

LAPORAN TUGAS BESAR
PREDIKSI PERGERAKAN HARGA SAHAM
MENGGUNAKAN *ARTIFICIAL NEURAL NETWORK*
(ANN)



Disusun Oleh:

Nama : 1. Abbas Adam Az-Zuhri
2. Cecep Triyandi
Kelompok : Terminale
Tema Pelatihan : Artificial Intelligence
Program Akademi : Fresh Graduate Academy

PROGRAM PELATIHAN *FRESH GRADUATE PROGRAM*
DIGITAL TALENT SCHOLARSHIP 2019
INSTITUT TEKNOLOGI BANDUNG

1. Pendahuluan

Pasar modal sebagai bagian dari suatu wadah yang memfasilitasi berbagai sarana dan prasarana kegiatan jual beli dan kegiatan terkait lainnya. Pasar modal merupakan pasar untuk berbagai instrumen keuangan jangka panjang yang bisa diperjualbelikan, baik surat utang (obligasi), ekuiti (saham), reksa dana, instrumen derivatif maupun instrumen lainnya. Di samping itu, pasar modal merupakan sarana pendanaan bagi perusahaan maupun institusi lainnya (misalnya pemerintah), dan sebagai sarana bagi kegiatan berinvestasi.^[1] Dalam perekonomian suatu negara, pasar modal memiliki peranan penting karena pasar modal menjalankan dua fungsi yaitu pertama sebagai sarana bagi pendanaan usaha atau sebagai sarana bagi perusahaan untuk mendapatkan dana dari masyarakat pemodal (investor) yang nantinya akan digunakan untuk pengembangan usaha, ekspansi, penambahan modal kerja dan lain-lain. Kedua, pasar modal menjadi sarana bagi masyarakat untuk berinvestasi pada instrumen keuangan seperti saham, obligasi, reksa dana, dan lain-lain.

Saham merupakan salah satu produk utama dalam instrumen pasar keuangan dan yang paling populer karena produk investasi yang dapat memberikan tingkat keuntungan yang menarik sehingga banyak dipilih para investor. Saham juga didefinisikan sebagai tanda penyertaan modal seseorang atau pihak (badan usaha) dalam suatu perusahaan sehingga dengan menyertakan modal tersebut maka pihak tersebut akan memiliki klaim atas pendapatan perusahaan, klaim atas asset perusahaan, dan berhak hadir dalam Rapat Umum Pemegang Saham (RUPS).^[2] Pendapatan yang diperoleh investor tergantung dari perusahaan yang menerbitkan saham. Jika emiten mampu menghasilkan keuntungan yang besar maka keuntungan yang diperoleh juga akan besar. Semakin tinggi keuntungan yang ditawarkan maka semakin tinggi resiko yang akan dihadapi dalam berinvestasi. Oleh karena itu perlu diprediksi pergerakan harga saham sekarang yang berdasarkan atas dasar harga saham sebelumnya.

Dalam memprediksi harga saham dapat dilakukan dengan memantau secara langsung pergerakan saham pada saat itu juga dengan melakukan analisis histori dari *trend* harga saham pada periode sebelumnya sehingga penelitian dapat dilakukan untuk menyelesaikan permasalahan prediksi saham, mulai dari penelitian dari sisi analisis ekonomi hingga *machine learning*.

Machine learning adalah bagian dari teknologi kecerdasan buatan yang lebih menitikberatkan pada konstruksi dan pelatihan sistem sehingga sistem dapat belajar dari data-data yang diberikan. Algoritma pembelajaran yang dimodelkan untuk memprediksi pergerakan indeks harga saham diantaranya adalah *Recurrent Neural Network*, *Long Short-Term Memory (LSTM)*, *Gate Recurrent Unit (GRU)* dimana metode tersebut termasuk dalam *supervised learning* dengan proses belajar yang dilakukan memerlukan sesuatu yang memiliki pengetahuan tentang lingkungan.

2. Studi Literatur

Penelitian ini merujuk ke beberapa penelitian terkait lainnya seperti *Predicting Stock Prices using LSTM*, *Stock Market Prices Movement Prediction with LSTM Neural Network*, *NSE Stock Market Prediction using Deep Learning Models*, dan *A Comparative Study of LSTM and DNN for Stock Market Forecasting* yang membahas tentang pembelajaran dari data *time-series*.

