



Perbandingan Model Long Short-Term Memory, Gated Recurrent Unit, dan Convolutional Neural Network untuk Peramalan Harga Emas

Sabrina Aziz Aulia*, Ahmad Muhamad Solihin, Rifky Surya Pratama, Ghofari Miftah

Universitas Negeri Semarang, Sekaran, Gunungpati, Semarang 50229, Indonesia

* Alamat Surel: sabrinaazizaulia@students.unnes.ac.id

Abstrak

Harga emas yang terus berubah meningkatkan risiko investasi yang merupakan bagian penting dari kondisi ekonomi, keuangan bank, dan pasar saham. Sehingga, peramalan harga emas diperlukan untuk dapat membantu dan mendukung investor keuangan dan bank sentral dalam mengambil keputusan yang tepat. Penelitian ini membahas secara komprehensif performa model prediksi *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), dan *Convolutional Neural Network* (CNN) untuk peramalan harga emas. Model dibangun dan dibandingkan pada tiga *time step*, yaitu 5 (1 minggu), 30 (1 bulan), dan 260 (1 tahun). Hasil dari penelitian menunjukkan bahwa proses pelatihan model GRU lebih efisien dibandingkan model lainnya pada semua *time step*. Berdasarkan akurasi, pada *time step* 5 dan 30, model GRU menunjukkan akurasi yang lebih baik dibandingkan model lainnya. Sedangkan pada *time step* 260, tidak ada perbedaan akurasi yang signifikan antara model LSTM dengan GRU yang mana lebih baik dibandingkan dengan model CNN. Sehingga, dapat disimpulkan bahwa model GRU adalah model yang paling efisien dan akurat pada *time step* 5, 30, dan 260.

Kata kunci: Emas, *Time Series*, LSTM, GRU, CNN

© 2025 Dipublikasikan oleh Jurusan Matematika, Universitas Negeri Semarang

1. Pendahuluan

Harga emas merupakan bagian penting dari kondisi ekonomi dan keuangan bank dan pasar saham (Ali et al., 2020). Fluktuasi harga emas selalu meningkatkan risiko investasi, sementara penyebab fluktuasi ini sangat rumit dan tren harga emas dipengaruhi oleh banyak faktor (Livieris et al., 2020). Dalam pasar yang volatil, kemampuan untuk meramalkan harga emas secara akurat memiliki nilai strategis yang tinggi bagi investor, bank sentral, dan pemerintah (Tholib et al., 2023). Akurasi prediksi harga emas dapat membantu dalam pengambilan keputusan investasi dan pengelolaan risiko (Hafiyya et al., 2022). Oleh karena itu, prediksi harga emas dapat membantu dan mendukung investor keuangan dan bank sentral dalam mengambil keputusan yang tepat untuk kebijakan investasi mereka dan mengelola risiko.

Model peramalan runtun waktu diklasifikasikan menjadi dua, yaitu model klasik dan model modern. Dua model klasik yang paling umum digunakan adalah *AutoRegressive Integrated Moving Average* (ARIMA) dan *Seasonal ARIMA* (SARIMA) (Liu & Wang, 2024; Surendra et al., 2021). Keunggulan model klasik adalah kemampuannya untuk bekerja dengan baik pada data dengan hubungan linear yang jelas, tetapi seringkali terbatas dalam menangani data yang terlalu kompleks atau memiliki dinamika non-linear (Liu & Wang, 2024). Sebaliknya, model modern lebih fleksibel dan mampu menangani hubungan non-linear dalam data runtun waktu (Jason, 2018). Beberapa model modern yang populer digunakan adalah *Deep Learning*, *Random Forest*, *XGBoost* (Casolaro et al., 2023; Fawaz, 2020; Jabeur et al., 2024).

Model *Deep Learning* yang paling populer digunakan dalam peramalan harga emas adalah *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), dan *Convolutional Neural Network* (CNN) (Kantar & Kilimci, 2023; Nurhambali et al., 2024; Primananda & Isa, 2021). Dalam penelitian yang

To cite this article:

Aulia, SA., Solihin, AM, Pratama, RS, & Miftah, G. (2025). Perbandingan Model Long Short-Term Memory, Gated Recurrent Unit, dan Convolutional Neural Network untuk Peramalan Harga Emas. *PRISMA, Prosiding Seminar Nasional Matematika 8*, 414-431

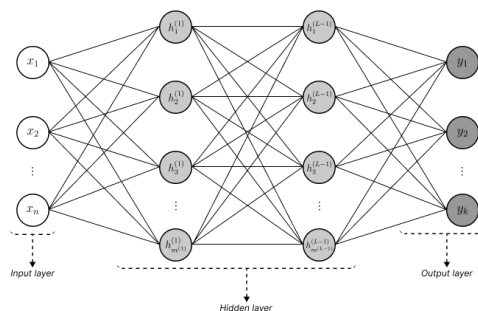
dilakukan oleh Nurhambali et al. (2024), hasil menunjukkan bahwa model LSTM memiliki kinerja yang baik pada peramalan harga emas. Penelitian yang dilakukan oleh Primananda & Isa (2021) menunjukkan bahwa model LSTM memiliki akurasi lebih baik dibandingkan model GRU pada data periode lebih dari tiga tahun. Namun, untuk data dengan periode kurang dari tiga tahun, model GRU lebih akurat dibandingkan model LSTM. Di sisi lain, penelitian yang dilakukan oleh Kantar & Kilimci (2023) menunjukkan bahwa model CNN memiliki performa yang lebih baik dibandingkan model LSTM dan GRU dalam memprediksi indeks harga emas. Sehingga, terdapat ketidakpastian terkait dengan model mana yang paling baik untuk digunakan dalam peramalan harga emas.

Fokus utama pada penelitian ini adalah untuk membuat model peramalan harga emas. Penelitian ini membahas secara komprehensif performa model peramalan harga emas. Penelitian ini juga berkontribusi dalam mengetahui perbandingan performa model LSTM, GRU, dan CNN dalam peramalan harga emas.

2. Metode

2.1. Deep Learning

Deep Learning yang pertama kali dikenalkan oleh (Ivakhnenko et al., 1967) merupakan suatu konsep *machine learning* yang dikembangkan dari model *Perceptron* yang terinspirasi dari jaringan syaraf biologis. Nama lain dari *Deep Learning* adalah *Deep Feedforward Networks*, disebut juga sebagai *Feedforward Neural Networks* (FFNN), atau *Multilayer Perceptrons* (MLPs) yang menjadi cikal bakal dari model *Deep Learning* (Courville, 2016). Model tersebut merupakan pengembangan dari model *Perceptron* dengan penambahan *hidden layer* yang berguna untuk memecahkan masalah non-linear yang lebih kompleks (Ivakhnenko et al., 1967). Model awal *Deep Learning* memiliki tiga komponen utama yaitu *input layer*, *hidden layer*, dan *output layer* yang dapat dilihat pada Gambar 1 (Courville, 2016).



Gambar 1. Diagram Deep Learning

Ketiga komponen tersebut adalah sebagai berikut. *Input layer* adalah layer yang berisi vektor *input* $x = (x_1, x_2, \dots, x_n)$ dengan n adalah banyaknya fitur x . *Hidden layer* adalah layer yang berisi gabungan proses *neuron-neuron* yang secara matematis dituliskan sebagai

$$h_j^{(l)} = f\left(\sum_{i=1}^n w_{ji}^{(l)} \cdot x_i + b_j^{(l)}\right), \quad (1)$$

dengan $h_j^{(l)}$ adalah nilai neuron ke- j pada hidden layer ke- l , $j = (0, 1, \dots, m^{(l)})$ dengan $m^{(l)}$ merupakan banyak *neuron* pada *hidden layer* ke- l , $l = (0, 1, \dots, L)$ dengan L merupakan banyaknya *hidden layer*, f merupakan fungsi aktivasi, $w_{ji}^{(l)}$ merupakan bobot pada vektor x_i dan *neuron* ke- j pada *hidden layer* ke- l , $b_j^{(l)}$ merupakan bias *neuron* ke- j pada *hidden layer* ke- l . *Output layer* adalah layer yang berisi vektor *output* $y = (y_1, y_2, \dots, y_k)$, dengan k merupakan banyaknya *output* dari model yang dituliskan sebagai

$$y_k = f\left(\sum_{j=1}^{m^{(L-1)}} w_{kj}^{(L)} \cdot h_j^{(L-1)} + b_k^{(L)}\right), \quad (2)$$

dengan y_k merupakan *output* ke- j , $j = (0, 1, \dots, m^{(L-1)})$ dengan $m^{(L-1)}$ merupakan banyak *neuron* pada *hidden layer* ke- $(L - 1)$ (terakhir), $w_{kj}^{(L)}$ merupakan bobot pada vektor x_k dan *neuron* ke- j pada *hidden layer* ke- L (terakhir), $h_j^{(L-1)}$ merupakan bobot pada vektor pada *neuron* ke- j pada *hidden layer* ke- $(L - 1)$ (terakhir), $b_k^{(L)}$ merupakan bias *neuron* ke- k *output layer*.

