

PENERAPAN METODE *GATED RECURRENT UNIT* UNTUK PREDIKSI HARGA TIMAH GLOBAL DENGAN OPTIMASI *ROOT MEAN SQUARE PROPAGATION* DAN *ADAPTIVE MOMENT ESTIMATION*

Rahayu Putri Andini^{1*}, Arief Rachman Hakim², Iut Tri Utami³

¹Mahasiswa Jurusan Statistika FSM Universitas Diponegoro

^{2,3}Staff Pengajar Jurusan Statistika FSM Universitas Diponegoro

*e-mail : rahayuputriandinii@gmail.com

DOI: 10.14710/J.GAUSS.XX.X.XX-XX

Article Info:

Received:

Accepted:

Available Online:

Keywords:

Tin; Forecasts; Gated Recurrent Unit; RMSProp; Adam.

Abstract: The mining and metal trading industry is one of the key contributors to the global economy. Tin as one of the basic mineral commodities is widely used in various sectors. A deep understanding of market behavior and predicting global tin prices is crucial in preparing for and mitigating risks for companies. This research aims to find the best optimization model and method and to determine the influence of applying momentum and bias correction in predicting tin prices. The dataset used in this research is the daily closing price of tin from January 1, 2019, to December 31, 2023. The prediction results from two optimization methods, Root Mean Square Propagation (RMSProp) and Adaptive Moment Estimation (Adam), were then compared to determine the best and most optimal model. The results showed that the model with the Adam optimization method at a batch size of 32, epoch 100, learning rate of 0.01, and 16 hidden units produced the best model with a MAPE value of 1.24%. The final results also showed that the combined application of momentum and bias correction in the Adam optimization method had an impact on the stability of the tin price prediction model.

1. PENDAHULUAN

Industri pertambangan dan perdagangan logam memiliki peran vital dalam perekonomian global, salah satu diantaranya yaitu timah (PT Timah, 2024). Timah telah digunakan secara luas dalam berbagai sektor seperti otomotif, elektronik, dan konstruksi. Keputusan Menteri ESDM Republik Indonesia pada 14 September 2023 yang menetapkan timah sebagai mineral kritis, menegaskan pentingnya timah untuk perekonomian dan pertahanan negara. Produksi timah dunia pada tahun 2023 mencapai 370.100 metrik ton atau turun sekitar 2,1% dari tahun sebelumnya. Indonesia sebagai salah satu produsen utama timah menyumbang kurang lebih 17,5% dari total produksi global. Hal tersebut membuktikan peran Indonesia dalam penyediaan bahan baku timah global (International Tin Association, 2024).

Berbagai dinamika global yang terjadi dalam lima tahun terakhir seperti pandemi Covid-19, konflik geopolitik, dan inflasi yang tinggi telah mempengaruhi ketidakstabilan harga komoditas termasuk timah (PT Timah, 2024). Harga timah global pada bursa London Metal Exchange (LME) mengalami kenaikan yang signifikan dan mencapai puncaknya pada kuartal pertama tahun 2022, tetapi kemudian mengalami penurunan tajam dan berfluktuasi hingga akhir 2023. Fluktuasi harga ini tidak hanya mempengaruhi ekonomi nasional, tetapi juga kinerja perusahaan yang terlibat dalam produksi dan perdagangan timah termasuk ekspor-impor (Kementerian Keuangan, 2024). Pemahaman yang mendalam tentang perilaku pasar dan prediksi harga timah global menjadi krusial dalam mempersiapkan dan mengurangi resiko bagi perusahaan (Munandar, *et al.*, 2016).

Analisis runtun waktu dapat diterapkan untuk memprediksi pergerakan harga pasar dimasa mendatang dengan didasarkan atas data historis yang tersedia. Model prediksi data runtun waktu dengan teknologi *machine learning* khususnya *deep learning* seperti *Gated Recurrent Unit* (GRU) memungkinkan untuk membangun model yang kompleks berdasarkan data historis tanpa asumsi statistik dan pembatasan tertentu (Cheng, *et al.*, 2022). GRU sebagai bagian dari *Recurrent Neural Network* (RNN) dirancang khusus untuk menangani *vanishing gradient* yang sering dialami oleh model RNN tradisional (Chung, *et al.*, 2014). Keberhasilan GRU dalam memprediksi data runtun waktu tidak terlepas dari peran metode optimasi. Metode optimasi atau *optimizer* dibutuhkan selama proses pelatihan GRU. Metode ini memiliki peran dalam memperbarui (*update*) dan mengoptimalkan parameter-parameter jaringan seperti bobot dan bias selama proses pelatihan (Kingma dan Ba, 2015). Metode optimasi RMSProp dan Adam merupakan beberapa metode optimasi yang dapat digunakan selama pelatihan GRU untuk mengoptimalkan parameter jaringan. Kedua metode ini dirancang untuk secara adaptif menyesuaikan laju pembelajaran untuk setiap parameter dengan mempertahankan rata-rata bergerak dari gradien. Metode Adam menerapkan momentum dan koreksi bias yang saling bekerjasama dalam mengatasi bias terhadap nol pada tahap awal pelatihan yang bisa terjadi dalam RMSProp (Kingma dan Ba, 2015). Perbandingan antara kedua metode optimasi tersebut masih perlu dievaluasi lebih lanjut terutama dalam hal stabilitas, sensitivitas, dan kecepatan konvergensi model khususnya dalam konteks prediksi harga timah global.