Pada penelitian yang berjudul *Predicting Stock Prices using LSTM*, penelitian tersebut menggunakan metode LSTM untuk memprediksi harga saham menggunakan 4 fitur data yaitu *open*, *high*, *low* dan *close* dengan *epoch* 500. Hasil dari penelitian tersebut yang diukur berdasarkan nilai RMSE didapatkan bahwa nilai RMSE untuk data training adalah 0,00983 dan data *training* adalah 0,00859.

Pada penelitian yang berjudul *Stock Market Prices Movement Prediction with LSTM Neural Network*, penelitian tersebut menggunakan metode LSTM untuk memprediksi harga saham akan naik atau tidak dalam waktu dekat. Hasil dari penelitian tersebut yang diukur berdasarkan nilai akurasi didapatkan bahwa nilai rata-rata akurasi adalah 55,9%.

Pada penelitian yang berjudul *NSE Stock Market Prediction using Deep Learning Models*, penelitian tersebut menggunakan metode MLP, RNN, LSTM dan CNN untuk memprediksi harga saham. Hasil dari penelitian tersebut didapat model *Deep Learning* (DL) mampu mengungguli model ARIMA untuk prediksi dari data *time-series*.

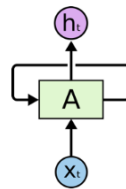
Pada penelitian yang berjudul *A Comparative Study of LSTM and DNN for Stock Market Forecasting*, penelitian tersebut menggunakan metode LSTM, RNN dan DNN untuk memprediksi harga saham dalam harian dan mingguan. Hasil dari penelitian tersebut didapatkan bahwa LSTM dan RNN mampu

mengungguli DNN dalam memprediksi harga saham mingguan karena lebih baik dalam proses data jangka panjang sedangkan untuk memprediksi harga saham harian LSTM, RNN dan DNN memberikan hasil yang sama.

3. Metodologi

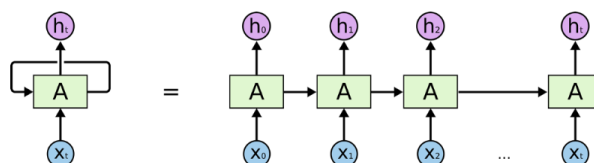
3.1. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah salah satu jenis *Artificial Neural Network* (ANN) yang pada bagian intinya (*cell*) terjadi pengulangan. Hal tersebut berarti bahwa output dari *cell* akan menjadi inputnya lagi.



Gambar 1. Struktur RNN.^[3]

Pada struktur jaringan saraf A seperti pada Gambar 1. terdapat beberapa input x_t dan menghasilkan nilai h_t . Pengulangan A memungkinkan informasi untuk diteruskan dari satu langkah jaringan ke langkah berikutnya. RNN dapat dianggap sebagai beberapa salinan dari jaringan yang sama, masing-masing menyampaikan pesan kepada jaringan berikutnya seperti yang ditunjukkan pada Gambar 2. RNN ini juga didesain secara khusus untuk mengatasi data berurutan.



Gambar 2. Struktur RNN setelah diuraikan.^[3]

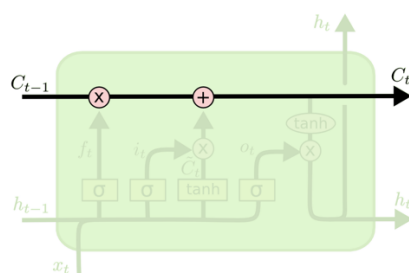
Semua proses komputasi pada RNN dilakukan pada pengulangan A. pada RNN yang paling sederhana, pengulangan A hanya berisi 1 neuron dengan fungsi aktivasi yang biasanya dipakai yaitu tanh atau sigmoid. Namun pada RNN ini mengalami masalah *vanishing gradient* seiring bertambah panjangnya data berurut yang ingin dilatih kepada RNN tersebut. *Vanishing gradient* adalah sebuah situasi dimana nilai gradient yang digunakan untuk memperbaharui *weight* pada neuron di sesi pelatihan ANN bernilai dan/atau

mendekati 0. Hal ini disebabkan karena penggunaan fungsi aktivasi sigmoid atau tanh yang memetakan input menjadi output yang jarak antar marginnya kecil yang memengaruhi sensitivitas layer neuron dalam memperbaharui *weight*. Dengan demikian, ketika errornya besar dan harus memperbaharui *weight* secara signifikan juga, yang akan terjadi nilai *weight* yang diperbaharui tidak signifikan. Semakin banyak layer yang ditumpuk maka semakin lambat pembelajaran (memperbaharui *weight*) yang dilakukan oleh neuron pada ANN.