Model tersebut merupakan pelopor dari model-model *Deep Learning* lainnya. Banyak model yang dikembangkan seperti LSTM, GRU, dan CNN yang akan digunakan dalam penelitian ini.

2.2. Long Short-Term Memory (LSTM)

Model *Long Short-Term Memory* (LSTM) dikembangkan pertama kali oleh (Hochreiter & Schmidhuber, 1997). Model LSTM dibuat sebagai sebuah solusi dari permasalahan *vanishing* dan *exploding gradient* yang ada pada model *Recurrent Neural Network* (RNN). LSTM merupakan salah satu model *Deep Learning* yang *hidden neuronnya* diubah menjadi sel LSTM yang dapat dilihat pada Gambar 2. Arsitektur model LSTM memiliki enam komponen utama yaitu *input layer*, *forget gate*, *input gate*, *cell state*, *output gate*, dan *output layer*. Kemudian untuk diagram model LSTM yang lengkap dapat dilihat pada Gambar 3. Berikut adalah model LSTM secara matematis.

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

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (4)$$

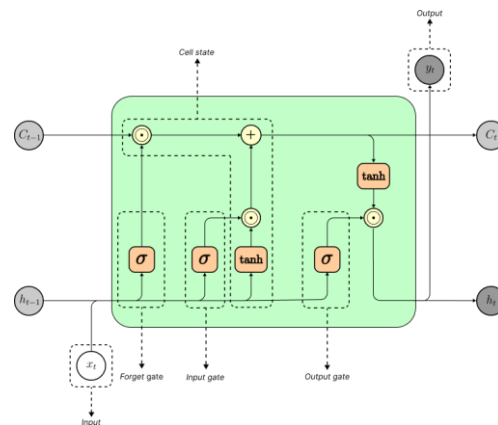
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (5)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (6)$$

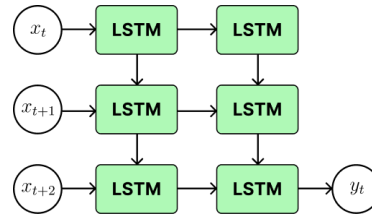
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{c}_t, \quad (7)$$

$$y_t = o_t \odot \tanh(C_t), \quad (8)$$

$$h_t = y_t. \quad (9)$$



Gambar 2. Diagram sel LSTM tunggal.



Gambar 3. Diagram LSTM terkoneksi.

Input layer adalah layer yang berisi vektor $x = (x_1, x_2, \dots, x_n)$ dengan n adalah banyaknya fitur x . *Forget gate* secara matematis dapat dilihat pada persamaan (3) dengan t adalah indeks waktu saat ini, f_t adalah nilai dari *forget gate*, σ adalah fungsi aktivasi sigmoid, W_f adalah bobot *forget gate* untuk vektor input x_t , U_f adalah bobot *forget gate* untuk neuron h_{t-1} , h_{t-1} adalah nilai neuron sebelumnya, dan b_f adalah bias dari *forget gate*. *Input gate* secara matematis dapat dilihat pada persamaan (4) dengan i_t adalah nilai dari *input gate*, W_i adalah bobot *input gate* untuk vektor input x_t , U_i adalah bobot *input gate* untuk vektor neuron h_{t-1} , dan b_i adalah bias dari *input gate*. *Output gate* secara matematis dapat dilihat pada persamaan (5) dengan o_t adalah nilai dari *output gate*, W_o adalah bobot *output gate* untuk vektor input x_t , U_o adalah bobot *output gate* untuk vektor neuron h_{t-1} , dan b_o adalah bias *output gate*. *Cell state* secara matematis dapat dilihat pada persamaan (6) dan (7) dengan \tilde{c}_t adalah kandidat dari *cell state*, W_c adalah bobot *cell state* untuk vektor input x_t , U_c adalah bobot *cell state* untuk vektor neuron h_{t-1} , b_c adalah bobot *cell state*, C_t adalah nilai dari *cell state* saat ini, \odot adalah *Hadamard product* (Horn & Johnson, 2012), dan C_{t-1} adalah nilai dari *cell state* sebelumnya. *Output layer* secara matematis dapat dilihat pada persamaan (8) dan (9) dengan y_t adalah nilai *output* saat ini dan jika nilai y_t dikirimkan lagi ke sel LSTM maka disebut neuron h_t .

LSTM masih populer digunakan hingga saat ini (Essai Ali et al., 2022; Murthy et al., 2020; Senanayake et al., 2022). Berbagai pengembangan LSTM juga diteliti salah satunya adalah GRU yang akan dibahas selanjutnya.

2.3. Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) adalah pengembangan dari model LSTM yang dikenalkan pertama kali oleh (Cho et al., 2014). Secara sederhana, GRU adalah model sederhana dari LSTM. GRU menyederhanakan *forget gate* dan *output gate* pada LSTM menjadi sebuah *reset gate* untuk dapat mencapai tujuan yang sama (Aggarwal, 2018). Arsitektur model GRU memiliki lima komponen utama yaitu *input layer*, *update gate*, *reset gate*, *hidden state*, dan *output layer* yang dapat dilihat pada Gambar 4. Secara matematis, model GRU dapat dituliskan sebagai

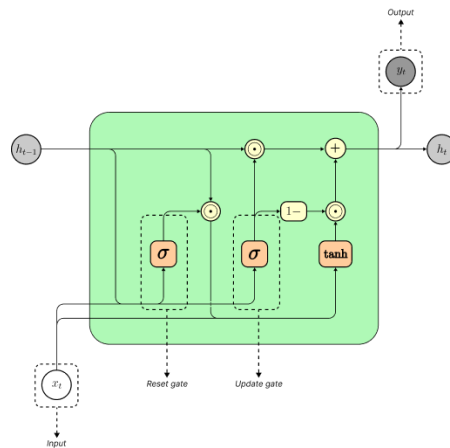
$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \quad (10)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (11)$$

$$\hat{h}_t = \sigma(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h), \quad (12)$$

$$y_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t, \quad (13)$$

$$h_t = y_t, \quad (14)$$



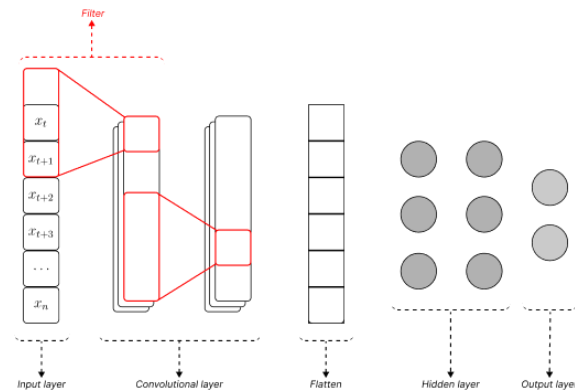
Gambar 4. Diagram sel GRU.

