

Perbandingan Algoritma *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU) untuk Prediksi Harga Emas Dunia

Marwondo, Taufik Hidayah

Informatika, Universitas Informatika dan Bisnis Indonesia
Email: marwondo@unibi.ac.id, taufik.h@student.unpad.ac.id

Abstrak

Harga emas dunia dapat berubah-ubah secara fluktuatif. Berbagai algoritma prediksi bisa diterapkan untuk mendapatkan akurasi prediksi dengan tepat, termasuk algoritma *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU). Di antara kedua algoritma tersebut, manakah algoritma yang memiliki performa terbaik dalam memprediksi harga emas dunia? *Dataset* yang digunakan berasal dari *website The London Bullion Market Association* (LBMA) yang berbentuk *time series* dari tahun 1968 sampai 2022. Parameter yang digunakan untuk perbandingan yaitu *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Error* (MAE). Setelah dilakukan proses Training dan Evaluasi dengan menguji data test dihasilkan sebuah analisa bahwa algoritma GRU memberikan performa yang lebih baik dibandingkan dengan LSTM karena memiliki nilai *loss/error* lebih rendah dalam memprediksi harga emas dunia.

Kata Kunci: LSTM, GRU, harga emas dunia.

Abstract

World gold prices can fluctuate fluctuating. Various prediction algorithms can be applied to obtain precise prediction accuracy, including the Long Short-Term Memory (LSTM) algorithm, and the Gated Recurrent Unit (GRU). Between the two algorithms, which algorithm has the best performance in predicting world gold prices? The dataset used comes from the website of The London Bullion Market Association (LBMA) in the form of a time series from 1968 to 2022. The parameters used for comparison are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error. (MAE). After carrying out the Training and Evaluation process by testing the test data, an analysis is produced that the GRU algorithm provides better performance compared to LSTM because it has a lower loss/error value in predicting world gold prices.

Keywords: LSTM, GRU, world gold prices.

1 PENDAHULUAN

Investasi merupakan suatu istilah ekonomi yakni penanaman modal dalam jangka panjang yang bertujuan untuk memperoleh keuntungan. Pada saat ini banyak sekali instrumen investasi yang dapat ditemukan dimulai deposito, saham, valas, properti, emas, bitcoin dan masih banyak lagi yang lainnya. Salah satu bentuk investasi yang

banyak diminati adalah emas. Penelitian yang dilakukan *World Gold Council* (WGC) pada tahun 2019 menunjukkan bahwa emas adalah pilihan utama investasi ketiga yang paling banyak dibeli secara konsisten, dengan 46% investor ritel global memilih produk emas, tepat di belakang rekening tabungan (78%) dan asuransi jiwa (54%).

Prediksi harga emas diperlukan supaya para investor mendapatkan laba sesuai dengan perencanaan atau untuk meminimalkan resiko. Prediksi yang akurat dapat membantu keputusan yang tepat. Dalam suatu investasi apapun bentuk instrumen dan jenis investasinya akan selalu ada resiko kerugian dalam menjalankannya. Resiko dapat diminimalisir dengan prediksi yang akurat.

Prediksi adalah suatu proses untuk memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan data-data atau informasi yang didapatkan dari masa lalu sampai sekarang yang, tujuannya agar selisih antara sesuatu yang terjadi tersebut dengan hasil perkiraan dapat diperkecil. Secara keilmuan, prediksi dapat dilakukan dengan berbagai cara, termasuk penggunaan algoritma. Berbagai algoritma dikembangkan untuk memberikan akurasi yang lebih baik.

Emas diperdagangkan setiap hari dan memiliki harga yang fluktuatif. Harga emas mengalami kenaikan dan penurunan juga setiap hari, sehingga harga emas termasuk jenis data *time series*. Beberapa penelitian dilakukan untuk menerapkan algoritma dalam memprediksi harga emas ini dan kebanyakan dari mereka menggunakan algoritma *Recurrent Neural Network* (RNN) seperti penelitian yang sudah dilakukan oleh Ilham Alkahfi dan Chiuloto (2021) dengan judul “Penerapan Model Gated Recurrent Unit Pada Masa Pandemi Covid-19 Dalam Melakukan Prediksi Harga Emas Dengan Menggunakan Model Pengukuran *Mean Square Error*” pada tahun 2021. Kemudian untuk RNN Long variasi *Short-Term Memory* (LSTM) sudah dilakukan oleh Prasetyo dkk. (2022) dengan judul “Prediksi Harga Emas Berdasarkan Data gold.org menggunakan Metode *Long Short Term Memory*” pada tahun 2022.