3.2. Long Short-Term Memory (LSTM)

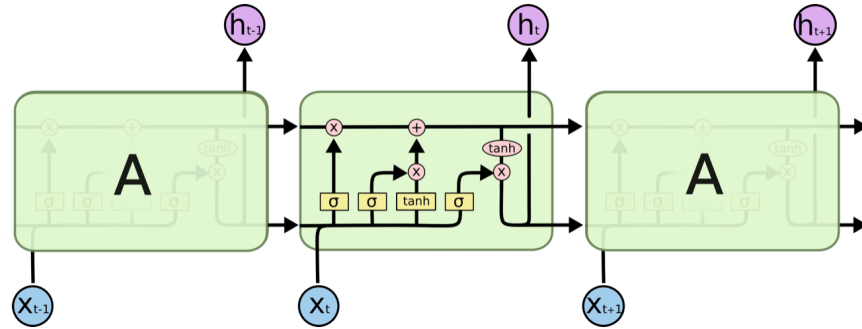
LSTM pertama kali dibuat oleh Sepp Hochreiter dan Jürgen Schmidhuber pada tahun 1997. LSTM menangani masalah *vanishing gradient* dalam pelatihan yang terjadi di *Recurrent Neural Network* (RNN) dasar. LSTM memiliki kemampuan untuk mengolah informasi lebih akurat.^[5] LSTM memiliki kelebihan dari pada RNN biasa, yaitu adanya arsitektur mengingat dan melupakan *output* yang akan diproses kembali menjadi *input*. Selain itu, kemampuan LSTM yang lain adalah dapat mempertahankan error yang terjadi ketika melakukan *backpropagation* sehingga tidak memungkinkan kesalahan meningkat.^[5]

Kunci utama pada LSTM adalah *cell state*. Pada LSTM, sebuah *cell state* berfungsi sebagai memori atau ingatan untuk sebuah *layer*. *Cell state* merupakan garis horizontal yang menghubungkan semua output layer pada LSTM seperti ditunjukkan pada Gambar 3.



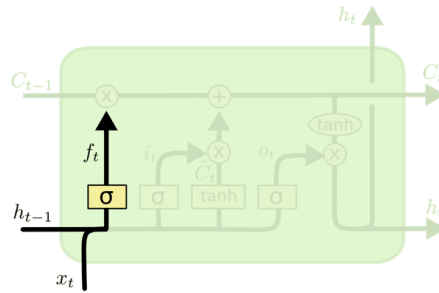
Gambar 3. *Cell state* pada LSTM.^[3]

Nilai *cell state* dimanipulasi menggunakan sebuah *gating system* atau sistem gerbang. Sistem gerbang ini terdiri atas beberapa arsitektur jaringan saraf tiruan (JST) sederhana untuk mengatur waktu sebuah data harus disimpan, digunakan, atau dilupakan. Berikut ini skema sederhana dari LSTM yaitu:



Gambar 4. Struktur LSTM.^[3]

Terdapat tiga jenis unit gerbang berbeda yang digunakan dalam LSTM, yaitu *input gate*, *forget gate*, dan *output gate*. Langkah pertama yaitu LSTM memutuskan informasi apa yang akan dihapus dari *cell state*. Keputusan ini dibuat oleh sigmoid layer yang bernama *forget gate*. Gerbang ini akan memproses h_{t-1} dan x_t sebagai input dan menghasilkan output berupa angka 0 atau 1 pada *cell state* C_{t-1} . Langkah pertama ditunjukkan pada Gambar 5.



Gambar 5. Alur informasi pada forget gate.^[3]

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

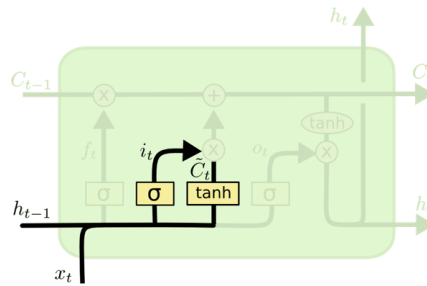
Dimana:

- f_t = *forget gate*
- σ = fungsi sigmoid
- W_f = nilai *weight* untuk *forget gate*
- h_{t-1} = nilai output sebelum orde ke-t
- x_t = nilai input pada order ke-t
- b_f = nilai bias pada *forget gate*

Forget gate berguna untuk menentukan sebuah memori pada waktu sebelumnya harus dilupakan atau tidak. *Forget gate* ini berguna untuk

mencegah *vanishing gradient* atau kasus sebaliknya *exploding gradient* (pembelajaran atau memperbaharui *weight* menjadi terlalu cepat atau besar).