Input layer adalah layer yang berisi vektor *input* $x = (x_1, x_2, \dots, x_n)$ dengan n adalah banyaknya fitur x . *Update gate* secara matematis dapat dilihat pada persamaan (10) dengan z_t adalah nilai dari *update gate, σ adalah fungsi aktivasi sigmoid, W_z adalah bobot *update gate* untuk vektor *input* x_t , x_t adalah nilai *input* saat ini, U_z adalah bobot *update gate* untuk vektor *input* x_t , dan b_z adalah bias *update gate*. *Reset gate* secara matematis dapat dilihat pada persamaan (11) dengan r_t adalah nilai dari *reset gate*, W_r adalah bobot *reset gate* untuk vektor *input* x_t , U_r adalah bobot *reset gate* untuk vektor *neuron* h_{t-1} , dan b_r adalah bias *reset gate*. *Hidden state* secara matematis dapat dilihat pada persamaan (12) dengan \hat{h}_t adalah nilai dari calon *hidden state*, W_h adalah bobot *hidden state* untuk vektor *input* x_t , U_h adalah bobot *hidden state* untuk vektor *neuron* $r_t \odot h_{t-1}$, dan b_h adalah bias *hidden state*. *Output layer* secara matematis dapat dilihat pada persamaan (13) dan (14) dengan y_t adalah nilai *output* saat ini dan jika nilai y_t dikirimkan lagi ke sel GRU maka disebut *neuron* h_t .*

Model GRU lebih sederhana dan mudah dalam implementasi dibandingkan dengan model LSTM. Namun, kedua model terbukti memiliki performa yang kurang-lebih sama tergantung pada tugas yang dihadapi (Aggarwal, 2018).

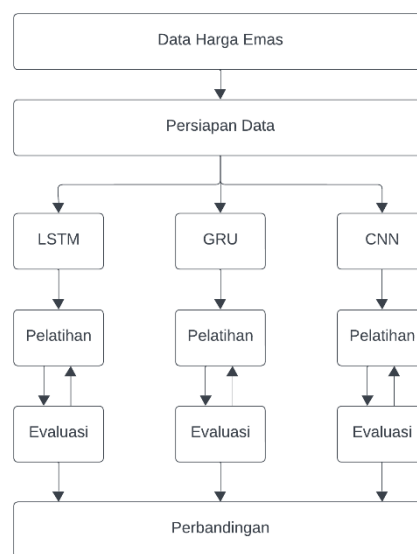
2.4. Convolutional Neural Network (CNN)

Konsep *Convolutional Neural Network (CNN)* pertama kali dikembangkan oleh Hubel D dan Wiesel T pada tahun 1960-an. Namun, penerapan model CNN pertama diterapkan oleh (Yann & Yoshua, 1995). CNN mampu mempelajari dan mengekstrak fitur secara otomatis dari data input mentah dari peramalan deret waktu (Jason, 2018). Terdapat lima komponen utama dalam CNN, yaitu (1) *input layer*, (2) *convolutional layer*, (3) *flatten*, (4) *hidden layer*, dan (5) *output layer* yang dapat dilihat pada Gambar 5. *Input layer* adalah layer yang berisi vektor *input* $x = (x_1, x_2, \dots, x_n)$ dengan n adalah banyaknya fitur x . *Convolutional layer* adalah *layer* yang menerapkan operasi konvolusi pada data input dengan menggunakan filter atau kernel kecil yang menggeser melintasi data untuk mendeteksi fitur atau pola tertentu pada data. *Flatten* merubah *output* dari *convolutional layer* menjadi vektor satu dimensi untuk digunakan pada proses selanjutnya yang sama dengan proses pada *Deep Learning*. *Hidden layer* dan *output layer* pada CNN adalah sama dengan *hidden layer* dan *output layer* yang ada pada model *Deep Learning*.



Gambar 5. Diagram CNN 1d.

2.5. Teknik analisis data



Gambar 6. Diagram teknik analisis data.

Ada beberapa tahapan dalam analisis data pada penelitian ini yang dapat dilihat pada Gambar 6. (1) pengumpulan data harga emas, (2) persiapan data, (3) pemilihan dan pembangunan model, (4) pelatihan model, (5) evaluasi model, dan (6) hasil atau perbandingan performa model.

Penjelasan dari tahapan-tahapan pada teknik analisis data adalah sebagai berikut. Pertama, data yang digunakan dalam penelitian adalah data historis harga emas dari tanggal 1 Januari 2020 s.d. 9 Oktober 2024 yang merupakan data sekunder yang diperoleh melalui website *finance.yahoo.com*. Data ini berisi harga emas harian (tanpa hari libur) pada saat “open”, “high”, “low”, “close”, “adj close”, dan “volume”. Data yang digunakan dalam penelitian adalah data “close”, yaitu harga terakhir yang diperdagangkan sebelum pasar tutup per hari.

Tahapan kedua adalah tahap persiapan data. Persiapan data ini meliputi pembersihan data, transformasi data, dan normalisasi data. Pembersihan data adalah membersihkan data-data yang kosong. Kemudian, transformasi data adalah mengubah data *time series* dari data menjadi data dengan tiga *time step* yang berbeda yaitu 5 (1 minggu), 30 (1 bulan), dan 260 (1 tahun). Setelah dibagi menjadi tiga *time step* data kemudian dibagi menjadi data *training* dan *testing* yang secara berturut-turut digunakan untuk melatih dan menguji akurasi model. Data *training* yang digunakan pada penelitian ini adalah data dari tahun 2020 s.d. tahun 2023. Data *testing* yang digunakan pada penelitian ini adalah data pada tahun 2024. Lalu, normalisasi data adalah mengubah data menjadi 0 – 1 untuk mempermudah komputasi.

Tahapan ketiga adalah pemilihan dan pembangunan model. Model yang digunakan dalam penelitian ini adalah model LSTM, GRU, dan CNN. Kemudian, nilai parameter dari model diperoleh menggunakan *bayesian optimization*. Setelah model dibentuk, selanjutnya model dilatih menggunakan data *training*. Model dilatih dengan cara iterasi dengan banyaknya iterasi disebut dengan *epochs*. Selanjutnya,

dilakukan pengujian model menggunakan data *testing*. Proses ini diulang hingga performa model dianggap optimal yang diukur menggunakan matriks evaluasi MSE, RMSE, MAE, dan MAPE.

Tahap selanjutnya adalah perbandingan model. Semua model yang telah optimal kemudian dibandingkan berdasarkan efisiensi pelatihan dan akurasi modelnya. Efisiensi pelatihan diukur menggunakan kurva pelatihan. Sedangkan akurasi diukur menggunakan matriks evaluasi MAPE.

2.6. Matriks Evaluasi

Performa model pada penelitian ini diuji dengan menggunakan matriks evaluasi. Adapun matriks evaluasi yang digunakan adalah *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE). Matriks evaluasi tersebut akan dijelaskan selanjutnya.

2.6.1. Mean Squared Error (MSE)

Mean Squared Error (MSE) adalah matriks evaluasi yang menghitung rata-rata dari selisih kuadrat antara nilai prediksi dengan nilai sebenarnya. MSE sering digunakan untuk mendeteksi *outlier* karena sensitif terhadap *outlier* (Chicco et al., 2021). MSE secara matematis didefinisikan sebagai

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (15)$$

dengan n adalah banyak data, y_i adalah nilai ke- i dengan $i = (1, 2, \dots, n)$, \hat{y}_i adalah nilai prediksi ke- i .

2.6.2. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) adalah matriks evaluasi yang menghitung akar dari MSE. RMSE secara matematis didefinisikan sebagai

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (16)$$

dengan n adalah banyak data, y_i adalah nilai ke- i dengan $i = (1, 2, \dots, n)$, \hat{y}_i adalah nilai prediksi ke- i .

2.6.3. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah matriks evaluasi yang menghitung rata-rata selisih absolut antara nilai prediksi dengan nilai sebenarnya. MAE secara matematis didefinisikan sebagai

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (17)$$

dengan n adalah banyak data, y_i adalah nilai ke- i dengan $i = (1, 2, \dots, n)$, \hat{y}_i adalah nilai prediksi ke- i .

