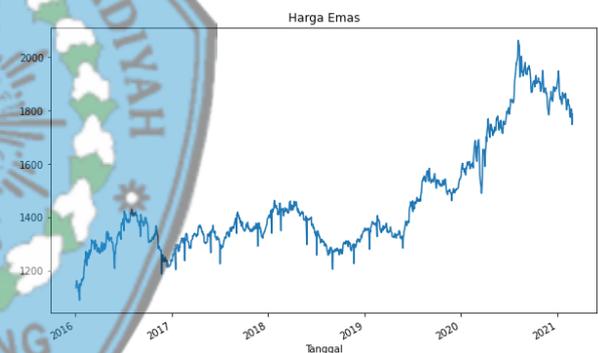
Data hasil dari ekstraksi fitur dan *ascending* ini dapat dilihat pada tabel berikut:

Tabel 4. Harga emas setelah *Preprocessing*

Tanggal	Harga Emas
04/01/2016	1135,5
05/01/2016	1138,4
06/01/2016	1151,9
07/01/2016	1162,7
08/01/2016	1151,7
⋮	
24/02/2021	1799,53
26/02/2021	1747,1
26/02/2022	1787,3

### Statistika Deskriptif

Sebelum menggunakan metode *Long Short Term Memory*, peneliti melakukan analisis deskriptif terlebih dahulu terhadap data historis harga emas untuk mendapatkan gambaran umum tentang harga emas selama 5 tahun terakhir dari 4 Januari 2016 hingga 26 Februari 2021.



Gambar 4. Grafik Harga Emas

Berdasarkan gambar grafik harga emas terlihat bahwa trend harga emas cenderung naik atau mengalami kenaikan. Dapat dilihat pada tanggal 6 Agustus 2020, emas mencapai harga tertinggi yaitu sebesar US\$ 2063,4 per *troy ounce*, harga emas terendah sekitar US\$ 1089,5 per *troy ounce* terjadi pada awal tahun yaitu 18 Januari 2016. Sedangkan untuk rata-rata harga emas dari tahun 1 Januari 2016 sampai dengan 26 Februari 2021 adalah sebesar US\$ 1456,09 per *troy ounce*.

### Analisis Recurrent Neural Network Long Short Term Memory

## 1. Normalisasi

Saat melakukan analisis LSTM, terlebih dahulu lakukan normalisasi kumpulan data untuk meminimalkan kesalahan, data aktual akan diubah menjadi rentang nilai dengan menggunakan rumus *min-max scaling* dalam rumus 2.15 yaitu menggunakan *range interval* [0,1].

Tabel 5. Hasil Normalisasi Data  $X_1 - X_3$  & Y

No.	$X_1$	$X_2$	$X_3$	Y
1	0,047	0,555	0,674	0,050
2	0,050	0,545	0,703	0,064
3	0,064	0,529	0,715	0,075
4	0,075	0,510	0,676	0,064
5	0,064	0,509	0,667	0,061
⋮	⋮	⋮	⋮	⋮
1325	0,704	0,813	0,146	0,725
1326	0,724	0,821	0,070	0,737
1327	0,736	0,840	0,124	0,737
1328	0,729	0,844	0,130	0,729
1329	0,675	0,857	0,122	0,717

## 2. Pembentukan data *Training* dan *Testing*

Data penelitian dibagi menjadi data *training* dan data *testing* dengan perbandingan 80% data *training* atau sebanyak 1063 dan 20% untuk data *testing* atau sebanyak 266.

Tabel 6. Data *Training* & *Testing*

Data	Persen	Jumlah
Training	80%	1063
Testing	20%	266

Penggunaan data *training* sebagian besar dilakukan untuk pembelajaran mesin atau algoritma pembelajaran dilatih dengan lebih baik melalui pola data dalam data *training*. Data *training* tersebut dilatih

dengan menggunakan metode *Long Short Term Memory*.