Selain itu, masih banyak penelitian-penelitian prediksi harga emas lainnya dengan menggunakan RNN baik itu variasi LSTM maupun GRU, namun dari penelitian-penelitian itu belum ada yang menggunakan keduanya dan membandingkan algoritma mana yang memberikan performa lebih baik dalam prediksi harga emas dunia.

Berdasarkan permasalahan diatas, penelitian kali ini akan melakukan prediksi harga emas dunia dengan menggunakan algoritma RNN variasi LSTM dan GRU kemudian akan dibandingkan antar keduanya variasi yang mana yang memberikan performa lebih baik dalam melakukan prediksi tersebut.

Batasan masalah agar masalah tidak meluas. Batasan untuk perbandingan algoritma untuk prediksi harga emas dunia dengan:

1. Data historis harga emas yang akan digunakan hanya berasal dari website *The London Bullion Market Association* (LBMA)
2. Data yang digunakan adalah data historis emas dari tahun 1968-2022 dengan jumlah data 13.838 data.
3. Hanya menggunakan algoritma LSTM dan GRU untuk melakukan prediksi

2 KAJIAN PUSTAKA

2.1 Prediksi

Prediksi adalah suatu proses yang sistematis untuk memperkirakan suatu hal di masa depan berdasarkan data-data atau informasi yang diperoleh dari masa lalu sampai saat ini.

Secara umum arti kata prediksi sendiri adalah tafsiran. Namun, dengan perkembangan ilmu pengetahuan yang melahirkan tehnik-tehnik tertentu, maka prediksi bukan lagi hanya sebatas tafsiran saja.

Metode atau teknik dalam prediksi merupakan cara memperkirakan apa yang akan terjadi pada masa depan secara sistematis dan pragmatis atas dasar data yang relevan pada masa yang lalu, sehingga dengan demikian metode prediksi diharapkan dapat memberikan objektivitas yang lebih besar.

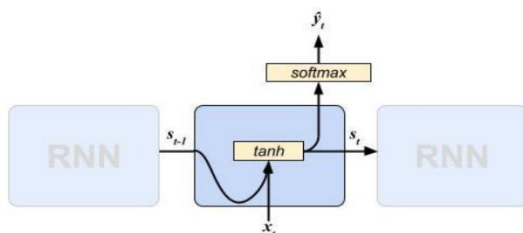
Metode dalam prediksi dapat memberikan cara pengerjaan yang teratur dan terarah, dengan demikian dapat dimungkinkannya penggunaan teknik penganalisaan yang lebih maju. Secara umum ada dua jenis prediksi yaitu:

1. Prediksi kualitatif adalah prediksi yang didasarkan atas pendapat suatu pihak, dan data yang diperoleh tidak bisa direpresentasikan secara tegas menjadi suatu angka atau nilai karena data bisa berupa deskripsi.
2. Prediksi kuantitatif merupakan kebalikan dari kualitatif, prediksi kuantitatif didasarkan atas data kuantitatif masa lalu (data historis) dan dapat dibuat dalam bentuk angka yang biasa disebut sebagai data *time series*.

2.2 Recurrent Neural Network (RNN)

RNN merupakan salah satu jenis arsitektur Jaringan Saraf Tiruan (JST) yang mana input dalam RNN ini diproses secara iteratif atau berulang. RNN ini juga termasuk kedalam kategori *Deep Learning* dikarenakan jenis kecerdasan buatan yang membutuhkan banyak lapisan atau *hidden layer* untuk memproses data inputan.

Ciri dari RNN ini sendiri adalah data yang digunakan sebagai inputan adalah data sekuensial atau linier. Sebuah data bisa dikatakan sekuensial apabila data tersebut *time series* atau apabila sampel data yang akan dibuat menjadi inputan tersebut diambil secara berurutan.



