

# Optimasi *Hyperparameter* Model GRU untuk Prediksi Harga Saham ANTAM

Subairi<sup>1</sup>, Anggraini Puspita Sari<sup>2,\*</sup>, Eka Prakarsa Mandyartha<sup>3</sup>

Program Studi Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia

<sup>1</sup>21081010019@student.upnjatim.ac.id; <sup>2</sup>anggraini.puspita.if@upnjatim.ac.id; <sup>3</sup>eka\_prakarsa.fik@upnjatim.ac.id

\*penulis korespondensi

## INFO ARTIKEL

### Sejarah Artikel

Diterima: 11 November 2025  
Direvisi: 9 Desember 2025  
Diterbitkan: 31 Desember 2025

### Kata Kunci

*Gated Recurrent Unit*  
*Grid search*  
Optimasi hyperparameter  
Prediksi Harga Saham

## ABSTRAK

Prediksi harga saham berperan penting meminimalisir kerugian akibat fluktuasi harga saham. Penelitian ini bertujuan untuk meningkatkan akurasi prediksi harga saham PT Aneka Tambang Tbk (ANTM) menggunakan model *Gated Recurrent Unit* (GRU) dengan optimasi *hyperparameter* melalui metode *Grid Search*. Model GRU dipilih karena mampu mengatasi permasalahan *vanishing gradient* dan efektif dalam mempelajari pola ketergantungan jangka panjang pada data deret waktu walaupun dengan arsitektur yang sederhana. Sementara itu, *Grid Search* digunakan karena memiliki keunggulan dalam menjelajahi ruang *hyperparameter* secara menyeluruh, sehingga setiap kombinasi parameter dapat diuji dan memungkinkan diperolehnya konfigurasi terbaik. Proses *Grid Search* dilakukan dengan ruang pencarian *hyperparameter* yang mencakup jumlah *units*, jumlah *epoch*, ukuran *batch*, serta variasi *optimizer*. Keunggulan utama penelitian ini terletak pada penerapan optimasi *hyperparameter* yang mampu meningkatkan efektivitas model GRU dalam menemukan konfigurasi terbaik, sehingga menghasilkan prediksi harga saham yang lebih akurat dan stabil. Evaluasi kinerja model menggunakan metrik RMSE, MAE, MAPE. Hasil penelitian menunjukkan bahwa model GRU dengan optimasi *Grid Search* menggunakan *optimizer* Adam memberikan performa yang optimal dengan nilai evaluasi RMSE sebesar 67.8805, MAE sebesar 45.6501, dan MAPE sebesar 2.2309%. Temuan ini membuktikan bahwa optimasi *hyperparameter* melalui *Grid Search* mampu meningkatkan akurasi prediksi model GRU pada data harga saham.

## PENDAHULUAN

Pasar saham menjadi salah satu instrumen investasi yang paling populer dan diminati, sebab menawarkan potensi keuntungan yang relatif tinggi dibandingkan dengan instrumen keuangan lainnya seperti deposito, obligasi, maupun reksa dana[1]. Namun, tingginya potensi keuntungan tersebut diiringi dengan tingkat risiko yang besar akibat fluktuasi harga saham yang sangat dinamis serta dipengaruhi oleh berbagai faktor ekonomi, politik, maupun psikologis[2][3]. Kompleksitas pergerakan harga saham ini menuntut adanya pendekatan analitis yang mampu mengenali pola data historis dan memprediksi pergerakan harga secara akurat. Pemanfaatan teknologi berbasis kecerdasan buatan (*artificial intelligence*), khususnya *deep learning*, menjadi alternatif yang efektif untuk menganalisis pola non-linear dan hubungan kompleks antar variabel pada data saham[4]. Salah satu konsep arsitektur *deep learning* yang terbukti unggul dalam pengolahan data berurutan (*time series*) adalah *Recurrent Neural Network* (RNN) beserta turunannya, yaitu *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU)[5]. Model GRU memiliki desain struktur yang lebih sederhana dari LSTM namun tetap efisien dalam mengatasi masalah *vanishing*

*gradient* serta mampu mempertahankan informasi jangka panjang melalui mekanisme *update gate* dan *reset gate*[6]. beberapa penelitian telah membuktikan bahwa GRU dapat memberikan akurasi prediksi yang lebih baik daripada algoritma regresi klasik seperti *linear regression* [7], dan juga terbukti dapat melampaui algoritma RNN, CNN, & LSTM dalam beberapa kasus[8]. Keunggulan inilah yang menjadikan GRU sangat sesuai untuk menganalisis data harga saham yang bersifat fluktuatif dan berurutan dari waktu ke waktu.