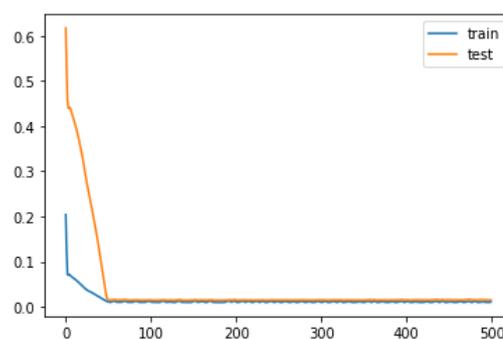
## 3. Penentuan Jumlah *Neuron* dan *Epoch*

Menentukan jumlah *neuron* yang optimal dalam *hidden layer* dan *epoch* adalah tugas penting dalam *Recurrent Neural Network Long Short Term Memory*. Ada banyak *neuron* pada *hidden layer* yang mungkin tidak mampu menangkap hubungan yang kompleks antara variabel target dan input. Namun, jika terlalu banyak *neuron* dialokasikan ke bagian jaringan yang tersembunyi, hal itu akan memperburuk kemampuan prediksi jaringan dan karena parameterisasi yang berlebihan (*overparameterization*) membuat data tidak terlihat. Oleh karena itu, sejumlah *neuron* dalam lapisan tersembunyi harus dicoba, dan kinerja arsitektur harus dinilai setelah setiap kali dijalankan dengan sejumlah *neuron* di lapisan tersembunyi.

Pada penelitian ini dibentuk jaringan yang berisi 3 variabel input dan 1 variabel output layer dengan jumlah *neuron* yang akan digunakan pada *hidden layer* untuk percobaan masing-masing adalah 10, 20, 30, 40, dan 50. Sedangkan untuk jumlah *epoch* akan digunakan sebesar 100, 500, dan 1000. Untuk mengetahui jumlah *neuron* dan jumlah *epoch* yang sesuai dapat dilihat dari nilai *loss* yang terkecil, dimana nilai *loss* merupakan nilai dari MAE, serta menggunakan optimasi parameter ADAM. *Mean Absolute Error* (MAE) merupakan pengukuran yang umum digunakan untuk prediksi *error* pada analisis *time series*. Alasan mengapa optimasi ADAM digunakan dalam penelitian ini adalah karena menggunakan teknik *bias correction*. Tidak ada aturan untuk menentukan jumlah *neuron* dan jumlah *epoch* sehingga jumlah *neuron* dan *epoch* didapatkan melalui percobaan sampai dengan mendapatkan yang paling optimal untuk memprediksi *time series*.

Tabel 7. hasil percobaan *neuron* dan *epoch*

No.	Neuron	Epoch	Error (MAE)
1		100	24,5896
2	10	500	27,4147
3		1000	29,1441
4		100	20,8285
5	20	500	22,7218
6		1000	22,7078
7		100	15,0189
8	30	500	14,5923
9		1000	15,9843
10		100	17,6587
11	40	500	15,1159
12		1000	16,9097
13		100	15,5947
14	50	500	16,8268
15		1000	17,3278



Gambar 5. Model Loss

Tabel 7. menunjukkan akurasi terbaik untuk jumlah *neuron* dan *epoch* adalah *neuron* sebanyak 30 sedangkan *epoch* (iterasi) sebanyak 500 yang menghasilkan nilai *error* terendah yaitu 14,5923. Jumlah *epoch* itu sendiri mewakili durasi proses pembelajaran yang dilakukan terhadap jaringan yang sedang diamati. Jumlah *epoch* yang terlalu sedikit mengakibatkan jaringan yang terbentuk bersifat terlalu general (umum), artinya kemampuan jaringan dalam mengenali pola terlalu sedikit atau tidak ada sama sekali. Sedangkan *epoch* yang terlalu banyak dapat mengakibatkan jaringan mengalami kondisi *overfit* atau jaringan terlalu bersifat terlalu spesifik terhadap data pelatihan seperti yang ditunjukkan pada tabel 6., untuk hasil terbaik bukan berada pada nilai *epoch* terbesar, melainkan dengan jumlah *neuron* sebanyak 30 dan *epoch* 500, arsitektur inilah yang akan digunakan dalam prediksi harga emas.

Pada gambar 4.2 menunjukkan hasil nilai *loss* yang merupakan nilai dari perhitungan *loss function* dari *training* dataset dan prediksi dari modelnya. Dan *validation loss* adalah nilai perhitungan *loss function* dari *validation* dataset dan prediksi dari modelnya dengan input data dari *validation* dataset. Semakin kecil nilai *error* yang diperoleh dari persamaan model *loss* maka semakin tinggi akurasi yang didapatkan.