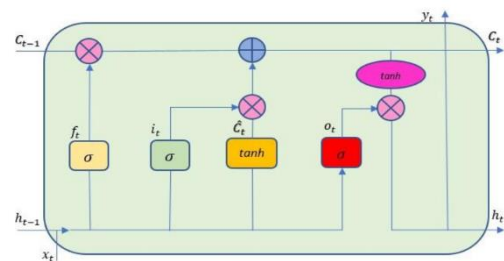
Gambar 1. Arsitektur Sederhana RNN

Bisa dilihat Gambar 1 dari arsitektur RNN tersebut bahwa output dari setiap hidden layer akan mengalami perulangan ke dirinya sendiri hal ini bertujuan untuk mendapatkan hasil output yang terbaik. Sehingga pada prinsipnya, RNN ini memiliki memori peristiwa yang telah terjadi di masa lampau dan itu akan digunakan kembali untuk memprediksi sesuatu di masa depan.

2.3 Long Short-Term Memory (LSTM)

RNN yang pertama kali diperkenalkan itu memiliki kelemahan, yakni pada metode ini sulit diterapkan apabila data yang akan menjadi inputan memiliki urutan yang panjang. Sehingga, untuk mengatasi masalah tersebut muncullah variasi LSTM yang merupakan pengembangan dari RNN sendiri untuk mengatasi masalah ketergantungan jangka panjang tersebut dengan menyediakan *cell state* memori ke *hidden layer*.

LSTM ini bisa mempelajari pola panjang dari data yang sekuensial, yang mana ini berguna untuk mencegah situasi gradien menghilang. Dikarenakan LSTM ini merupakan pengembangan dari RNN, maka LSTM ini masih memiliki konsep yang sama dengan RNN, bedanya terletak di isi dari selnya.



Gambar 2. Arsitektur LSTM

Ada beberapa langkah atau tahapan pada LSTM untuk memproses inputan datanya yaitu:

1. *Forget Gate* (f_t), pada tahap ini informasi yang kurang dibutuhkan atau tidak terlalu memiliki makna terhadap kasus yang diolah akan dihilangkan menggunakan fungsi sigmoid.

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

2. *Input Gate* (i_t), pada proses ini akan memilah dan menentukan informasi tertentu yang akan diperbarui ke bagian cell state dengan menggunakan fungsi aktivasi sigmoid. Menggunakan persamaan

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c).$$

3. *Cell State* (c_t), proses untuk memperbarui cell state lama c_{t-1} menjadi cell state baru dengan persamaan c_t .

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t.$$

4. *Output Gate* (o_t), proses menjalankan sigmoid untuk menghasilkan nilai output pada hidden state dan menempatkan cell state pada tanh, kemudian menghasilkan nilai output sigmoid dan nilai output tanh kedua hasil aktivasi tersebut dilakukan perkalian sebelum menuju langkah selanjutnya.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o),$$

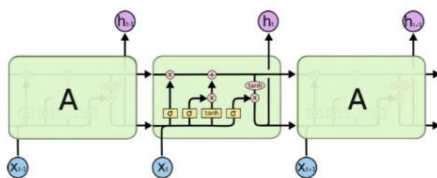
$$h_t = o_t * \tanh(c_t).$$

2.4 Gated Recurrent Unit (GRU)

Sama halnya dengan LSTM, GRU juga merupakan pengembangan dari RNN yang bertujuan untuk menghasilkan setiap recurrent unit agar bisa menangkap hubungan (dependensi) pada skala waktu yang berbeda-beda secara adaptif.

Arsitektur GRU sendiri hampir sama dengan LSTM dalam GRU juga menggunakan sistem gerbang atau *gate*, namun arsitektur GRU lebih sederhana dibandingkan dengan LSTM.

LSTM menggunakan *cell state*, tetapi GRU tidak menggunakan cell state, namun memanfaatkan hidden state untuk menyimpan informasi. *Reset gate* dalam GRU menentukan informasi baru harus dilupakan atau tidak, sedangkan *update gate* untuk mengingat.



Gambar 3. Arsitektur GRU

Hampir sama dengan LSTM, GRU juga mempunyai tiga jenis gates antara lain ialah

forget gate, *input gate*, serta *output gate* dengan persamaan seperti berikut:

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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c),$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t,$$

$$o_t = (W_o \cdot [h_{t-1}, x_t] + b_o)h_t$$

$$= o_t * \tanh(C_t).$$

2.5 Harga Emas Dunia

Dimulai pada tahun 1968, harga emas yang dijadikan patokan oleh seluruh dunia adalah harga emas yang berdasarkan standar pasar emas London. Digunakan secara internasional, harga emas dihitung dalam satuan *troy ounce* yang jumlahnya *troy ounce* = 31,1035 gram.