Kualitas prediksi GRU sangat dipengaruhi oleh pemilihan *hyperparameter* yang tepat. Berbagai penelitian terdahulu menunjukkan bahwa model berbasis RNN, LSTM, dan GRU memiliki kemampuan kuat dalam mempelajari pola non-linear pada data saham namun sebagian besar studi tersebut menggunakan konfigurasi *hyperparameter* default atau hanya menguji beberapa skenario terbatas sehingga performa model belum optimal. Penelitian Sari [9][10] dan Haromainy [3] menegaskan bahwa akurasi model berbasis RNN sangat sensitif terhadap pengaturan *hyperparameter*, sementara tinjauan sistematis oleh Lin dan Marques [11] menekankan bahwa tidak ada satu set *hyperparameter* yang bersifat universal untuk semua kondisi pasar saham maupun periode waktu tertentu. Maka, *hyperparameter* perlu disesuaikan, diverifikasi, diperbarui, dan di-*fine-tune* secara berkala guna memastikan ketahanan (*robustness*) dan akurasi model tetap terjaga. Oleh karena itu, penerapan teknik optimasi *hyperparameter* menjadi krusial untuk memperoleh konfigurasi yang paling efektif. Metode optimasi *hyperparameter* terbukti mampu meningkatkan kinerja model, seperti pada penelitian oleh Hoque [12] yang menunjukkan bahwa optimasi *hyperparameter* dapat meningkatkan akurasi prediksi. Salah satu metode optimasi yang umum untuk diterapkan adalah *Grid Search*, yaitu teknik pencarian kombinasi *hyperparameter* secara sistematis. Beberapa penelitian membuktikan bahwa penerapan *Grid Search* pada beberapa model prediksi harga saham [12], dan juga spesifik pada model berbasis RNN berhasil meningkatkan akurasi secara signifikan. Hal ini menjadi dasar pemilihan *Grid Search* dalam optimasi model GRU pada penelitian ini.

Penelitian ini berfokus pada pengembangan model prediksi harga saham PT Aneka Tambang Tbk (ANTM) berbasis GRU dengan optimasi *hyperparameter* menggunakan *Grid Search*. Saham ANTM dipilih karena memiliki tingkat volatilitas yang tinggi, dipengaruhi oleh dinamika harga komoditas global, serta termasuk saham dengan volume perdagangan besar di BEI. Karakteristik ini menjadikan ANTM kasus uji yang relevan untuk mengevaluasi kemampuan GRU dalam menangkap pola pergerakan harga yang kompleks dan non-linear. Proses optimasi dilakukan untuk memperoleh kombinasi parameter terbaik, seperti jumlah *neuron*, *batch size*, *learning rate*, dan jumlah *epoch*, yang berdampak pada peningkatan akurasi dan kestabilan prediksi. Selain itu, penelitian ini diharapkan memberikan manfaat dalam meningkatkan efisiensi proses analisis pasar saham, khususnya pada saham ANTM yang dikenal memiliki volatilitas tinggi dan menjadi salah satu saham unggulan di sektor pertambangan. Hasil penelitian ini diharapkan dapat menjadi referensi bagi investor maupun analis dalam merumuskan keputusan investasi yang lebih rasional dan berbasis data, serta memperkaya kajian penerapan *deep learning* dalam bidang keuangan, khususnya dalam prediksi harga saham di pasar modal Indonesia.

## METODE

Penelitian ini dilakukan dengan serangkaian proses seperti pada Gambar 1. Proses dimulai dari akuisisi data, *preprocessing* data, pembuatan arsitektur model GRU-*Grid search*, pelatihan model dan terakhir evaluasi model.



Gambar 1. Alur Metode Penelitian

### Akuisisi Data

Data yang digunakan dalam penelitian ini diperoleh dengan mengunduh dari situs Investing.com, dengan objek penelitian berupa data historis harian harga saham PT Aneka Tambang Tbk (ANTM) periode 26 oktober 2018 hingga 26 oktober 2025, terhitung sebanyak 1690 data. Dataset mencakup atribut *Date*, *Price*, *Open*, *High*, *Low*, *Volume*, dan *Change* yang merepresentasikan pergerakan harga saham harian. Fitur *Price* atau harga penutupan digunakan sebagai fitur target karena mencerminkan hasil akhir dari aktivitas perdagangan saham pada suatu hari, dianggap paling mewakili kondisi pasar yang sebenarnya, serta memberikan gambaran yang lebih stabil pada pergerakan saham dari pada fitur yang lain[11]. Gambaran umum dari data yang digunakan dalam penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Dataset

<i>Date</i>	<i>Price</i>	<i>Open</i>	<i>High</i>	<i>Low</i>	<i>Vol.</i>	<i>Change %</i>
10/26/2018	725	725	730	720	35.08M	0.69%
10/29/2018	705	725	730	705	30.06M	-2.76%
10/30/2018	675	700	705	670	99.35M	-4.26%
10/31/2018	680	680	685	665	50.38M	0.74%
11/01/2018	685	680	705	680	92.37M	0.74%
...	...	...	...	...	...	...
10/20/2025	3270	3450	3450	3270	160.78M	-5.22%
10/21/2025	3290	3340	3350	3270	96.57M	0.61%
10/22/2025	3170	3150	3230	3120	114.87M	-3.65%
10/23/2025	3190	3180	3210	3170	66.75M	0.63%
10/24/2025	3130	3190	3210	3120	84.45M	-1.88%