Langkah kedua yaitu memutuskan informasi apa yang akan disimpan di *cell state*. Pada langkah ini ada dua bagian, pertama sigmoid layer yang bernama *input gate* memutuskan nilai mana yang akan diperbaharui. Kemudian, layer tanh membuat satu kandidat dengan nilai baru \tilde{C}_t yang dapat ditambahkan ke *cell state*. Kemudian output dari *input gate* dan layer tanh akan digabungkan untuk memperbaharui *cell state*. Langkah kedua ditunjukkan pada Gambar 6.



Gambar 6. Alur informasi pada input gate.^[3]

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \end{aligned} \quad (2)$$

Dimana:

i_t = input gate

σ = fungsi sigmoid

W_i = nilai *weight* untuk *input gate*

W_c = nilai *weight* untuk *cell gate*

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

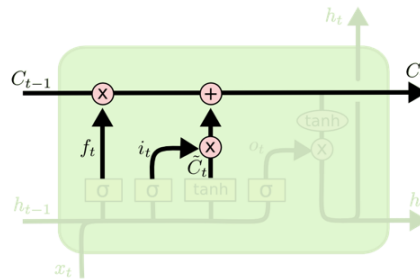
x_t = nilai input pada order ke-t

b_i = nilai bias pada *input gate*

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

Input gate adalah layer neuron dengan fungsi aktivasi sigmoid dan sebelumnya yang mempunyai fungsi aktivasi tanh. *Input gate* berfungsi untuk menentukan sebuah masukan yang akan ditambahkan ke dalam memori *cell state* saat itu atau tidak yang berarti bagian mana yang akan diperbarui. Secara harfiah *cell state* akan diperbarui dengan output dari *gate* ini.

Langkah ketiga yaitu memperbaharui *cell state* yang lama C_{t-1} menjadi *cell state* baru C_t seperti pada Gambar 7. Kemudian dengan mengkalikan state lama dengan f_t untuk menghapus informasi yang sudah ditentukan sebelumnya pada *forget gate*. Kemudian ditambahkan dengan $i_t * \tilde{C}_t$ yang merupakan nilai baru dan digunakan untuk memperbaharui state.



Gambar 7. Alur membuat *cell state* baru.^[3]

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3)$$

Dimana:

C_t = *cell state*

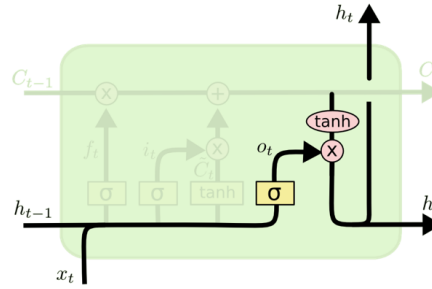
f_t = *forget gate*

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

i_t = *input gate*

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

Langkah keempat yaitu bertujuan untuk memutuskan hasil output seperti pada Gambar 8. Output harus sesuai dengan *cell state* yang telah diproses terlebih dahulu. Pertama sigmoid layer memutuskan bagian dari *cell state* yang menjadi output. Kemudian output dari *cell state* dimasukkan ke dalam tanh layer (untuk mengganti nilai menjadi antara -1 dan 1) dan dikalikan dengan sigmoid gate agar output yang diperoleh sesuai dengan apa yang kita putuskan sebelumnya.



Gambar 8. Alur informasi pada output gate.^[3]

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (4)$$

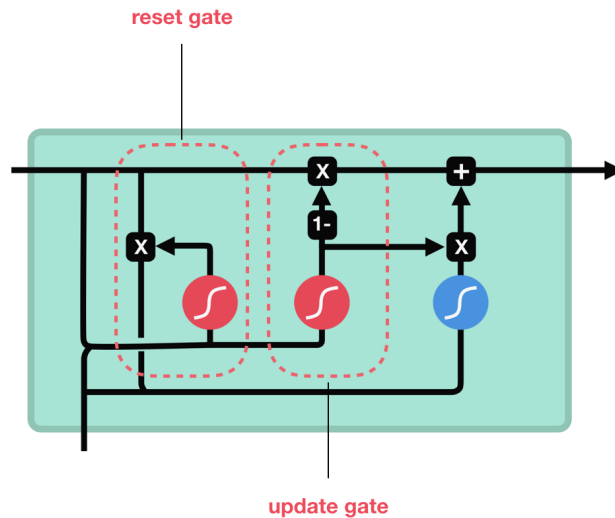
Dimana:

- o_t = output gate
- σ = fungsi sigmoid
- W_o = nilai weight untuk output gate
- h_{t-1} = nilai output sebelum orde ke-t
- x_t = nilai input pada order ke-t
- b_o = nilai bias pada output gate
- h_t = nilai output order ke-t
- C_t = cell state

Output gate adalah layer neuron dengan fungsi aktivasi sigmoid di paling kanan pada jajaran layer neuron. *Output gate* berguna untuk menentukan seberapa besar pengaruh memori *cell state* terhadap hasil prediksi yang akan dihasilkan. *Output gate* ini tidak berkontribusi pada *cell state* tetapi *gate* ini membedakan antara *cell state* dan output yang sebenarnya.

3.3. Gate Recurrent Unit (GRU)

GRU pertama kali diperkenalkan oleh Chung et al. pada tahun 2014. GRU merupakan salah satu arsitektur dari RNN yang membuat *recurrent unit* dapat menangkap *dependencies* secara aditif dalam skala waktu berbeda-beda. GRU dapat dikategorikan sebagai versi LSTM yang sederhana. GRU memiliki dua komponen pengatur alur informasi yaitu *reset gate* dan *update gate* seperti ditunjukkan pada Gambar 9.



Gambar 9. Struktur GRU.^[7]

Reset gate berperan untuk menentukan penggabungan *input* baru dengan informasi di masa lalu. Sedangkan *update gate* berperan untuk menentukan banyaknya informasi di masa lalu yang harus disimpan.

4. Implementasi

Pada penelitian ini digunakan algoritma *Long Short-Term Memory* (LSTM) untuk memprediksi harga saham PT Bank Negara Indonesia, Tbk. (BBNI) yang tercatat di Bursa Efek Indonesia (BEI). Dengan menggunakan data histori harga saham selama 15 tahun dan melakukan pemilihan atribut tertentu yang dianggap dapat meningkatkan nilai akurasi.

4.1 Pemahaman Data (*Data Understanding*)

Pada tahap ini dilakukan proses pengumpulan data. Penelitian ini menggunakan data *time-series* perusahaan PT. Bank Negara Indonesia, Tbk (BBNI) yang diperoleh dari website finance.yahoo.com. Data yang diperlukan pada penelitian ini adalah sebagai berikut,

- a) Data sekunder yang terdiri dari data harga saham BBNI dari 23 Desember 2003 sampai dengan 9 Agustus 2019 yang berjumlah 3896 data. Porsi untuk 80% *data training*, 10 % *data validation* 10%, dan 10% *data testing* 10%. Kemudian diasumsikan harga saham hanya dipengaruhi oleh harga saham pada hari sebelumnya.

- b) Data primer yang terdiri dari data yang diperoleh dari hasil komputasi model prediksi.

Setelah data diperoleh maka akan dilakukan analisa statistik sederhana untuk memperoleh informasi (*insight*) tentang data tersebut. Potongan program ditunjukkan pada Gambar 10.

```
16 # function describe data
17 def describe_data(df):
18     describe = df.describe()
19     return describe
```

Gambar 10. Program analisa statistik

Setelah analisa statistik maka akan dilakukan *correlation* data untuk seleksi setiap atribut dengan membuang atribut yang tidak diperlukan. Potongan program ditunjukkan pada Gambar 11.

```
21 # function correlation data
22 def correlation_data(df):
23     correlations = df.corr(method='pearson')
24     return correlations
```

Gambar 11. Program *correlation* data

Setelah *correlation* data maka akan dilakukan *skew* untuk mengukur asimetri distribusi data normal. Potongan program ditunjukkan pada Gambar 12.

```
26 # function skew data
27 def skew_data(df):
28     skew = df.skew()
29     return skew
```