Penetapan harga emas saat itu (*on spot*) maupun penjualan yang akan datang (*forward*) ditentukan oleh “Tim Lima” yang mewakili para dealer emas serta bank-bank yang sangat berpengaruh dan paling besar di dunia, yakni N.M Rothschild (sebagai kepala tim), Societe General, Hongkong Shanghai Bank HSBC, Scotia Mocatta, dan Deutsche Bank.

Perubahan harga emas terjadi karena ada beberapa faktor yang mempengaruhi harga emas menurut Suharto (2013) diantaranya:

1. Kenaikan inflasi yang melebihi perkiraan, dimana ketika prediksi tingkat inflasi meleset dan malah melebihi yang diperkirakan, maka harga emas akan melonjak tinggi.
2. Terjadi kepanikan finansial seperti krisis moneter pada tahun 1998 dan 2008, ketika itu harga emas meroket tidak terkendali dikarenakan masyarakat enggan menyimpan uang kertas dan lebih memilih dalam bentuk emas.
3. Harga minyak naik secara signifikan, ketika harga minyak mentah dunia mengalami lonjakan signifikan, maka harga emas juga akan ikut naik meski dampaknya tidak terjadi seketika.
4. Demand terhadap emas, seperti halnya dalam hukum ekonomi jika demand

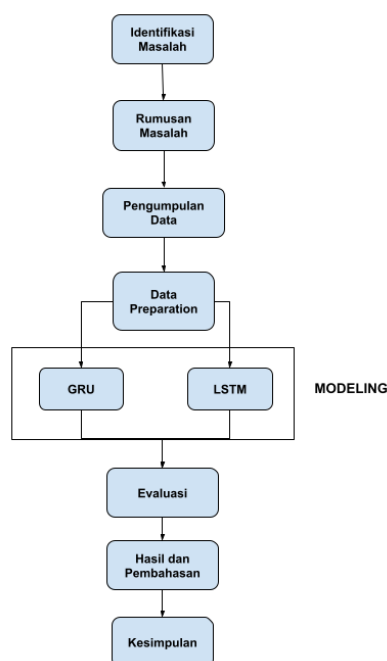
tinggi maka harga akan naik dan begitu juga sebaliknya.

5. Kondisi politik dunia, ketika ketegangan politik dunia yang terjadi membuat suhu politik dunia meminninggi sehingga mengakibatkan ketidakpastian ekonomi. Hal inilah yang menyebabkan kenaikan harga emas.

3 METODE PENELITIAN

3.1 Alur Penelitian

Penelitian ini adalah penelitian komparatif atau perbandingan, yang akan membandingkan algoritma mana yang lebih baik secara performa dalam melakukan prediksi harga emas dunia antara LSTM dan GRU. Secara garis besar, alur penelitian dapat dijelaskan melalui Gambar 5.



Gambar 4. Skema Penelitian.

3.2 Pengumpulan Data

Data yang dikumpulkan adalah data kuantitatif berasal dari situs yang menyediakan data historis harga emas dunia dari waktu ke waktu sejak 1968 yakni *website The London Bullion Market Association (LBMA)*.

Data yang diambil untuk penelitian ini dimulai dari 1 Januari 1968 sampai dengan 30 September 2022, dengan jumlah data sekitar 13.838 data yang terkumpul yang berisi harga open AM dan PM.

3.3 Data Preparation

Data preparation adalah sebuah proses penggabungan data, penataan dan lain-lain sehingga data bisa digunakan dalam prediksi harga emas dunia dengan menggunakan LSTM dan GRU.

3.4 Modeling

Pada tahap ini dilakukan implementasi, di mana dimulai merancang arsitektur algoritma kemudian diterapkan pada *tools* seperti python untuk melakukan training kemudian melakukan prediksi.

3.5 Evaluasi

Evaluasi adalah langkah terakhir, dimana hasil prediksi dari algoritma dibandingkan dengan data testing kemudian memvisualisasikan dan menghitung galat atau *error*. Parameter untuk menghitung loss/error menggunakan 3 parameter, yakni *Mean Square Error (MSE)*, *Mean Absolute Error (MAE)*, dan *Root Mean Square Error (RMSE)*.

4 HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Data historis harga emas telah dikumpulkan oleh LBMA dari waktu ke waktu, data tersebut bisa diunduh secara gratis di situs resminya. Dalam penelitian kali ini akan menggunakan data historis harga emas dari LBMA dari mulai 1 Januari 1968 sampai 30 September.