### Preprocessing data

Tahap *preprocessing* diterapkan untuk mempersiapkan data sehingga dapat diolah dengan optimal pada saat pelatihan model GRU. Proses ini diawali dengan pembersihan data dari karakter-karakter yang tidak diinginkan seperti “M” dan “B” pada fitur Volume dan penyesuaian tipe data agar seragam, termasuk konversi kolom *Date* menjadi format waktu (*datetime*) dan kolom-kolom lainnya menjadi tipe data *float*. Selanjutnya dilakukan proses pembagian data menjadi data untuk pelatihan sebesar 70% dan data untuk pengujian sebesar 30% secara berurutan berdasarkan waktu untuk menjaga kontinuitas pola deret waktu. Proses pembagian data mempertimbangkan prinsip *temporal validation*, di mana data dibagi menjadi dua bagian dengan tetap mempertahankan kesesuaian urutannya tanpa melakukan pengacakan, karena data masa depan tidak boleh memengaruhi proses pelatihan agar model dapat dievaluasi secara realistis terhadap data yang benar-benar belum pernah dilihat.

Tabel 2. Pembagian data

Data	Periode	Jumlah data
Pelatihan	26/10/2018 – 06/09/2023	1153
Pengujian	07/09/2023 – 24/10/2025	477

Tahap berikutnya adalah normalisasi data menggunakan metode skala *minmax* supaya seluruh fitur pada dataset berada dalam rentang dan skala yang sama [0, 1] untuk memudahkan dalam analisis dan melatih model. Proses *fitting* atau perhitungan nilai minimum dan maksimum yang digunakan pada normalisasi hanya dilakukan pada data pelatihan, kemudian nilai tersebut digunakan untuk mentransformasikan data pelatihan dan pengujian. Hal tersebut bertujuan untuk menghindari *data leakage*, yaitu suatu keadaan dimana model secara tidak sengaja mengetahui sebaran data awal sehingga membuat proses evaluasi dengan data uji menghasilkan bias dan tidak menunjukkan performa model yang sebenarnya. Rumus normalisasi skala *minmax* terdapat pada Persamaan (1).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

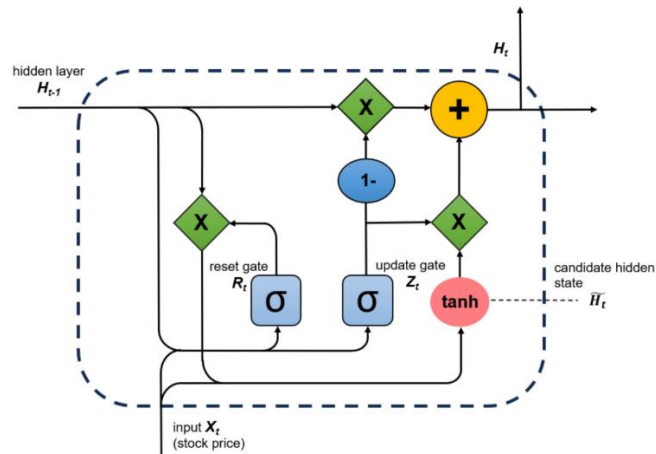
Proses terakhir yaitu pembentukan data sekuens menggunakan pendekatan *sliding window* agar data dapat diolah sebagai deret waktu oleh model GRU, di mana setiap sekuens terdiri dari nilai historis  $n$  atau beberapa hari sebelumnya untuk kemudian memprediksi harga saham pada keesokan harinya[13]. Pada penelitian ini digunakan *sequence length* sebesar 30 hari, yang berarti setiap sampel input terdiri dari 30 nilai historis berturut-turut untuk memprediksi harga pada hari ke-31. *Window* yang digunakan bersifat statis dan tidak menerapkan *padding*, oleh karena itu, data dengan panjang kurang dari 30 hari tidak disertakan dalam pelatihan. proses pembentukan data sekuens dengan metode *sliding window* diilustrasikan pada Gambar 2.



Gambar 2. Metode *sliding window*

### **Gated Recurrent Unit (GRU)**

GRU merupakan salah satu varian dari arsitektur RNN yang dikembangkan untuk mencegah terjadinya masalah *vanishing gradient*, namun dengan rancangan yang lebih ringkas dibandingkan varian pendahulunya, LSTM[14]. Pada arsitektur ini terdapat dua gerbang utama, yaitu *update gate* dan *reset gate*, yang berfungsi mengendalikan aliran informasi di dalam jaringan. Mekanisme tersebut memungkinkan GRU untuk menentukan bagian informasi dari data masa lalu yang perlu dipertahankan maupun diabaikan[15]. Dalam penerapannya pada peramalan deret waktu, GRU mampu mempelajari hubungan ketergantungan jangka panjang secara efektif, tetapi tetap menawarkan proses pelatihan yang lebih cepat dan efisien[16], sehingga model dapat menghasilkan prediksi yang baik dengan kompleksitas komputasi yang lebih rendah.