Nilai bobot dan bias yang didapatkan dari 30 *neuron* dan 500 *epoch* dapat dilihat pada tabel 8 dan tabel 9

Tabel 8. Hasil bobot dari setiap *neuron* variabel harga emas

No.	$W_i$	$W_f$	$W_c$	$W_o$
1	-	-	-	-
2	0,1195	0,0707	0,5491	0,0847
3	0,0421	-	-	-
4	1	0,1468	0,2464	0,0013
5	0,2856	0,1225	-	0,3504
6	8	6	0,4552	4
7	-	0,0300	0,5422	0,008
8	0,3656	6	8	
9	0,1631	0,0924	0,3462	0,1461
10	3	9	5	2
11	⋮	⋮	⋮	⋮
12	0,0368	0,1863	-	0,0314
13	8	3	0,2793	2
14	-	0,0860	0,1540	0,0412
15	0,1228	5	4	3
16	0,1773	-	-	0,4199
17	8	0,1035	0,2764	5
18	0,1844	-	0,3740	0,1967
19	4	0,1334	4	1

30	0,1413	-0,005	0,4511	0,5902
	1		4	8

Tabel 9. bias untuk setiap *neuron*

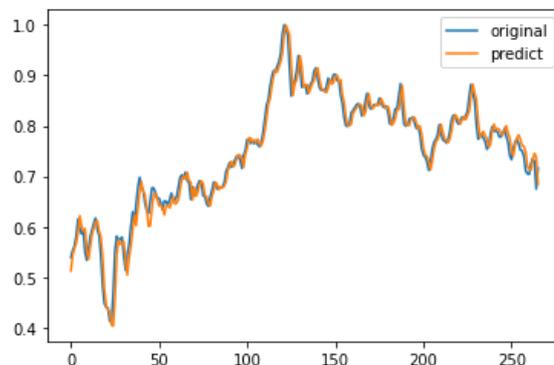
No.	b <sub>i</sub>	b <sub>f</sub>	b <sub>c</sub>	b <sub>o</sub>
1	0,25436	1.	-0,0182	0,25562
2	0,02692	1.	0,01202	0,02713
3	0,14403	1.	-0,0046	0,14691
4	0,22463	1.	0,0126	0,22325
5	0,04214	1.	-0,0094	0,04445
⋮	⋮	⋮	⋮	⋮
26	0,01581	1.	0,00674	0,01377
27	-0,0678	1.	-0,036	-0,0732
28	0,03926	1.	-0,0006	0,03085
29	0,04318	1.	-0,001	0,04862
30	0,3818	1.	0,00843	0,404

Pada  $W_i[0]$ ,  $W_f[0]$ ,  $W_c[0]$ , dan  $W_o[0]$  adalah bobot dari variabel pertama (harga emas) untuk *input gate*, *forget gate*, *cell gate*, dan *output gate*. *Input gate* merupakan pintu gerbang yang menentukan banyak input jaringan saat disimpan dalam status sel, *forget gate* merupakan gerbang untuk menentukan informasi status sel mana yang akan dibuang dari model atau *memory cell*, sedangkan *output gate* merupakan gerbang yang mengontrol seberapa banyak keadaan sel saat ini dibuang. Banyaknya bobot sesuai dengan banyaknya *neuron* yang digunakan adalah 30, dan masing-masing variabel memiliki bobotnya masing-masing. Sehingga ketika dijumlahkan terdapat 90 bobot untuk masing-masing *input gate*, *forget gate*, *cell state*, dan *output gate*. Sedangkan untuk tabel 4.6  $b_i$ ,  $b_f$ ,  $b_c$ , dan  $b_o$  adalah bias untuk input *input gate*, *forget gate*, *cell gate*, dan *output gate*. Banyaknya bias sesuai dengan banyak *neuron* yang digunakan yaitu 30 jadi jika dijumlahkan terdapat sebanyak 120 bias secara keseluruhan.

### Prediksi Harga Emas

Setelah mendapatkan arsitektur terbaik untuk prediksi, dengan menggunakan data *training* 80% dan data *testing* 20% serta

menggunakan *neuron* sebanyak 30 dan *epoch* sebanyak 500, maka didapatkan grafik perbandingan data aktual dari harga emas dengan data prediksi harga emas yang ditunjukkan pada gambar 6.