Data-data mentah yang sudah dikumpulkan oleh LBMA didapatkan dalam bentuk file Excel, kemudian dari file Excel tersebut untuk memudahkan dalam proses berikutnya maka diubahlah menjadi format DataFrame.

0	DATE	AM	PM
1	2022-09-30	1672.75	1671.75
2	2022-09-29	1646.6	1654.8
3	2022-09-28	1618.2	1652.15
4	2022-09-27	1632.6	1634.3
...
13833	1968-01-08	35.14	NaN
13834	1968-01-05	35.14	NaN
13835	1968-01-04	35.14	NaN
13836	1968-01-03	35.16	NaN
13837	1968-01-02	35.18	NaN

Gambar 5. Data historis harga emas dalam format Data Frame.

Empat data teratas adalah harga terbaru yang diambil yaitu dimulai 30 September sedangkan lima data terakhir adalah harga pada tanggal yang dimulai 1 Januari 1968, dengan jumlah data 13.838.

DATE berisi data *time series* untuk harga emas ini, AM dan PM merupakan harga open dalam USD yang diambil berdasarkan waktu AM (10:30 GMT) dan PM (15:00 GMT) Amerika.

4.2 Data Preparation

4.2.1 Data Cleaning (Pembersihan Data)

Data cleaning dilakukan untuk menghilangkan data-data yang tidak perlu, atau menghilangkan outlier pada data. Namun, pada penelitian kali ini hanya akan menghilangkan baris pertama pada data, dikarenakan itu bisa dikatakan kesalahan.

	DATE	AM	PM
2	2022-09-29	1646.6	1654.8
3	2022-09-28	1618.2	1652.15
4	2022-09-27	1632.6	1634.3
5	2022-09-26	1647	1643.35
6	2022-09-23	1661.45	1643.55

Gambar 6. Hasil setelah *data cleaning*.

Menghapus data baris pertama pada data ini dilakukan dengan sebuah bantuan method, yaitu `DataFrame.drop()`.

4.2.2 Scrubbing

Agar tidak terjadi error jika dilakukan pengoperasian matematika pada data, maka harus dipastikan terlebih dahulu tipe data apa saja yang terdapat pada data. Karena seperti yang kita ketahui jika data kita berupa angka namun bertipe data string maka data tersebut tidak bisa dilakukan operasi matematika karena dianggap sebagai string.

```
DATE    object
AM      object
PM      object
dtype: object
```

Gambar 7. Hasil pengecekan tipe data.

Dari hasil pengecekan digambar 8, terlihat bahwa semuanya bertipe data object. Jika hal ini dibiarkan maka kemungkinan akan terjadi error kedepannya, oleh karena itu untuk mengatasi hal tersebut, penulis menggunakan *method* dari *library* pandas untuk mengkonversi tipe data DATE menjadi tipe data *datetime* AM dan PM menjadi float karena mengandung koma didalamnya.

```
df['DATE'] = pd.to_datetime(df['DATE'])
df['AM'] = df['AM'].astype(float)
df['PM'] = df['PM'].astype(float)
df.dtypes

DATE    datetime64[ns]
AM      float64
PM      float64
dtype: object
```

Gambar 8. Hasil setelah dikonversi tipe data.

Dikarenakan terdapat dua harga dalam data, yaitu AM yaitu data yang diambil berdasarkan harga open di AM (10:30 GMT) dan PM (15:00 GMT). Oleh karena itu akan diambil angka rata-rata diantara keduanya.

	DATE	AVRG
2	2022-09-29	1650.700
3	2022-09-28	1635.175
4	2022-09-27	1633.450
5	2022-09-26	1645.175
6	2022-09-23	1652.500

Gambar 9. Hasil setelah diambil rata-rata.

Didalam *library* pandas sangat mudah untuk mengambil nilai rata-rata, yaitu dengan menggunakan *method* `pandas.mean(axis=1, skipna=True)`, parameter `axis=1` menandakan bahwa yang akan diambil nilai-rata nya adalah dari kolom yang dimana AM dan PM. Sedangkan, `skipna=True` untuk melewati jika ada field yang null atau kosong.