Gambar 3. Arsitektur blok GRU[17]

Gambar 3 merupakan gambar arsitektur GRU,  $X_t$  berperan sebagai data input saat waktu ke- $t$ , sedangkan  $H_{t-1}$  merupakan *hidden state* yang dibawa dari langkah sebelumnya. Keduanya dikombinasikan melalui bobot tertentu dan hasilnya dilewatkan ke fungsi aktivasi sigmoid. Di dalam proses ini, *update gate* ( $Z_t$ ) berfungsi mengendalikan seberapa banyak informasi dari masa lalu yang tetap dipertahankan, sehingga membantu mengatasi masalah *vanishing gradient*. Perhitungan *update gate* ditunjukkan pada persamaan (2) dengan  $W_z$  sebagai bobot/*weight* dan  $b_z$  sebagai bias untuk *update gate*[18], [19].

$$Z_t = \sigma (W_z \cdot [H_{t-1}, x_t] + b_z) \quad (2)$$

Sementara itu, *reset gate* ( $R_t$ ) menentukan bagian informasi lama yang perlu diabaikan. Dihitung dengan persamaan (3) dengan  $W_r$  sebagai bobot/*weight* dan  $b_r$  sebagai bias untuk *reset gate*[18], [19].

$$R_t = \sigma (W_r \cdot [H_{t-1}, x_t] + b_r) \quad (3)$$

Nilai  $\tilde{H}_t$  atau memori baru dihasilkan dari gabungan input dan state sebelumnya yang telah diatur oleh reset gate, kemudian diolah melalui fungsi aktivasi tanh, seperti persamaan (4) dengan  $W_h$  sebagai bobot/*weight* dan  $b_h$  sebagai bias untuk kandidat *hidden state*[18], [19].

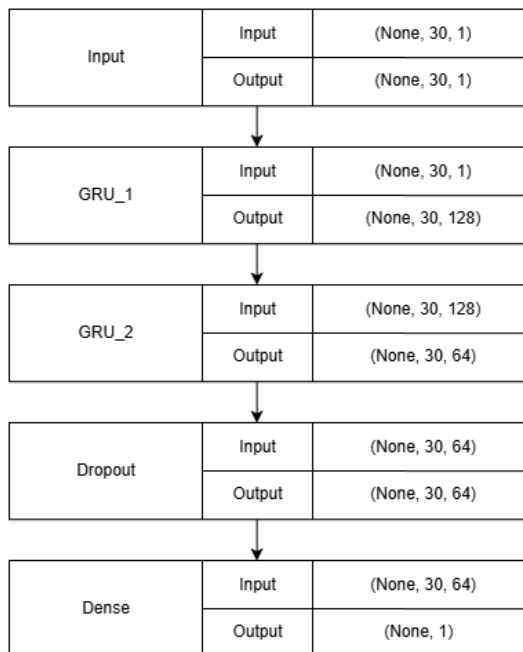
$$\tilde{H}_t = \tanh (W_h \cdot [R_t \cdot H_{t-1}, x_t] + b_h) \quad (4)$$

Akhirnya,  $H_t$  dihitung dengan persamaan (5) yang menggabungkan memori lama dan memori baru melalui operasi elemen-per-elemen (Hadamard product), sehingga hanya informasi yang relevan saja yang diteruskan ke langkah berikutnya[18], [19].

$$H_t = (1 - Z_t) \odot H_{t-1} + Z_t \odot \tilde{H}_t \quad (5)$$

GRU kemudian dirangkai bersama *layer input*, *dropout*, dan *layer Dense* sebagai *output*, sehingga arsitektur final model terdiri atas *layer input* yang menerima sekuens data, diikuti oleh dua layer GRU dengan jumlah unit 128 dan 64 secara berurutan untuk mengekstraksi pola temporal, *layer dropout* dengan tingkat 0,2 untuk mengurangi

*overfitting*, dan akhirnya layer *Dense* yang memetakan representasi fitur menjadi nilai prediksi.