Pengaplikasian metode GRU dalam meramalkan data runtun waktu telah dilakukan di berbagai bidang. Dutta (2020) melakukan prediksi harga bitcoin menggunakan metode GRU. Analisis dilakukan dengan menggunakan data selama 10 tahun terakhir. Hasil penelitian menunjukkan bahwa metode GRU menghasilkan nilai RMSE sebesar 0,017. Shen *et al.* (2018) menggunakan GRU dengan metode optimasi RMSProp untuk memprediksi sinyal trading dari 3 saham yang berbeda yaitu HSI, DAX, dan S&P 500. Hasil penelitian menunjukkan tingkat akurasi dari ketiga dataset rata-rata sebesar 51,4%. Saputra (2023), menerapkan metode GRU untuk memprediksi harga minyak mentah dunia. Nilai MAPE yang dihasilkan dalam penelitian tersebut sebesar 2,25% sehingga dapat dikatakan bahwa model memiliki kemampuan prediksi yang sangat baik.

Penelitian ini dilakukan dengan tujuan untuk memprediksi harga timah global di LME menggunakan metode GRU. Penelitian berfokus pada pencarian model terbaik dengan mengoptimalkan *hyperparameter* dan pemilihan metode optimasi yang optimal dari dua metode optimasi yaitu RMSProp dan Adam serta mengetahui pengaruh penambahan momentum dan koreksi bias pada Adam dalam prediksi harga timah global.

2. TINJAUAN PUSTAKA

Jaringan saraf tiruan atau *Artificial Neural Network* (ANN) bekerja dengan mengolah informasi untuk mengidentifikasi pola-pola dalam data. Hal tersebut dimaksudkan untuk memahami hubungan yang kompleks antara *input* dan *output* dengan meniru kemampuan belajar dan adaptasi otak manusia. Secara matematis cara kerja jaringan syaraf tiruan digambarkan terdiri dari input $x_1, x_2, \dots, x_t \in \mathbb{R}^n$ dan bobot yang menyertainya $W_1, W_2, \dots, W_t \in \mathbb{R}^{m \times n}$, serta adanya fungsi aktivasi f (Puspitaningrum, 2006).

$$\text{Output} = f(W_t x_t + b) \quad (1)$$

Keterangan:

f : fungsi aktivasi

W : matriks bobot berukuran $m \times n$

x_t : nilai *input* pada waktu ke t berukuran $n \times 1$
 b : nilai *bias* berukuran $m \times 1$

Fungsi aktivasi berperan dalam menentukan apakah informasi dari unit *input* jaringan akan diteruskan ke unit jaringan lain atau tidak agar dapat berfungsi dengan efektif (Googfellow, *et al.*, 2016). Fungsi aktivasi linear merupakan fungsi yang mengembalikan nilai yang sama dengan input yang diberikan. Secara matematis, fungsi linear dituliskan sebagai berikut

$$f(x) = x \quad (2)$$

Fungsi aktivasi sigmoid merupakan fungsi nonlinear yang menghasilkan nilai pada skala 0 hingga 1. Fungsi *sigmoid* didefinisikan dengan persamaan

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Fungsi aktivasi tangen hiperbolik merupakan fungsi aktivasi nonlinear yang berpusat pada 0 dan menghasilkan nilai dengan rentang -1 hingga 1. Fungsi *tanh* memiliki persamaan

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

Recurrent Neural Network (RNN) merupakan salah satu jenis arsitektur *Artificial Neural Network* (ANN) yang dirancang khusus untuk menangani data berurutan (*sequential data*) (Googfellow, *et al.*, 2016). RNN tidak hanya menggunakan *input* saat ini (x_t) tetapi juga *input* sebelumnya (h_{t-1}), sehingga antar *input* saling berhubungan dan dapat memberikan informasi ke *hidden layer* (Schmidt, 2019). Secara matematis, sel unit RNN dapat ditulis dalam persamaan berikut:

$$h_t = f(Wx_t + Uh_{t-1} + b_h) \quad (5)$$

$$\hat{y}_t = f(Vh_t + b) \quad (6)$$

GRU pertama kali diperkenalkan pada tahun 2014 oleh Kyunghyun Cho dkk dengan tujuan untuk menangani *vanishing gradient* yang sering dialami oleh RNN (Chung, *et al.*, 2014). GRU memiliki 2 gerbang/*gate* yang digunakan untuk mengontrol alur informasi, yaitu *update gate* dan *reset gate*. *Update gate* merupakan gerbang yang membantu model dalam menentukan banyaknya informasi dari masa lalu (*timesteps* sebelumnya) disimpan dan digunakan untuk menghasilkan *output* pada *timesteps* selanjutnya (Cho, *et al.*, 2014). *Update gate* diformulasikan dalam persamaan sebagai berikut:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (7)$$