Gambar 12. Program *skew* data

4.2 Pengolahan Data (*Data Preprocessing*)

Pada tahap ini akan dilakukan pengolahan data agar hasil yang diinginkan dapat tercapai. Terdapat beberapa tahap pengolahan data dimana langkah-langkah yang dilakukan adalah sebagai berikut:

a) *Data Cleaning*

Data cleaning merupakan langkah untuk menghilangkan atribut yang tidak lengkap, tidak konsisten, dan tidak rapi yang dapat mengakibatkan algoritma tidak sesuai serta dapat mengakibatkan program berjalan sangat lama. Dalam penelitian ini menggunakan teknik *missing value* untuk data

cleaning. Data yang diperoleh pada penelitian ini 3743 (BBNI) data namun tidak semua data digunakan karena ada beberapa data yang tidak adanya volume perdagangan, berisikan jumlah dividen, dan bernilai NaN sehingga perlu dilakukan *data cleaning* dengan membuang data tersebut. Dengan demikian data yang digunakan setelah proses *data cleaning* 3723 data BBNI. Potongan program ditunjukkan pada Gambar 13.

```
10 # function drop data zero or NaN
11 def drop_data(df):
12     df = df[df['Volume'] != 0]
13     df = df[df['Volume'].notnull()]
14     return df
```

Gambar 13. Program *cleanning* data

b) *Data Transformation*

Pada tahap ini dilakukan normalisasi. Normalisasi merupakan salah satu teknik yang dilakukan pada *data transformation*. Tujuan dilakukannya normalisasi adalah agar proses pengolahan data dapat berjalan dengan cepat dan dalam bentuk sederhana. Potongan program ditunjukkan pada Gambar 14.

```
26 # function min-max normalization
27 def normalization_data(df):
28     min_max_scaler = sklearn.preprocessing.MinMaxScaler()
29     df['Open'] = min_max_scaler.fit_transform(df['Open'].values.reshape(-1,1))
30     df['High'] = min_max_scaler.fit_transform(df['High'].values.reshape(-1,1))
31     df['Low'] = min_max_scaler.fit_transform(df['Low'].values.reshape(-1,1))
32     df['Close'] = min_max_scaler.fit_transform(df['Close'].values.reshape(-1,1))
33     df['Volume'] = min_max_scaler.fit_transform(df['Volume'].values.reshape(-1,1))
34     return df
```