2.6.4. Mean Absolute Percentage Error (MAPE)

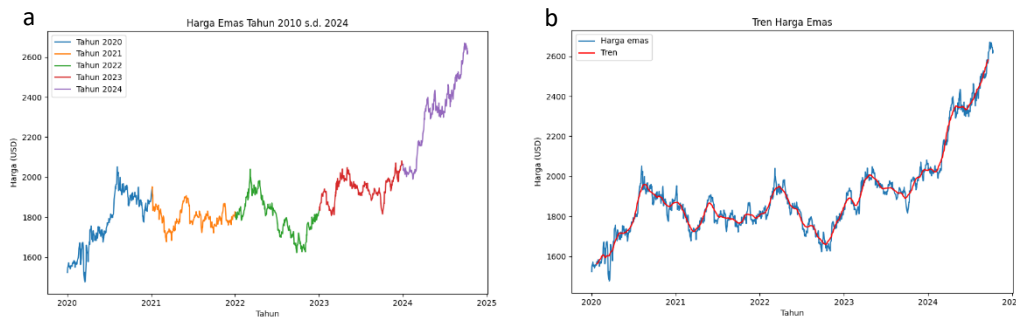
Mean Absolute Percentage Error (MAPE) memberikan interpretasi yang sangat intuitif dalam hal kesalahan relatif [Chicco, 2021]. MAPE secara matematis didefinisikan sebagai

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i|} \times 100\% \quad (18)$$

dengan n adalah banyak data, y_i adalah nilai ke- i dengan $i = (1, 2, \dots, n)$, \hat{y}_i adalah nilai prediksi ke- i .

3. Hasil dan Pembahasan

Penelitian ini memberikan gambaran data harga emas dunia dari tanggal 1 Januari 2020 s.d. 9 Oktober 2024 yang merupakan data sekunder yang diambil melalui situs web *finance.yahoo.com*. Visualisasi data dan tren harga emas dapat dilihat pada Gambar 7. Terlihat bahwa harga emas memiliki tren yang terus naik dari tahun 2020 s.d. 2024 dan terdapat lonjakan yang pesat dari awal tahun 2024. Berdasarkan Gambar 7, terlihat jelas bahwa data harga emas menunjukkan pola yang tidak teratur. Karakter data tersebut merupakan hal yang perlu diperhatikan untuk memilih model peramalan yang tepat.



Gambar 7. (a) Harga emas tahun 2010 s.d. 2024; (b) Tren harga emas tahun 2010 s.d. 2024.

Fokus dari penelitian ini adalah untuk membangun dan mengetahui performa model peramalan harga emas menggunakan LSTM, GRU, dan CNN. Masing-masing model dibuat dalam tiga *time step*, yaitu 5 (1 minggu), 30 (1 bulan), dan 260 (1 tahun). Parameter dari masing-masing model dapat dilihat pada Tabel 1. Fungsi optimasi yang digunakan adalah Adam dengan parameter *learning rate*-nya dapat dilihat pada Tabel 1. Nilai parameter dari model-model tersebut diperoleh melalui proses *bayessian optimization* yang dapat dilihat pada (Aslam et al., 2021).

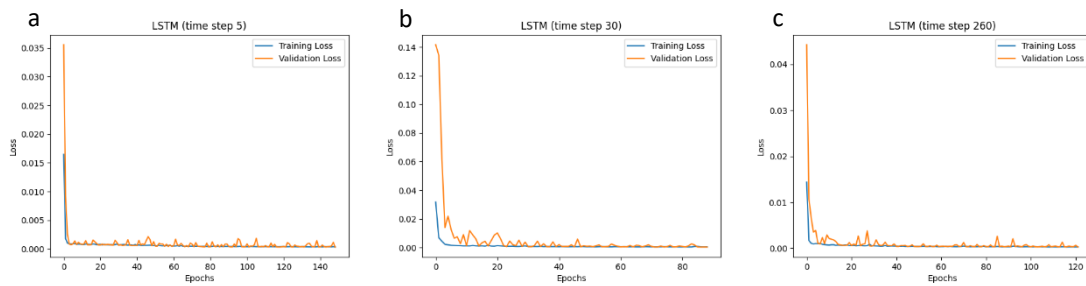
Tabel 1. Nilai parameter model.

<i>Time step</i>	Model	Layer	Dense	Dropout rate	Learning rate	Batch size
5	LSTM	2	169	0.16068	0.00071	38
5	GRU	1	103	0.18421	0.00626	105
5	CNN	1	111	0.12501	0.00178	16
30	LSTM	4	197	0.11057	0.00257	81
30	GRU	2	60	0.25701	0.00125	23
30	CNN	1	71	0.10430	0.00314	112
260	LSTM	2	151	0.13305	0.00147	37
260	GRU	2	123	0.16627	0.00239	80
260	CNN	1	125	0.13725	0.00035	23

Model-model tersebut kemudian dilatih secara iteratif menggunakan data *training* dan diuji menggunakan data *testing*. Data *training* merupakan data dari tahun 2020 s.d. 2023 dan data testing merupakan data tahun 2024. Banyaknya iterasi pelatihan dinamakan *epochs*. Lalu, performa model diperoleh menggunakan matriks evaluasi MSE, RMSE, MAE, dan MAPE.

3.1. Performa Model LSTM

Visualisasi pelatihan model atau kurva pelatihan dapat dilihat pada Gambar 8. Dapat dilihat bahwa untuk *time step* 5, model mencapai titik optimal pada *epochs* 149. Untuk *time step* 30, model mencapai titik optimal pada *epochs* 89. Dan untuk *time step* 260, model mencapai titik optimal pada *epochs* 122. Pada semua *time step*, model LSTM cukup cepat untuk konvergen. Meskipun demikian, ada sedikit keterlambatan pada *time step* 30 pada awal iterasi.

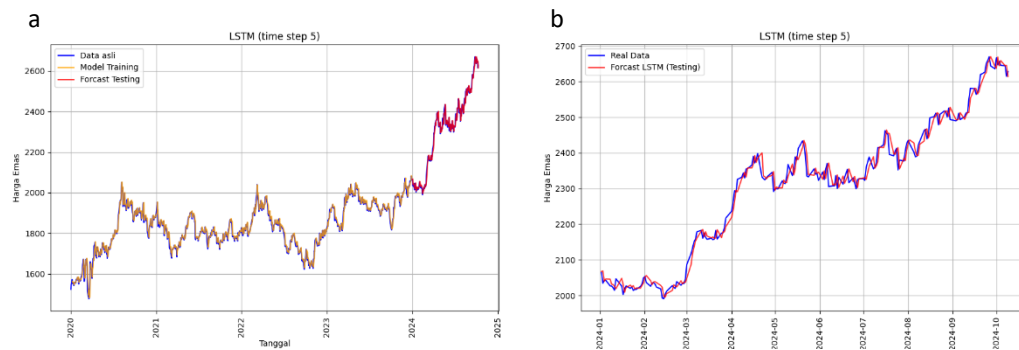


Gambar 8. (a) Kurva pelatihan untuk *time step* 5; (b) Kurva pelatihan untuk *time step* 30; (c) Kurva pelatihan untuk *time step* 260.

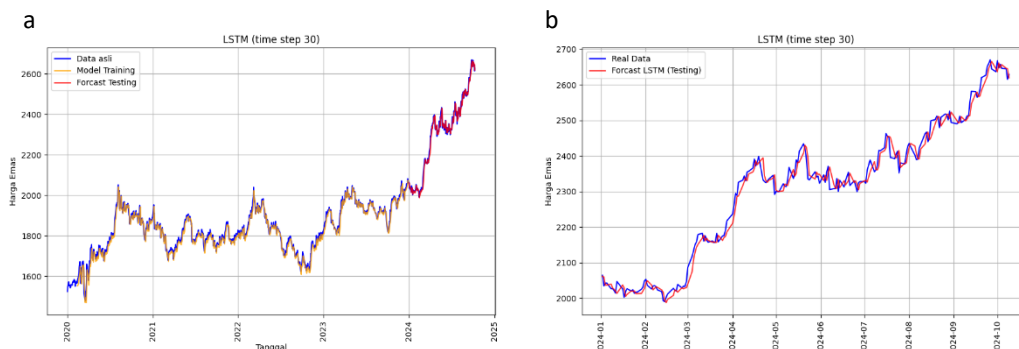
Performa model LSTM dapat dilihat pada Tabel 2. Terlihat bahwa pada *time step* 5, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 438.06, 16.07, 20.98, dan 0.00692. Pada *time step* 30, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 459.60, 16.67, 21.44, dan 0.00719. Pada *time step* 260, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 437.20, 16.05, 20.91, dan 0.00692. Visualisasi akurasi model LSTM untuk *time step* 5, 30, dan 260 secara berturut-turut data dapat dilihat pada Gambar 9, Gambar 10, dan Gambar 11.