Reset gate merupakan gerbang yang digunakan untuk menentukan seberapa banyak informasi dari *timesteps* sebelumnya yang harus dihapus atau dilupakan (Cho, *et al.*, 2014). *Reset gate* diformulasikan dalam persamaan sebagai berikut:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (8)$$

Kandidat *hidden state* digunakan untuk menyimpan informasi yang relevan dari unit sebelumnya (Chung, *et al.*, 2014). Kandidat *hidden state* diformulasikan dalam persamaan sebagai berikut:

$$\tilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \quad (9)$$

Hidden state berfungsi untuk menyimpan informasi dari unit saat ini dan akan diteruskan ke jaringan pada *timesteps* selanjutnya (Chung, *et al.*, 2014).

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \quad (10)$$

Nilai *output* \hat{y}_t atau nilai prediksi dari model GRU dihitung dengan menggunakan nilai h_t yang didapat sebelumnya. Nilai \hat{y}_t yang didapat kemudian digunakan untuk menghitung nilai *loss*. *Loss function* yang digunakan adalah *Mean Square Error* (MSE).

$$L = MSE = \frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2 \quad (11)$$

Proses jaringan propagasi mundur (*Backpropagation Throught Time/ BPTT*) pada GRU dilakukan dengan menghitung gradien *loss function* terhadap setiap parameter pada model GRU, yang meliputi bobot pada *update gate*, bobot *reset gate*, bobot *hidden state* dan bias. Gradien yang diperoleh kemudian akan digunakan untuk memperbarui setiap parameter melalui metode optimasi. Proses BPTT dilakukan dengan menggunakan metode aturan rantai (*chain rule*) (Li, 2016).

Metode optimasi atau *optimizer* berperan dalam menyesuaikan parameter model selama proses pelatihan yang didasarkan pada nilai *loss function*. Penyesuaian parameter ini bertujuan untuk memperoleh model dalam bentuk yang paling akurat. *Root Mean Square Propagation* (RMSProp) merupakan salah satu metode optimasi yang bekerja dengan menghitung rata-rata bergerak dari gradien kuadrat untuk setiap parameter jaringan. Hal ini dilakukan untuk menyesuaikan laju pembelajaran (*learning rate*) agar lebih adaptif terhadap perubahan dalam gradien. Nilai γ yang disarankan adalah 0.9 (Tieleman dan Hinton, 2012).

$$\mathbf{v}_t = \gamma \mathbf{v}_{t-1} + (1 - \gamma)(\mathbf{g}_t)^2 \quad (12)$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \frac{\eta}{\sqrt{\mathbf{v}_t + \epsilon}} \mathbf{g}_t \quad (13)$$

Adaptive Moment Estimation (Adam) bekerja dengan mengadaptasi *learning rate* menggunakan estimasi momen pertama dan momen kedua dari gradien untuk setiap bobot jaringan syaraf (Kingma dan Ba, 2015). Metode optimasi Adam didesain dengan menggabungkan momentum dan RMSProp. Berikut merupakan persamaan matematis dari momen pertama dan kedua dalam metode optimasi Adam.

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t \quad (14)$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2 \quad (15)$$

Adam juga menerapkan koreksi bias dengan tujuan untuk menghindari bias terhadap 0 selama awal iterasi (Kingma dan Ba, 2015).

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t} \quad (16)$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t} \quad (17)$$

Sehingga, perhitungan *update* parameter baru (bobot dan bias) adalah sebagai berikut:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t + \epsilon}} \quad (18)$$

Tahap perhitungan GRU dimulai dengan tahap data *pre-processing*. Tahap ini terdiri atas 2 langkah yaitu normalisasi data dan *splitting* data. Normalisasi data dilakukan dengan menggunakan *Min Max Scaller* dengan persamaan (16).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (19)$$

Langkah lainnya dalam tahap data *preprocessing* adalah *data splitting*. *Data splitting* dilakukan untuk membagi data menjadi data *training* dan data *testing*. Tahap selanjutnya yaitu dengan menginisialisasi *hyperparameter* yang akan digunakan dalam membangun model GRU. Proses *tuning hyperparameter* atau penyesuaian *hyperparameter* dilakukan untuk menemukan kombinasi *hyperparameter* yang menghasilkan model dengan kinerja terbaik dan paling optimal. *Hyperparameter* ini antara lain *timesteps*, *epochs*, *batch size*, *hidden unit (neuron)*, *optimizer*, *learning rate* dan *dropout*.

a. *Learning Rate*, berperan dalam ketelitian dan efektifitas suatu jaringan (Googfellow, et al., 2016).