	DATE	AVRG
13837	1968-01-02	35.180
13836	1968-01-03	35.160
13835	1968-01-04	35.140
13834	1968-01-05	35.140
13833	1968-01-08	35.140
...
6	2022-09-23	1652.500
5	2022-09-26	1645.175
4	2022-09-27	1633.450
3	2022-09-28	1635.175
2	2022-09-29	1650.700

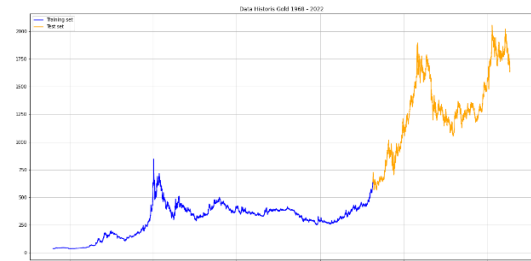
Gambar 10. Hasil pengurutan data.

Data-data historis emas dunia ini, dimulai dengan data yang paling terbaru. Jika tidak diurutkan maka hasil prediksi akan memprediksi terbalik yakni akan memprediksi harga untuk tahun 1968, sedangkan yang kita mau adalah memprediksi untuk masa depan.

Sehingga untuk itu dengan menggunakan *method* `pandas.sort_values(by='DATE')` untuk mengurutkan data dari yang terlama sampai terbaru seperti ditunjukkan di Gambar 10.

4.2.3 Pembagian Data

Data akan dibagi menjadi 70% untuk data training dan 30% untuk data testing. Sehingga dari total 13.838 data, 9.685 data akan dibuat menjadi data training dan sisanya 4.151 data untuk testing untuk semua model algoritma.



Gambar 11. Pembagian Data Training dan Testing

4.2.4 Normalisasi Data

Normalisasi data adalah sebuah teknik untuk membuat data memiliki rentang nilai yang sama tidak ada yang terlalu besar maupun terlalu kecil dan normalisasi ini memiliki tujuan menghindari informasi yang berlebihan dalam data dan memperbaharui anomaly (Bahmani, Naghibzadeh, dan Bahmani, 2014).

Banyak teknik untuk normalisasi namun pada penelitian kali ini, penulis akan menggunakan Min-Max, untuk cara kerjanya sangat sederhana yakni setiap nilai pada sebuah fitur dikurangi dengan nilai minimum fitur tersebut, kemudian dibagi dengan rentang nilai atau nilai maksimum dikurangi nilai minimum dari fitur tersebut.

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$$

Sehingga nilai baru yang dihasilkan akan bernilai antara 0 sampai 1 seperti Gambar 12 berikut.

```
array([[0.00051125],
       [0.00048661],
       [0.00046197],
       [0.00046197],
       [0.00046197]])
```

Gambar 12. Sample data yang sudah dinormalisasi

Dikarenakan normalisasi kali ini menggunakan bantuan *library* sklearn yaitu *MinMaxScaler* sehingga data input maupun output harus dalam bentuk array, dan *time series* atau data waktu tidak di ikut sertakan karena sebenarnya setiap data sudah sequence mewakili *time series*. Gambar 12 merupakan 5 *sample* hasil normalisasi dalam bentuk array.

4.2.5 Reshaping Data

Reshaping data adalah sebuah proses tambahan yaitu merubah dimensi array dari hasil normalisasi yang berupa array. Hal ini diperlukan karena untuk model yang akan digunakan menggunakan bantuan *library* keras, dan model dari keras ini hanya menerima inputan untuk X berupa array 3D karena merepresentasikan time series, jadi axisnya (instance, timestamp, future) dan Y atau target nya array 2D.

```
Shape of X_train after reshape : (9625, 60, 1)
Shape of y_train after reshape : (9625, 1)
```

Gambar 13. Hasil setelah array di reshape.

Karena berikutnya akan memasuki modeling maka data dibagi lagi menjadi *X_train* dan *y_train*. Proses reshape sendiri menggunakan bantuan *library* numpy dimana numpy memiliki sebuah method untuk reshape atau merubah dimensi array.

Perintah yang digunakan dalam numpy ini yakni, `numpy.reshape(array,(dimensi))`, jadi untuk parameter pertama dimasukan array yang akan di ubah dimensinya kemudian untuk parameter kedua yaitu dimensinya sendiri untuk array.

Untuk array 3D dimensi pertama itu adalah jumlah data yaitu 9.625, yang kedua adalah timestamp yaitu 60 yang berarti untuk memprediksi sebuah membutuhkan 60 data sebelumnya dan dimensi yang terakhir ialah future.