Tabel 2. Performa model LSTM

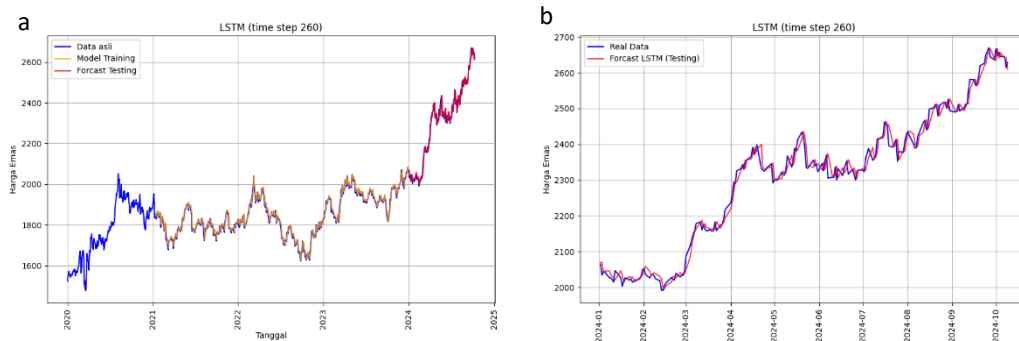
<i>Time step</i>	MSE		MAE		RMSE		MAPE	
	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
5	368.59	438.06	13.65	16.07	19.20	20.93	0.00752	0.00692
30	501.21	459.60	17.14	16.67	22.39	21.44	0.00941	0.00719
260	302.46	437.20	12.98	16.05	17.39	20.91	0.00705	0.00692



Gambar 9. (a) Akurasi model LSTM pada data *training* dan *testing*; (b) Akurasi model LSTM pada data *testing*.



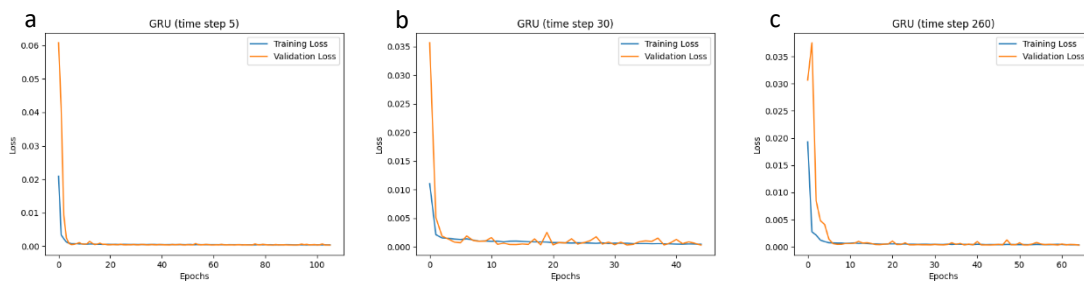
Gambar 10. (a) Akurasi model LSTM pada data *training* dan *testing*; (b) Akurasi model LSTM pada data *testing*.



Gambar 11. (a) Akurasi model LSTM pada data *training* dan *testing*; (b) Akurasi model LSTM pada data *testing*.

3.2. Performa Model GRU

Visualisasi pelatihan model atau kurva pelatihan dapat dilihat pada Gambar 12. Dapat dilihat bahwa untuk *time step* 5, model mencapai titik optimal pada *epochs* 106. Untuk *time step* 30, model mencapai titik optimal pada *epochs* 45. Dan untuk *time step* 260, model mencapai titik optimal pada *epochs* 65. Pada semua *time step*, model GRU sangat cepat untuk konvergen. Meskipun demikian, ada sedikit keteringgalan pada *time step* 260 pada awal iterasi.

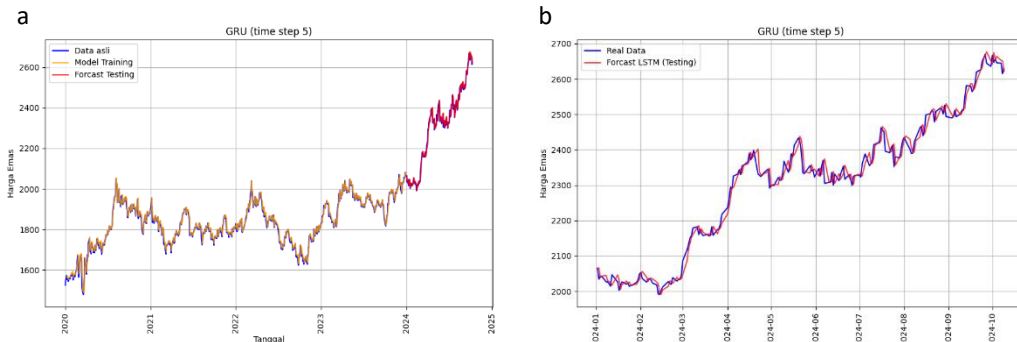


Gambar 12. (a) Kurva pelatihan untuk *time step* 5; (b) Kurva pelatihan untuk *time step* 30; (c) Kurva pelatihan untuk *time step* 260.

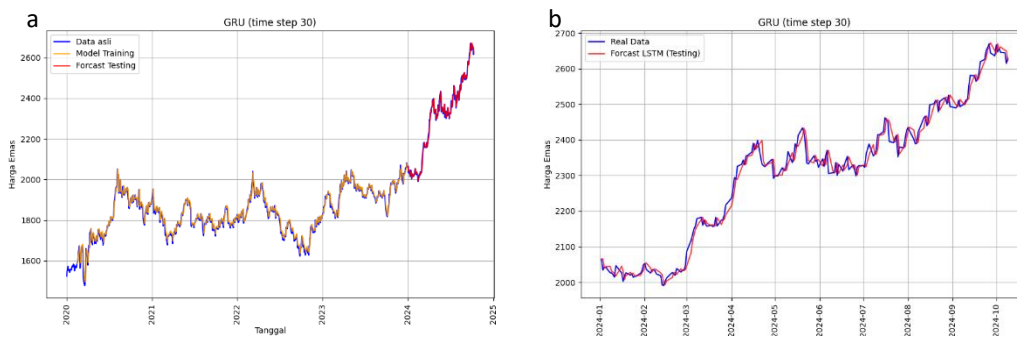
Performa model GRU dapat dilihat pada Tabel 3. Terlihat bahwa pada *time step* 5, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 442.01, 15.89, 21.02, dan 0.00685. Pada *time step* 30, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 440.96, 16.10, 21.00, dan 0.00693. Pada *time step* 260, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 443.97, 16.13, 21.07, dan 0.00695. Visualisasi akurasi model GRU untuk *time step* 5, 30, dan 260 secara berturut-turut data dapat dilihat pada Gambar 13, Gambar 14, dan Gambar 15.

Tabel 3. Performa model GRU.

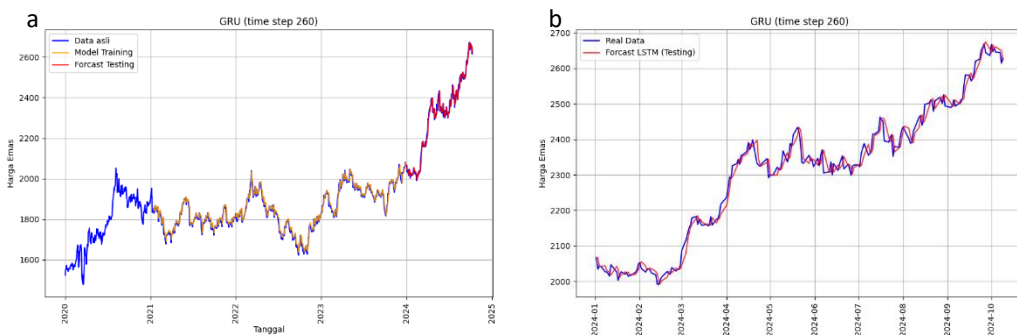
<i>Time step</i>	MSE		MAE		RMSE		MAPE	
	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
5	370.62	442.01	13.82	15.89	19.25	21.02	0.00762	0.00685
30	387.39	440.96	14.20	16.10	19.68	21.00	0.00782	0.00693
260	310.21	443.97	13.22	16.13	17.61	21.07	0.00720	0.00695



Gambar 13. (a) Akurasi model GRU pada data *training* dan *testing*; (b) Akurasi model GRU pada data *testing*.