- b. *Hidden unit*, merupakan *hyperparameter* yang menentukan seberapa banyak unit atau sel yang terdapat dalam jaringan GRU. *Hidden unit* menyimpan dan memproses informasi selama pemberian input yang berurutan (Garcia, 2021).
- c. *Epoch*, merupakan banyaknya jumlah iterasi yang terjadi. Satu *epoch* artinya satu kali iterasi melalui seluruh dataset pelatihan oleh suatu model (Sharma, 2017).
- d. *Batch size*, merupakan banyaknya sampel data yang diproses dalam satu iterasi (epoch) pelatihan (Sharma, 2017).
- e. *Dropout* merupakan teknik regularisasi yang dikembangkan oleh Geoffrey Hinton dengan tujuan untuk mengurangi *overfitting*. Dropout bekerja dengan mengabaikan beberapa neuron secara acak di setiap *hidden layer* selama proses pelatihan (Srivastava, *et.al*, 2014).

Proses pelatihan dan prediksi data dilakukan berdasarkan kombinasi *hyperparameter* yang telah dinisialisasi sebelumnya. Data yang digunakan dalam penelitian merupakan data yang sebelumnya sudah melalui tahap normalisasi, sehingga hasil prediksi bukan berada pada skala aslinya. Data tersebut perlu untuk dikembalikan ke dalam bentuk semula (denormalisasi data) dengan persamaan (17)

$$x = x'(\max(x) - \min(x)) + \min(x) \quad (20)$$

Kriteria evaluasi yang digunakan dalam penelitian ini yaitu *Mean Absolute Percentage Error* (MAPE). MAPE adalah ukuran persentase kesalahan rata-rata secara mutlak. Semakin kecil persentase kesalahan (*percentage error*) pada MAPE, semakin akurat hasil prediksi yang dihasilkan (Klimberg, *et al.*, 2010).

$$MAPE = \left(\frac{100\%}{n} \right) \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \quad (21)$$

3. METODE PENELITIAN

Data yang digunakan dalam penelitian ini adalah data historis harga penutupan harian timah global di bursa London Metal Exchange (LME) dari 1 Januari 2019 hingga 31 Desember 2023. Jumlah total data sebanyak 1262 data dengan variabel penelitian yang digunakan dalam prediksi harga harian timah global adalah sebagai berikut:

- a. Harga Timah, merepresentasikan harga penutupan (*closing price*) harian timah di bursa London Metal Exchange (LME).
- b. Tanggal, merepresentasikan tanggal spesifik di mana harga timah dicatat. Harga timah LME yang tercatat merupakan harga pada hari kerja pada periode 1 Januari 2019 hingga 31 Desember 2023.

Microsoft Excel dan bahasa pemrograman *python* digunakan dalam proses pengolahan data yang diakses melalui *platform cloud computing Google Collab*. Berikut merupakan tahapan analisis yang dilakukan:

1. Melakukan input data yang digunakan dalam penelitian.
2. Melakukan tahap analisis data eksplorasi.
3. Melakukan tahap *preprocessing* data berupa normalisasi data menggunakan persamaan (19) dan *splitting* data untuk data *training*, data *validation* dan data *testing*.
4. Menentukan metode optimasi yang akan digunakan.
5. Menentukan *hyperparameter* untuk pemodelan GRU berupa *learning rate*, *hidden unit*, *batch size*, *epoch* dan *dropout rate*.
6. Melakukan pemodelan pada data training dengan metode GRU yang terdiri atas tahap berikut:
 - a. Menghitung *update gate* menggunakan persamaan (7).
 - b. Menghitung *reset gate* menggunakan persamaan (8).

- c. Menghitung kandidat *hidden state* menggunakan persamaan (9).
- d. Menghitung *hidden state* menggunakan persamaan (10)
- e. Menghitung nilai output menggunakan persamaan (6)
- f. Melakukan *backpropagation through time* (BPTT) dengan diawali menghitung nilai *loss* menggunakan persamaan (11)
- g. *Update* parameter jaringan dengan metode RMSProp atau Adam. Perhitungan parameter baru pada RMSProp melibatkan perhitungan dengan persamaan (12) dan (13). Parameter baru pada metode optimasi Adam dihitung dengan melibatkan persamaan (14), (15), (16), (17) dan (18).
7. Melakukan validasi model dengan data *validation*.
8. Melakukan evaluasi kinerja model menggunakan data testing dengan kriteria evaluasi MAPE menggunakan persamaan (21).
9. Melakukan prediksi dan denormalisasi data hasil prediksi menggunakan persamaan (20).

4. HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini adalah data harga penutupan timah global harian dari bursa London Metal Exchange (LME) selama periode 1 Januari 2019 hingga 31 Desember 2023. Total terdapat 1262 data yang mewakili harga penutupan timah selama periode lima tahun. Harga penutupan timah harian pada periode tersebut memiliki rata-rata sebesar USD 24729,72 per metrik ton dengan standar deviasi yang tercatat sebesar USD 7823,73 per metrik ton. Harga timah terendah tercatat pada 15 Januari 2020 yaitu sebesar USD 13.205 per metrik ton dan harga tertinggi mencapai USD 48.650 per metrik ton pada 8 Maret 2022. Prediksi harga timah dengan metode *gated recurrent unit* dilakukan dalam tahapan sebagai berikut.