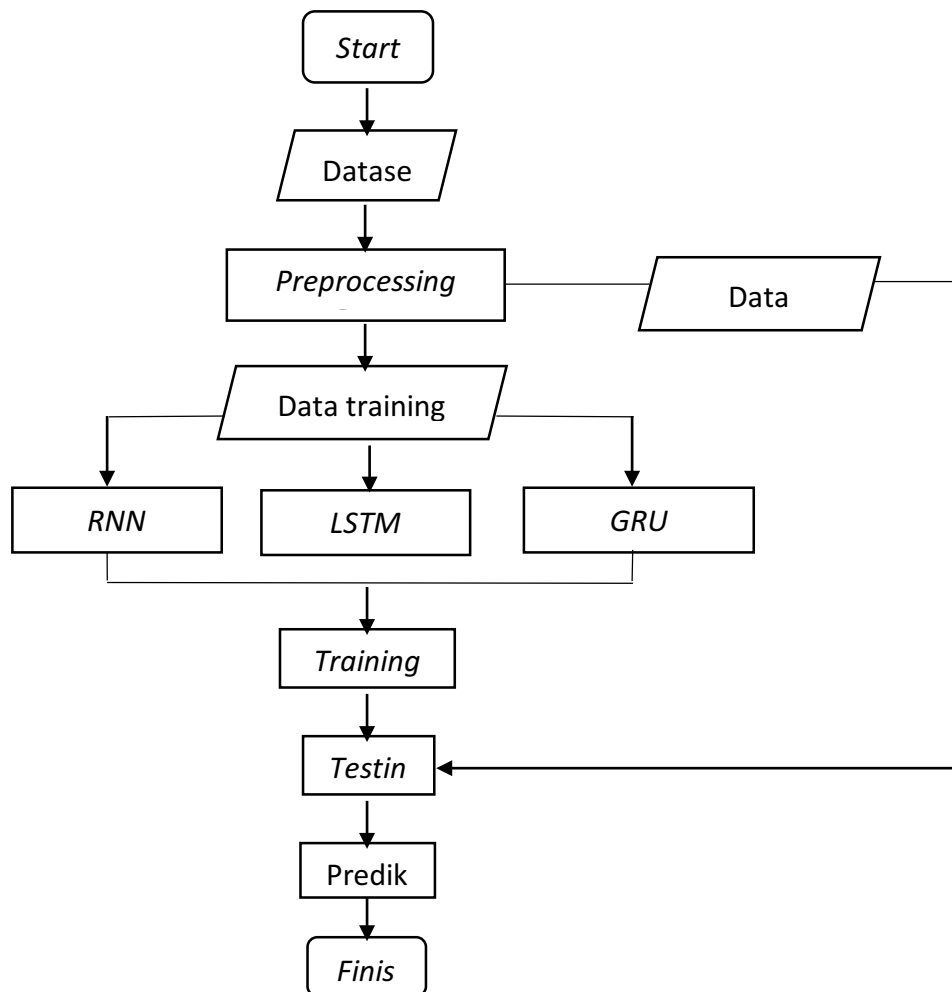
Gambar 14. Program normalisasi data

c) *Pembagian Data (Data Splitting)*

Pembagian data ini dibagi menjadi tiga bagian, yaitu data training, data testing, dan data validasi. Dari data saham harian yang dikumpulkan selama periode 15 tahun diperoleh 3723 data BBNI. Data tersebut dibagi menjadi 8:1:1 untuk data training:data testing:data validasi.

4.3 Pemodelan

Pada tahap ini disebut juga sebagai tahap pembelajaran. Tahap ini merupakan gambaran dari rangkaian tahapan untuk melakukan pelatihan dan pengujian terhadap data yang sudah terkumpul yang kemudian diolah. Pembuatan model pada tahap ini menggunakan algoritma LSTM karena data yang diolah merupakan data *time-series*. Sistem yang akan diimplementasikan mempunyai tujuan utama untuk memprediksi pergerakan harga saham yang diklasifikasikan menjadi pergerakan naik atau pergerakan turun. Hasil keluaran dari sistem akan dihitung akurasi dengan cara membandingkannya dengan data pergerakan indeks harga saham sebenarnya yang telah melalui *preprocessing* terlebih dahulu sehingga data yang menjadi target dari sistem dapat dibandingkan dengan hasil keluaran sistem. Tahapan sederhana penelitian yang akan dilakukan untuk membangun sistem prediksi menggunakan LSTM dapat dilihat pada Gambar 15.



Gambar 15. Alur proses prediksi pergerakan harga saham menggunakan ANN

4.4 Evaluasi

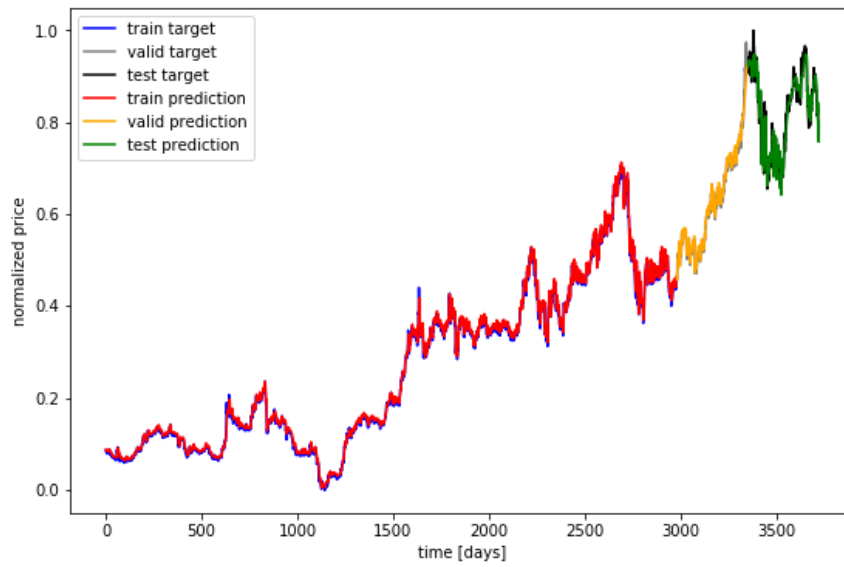
Setelah diperoleh model dari hasil training maka akan dilakukan evaluasi terhadap model tersebut dengan menggunakan variasi *epochs* sejumlah 50 *epochs* dan 100 *epochs*. Kemudian untuk mengukur akurasi akan menggunakan *Root Mean Squared Error* (RMSE) dan *Mean Squared Error* (MSE) dengan menguji model prediksi yang dianggap paling optimal yang terbentuk dengan algoritma RNN, LSTM dan GRU.