Gambar 4. Arsitektur final model prediksi saham

### Optimasi *hyperparameter*

Variabel yang disebut *hyperparameter* adalah parameter yang mengatur bagaimana model belajar, seperti jumlah *neuron*, *batch size*, *learning rate*, dan jumlah *epoch* pada pada pelatihan model. Pentingnya proses tuning *hyperparameter* terletak pada pengaruhnya terhadap kinerja model, karena pemilihan kombinasi *hyperparameter* yang tepat dapat secara signifikan meningkatkan hasil prediksi yang dihasilkan model[19]. *Grid Search* merupakan metode *optimasi hyperparameter* yang umum digunakan untuk menentukan kombinasi parameter optimal untuk suatu model. Teknik ini bekerja dengan mengevaluasi seluruh kombinasi nilai *hyperparameter* yang telah ditentukan dalam bentuk grid secara sistematis[20]. Meskipun *Grid Search* dikenal memiliki biaya komputasi yang lebih tinggi dibandingkan pendekatan *random search* atau algoritma heuristik, metode ini tetap dipilih karena memberikan eksplorasi ruang *hyperparameter* yang menyeluruh, sehingga peluang menemukan konfigurasi optimal menjadi lebih tinggi. Pendekatan tanpa optimasi *hyperparameter* terbukti menghasilkan performa model yang kurang stabil, sebagaimana ditunjukkan pada beberapa penelitian terdahulu. Misalnya, Khairunnisa [1] dan Tuttur [2] menggunakan model prediksi tanpa optimasi sistematis, sehingga performa yang dihasilkan sangat bergantung pada pemilihan parameter awal dan sering kali tidak mencerminkan kemampuan terbaik dari model yang digunakan. Berdasarkan landasan tersebut, *Grid Search* dipandang sebagai pendekatan yang tepat pada penelitian ini karena mampu memberikan pencarian parameter yang sistematis dan lengkap, terutama pada model GRU yang memiliki jumlah parameter relatif terbatas sehingga ruang pencarian masih dapat dievaluasi secara efisien.

### Metrik Evaluasi

Untuk memastikan evaluasi kinerja yang objektif, model perlu diuji dengan metrik evaluasi. Setiap konfigurasi terlebih dahulu dilatih, kemudian dievaluasi menggunakan data

uji untuk melihat kualitas generalisasi model. Penentuan kombinasi *hyperparameter* terbaik didasarkan pada hasil performa model ketika diuji pada data uji, sehingga diperoleh konfigurasi yang tidak hanya unggul dalam pelatihan tetapi juga stabil dalam melakukan prediksi pada data baru.

Penelitian ini menggunakan tiga metrik evaluasi, yaitu *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE), untuk menilai tingkat akurasi model. Semua indikator tersebut digunakan untuk menghitung besar kesalahan antara nilai yang sebenarnya  $y_t$  dan nilai hasil prediksi  $\hat{y}_t$  pada model regresi. Secara umum, semakin kecil nilai MAE, MSE, dan RMSE, maka semakin baik kualitas pemodelan atau kecocokan model terhadap data. Perhitungan MAE, RMSE, dan MAPE secara berurutan terdapat pada Persamaan (6), (7), dan (8)[8], [21].

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (6)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (7)$$

$$MAPE (\%) = \frac{1}{n} \sum_{t=1}^n \left| \frac{\hat{y}_t - y_t}{\hat{y}_t} \right| \times 100 \quad (8)$$

## HASIL DAN PEMBAHASAN

Bagian ini menampilkan hasil pengujian yang telah dilakukan. Pada tahap sebelumnya, dilakukan percobaan dengan beberapa variasi rasio pembagian data, yaitu 90:10, 80:20, dan 70:30. Dari hasil tersebut, rasio 70:30 memberikan performa terbaik. Selanjutnya percobaan panjang sekuens dengan variasi 10, 20, dan 30, ditemukan panjang sekuens 30 memberikan performa terbaik. Selanjutnya, percobaan jumlah *hidden layer* pada model GRU dengan variasi jumlah 2, 3, dan 4 *hidden layer* menunjukkan bahwa dua *hidden layer* memberikan hasil paling optimal. Selibuhnya Uji coba *learning rate* menunjukkan bahwa *learning rate* 0,001 merupakan yang paling sesuai dalam pelatihan model. Berdasarkan hasil percobaan tersebut dilakukan percobaan lanjutan untuk menentukan konfigurasi model dengan melakukan percobaan pada beberapa variasi konfigurasi units dengan tambahan atau tanpa tambahan lapisan *dropout*.

Tabel 3. Perbandingan konfigurasi model

<i>Model</i>	<i>Units</i>	<i>Dropout</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAPE</i>
A	32-32	Tanpa <i>dropout</i>	54.0959	86.5471	2.4466
B	64-64		64.1142	97.2344	2.9283
C	128-128		61.1117	98.61	2.6917
D	32-32	<i>Dropout</i> 0,2	51.9571	80.3499	2.4357
E	64-64		54.1193	80.9838	2.5841
F	128-128		<b>51.7777</b>	<b>79.0656</b>	<b>2.4593</b>

Pada Tabel 4 ditunjukkan bahwa model dengan penggunaan *dropout* bobot 0,2 melampaui semua akurasi model tanpa penggunaan *dropout* pada berbagai variasi *units hidden layer* dengan akurasi terbaik diperoleh oleh units 128-128 diikuti *layer dropout* dengan bobot 0,2 dengan hasil MAE 51,7777, nilai RMSE 79,0656 dan nilai MAPE 2,4593%. berdasarkan hasil percobaan tersebut, pengujian selanjutnya menggunakan grid search difokuskan untuk observasi ruang pencarian *hyperparameter* pada jumlah *units* masing-masing *layer*, ukuran *batch*, serta jumlah *epoch* untuk proses optimasi menggunakan

*Grid Search*. Ruang *hyperparameter* yang akan diobservasi oleh metode *Grid search* ditunjukkan pada Tabel 4.