Tahap *data preprocessing* terdiri atas 2 langkah yaitu normalisasi data dan *data splitting*. Normalisasi dilakukan dengan tujuan untuk membuat data berada pada rentang yang seragam. Metode *Min Max Scaler* digunakan dalam proses normalisasi data pada penelitian ini.

Tabel 1. Data Hasil Normalisasi

Tanggal	Close	Close Scaled
02/01/2019	19482,50	0,17606
03/01/2019	19535,00	0,17754
04/01/2019	19585,00	0,17895
...
27/12/2023	25644,00	0,35093
28/12/2023	25680,00	0,35195
29/12/2023	25415,00	0,34447

Tahap *data splitting* dilakukan untuk membagi data menjadi data *training* (data pelatihan), data *validation* (data validasi) dan data *testing* (data pengujian). Proporsi untuk masing-masing data *training* dan data *testing* pada penelitian ini yaitu 90:10, sedangkan data *validation* diperoleh dengan membagi kembali data *training* dengan proporsi 90:10, sehingga jumlah data *training* menjadi 1021 data, data *validation* 114 data dan data *testing* 127 data. Struktur dan dimensi dari data masukan berupa (*sample, timesteps, feature*) sedangkan dimensi dari data target yaitu (*samples, feature*). Prediksi data target pada

penelitian ini dilakukan dengan mempertimbangkan data harga harian timah selama 20 hari kerja sebelumnya, sehingga dimensi data seperti ditunjukkan dalam Tabel 2.

Tabel 2. Dimensi data *input* dan data target

Pembagian Data	Dimensi
<i>X training</i>	(1115, 20, 1)
<i>y training</i>	(1115,1)
<i>X validation</i>	(94, 20, 1)
<i>y validation</i>	(94,1)
<i>X testing</i>	(107, 20, 1)
<i>y testing</i>	(107, 1)

Model arsitektur GRU dibangun dengan 3 lapisan (*layer*) utama yaitu lapisan *input*, lapisan tersembunyi (*hidden layer*) dan lapisan *output*. Proses *tuning hyperparameter* atau penyesuaian *hyperparameter* dilakukan untuk menemukan kombinasi *hyperparameter* yang menghasilkan model dengan kinerja paling optimal. Tabel 3 merupakan kombinasi *hyperparameter* yang digunakan dalam *tuning hyperparameter* pada penelitian.

Tabel 3. Kombinasi *Hyperparameter*

Kombinasi <i>Hyperparameter</i>	
<i>Timestep</i>	[20]
<i>Epoch</i>	[50, 100]
<i>Batch Size</i>	[32, 64, 128]
<i>Learning Rate</i>	[0,01, 0,001, 0,0001]
<i>Hidden Unit</i>	[16, 32, 64]

Teknik regularisasi *dropout* juga diterapkan dalam model guna menghindari *overfitting*. *Dropout rate* yang digunakan dalam penelitian ini sebesar 0,3 artinya 30% dari unit dalam lapisan jaringan akan dinonaktifkan (*dropout*) secara acak selama setiap iterasi pelatihan.

Proses pemodelan GRU dilakukan dengan menggunakan konfigurasi *hyperparameter* yang telah ditetapkan sebelumnya. Percobaan dilakukan dengan 54 kombinasi *hyperparameter* untuk setiap metode optimasi. Setiap proses *training* maupun proses validasi didapatkan nilai *loss* yang menunjukkan seberapa jauh perbedaan nilai prediksi model dari nilai sebenarnya.

Tabel 4. Nilai *Training Loss* dan *Validation Loss*

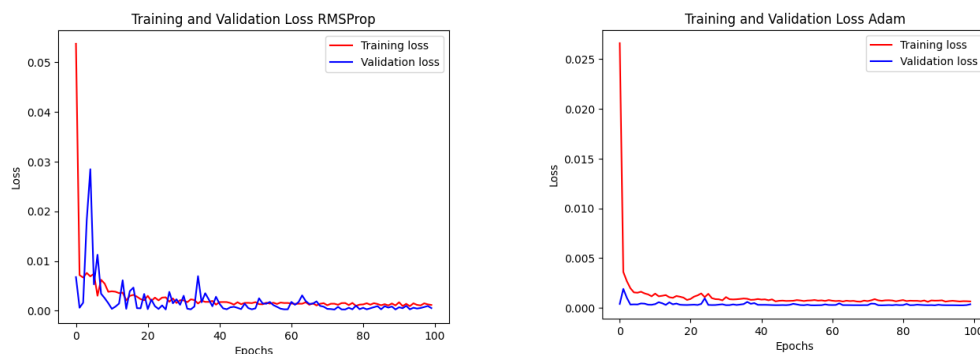
Hyperparameter			RMSProp		Adam		
Batch Size	Epochs	Learning Rate	Units	Epoch Stopped	Val Loss	Epoch Stopped	Val Loss
32	50	0,01	16	3	0,00053	7	0,00032
			32	4	0,00038	6	0,00033
			64	4	0,00040	7	0,00030
		0,001	16	8	0,00064	5	0,00064
			32	23	0,00051	4	0,00062
			64	4	0,00076	12	0,00045
			0,0001	16	12	0,00054	24