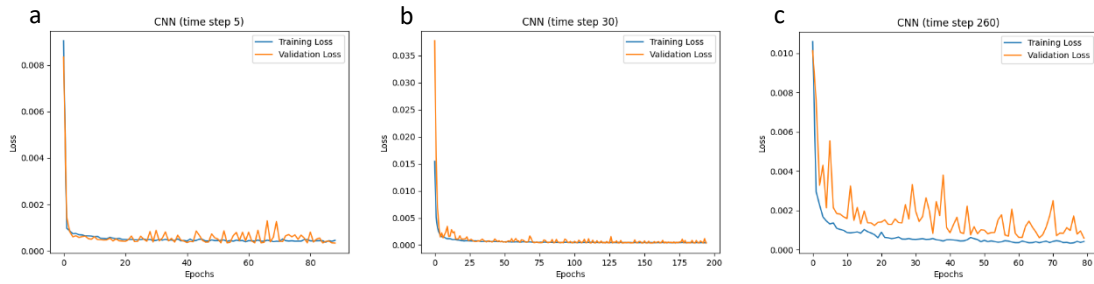
Gambar 14. (a) Akurasi model GRU pada data *training* dan *testing*; (b) Akurasi model GRU pada data *testing*.



Gambar 15. (a) Akurasi model GRU pada data *training* dan *testing*; (b) Akurasi model GRU pada data *testing*.

3.3. Performa Model CNN

Visualisasi pelatihan model atau kurva pelatihan dapat dilihat pada Gambar 16. Dapat dilihat bahwa untuk *time step* 5, model mencapai titik optimal pada *epochs* 89. Untuk *time step* 30, model mencapai titik optimal pada *epochs* 195. Dan untuk *time step* 260, model tidak dapat mencapai titik optimal dengan *epochs* 80. Pada *time step* 5, pelatihan model tidak konvergen terutama untuk *validation loss*-nya. Pada *time step* 30, pelatihan model cukup cepat untuk konvergen. Sedangkan pada *time step* 260, pelatihan model CNN konvergen tetapi membutuhkan waktu yang lebih lama.

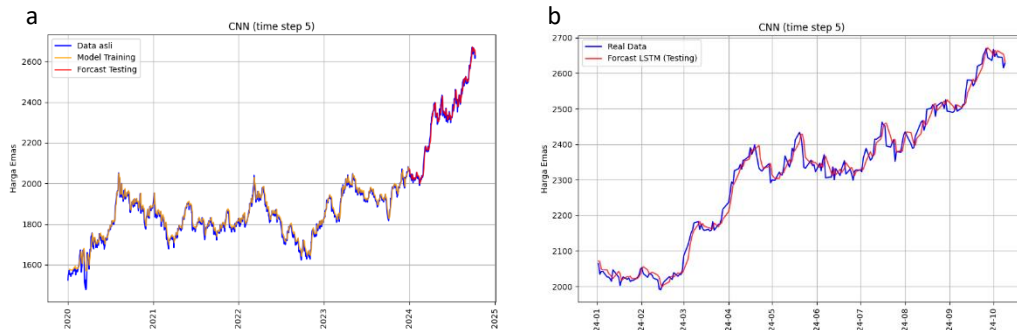


Gambar 16. (a) Kurva pelatihan untuk *time step* 5; (b) Kurva pelatihan untuk *time step* 30; (c) Kurva pelatihan untuk *time step* 260.

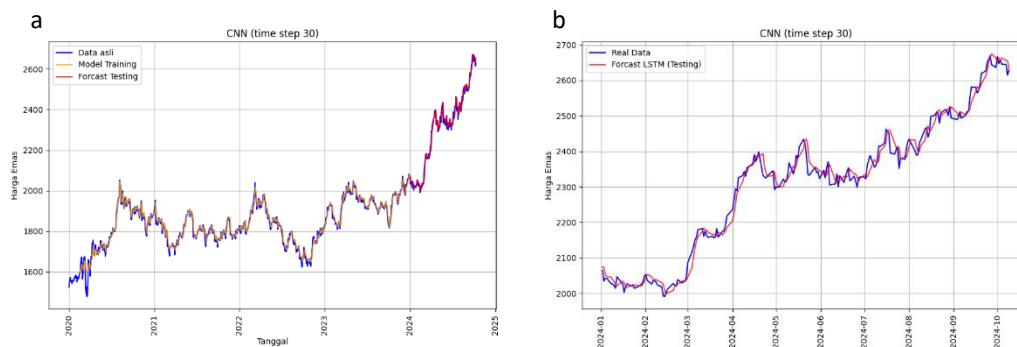
Performa model CNN dapat dilihat pada Tabel 4. Terlihat bahwa pada *time step* 5, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 469.65, 16.87, 21.67, dan 0.00728. Pada *time step* 30, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 486.57, 17.14, 22.06, dan 0.00739. Pada *time step* 260, nilai MSE, MAE, RMSE, dan MAPE secara berturut-turut sebesar 831.30, 23.18, 27.29, dan 0.01016. Visualisasi akurasi model CNN untuk *time step* 5, 30, dan 260 secara berturut-turut data dapat dilihat pada Gambar 17, Gambar 18, dan Gambar 19.

Tabel 4. Performa model CNN

<i>Time step</i>	MSE		MAE		RMSE		MAPE	
	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
5	476.53	469.65	15.97	16.87	21.83	21.67	0.00887	0.00728
30	460.73	486.57	15.11	17.14	21.46	22.06	0.00835	0.00739
260	542.99	831.30	18.60	23.18	23.30	28.83	0.01007	0.01016



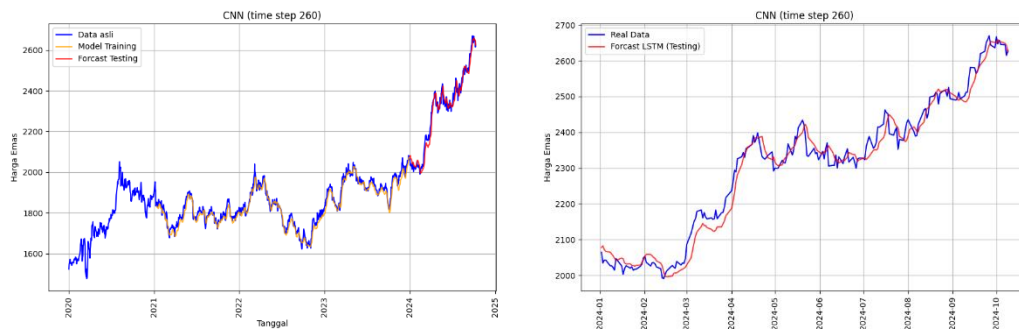
Gambar 17. (a) Akurasi model CNN pada data *training* dan *testing*; (b) Akurasi model CNN pada data *testing*.



Gambar 18. (a) Akurasi model CNN pada data *training* dan *testing*; (b) Akurasi model CNN pada data *testing*.

a

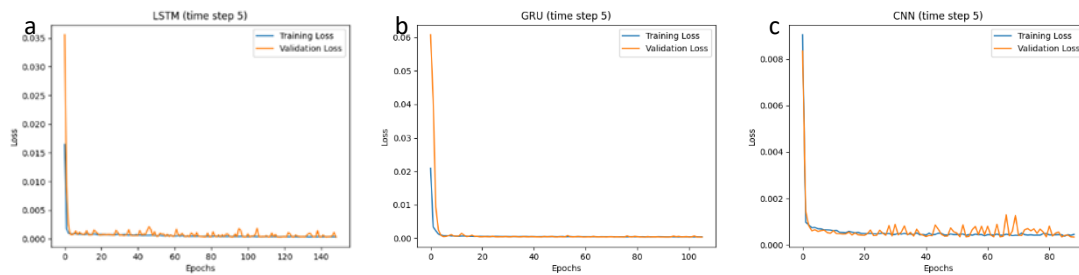
b



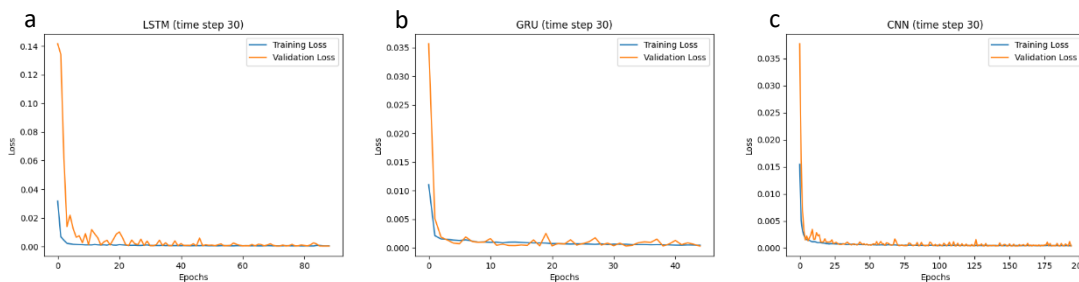
Gambar 19. (a) Akurasi model CNN pada data *training* dan *testing*; (b) Akurasi model CNN pada data *testing*.