Tabel 4. Ruang *Hyperparameter*

<i>Hyperparameter</i>	Nilai
<i>Units 1</i>	32, 64, 128
<i>Units 2</i>	32, 64, 128
Ukuran <i>batch</i>	8, 16, 32
<i>Epochs</i>	75, 100

Berdasarkan ruang *hyperparameter* pada Tabel 4 akan dilakukan proses optimasi pada model GRU. Proses optimasi dengan *Grid search* dilakukan sebanyak kombinasi yang ada pada Tabel 4, yaitu 54 kombinasi. Setiap kombinasi dievaluasi menggunakan *3-fold cross validation* untuk menghasilkan model yang memiliki kemampuan generalisasi data yang baik. Perlu ditegaskan bahwa penerapan *3-fold cross validation* pada penelitian ini tidak menggunakan skema pengacakan data (*random shuffling*) sebagaimana pada *cross validation* konvensional, tetapi menggunakan pembagian berurutan berdasarkan waktu (*blocked temporal folds*), dimana setiap bagian tetap mempertahankan sifat deret waktu di mana data pelatihan selalu berada pada periode waktu yang lebih awal dibandingkan data validasi sehingga tidak terjadi *temporal leakage* dan validasi tetap mencerminkan skenario prediksi dunia nyata. Total proses pelatihan yang dilakukan sepanjang optimasi adalah 162 iterasi, yaitu hasil dari 54 kombinasi *hyperparameter* yang masing-masing divalidasi menggunakan 3 fold temporal.

Untuk menguji keefektifan model yang dihasilkan, dilakukan perbandingan antara model GRU dan GRU dengan optimasi *Grid Search* menggunakan variasi jenis *optimizer* (Adam, SGD, dan RMSprop). Evaluasi kinerja dilakukan dengan menggunakan metrik RMSE, MAE, dan MAPE, sebagaimana ditunjukkan pada Tabel 4.

Tabel 5. Hasil pengujian

<i>Metode</i>	<i>Optimizer</i>	<i>MAE</i>	<i>RMSE</i>	<i>MAPE</i>
GRU	Adam	51.7777	79.0656	2.4593
	RMSProp	64.5809	88.0648	3.2641
	SGD	56.0892	88.8476	2.6345
GRU- <i>Grid search</i>	<b>Adam</b>	<b>45.6501</b>	<b>67.8805</b>	<b>2.2309</b>
	RMSProp	55.1797	71.8045	2.9283
	SGD	48.9893	76.6049	2.3492

Berdasarkan hasil pengujian pada Tabel 4, model GRU menghasilkan nilai *error* terendahnya dengan menggunakan *optimizer* Adam, dengan MAE sebesar 54,1193, RMSE sebesar 80,9838, nilai MAPE 2,5841%. Sementara itu Model GRU yang dioptimasi dengan *Grid search* menghasilkan *error* terendahnya pada pengujian menggunakan *optimizer* Adam dengan memperoleh nilai MAE sebesar 45,6501, nilai RMSE sebesar 67,8805, dan nilai MAPE sebesar 2,2309%. Adam cenderung memberikan hasil yang lebih baik karena menggunakan mekanisme *adaptive learning rate*, di mana laju pembelajaran disesuaikan secara otomatis untuk setiap parameter sehingga proses konvergensi menjadi lebih cepat dan stabil. *Optimizer* ini juga memanfaatkan momentum melalui estimasi rata-rata gradien pertama dan kedua, sehingga dapat menghindari osilasi tajam dan memberikan pembaruan bobot yang lebih halus dibandingkan *optimizer* SGD dan RMSProp. Selain itu, karakteristik Adam yang adaptif sangat cocok untuk data harga saham yang fluktuatif, karena mampu

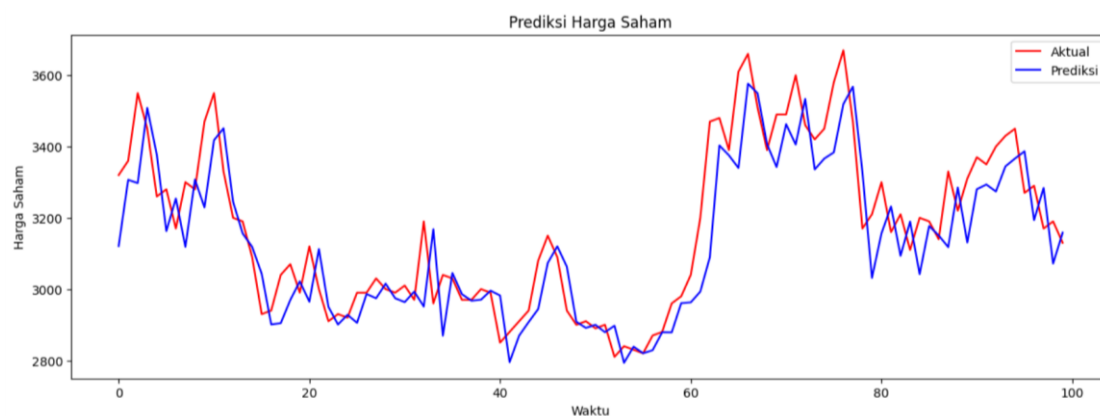
menyesuaikan kecepatan pembelajaran terhadap perubahan pola gradien yang terjadi pada data. Keunggulan *optimizer* Adam bukan hanya terlihat dari nilai *error* yang lebih rendah, tetapi juga dari kemampuannya menjaga stabilitas pelatihan pada data dengan volatilitas tinggi seperti saham ANTM. Dengan hasil tersebut dapat disimpulkan bahwa pemilihan *optimizer* yang tepat berperan penting terhadap kualitas model yang dilatih, karena setiap *optimizer* memiliki mekanisme pembaruan bobot yang berbeda sehingga menghasilkan performa pelatihan yang tidak sama.