4.3 Reshaping Data

Memodelkan atau meng-implementasi algoritma LSTM dan GRU pada penelitian kali ini akan menggunakan *tools* yaitu bahasa pemrograman python versi 3 yang mana dalam prosesnya juga akan memakai *library* TensorFlow dan *library* keras sebagai API untuk mempermudah dalam membuat model.

Dalam implementasi ini, untuk semua algoritma akan menggunakan 4 *hidden layer*, setiap hidden layer terdapat 64 units, memakai fungsi aktivasi tanh untuk setiap unit, memakai optimasi adam untuk menga *update* bobot, dan melakukan 150 epoch/iterasi training.

Sehingga dari ketentuan tersebut jika di implementasikan kedalam algoritma LSTM dan GRU dalam bahasa pemrograman python akan seperti berikut ini.

```
#LSTM Arsitektur
lstm = Sequential()
lstm.add(
    LSTM(units = 64,
         activation = "tanh",
         return_sequences = True,
         input_shape = (X_train.shape[1],1))
)
lstm.add(
    Dropout(0.2)
)
# adding second layer and dropout regularization
lstm.add(
    LSTM(units = 64,
         activation = "tanh",
         return_sequences = True)
)
lstm.add(
    Dropout(0.2)
)
# adding third layer and dropout regularization
lstm.add(
    LSTM(units = 64,
         activation = "tanh",
         return_sequences = True)
)
lstm.add(
    Dropout(0.2)
)
# adding fourth layer and dropout regularization
lstm.add(
    LSTM(units = 64,
         activation = "tanh")
)
lstm.add(
    Dropout(0.2)
)
# adding the output layer
lstm.add(Dense(units = 1))
# compiling RNN
lstm.compile(
    optimizer = "adam",
    loss = "mean_squared_error",
    metrics = ["accuracy"])
# fitting the model
model_lstm = lstm.fit(X_train, y_train, epochs = 150, batch_size = 100)
```

Gambar 14. Hasil implementasi LSTM

```
#GRU Arsitektur
gru = Sequential()
gru.add(
    GRU(units = 64,
        activation = "tanh",
        return_sequences = True,
        input_shape = (X_train.shape[1],1))
    )
gru.add(
    Dropout(0.2)
    )
# adding second layer and dropout regularization
gru.add(
    GRU(units = 64,
        activation = "tanh",
        return_sequences = True)
    )
gru.add(
    Dropout(0.2)
    )
# adding third layer and dropout regularization
gru.add(
    GRU(units = 64,
        activation = "tanh",
        return_sequences = True)
    )
gru.add(
    Dropout(0.2)
    )
# adding fourth layer and dropout regularization
gru.add(
    GRU(units = 64,
        activation = "tanh")
    )
gru.add(
    Dropout(0.2)
    )
# adding the output layer
gru.add(Dense(units = 1))
# compiling RNN
gru.compile(
    optimizer = "adam",
    loss = "mean_squared_error",
    metrics = ["accuracy"])
# fitting the model
model_gru = gru.fit(X_train, y_train, epochs = 150, batch_size = 100)
```

Gambar 15. Hasil implementasi GRU.

Setelah selesai implementasi algoritma, selanjutnya adalah melakukan training kepada data yang sudah dilakukan *Reshaping* menjadi data array 3 Dimensi. Proses training untuk masing-masing algoritma kurang lebih memakan waktu selama 1 jam 3 menit untuk 1 algoritma.

4.4 Evaluasi

Ketika sudah dilakukan proses training, maka untuk melakukan evaluasi semua algoritma akan melakukan prediksi dengan data *test* yang sudah dibagi sebelumnya yaitu 4.151 data untuk testing sisa dari data training.

Pada tahap evaluasi akan menghitung nilai *loss/error*, dimana nilai *loss/error* menghitung selisih data antara data prediksi dan data yang sebenarnya kemudian supaya bisa dibandingkan kedepannya maka akan dipakai parameter-parameter penghitung *loss/error*.

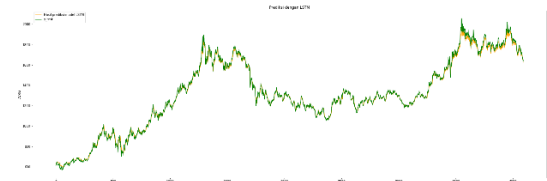
Ada tiga parameter untuk menghitung nilai *loss/error* yaitu *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE),

dan *Mean Absolute Error*. Setelah dilakukan perhitungan nilai *loss/error* maka didapati hasil seperti tabel berikut.