Gambar 6. Grafik perbandingan data aktual dan data prediksi harga emas

Gambar 6. ini menunjukkan bahwa dengan menggunakan model yang terbentuk menghasilkan *output* yang sesuai, terlihat pada grafik dimana pola garis testing dengan pola garis hampir sama dan tidak jauh berbeda. Langkah selanjutnya adalah dilakukannya denormalisasi data untuk dapat menampilkan hasil prediksi dan nilai prediksi aktual.

Tabel 10. Hasil Prediksi Harga Emas

No.	Tanggal	Data Aktual	Hasil Prediksi dalam Min-Max Scaler	Hasil Prediksi
1	18 Februari 2020	1615,2	0,51278	1589,27
2	19 Februari 2020	1628,83	0,54359	1619,28
3	20 Februari 2020	1637	0,55825	1633,56
4	21 Februari 2020	1657,05	0,56721	1642,29
5	24 Februari 2020	1688,88	0,58843	1662,95
⋮	⋮	⋮	⋮	⋮
262	22 Februari	1795,35	0,71228	1783,58

	2021			
263	23 Februari 2021	1806,23	0,732	1802,78
264	24 Februari 2021	1799,53	0,7452	1815,64
265	25 Februari 2021	1747,1	0,73853	1869,15
266	26 Februari 2021	1787,3	0,68436	1756,38

Tabel 10. merupakan hasil perhitungan prediksi harga emas. Hasil prediksi tersebut memiliki tingkat akurasi sebesar 98,91% berdasarkan nilai *Mean Absolute Percentage Error* (MAPE), dengan nilai MAPE sebesar 1,09% yang dimana nilai tersebut menunjukkan bahwa hasil prediksi sangat baik.

## SIMPULAN dan SARAN

### Simpulan

Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan prediksi harga emas dengan metode RNN-LSTM ini data *training* 80% dan data *testing* 20% serta menggunakan jumlah *neuron* 10, 20, 30, 40, dan 50, sedangkan jumlah *epoch* yang digunakan yaitu sebanyak 100, 500, dan 1000, berdasarkan nilai *error* (MAE) didapatkan arsitektur terbaik untuk prediksi yaitu dengan menggunakan *neuron* sebanyak 30 dan *epoch* sebanyak 500. Pada pemodelan terbaik menghasilkan prediksi harga emas pada tanggal 18 Februari 2020 sampai dengan 26 Februari 2021. Dari hasil tersebut terlihat jika hasil prediksi harga emas tertinggi terjadi pada tanggal 7 Agustus 2020 sebesar US\$ 2063,11 per *troy ounce* dan prediksi harga emas terendah terjadi pada tanggal 23 Maret 2020 yaitu sebesar US\$ 1482,66 per *troy ounce*. Sedangkan dari hasil perhitungan prediksi harga emas didapatkan hasil prediksi tersebut memiliki tingkat akurasi sebesar 98,91% berdasarkan nilai *Mean Absolute Percentage Error* (MAPE), dengan nilai MAPE sebesar 1,09% yang dimana nilai tersebut

menunjukkan bahwa hasil prediksi sangat baik.

### Saran

Saran yang diberikan oleh peneliti untuk penelitian selanjutnya tentang metode RNN-LSTM adalah bisa menggunakan dan membandingkan dengan optimasi lain (misalnya AdaMax, RMSprop, dll) sehingga diperoleh hasil prediksi yang lebih akurat. Serta diharapkan penelitian selanjutnya dapat menambahkan variabel lain yang mempengaruhi naik turunnya harga emas dunia.

### Daftar Pustaka

- Abdul, Halim. (2005). Analisis Investasi. Edisi Kedua. Jakarta: Salemba Empat. Alfabeta.
- Arfan Adhib, Lusiana ETP. (2020). Perbandingan Algoritma Long Short-Term Memory dengan SVR pada Prediksi Harga Saham di Indonesia. Universitas Gunadarma. Jurnal Pengkajian dan Penerapan Teknik Informatika, Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018.
- Aulia, Nilda. (2020). Prediksi Harga Ethereum Berdasarkan Informasi Blockchain Menggunakan Metode Long Short Term Memory. Tugas Akhir. Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia Yogyakarta.
- Colah. (2015). Retrieved from Colahsblog: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Dharma, S., Putera, A., & Ardana, P. D. (2011). Artificial Neural Networks Untuk Pemodelan Curah Hujan-Limpasan pada Daerah Aliran Sungai (DAS) di Pulau Bali. Jurnal Bumi Lestari, Volume 11 No. 1, Pebruari 2011, hlm. 9-22.
- Dipraja, S. (2011) Siapa Bilang Investasi Emas Butuh Modal Gede? (Rp 200 Ribu Bisa Kok!). Jakarta: PT. Tangga