64	100	0,01	32	11	0,00058	10	0,00065
			64	6	0,00070	6	0,00083
			16	6	0,00090	8	0,00030
			32	13	0,00032	7	0,00031
			64	8	0,00029	5	0,00034
			16	12	0,00045	6	0,00047
		0,001	32	3	0,00060	4	0,00055
			64	5	0,00077	3	0,00068
			16	19	0,00052	24	0,00054
			32	13	0,00059	11	0,00069
		0,0001	64	6	0,00077	6	0,00081
			16	5	0,00039	5	0,00036
64	50	0,01	32	3	0,00044	9	0,00030
			64	3	0,00133	7	0,00033
			16	10	0,00066	8	0,00063
		0,001	32	10	0,00069	7	0,00066
			64	11	0,00072	22	0,00053
			16	31	0,00092	43	0,00076
		0,0001	32	22	0,00049	17	0,00053
			64	9	0,00063	10	0,00066
			16	3	0,00130	8	0,00029
	100	0,01	32	4	0,00052	13	0,00030
			64	4	0,00447	9	0,00029
			16	4	0,00055	7	0,00048
		0,001	32	13	0,00061	6	0,00060
			64	3	0,00071	14	0,00054
			16	55	0,00060	58	0,00068
		0,0001	32	14	0,00054	13	0,00056
			64	13	0,00069	11	0,00078
128	50	0,01	16	6	0,00054	11	0,00033
			32	3	0,00049	17	0,00031
			64	6	0,00046	5	0,00034
		0,001	16	9	0,00068	13	0,00067
			32	8	0,00070	13	0,00065
			64	3	0,00081	9	0,00065
		0,0001	16	Tidak Konvergen	0,00099	Tidak konvergen	0,00156
			32	44	0,00062	34	0,00073
			64	22	0,00066	20	0,00072
	100	0,01	16	7	0,00053	20	0,00031
			32	5	0,00062	4	0,00044
			64	3	0,00065	5	0,00042
		0,001	16	9	0,00082	12	0,00073
			32	6	0,00072	5	0,00085
			64	4	0,00072	7	0,00057
		0,0001	16	Tidak Konvergen	0,00111	Tidak konvergen	0,00120
			32	35	0,00057	33	0,00063
			64	22	0,00067	19	0,00075

Model dengan metode optimasi RMSProp memiliki nilai *validation loss* terendah sebesar 0,00029 dengan kombinasi *batch size* 32, *epoch* 100, *learning rate* 0,01, dan jumlah *hidden units* 64. Nilai *validation loss* terendah pada model dengan metode optimasi Adam diperoleh sebesar 0,00029 dengan konfigurasi *hyperparameter batch size* 64, *epoch* 100, *learning rate* 0,01, dan jumlah *hidden unit* 64. Model dengan metode optimasi Adam cenderung lebih sensitif terhadap perubahan *learning rate*. Nilai *learning rate* yang kecil menyebabkan nilai *loss* pada model semakin tinggi.

Berdasarkan Tabel 4, model dengan optimasi RMSProp dan Adam paling cepat mencapai konvergensi pada iterasi ke 3 dan terdapat dua model baik pada metode optimasi RMSProp maupun Adam yang tidak mencapai konvergensi dalam jumlah *epoch* yang telah ditentukan. Berdasarkan hasil tersebut diketahui bahwa mayoritas model RMSProp lebih cepat mencapai konvergensi dibanding Adam. Namun, pada kondisi ketika nilai *learning rate* sangat kecil (0,0001), baik model RMSProp maupun Adam membutuhkan waktu lebih lama untuk mencapai konvergensi.

Analisis grafik fungsi *loss* juga dapat digunakan dalam mengamati kecepatan konvergensi dan stabilitas model dengan memantau perubahan nilai *error* selama *epochs* berlangsung. Penurunan yang konsisten dalam nilai *loss* menunjukkan bahwa model sedang belajar dengan baik, sedangkan fluktuasi yang besar bisa mengindikasikan masalah seperti ketidakstabilan model.