3.4. Perbandingan Performa Model LSTM, GRU, dan CNN

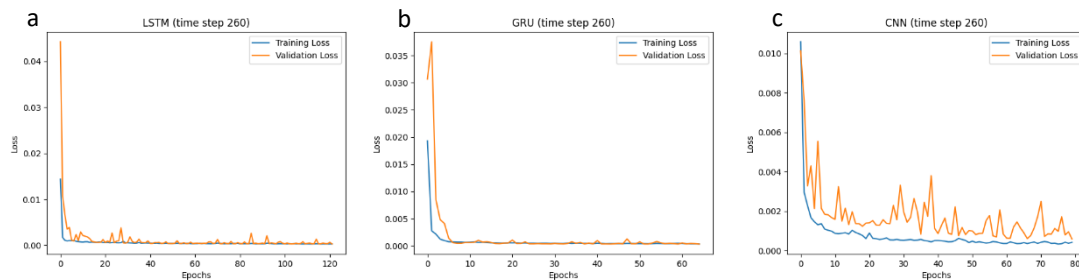
Perbandingan efisiensi pelatihan model berdasarkan *time step* 5, 30, dan 260 secara berturut-turut dapat dilihat pada Gambar 20, Gambar 21, dan Gambar 22. Terlihat bahwa model GRU adalah model yang paling cepat untuk optimal pada semua *time step*. Model GRU hanya memerlukan epochs 106, 45, dan 65 untuk mencapai titik optimal secara berturut-turut pada *time step* 5, 30, dan 260. Sehingga, model GRU merupakan model yang paling efisien pada semua *time step*. Disisi lain, model CNN merupakan model yang paling tidak efisien pada semua *time step*.



Gambar 20. (a) Kurva pelatihan LSTM; (b) Kurva pelatihan GRU; (c) Kurva pelatihan CNN.



Gambar 21. (a) Kurva pelatihan LSTM; (b) Kurva pelatihan GRU; (c) Kurva pelatihan CNN.



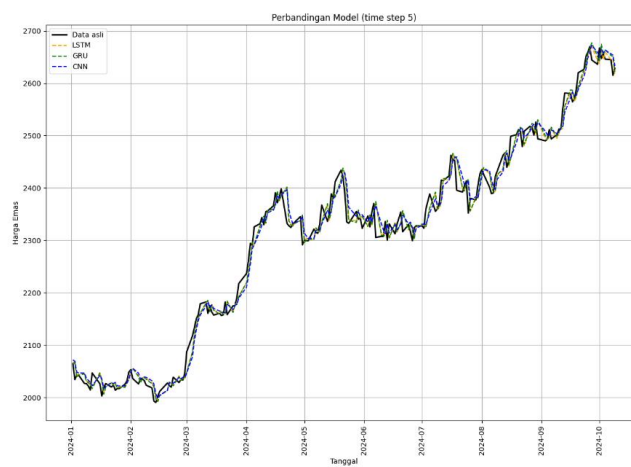
Gambar 22. (a) Kurva pelatihan LSTM; (b) Kurva pelatihan GRU; (c) Kurva pelatihan CNN.

Akurasi model dibandingkan berdasarkan nilai MAPE-nya. Perbandingan akurasi model berdasarkan *time step* 5, 30, dan 260 dapat dilihat pada Tabel 5. Pada *time step* 5 dan 30, model GRU merupakan

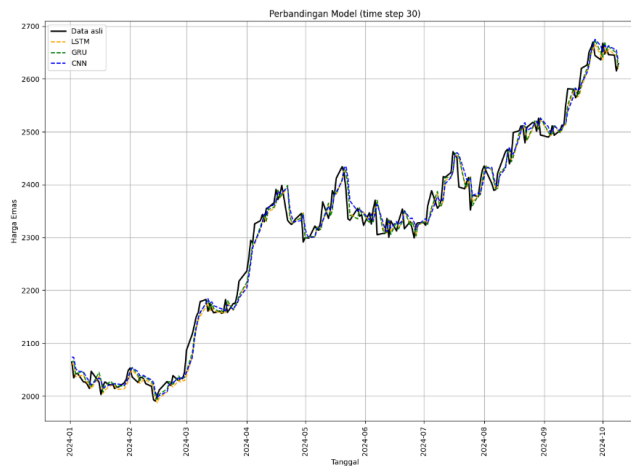
model yang paling akurat dengan nilai MAPE secara berturut-turut 0.00685 dan 0.00693. Sedangkan pada *time step* 260, model LSTM merupakan model paling akurat dengan nilai MAPE 0.00692. Perbandingan akurasi ini juga dapat dilihat pada Gambar 23, Gambar 24, dan Gambar 25 secara berturut-turut untuk *time step* 5, 30, dan 260.

Tabel 5. Perbandingan performa model.

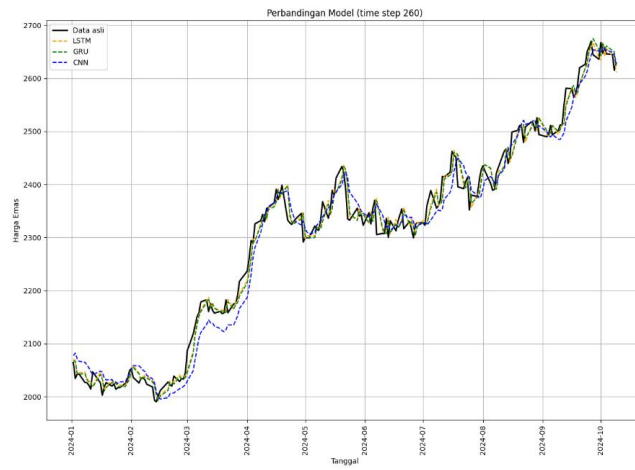
Model	Time step 5		Time step 30		Time step 260	
	Training	Testing	Training	Testing	Training	Testing
LSTM	0.00692	0.00719	0.00692	0.00692	0.00719	0.00692
GRU	0.00685	0.00693	0.00695	0.00685	0.00693	0.00695
CNN	0.00728	0.00739	0.00972	0.00728	0.00739	0.00972



Gambar 23. Perbandingan model untuk *time step* 5.



Gambar 24. Perbandingan model untuk *time step* 30.



Gambar 25. Perbandingan model untuk *time step* 260.

4. Simpulan

Berdasarkan hasil analisis, diperoleh beberapa kesimpulan terkait dengan data harga emas, performa model, dan perbandingan model. Pertama, harga emas pada tanggal 1 Januari 2020 s.d. 9 Oktober 2024 menunjukkan tren naik dan terdapat lonjakan harga mulai dari awal tahun 2024.

Kedua, performa model LSTM. Pelatihan model LSTM dinilai cukup efisien dan berhasil mencapai titik optimal pada semua *time step* dengan *epochs* sebanyak 149, 89, dan 122 berturut-turut pada *time step* 5, 30, dan 260. Kemudian, model LSTM sangat akurat dan cukup konsisten pada semua *time step* dengan nilai MAPE sebesar 0.00692, 0.00719, dan 0.00692 berturut-turut pada *time step* 5, 30, dan 260. Sehingga dapat disimpulkan bahwa model LSTM cukup efisien dan cukup akurat untuk memprediksi harga emas.