- Pustaka.
- Fachruddin, M. Azzam. (2019). Implementasi Metode Elman Recurrent Neural Network (ERNN) Untuk Prediksi Harga Emas. Tugas Akhir. Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau Pekanbaru.
- Ghiffary, Ghardapaty Ghaly. (2020). Penerapan Artificial Neural Network Menggunakan Software R. <https://medium.com/@17611083/penerapan-artificial-neural-network-menggunakan-software-r-f8cfee4dfc38>
- Habibi, M. Y., & Riksakomara, E. (2017). Peramalan Harga Garam Konsumsi Menggunakan Artificial Neural Network Feedforward-Backpropagation. *TEKNIK ITS*, 307.
- Hansson, Magnus. (2017). On stock return prediction with LSTM networks. Seminar: 1st of June 2017, 4:15 pm, EC1:270 Lund, Department of Economics Lund University.
- IndoML. (2018). Pengenalan Long Short Term Memory (LSTM) dan Gated Recurrent Unit (GRU) – RNN Bagian 2. Belajar Pembelajaran Mesin Indonesia: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>
- Kumar, Jitendra & dkk. (2018). Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters. *Procedia Computer Science* 125 (2018) 676-682.
- Kurniawansyah, A. S. (2018). Implementasi Metode Artificial Neural Network dalam Memprediksi Hasil Ujian Kompetensi Kebidanan. *Pseudocode*, 38.
- Kusumadewi, Felasufah. (2014). Peramalan Harga Emas Menggunakan Feedforward Neural Network dengan Algoritma Backpropagation. Skripsi. Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta.
- Larasati, Kinanti Dhea. (2020). Prediksi Harga Bitcoin Berdasarkan Informasi Blockchain Menggunakan Metode Long-Short Term Memory. Tugas Akhir. Program Studi Statistika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Islam Indonesia Yogyakarta.
- Ma, X. (2015). Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data. *Transportation Research Part C*, 191. ScienceDirect, Elsevier.
- Maya Apriyanti. (2012). Anti Rugi dengan Berinvestasi Emas. Yogyakarta: Pustaka Baru Press.
- Mulyaningsih, Widya Sri. (2019). Machine Learning 2 (Artificial Neural Network). <https://medium.com/@16611021/machine-learning-2-artificial-neural-network-b13f2e92bbd6>
- Qiu J, Wang B, Zhou C. (2020). Forecasting Stock Prices with Long-Short Term Memory Neural Network Based On Attention Mechanism. *PLOS ONE* 15(1): e0227222.
- Rizal, Ahmad Ashril. (2018). Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory. *ResearchGate* hal 4-5.
- Rizki Muhammad, Basuki Setio, Azhar Yufis. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory Untuk Prediksi Curah Hujan Kota Malang. *Teknik Informatika*, Universitas Muhammadiyah Malang: *REPOSITOR*, Vol. 2, No. 3, Maret 2020, Pp. 331-338.
- Salim, J. (2010). Jangan Investasi Emas Sebelum Baca Buku Ini! Jakarta:

Visimedia.

Sherstinsky, Alex. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. Elsevier journal, Vol. 404, March 2020.

Tian C, Ma J, Zhang C, Zhan P. (2018). A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. MDPI, *Energies* 2018, 11, 3493; doi:10.3390/en11123493.

Warmansyah, Julio & Hilpiah, Dida. (2019). Penerapan metode fuzzy sugeno untuk prediksi persediaan bahan baku. *Jurnal Ilmiah Teknologi-Informatika & Sains TEKNOIS*, Volume 9 Number 2 November 2019 Page. 12-20.

Yin, C. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Network. *IEE*, 21955-21956. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8066291>.

Zaman, Lukman, dkk. (2019). Analisis Kinerja LSTM dan GRU sebagai Model Generatif untuk Tari Remo. *JNTETI*, Vol. 8, No. 2 Mei 2019