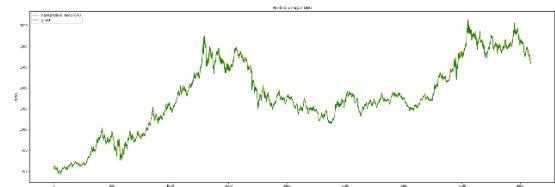
Tabel 1. Hasil Perhitungan parameter nilai *loss/error*

No	Parameter	LSTM	GRU
1	MSE	0,00017	0,00011
2	RMSE	0,1143	0,1087
3	MAE	0,00962	0,00752

Nilai-nilai yang ada pada tabel tersebut mengartikan bahwa semakin kecil nilainya maka performa dari algoritma tersebut semakin baik. Jika Hasil prediksi setiap algoritma dibuatkan dalam bentuk grafik maka akan seperti berikut ini.



Gambar 16. Hasil prediksi dengan algoritma LSTM vs data sebenarnya.



Gambar 17. Hasil prediksi dengan algoritma GRU vs data sebenarnya.

5 SIMPULAN

Berdasarkan hasil perbandingan baik itu ketika saat proses training maupun pada saat evaluasi, algoritma GRU secara konsisten memiliki nilai *loss/error* lebih kecil dibandingkan dengan dengan algoritma LSTM. Sehingga penulis menyimpulkan bahwa algoritma GRU memberikan performa lebih baik untuk memprediksi harga emas dunia.

Hal ini juga sejalan dengan apa yang telah dilakukan penelitian terdahulu yaitu Sofi dkk. (2021) dan Nilsen (2022) yang juga telah

melakukan prediksi menggunakan algoritma LSTM dan GRU ini, namun mereka melakukan prediksi untuk harga saham-saham di Indonesia dan hasilnya algoritma GRU memiliki performa lebih baik dibandingkan dengan LSTM.

Algoritma GRU mendapatkan performa lebih baik dibandingkan dengan LSTM dalam melakukan prediksi di harga saham dan harga emas dunia. Karena pada dasarnya harga saham dan harga emas dunia memiliki kesamaan yaitu ada time series, ada harga *open* ataupun harga *close* sehingga karena harga saham dan harga emas dunia identik, maka algoritma GRU memberikan performa lebih baik untuk kasus tersebut.

DAFTAR PUSTAKA

- Aulia, A., Elhanafi, A. M., & Dafitri, H. (2021, October). Implementasi Algoritma Gated Recurrent Unit Dalam Melakukan Prediksi Harga Kelapa Sawit Dengan Memanfaatkan Model Recurrent Neural Network (RNN). In *Prosiding SNASTIKOM: Seminar Nasional Teknologi Informasi & Komunikasi Paper* (pp. 288-294).
- Alkahfi, I., & Chiuloto, K. (2021). Penerapan Model Gated Recurrent Unit Pada Masa Pandemi Covid-19 Dalam Melakukan Prediksi Harga Emas Dengan Menggunakan Model Pengukuran Mean Square Error. *Snastikom Ke*, 8, 225-232.
- Farhah, A., Prasasti, A. L., & Paryasto, M. W. (2021). Implementasi Recurrent Neural Network dalam Memprediksi Kepadatan Restoran Berbasis LSTM. *Jurnal Media Informatika Budidarma*, 5(2), 524-531.
- Idham, I., Akbar, M. G. T., Panggabean, S., & Noor, M. (2022). Perbandingan Prediksi Harga Saham Dengan Menggunakan LSTM GRU Dengan Transformer. *Smart Comp: Jurnalnya Orang Pintar Komputer*, 11(1), 44-47.
- Nilsen, A. (2022). Perbandingan Model RNN, Model LSTM, dan Model GRU dalam Memprediksi Harga Saham-Saham LQ45. *Jurnal Statistika dan Aplikasinya*, 6(1), 137-147.
- Prasetyo, V. R., Axel, S., Soebroto, J. T., Sugiarto, D., Winatan, S. A., & Njudang, S. D. (2022). Gold Price Prediction Based on Gold. org Data Using the Long Short Term Memory Method. *SISTEMASI*, 11(3), 623-629.
- Sofi, K., Sunge, A. S., Riady, S. R., & Kamalia, A. Z. (2021). Perbandingan algoritma linear regression, LSTM, dan GRU dalam memprediksi harga saham dengan model time series. *PROSIDING SEMINASTIKA*, 3(1), 39-46.