Pada semua *optimizer* yang diuji, model GRU yang telah dioptimasi *Grid Search* konsisten menunjukkan nilai *error* yang lebih rendah dibandingkan GRU standar. Maka, dapat disimpulkan bahwa *Grid Search* tidak hanya efektif dalam menemukan *hyperparameter* terbaik pada konfigurasi tertentu, tetapi juga konsisten meningkatkan performa GRU pada berbagai jenis *optimizer*. Dengan penyesuaian *hyperparameter* yang lebih sesuai dengan karakteristik data, model GRU dapat mempelajari pola historis harga saham secara lebih optimal dan menghasilkan prediksi yang lebih akurat. Adapun konfigurasi *hyperparameter* terbaik yang diperoleh dari proses *Grid Search* ditunjukkan pada Tabel 6.

Tabel 6. Konfigurasi model GRU terbaik

<i>Hyperparameter</i>	<i>Optimizer</i>
Pembagian data	70:30
Panjang sekuens	30
Jumlah <i>hidden layer</i>	2
<i>Learning rate</i>	0,001
<i>Epochs</i>	75
Ukuran <i>batch</i>	8
<i>Units</i>	128-64
<i>Dropout</i>	0,2
<i>Optimizer</i>	Adam

Untuk memberikan gambaran yang lebih mengenai kinerja model, dilakukan perbandingan visual nilai aktual dengan nilai yang diprediksi oleh model terbaik., yaitu GRU yang telah dioptimasi dengan *Grid Search*, ditampilkan pada Gambar 5. Visualisasi tersebut menunjukkan bahwa pola prediksi yang dihasilkan model dapat mengikuti pergerakan data aktual dengan sangat baik, baik pada tren kenaikan maupun penurunan.



Gambar 5. Visualisasi nilai aktual vs prediksi

## KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa optimasi *hyperparameter* menggunakan metode *Grid Search* terbukti mampu meningkatkan performa model GRU dalam memprediksi harga saham ANTM. Model GRU yang telah dioptimasi menunjukkan nilai *error* yang lebih rendah dibandingkan model tanpa optimasi pada seluruh jenis *optimizer* yang diuji, dengan *optimizer* Adam memberikan performa terbaik berdasarkan indikator evaluasi. Pada model GRU-Adam dengan hasil optimasi, diperoleh nilai MAE sebesar 45,6501, RMSE sebesar 67,8805, dan MAPE sebesar 2,2309%. Temuan ini mengindikasikan bahwa pemilihan kombinasi *hyperparameter* yang tepat, seperti jumlah *units*, ukuran *batch*, jumlah *epoch*, dan jenis *optimizer*, memiliki peran penting dalam meningkatkan kemampuan model dalam mempelajari pola historis pergerakan harga saham serta menghasilkan prediksi yang lebih akurat. Proses optimasi model dengan *Grid search* bekerja dengan mengevaluasi seluruh kombinasi *hyperparameter*, sehingga penetapan ruang pencarian harus seminimal dan seefektif mungkin untuk menghemat sumber daya komputasi. Untuk penelitian selanjutnya, disarankan menggunakan metode optimasi lain seperti mencoba pendekatan *Genetic Algorithm* yang dapat menjelajahi ruang parameter yang lebih luas dengan lebih efisien sehingga berpotensi menemukan konfigurasi model yang lebih optimal.

## REFERENSI

- [1] A. Khairunnisa, L. Muflikhah, and B. D. Setiawan, "Pemodelan Prediktif Harga Saham Menggunakan Simple Moving Average Dengan Metode Long Short-Term Memory," 2025. doi: 10.35940/ijrteB3048.078219.
- [2] A. Tuter, R. Manik, and S. Andayani, "Pemodelan Prediktif Harga Saham Pada Bank ABC Menggunakan K-Nearest Neighbors Dengan Pendekatan Tren," *JSI : Jurnal Sistem Informasi (E-Journal)*, vol. 16, no. 2, p. 2024, 2024, doi: 10.18495/jsi.v16i2.177.
- [3] M. M. Al Haromainy, D. A. Prasetya, and A. P. Sari, "Improving Performance of RNN-Based Models With Genetic Algorithm Optimization For Time Series Data," *TIERS Information Technology Journal*, vol. 4, no. 1, pp. 16–24, Jun. 2023, doi: 10.38043/tiers.v4i1.4326.
- [4] F. Ricchiuti and G. Sperli, "An Advisor Neural Network framework using LSTM-based Informative Stock Analysis," *Expert Syst Appl*, vol. 259, Jan. 2025, doi: 10.1016/j.eswa.2024.125299.
- [5] Z. Xu, "Factors Influencing the Predictive Performance of the LSTM Model on Stock Prices and Its Application in Forecasting U.S. Technology Sector Stock Prices — Sample Time Span, Time Window Length, Feature Selection, and Prediction Days," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2024, pp. 370–375. doi: 10.1145/3700906.3700964.