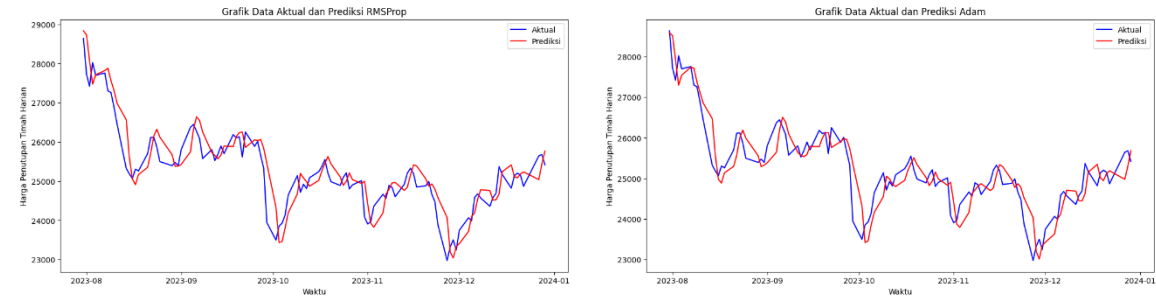
Gambar 1. Grafik Fungsi Loss

Gambar 1 merupakan grafik *loss* dari kedua metode optimasi dengan mengambil *hyperparameter* yang sama yaitu *batch size* 64, *epoch* 100, *learning rate* 0,01, dan *hidden unit* 64. Berdasarkan grafik *loss* diatas, dapat diketahui bahwa model dengan metode optimasi RMSProp memperlihatkan kecepatan konvergensi yang sangat cepat pada awal *training*, sementara grafik *validation loss* menunjukkan fluktuasi khususnya pada awal *epoch* sebelum akhirnya mulai stabil. Grafik *loss* pada model dengan metode optimasi Adam juga menunjukkan kecepatan konvergensi yang tinggi pada awal pelatihan. *Training loss* menurun secara signifikan dalam beberapa epoch pertama dan mencapai stabilitas dengan cepat. *Validation loss* pada model Adam juga cepat mencapai nilai minimum dan menunjukkan stabilitas yang lebih baik jika dibandingkan dengan model RMSProp.

Penambahan momentum dan koreksi bias pada metode optimasi Adam menunjukkan pengaruhnya dalam membantu dalam mencapai stabilitas yang lebih baik pada data *training* maupun data *testing*. Koreksi bias membantu dalam menstabilkan model pada awal iterasi

sedangkan momentum berperan dalam mengurangi fluktuasi model. Hal tersebut ditunjukkan dari grafik dimana Adam menurunkan nilai *loss* dan mencapai stabilitas lebih cepat pada awal iterasi, dibanding pada grafik RMSProp yang masih berfluktuasi bahkan hingga pertengahan iterasi.

Model terbaik yang telah dipilih pada masing-masing metode optimasi berdasarkan nilai *validation loss* terkecil kemudian dievaluasi. Model dengan optimasi RMSProp diperoleh model terbaik pada *batch size* 32, *epoch* 100, *learning rate* 0,01, dan jumlah *hidden unit* 64 sedangkan pada model dengan optimasi Adam diperoleh model terbaik pada *batch size* 64, *epoch* 100, *learning rate* 0,01, dan jumlah *hidden unit* 64. Hasil prediksi data *testing* dari kedua model terbaik kemudian divisualisasikan dalam plot untuk mempermudah analisis perbandingan hasil prediksi. Gambar 2 menunjukkan plot perbandingan antara data aktual dengan data prediksi dari hasil model terbaik dengan optimasi RMSProp dan Adam.



Gambar 2. Plot Data Aktual dan Data Prediksi

Evaluasi model dilakukan dengan menggunakan metrik evaluasi MAPE, diperoleh nilai MAPE untuk prediksi data *testing* pada model terbaik dengan optimasi RMSProp mencapai 1,34%, sedangkan untuk model terbaik dengan optimasi Adam menghasilkan nilai MAPE sebesar 1,29%. Nilai MAPE tersebut menunjukkan bahwa metode Adam masih lebih unggul dibanding metode RMSProp untuk digunakan dalam prediksi harga timah global.

Model terbaik dari metode optimasi Adam kemudian digunakan untuk melakukan prediksi pada periode yang akan datang. Hasil dari prediksi harga penutupan timah harian untuk 20 hari kedepan ditampilkan dalam Tabel 5.

Tabel 5. Hasil Prediksi 20 Hari Kedepan

Dates	Prediksi	Dates	Prediksi
02/01/2024	25591,90	16/01/2024	25262,27
03/01/2024	25502,64	17/01/2024	25238,67
04/01/2024	25473,15	18/01/2024	25215,65
05/01/2024	25444,40	19/01/2024	25193,19
08/01/2024	25416,53	22/01/2024	25171,30
09/01/2024	25389,30	23/01/2024	25149,95
10/01/2024	25362,69	24/01/2024	25129,12
11/01/2024	25336,67	25/01/2024	25108,82
12/01/2024	25311,26	26/01/2024	25089,02
15/01/2024	25286,46	29/01/2024	25069,71

5. KESIMPULAN

Analisis data dengan menggunakan metode GRU dengan metode optimasi RMSProp dan Adam dilakukan dengan proporsi data *training* sebesar 90% dan data *testing* sebesar 10%. Sebanyak 10% dari data *training* digunakan sebagai data *validation*. Pada proses pelatihan GRU, digunakan berbagai konfigurasi *hyperparameter* yaitu *batch size* (32, 64, 128), *epoch* (50, 100), *learning rate* (0,01, 0,001, 0,0001), dan *hidden units* (16, 32, 64). Jumlah *timesteps* yang digunakan yaitu 20.