Ketiga, performa model GRU. Pelatihan model GRU dinilai sangat efisien dan berhasil mencapai titik optimal pada semua *time step* dengan *epochs* sebanyak 106, 45, dan 65 berturut-turut pada *time step* 5, 30, dan 260. Kemudian, model GRU sangat akurat dan sangat konsisten pada semua *time step* dengan nilai MAPE sebesar 0.00685, 0.00693, dan 0.00695 berturut-turut pada *time step* 5, 30, dan 260. Sehingga dapat disimpulkan bahwa model GRU sangat efisien dan sangat akurat untuk memprediksi harga emas.

Ketiga, performa model CNN. Pelatihan model CNN dinilai kurang efisien dan hanya berhasil mencapai titik optimal pada *time step* 5 dan 30 dengan *epochs* berturut-turut sebanyak 89 dan 195. Kemudian, model CNN kurang akurat dengan nilai MAPE sebesar 0.00728, 0.00739, dan 0.01016 berturut-turut pada *time step* 5, 30, dan 260. Sehingga dapat disimpulkan bahwa model CNN kurang efisien dan kurang akurat untuk memprediksi harga emas.

Keempat, perbandingan performa model LSTM, GRU, dan CNN. Berdasarkan pelatihannya, model CNN merupakan model yang paling cepat untuk mencapai titik optimal pada *time step* 5 dengan *epochs* 89. Sedangkan untuk *time step* 30 dan 260, model GRU merupakan model yang paling cepat untuk mencapai titik optimal dengan *epochs* berturut-turut 106 dan 45. Berdasarkan akurasinya, untuk *time step* 5 dan 30, model GRU merupakan model yang paling akurat dibandingkan dengan model yang lain dengan nilai MAPE berturut-turut 0.00685 dan 0.00693. Sedangkan untuk *time step* 260, model LSTM menunjukkan nilai MAPE tertinggi sebesar 0.00692 yang hanya selisih 3 dari nilai MAPE model GRU pada *time step* yang sama, yaitu sebesar 0.00695.

Berdasarkan analisis diperoleh hasil dari penelitian menunjukkan bahwa proses pelatihan model GRU lebih efisien dibandingkan model lainnya pada semua *time step*. Berdasarkan akurasinya, pada *time step* 5 dan 30, model GRU menunjukkan akurasi yang lebih baik dibandingkan model lainnya. Sedangkan pada *time step* 260, tidak ada perbedaan akurasi yang signifikan antara model LSTM dengan GRU yang mana lebih baik dibandingkan dengan model CNN. Sehingga, dapat disimpulkan bahwa model GRU adalah model yang paling efisien dan akurat untuk peramalan harga emas pada *time step* 5, 30, dan 260.

Daftar Pustaka

- Aggarwal, C. C. (2018). *Neural networks and deep learning : a textbook / by Charu C. Aggarwal. Neural Networks and Deep Learning : A Textbook*, 497. <https://play.google.com/store/books/details?id=achqDwAAQBAJ>
- Ali, R., Mangla, I. U., Rehman, R. U., Xue, W., Naseem, M. A., & Ahmad, M. I. (2020). Exchange rate, gold price, and stock market nexus: A quantile regression approach. *Risks*, 8(3), 1–16. <https://doi.org/10.3390/risks8030086>
- Aslam, M., Lee, S. J., Khang, S. H., & Hong, S. (2021). Two-Stage Attention over LSTM with Bayesian Optimization for Day-Ahead Solar Power Forecasting. *IEEE Access*, 9, 107387–107398. <https://doi.org/10.1109/ACCESS.2021.3100105>
- Casolaro, A., Capone, V., Iannuzzo, G., & Camastra, F. (2023). Deep Learning for Time Series Forecasting: Advances and Open Problems. *Information (Switzerland)*, 14(11). <https://doi.org/10.3390/info14110598>
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, 1–24. <https://doi.org/10.7717/PEERJ-CS.623>
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. <https://doi.org/10.3115/v1/d14-1179>
- Courville, I. G. and Y. B. and A. (2016). Deep learning 简介 一、什么是 Deep Learning ? . *Nature*, 29(7553), 1–73. <http://deeplearning.net/>
- Essai Ali, M. H., Abdel-Raman, A. B., & Badry, E. A. (2022). Developing Novel Activation Functions Based Deep Learning LSTM for Classification. *IEEE Access*, 10(September), 97259–97275. <https://doi.org/10.1109/ACCESS.2022.3205774>
- Fawaz, H. I. (2020). Deep learning for time series classification. <http://arxiv.org/abs/2010.00567>
- Hafiyya, N., Virgantari, F., & Widyastiti, M. (2022). Implementasi Metode Fuzzy Time Series Pada Peramalan Harga Emas Di Indonesia. *Interval: Jurnal Ilmiah Matematika*, 2(2), 94–103. <https://doi.org/10.33751/interval.v2i2.6517>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Horn, R. A., & Johnson, C. R. (2012). *Matrix Analysis*. In *Matrix Analysis*. <https://doi.org/10.1017/cbo9781139020411>
- Ivakhnenko, A. G., Lapa, V. G., & Valentin Grigor'evich Lapa. (1967). *Cybernetics_and_Forecasting_Techniques*. American Elsevier Publishing Company.
- Jabeur, S. Ben, Mefteh-Wali, S., & Viviani, J. L. (2024). Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Annals of Operations Research*, 334(1–3), 679–699. <https://doi.org/10.1007/s10479-021-04187-w>
- Jason, B. (2018). Deep Learning for Time Series Forecasting (Top Reference). *MI*, 1(1), 1–50.
- Kantar, O., & Kilimci, Z. H. (2023). Deep learning based hybrid gold index (XAU/USD) direction forecast model. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38(2), 1117–1128. <https://doi.org/10.17341/gazimmfd.888456>
- Liu, X., & Wang, W. (2024). Deep Time Series Forecasting Models: A Comprehensive Survey. *Mathematics*, 12(10). <https://doi.org/10.3390/math12101504>
- Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32(23), 17351–17360. <https://doi.org/10.1007/s00521-020-04867-x>

- Murthy, G. S. ., Allu, S. R., Andhavarapu, B., Bgadi, M., & Belusonti, M. (2020). Text based Sentiment Analysis using Long Short Term Memory (LSTM). *International Journal of Engineering Research & Technology (IJERT)*, 9(05), 299–303.
- Nurhambali, M. R., Angraini, Y., & Fitrianto, A. (2024). Implementation of Long Short-Term Memory for Gold Prices Forecasting. *Malaysian Journal of Mathematical Sciences*, 18(2), 399–422. <https://doi.org/10.47836/mjms.18.2.11>
- Primananda, S. B., & Isa, S. M. (2021). Forecasting Gold Price in Rupiah using Multivariate Analysis with LSTM and GRU Neural Networks. *Advances in Science, Technology and Engineering Systems Journal*, 6(2), 245–253. <https://doi.org/10.25046/aj060227>
- Senanayake, S., Pradhan, B., Alamri, A., & Park, H. J. (2022). A new application of deep neural network (LSTM) and RUSLE models in soil erosion prediction. *Science of the Total Environment*, 845(May), 157220. <https://doi.org/10.1016/j.scitotenv.2022.157220>
- Surendra, J., Rajyalakshmi, K., Apparao, B. V., Charankumar, G., & Dasore, A. (2021). Forecast and trend analysis of gold prices in India using auto regressive integrated moving average model. *Journal of Mathematical and Computational Science*, 11(2), 1166–1175. <https://doi.org/10.28919/jmcs/5250>
- Tholib, A., Agusmawati, N. K., & Khoiriyah, F. (2023). PREDIKSI HARGA EMAS MENGGUNAKAN METODE LSTM DAN GRU. *Jurnal Informatika Dan Teknik Elektro Terapan*. <https://doi.org/10.23960/jitet.v11i3.3250>
- Yann, L., & Yoshua, B. (1995). Convolutional Networks for Images, Speech, and Time-Series. *The Handbook of Brain Theory and Neural Networks*, 255–258. https://www.researchgate.net/profile/Yann_Lecun/publication/2453996_Convolutional_Networks_for_Images_Speech_and_Time-Series/links/0deec519dfa2325502000000.pdf%0Ahttps://www.researchgate.net/profile/Yann_Lecun/publication/2453996_Convolutional_Networks_fo