- [6] A. Viéitez, M. Santos, and R. Naranjo, "Machine learning Ethereum cryptocurrency prediction and knowledge-based investment strategies," *Knowl Based Syst*, vol. 299, Sep. 2024, doi: 10.1016/j.knosys.2024.112088.
- [7] Khalis Sofi, Aswan Supriyadi Sunge, Sasmitoh Rahmad Riady, and Antika Zahrotul Kamalia, "Perbandingan Algoritma Linear Regression, Lstm, Dan Gru Dalam Memprediksi Harga Saham Dengan Model Time Series," *Seminastika*, vol. 3, no. 1, pp. 39–46, 2021, doi: 10.47002/seminastika.v3i1.275.
- [8] D. M. Teixeira and R. S. Barbosa, "Stock Price Prediction in the Financial Market Using Machine Learning Models," *Computation*, vol. 13, no. 1, Jan. 2025, doi: 10.3390/computation13010003.
- [9] A. P. Sari, T. Yasuno, D. A. Prasetya, and M. M. Al Haromany, "Forecasting System of Wind Speed and Direction by Neural Network," in *Proceeding - IEEE 9th Information Technology International Seminar, ITIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ITIS59651.2023.10420216.
- [10] A. P. Sari, D. A. Prasetya, T. Yasuno, A. N. Sihananto, M. M. Al Haromany, and W. S. J. Saputra, *Forecasting Model of Wind Speed and Direction by Convolutional Neural Network - Deep Convolutional Long Short Term Memory*. IEEE, 2022. doi: 10.1109/ITIS57155.2022.10010005.
- [11] C. Y. Lin and J. A. Lobo Marques, "Stock market prediction using artificial intelligence: A systematic review of systematic reviews," Jan. 01, 2024, *Elsevier Ltd*. doi: 10.1016/j.ssaho.2024.100864.
- [12] K. E. Hoque and H. Aljamaan, "Impact of hyperparameter tuning on machine learning models in stock price forecasting," *IEEE Access*, vol. 9, pp. 163815–163830, 2021, doi: 10.1109/ACCESS.2021.3134138.
- [13] I. Hariyanti, V. Hafizh, C. Putra, A. R. Raharja, and K. Kunci, "PREDIKSI HARGA SAHAM BBKA MENGGUNAKAN METODE LONG SHORT-TERM MEMORY DAN GATED RECURRENT UNIT," *Jurnal Responsif: Riset Sains &*, vol. 7, 2025, doi: 10.51977/jti.v7i1.1901.
- [14] D. Kurniasari, M. E. Nuraini, and R. K. Nisa, "A Case Study: Comparison of LSTM and GRU Methods for Forecasting Oil, Non-Oil, and Gas Export Values in Indonesia," vol. 17, no. 2, pp. 127–138, 2024, doi: 10.15408/jti.v17i2.39098.
- [15] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, p. 517, Aug. 2024, doi: 10.3390/info15090517.
- [16] A. Arwansyah, S. Suryani, H. SY, U. Usman, A. Ahyuna, and S. Alam, "Time Series Forecasting Menggunakan Deep Gated Recurrent Units," *Digital Transformation Technology*, vol. 4, no. 1, pp. 410–416, 2022, doi: 10.47709/digitech.v4i1.4141.
- [17] C. Fan and X. Zhang, "Stock price nowcasting and forecasting with deep learning," *J Intell Inf Syst*, pp. 0–19, 2024, doi: 10.1007/s10844-024-00908-2.
- [18] K. Alkhatib, H. Khazaleh, H. A. Alkhazaleh, A. R. Alsoud, and L. Abualigah, "A New Stock Price Forecasting Method Using Active Deep Learning Approach," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 8, no. 2, Jun. 2022, doi: 10.3390/joitmc8020096.
- [19] U. Singh, S. Tamrakar, K. Saurabh, R. Vyas, and O. P. Vyas, "Optimizing Hyperparameters of Deep Learning Models for Stock Price Prediction," in *2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICCCNT61001.2024.10724813.
- [20] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *International Journal of Computers and Applications*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.
- [21] Y. Wang and K. Yan, "Machine learning-based quantitative trading strategies across different time intervals in the American market," *Quantitative Finance and Economics*, vol. 7, no. 4, pp. 569–594, 2023, doi: 10.3934/QFE.2023028.