Model terbaik berdasarkan nilai *validation loss* terendah pada RMSProp diperoleh pada model dengan konfigurasi *hyperparameter batch size* 32, *epoch* 100, *learning rate* 0,01, dan jumlah *hidden unit* 64 yang menghasilkan nilai minimum *validation loss* sebesar 0,00029 dan nilai MAPE sebesar 1,34%. Model dengan optimasi Adam diperoleh model terbaik pada *batch size* 64, *epoch* 100, *learning rate* 0,01, dan jumlah *hidden unit* 64 dengan nilai *validation loss* sebesar 0,00029 dan nilai MAPE sebesar 1,29%. Metode optimasi Adam cenderung lebih sensitif terhadap perubahan *learning rate* dan membutuhkan waktu lebih lama dalam mencapai konvergensi jika dibandingkan dengan metode optimasi RMSProp.

Berdasarkan analisis grafik *loss function* pada model terbaik, model dengan metode optimasi Adam cenderung lebih stabil terutama pada awal iterasi. Ini dipengaruhi oleh penggunaan momentum dan bias koreksi pada yang secara bersama-sama menstabilkan proses pelatihan terutama pada awal iterasi dan mengurangi fluktuasi.

DAFTAR PUSTAKA

- Cheng, C.-H., Tsai, M.-C., & Chang, C. (2022). A TimeSeries Model Based on Deep Learning and Integrated Indicator Selection Method for Forecasting Stock Prices and Evaluating Trading Profits. *Journal of MDPI*. doi:<https://doi.org/10.3390/systems10060243>
- Cho, K., Merriënboer, B. v., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (hal. 1724-1734). Doha. doi:<https://doi.org/10.3115/v1/D14-1179>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS 2014 Workshop on Deep Learning*.
- Dutta, A. (2020). A Gated Recurrent Unit Approach to Bitcoin Price Prediction. *Journal of Risk and Financial Management*.
- Garcia, R., Munz, T., & Weiskopf, D. (2021). Visual analytics tool for the interpretation of hidden states in recurrent neural networks. *Journal of Visual Computing for Industry, Biomedicine, and Art*.
- Googfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press.

- International Tin Association. (2024). Global tin production sees modest decline in 2023. Retrieved from *internationaltin.org*: <https://www.internationaltin.org/global-tin-production-sees-modest-decline-in-2023/>
- Kementerian Keuangan. (2024). Konferensi Pers Kinerja Fiskal Dan Ekonomi Regional Babel Tahun 2023. Retrieved from *DJPb Kemenkeu*: <https://djp.b.kemenkeu.go.id/kanwil/babel/id/data-publikasi/berita-terbaru/3007-konferensi-pers-kinerja-fiskal-dan-ekonomi-regional-babel-tahun-2023.html>
- Kingma, D. P., & Ba, J. L. (2015). ADAM: A Method for Stochastic Optimization . *Journal of ICLR*.
- Klimberg, R. K., Boyle, K. J., & Sillup, G. P. (2010). *Forecasting performance measures - What are their practical meaning*. Philadelphia: Emerald Group Publishing Limited.
- Li, M. (2016). *A Tutorial On Backward Propagation Through Time (BPTT) In The Gated Recurrent Unit (GRU) RNN*. Carnegie Mellon University. doi:10.13140/RG.2.2.32858.98247
- Munandar, A. I., Siregar, H., Andati, T., & Anggraeni, L. (2016). Volatilitas Harga Timah. *Jurnal Ilmiah Manajemen*, 171-180.
- PT Timah. (2024). *Laporan Tahunan 2023: Memastikan Strategi untuk Mengoptimalkan Kinerja*. Pangkajene: PT Timah Tbk.
- Puspitaningrum, D. (2006). *Pengantar Jaringan Saraf Tiruan*. Yogyakarta: Penerbit Andi.
- Saputra, N. W. (2023). Penerapan Deep Learning Menggunakan Gated Recurrent Unit Untuk Memprediksi Harga Minyak Mentah Dunia. *Jurnal BITS*, 86-94.
- Schmidt, R. M. (2019). Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. Retrieved from *ArXiv*: <https://arxiv.org/abs/1912.05911>
- Sharma, S. (2017). Epoch vs Batch Size vs Iterations. Retrieved from *Towards Data Science*: <https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9>
- Shen, G., Tan, Q., Zhang, H., Zeng, P., & Jianjun, X. (2018). Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Prediction. *Procedia Computer Science*, 895-903.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 1929-1958.
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5 rmsprop: Divide the gradient by a running average of its recent magnitude. Retrieved from *COURSERA: Neural Networks for Machine Learning*.