5. Hasil dan Pembahasan

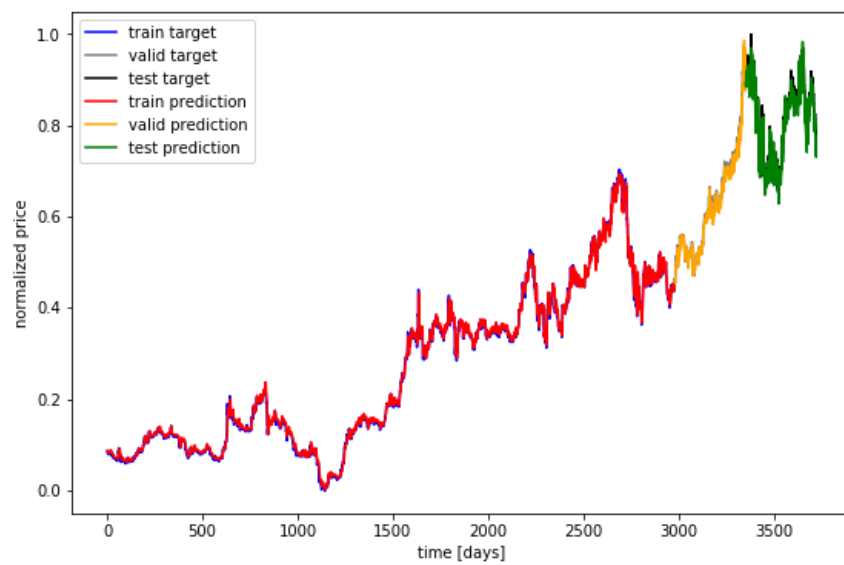
Pengujian pada penelitian ini adalah menganalisa parameter yang digunakan terhadap masing-masing model dari ANN untuk melihat nilai MSE dan RMSE yang optimal. Parameter yang diuji berupa nilai maksimal *epochs* pada masing-masing model yaitu RNN, LSTM dan GRU. Parameter yang akan diuji menggunakan model RNN, LSTM, dan GRU dengan variasi *epochs*: 50 dan 100.

5.1 Pengujian 1

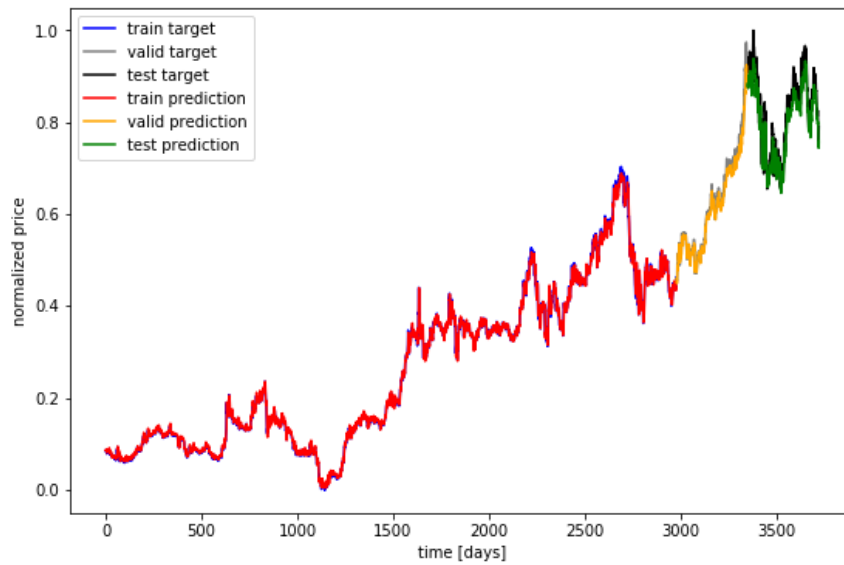
Pada pengujian ini dilakukan pengujian pada masing-masing model yaitu RNN, LSTM dan GRU dengan parameter *epochs* adalah 50 *epochs*. Parameter lainnya adalah 100 neuron, 100 *batch size*, 5 *hidden layers* dan *learning rate* 0.0001. Kemudian analisa hasil model yang telah divalidasi melalui visualisasi grafik berdasarkan nilai *close* model GRU, LSTM, dan RNN yang ditunjukkan oleh Gambar 16., Gambar 17., dan Gambar 18. dengan epoch 50.



Gambar 16. Pergerakan harga saham pada GRU berdasarkan nilai *close* dengan epoch 50.

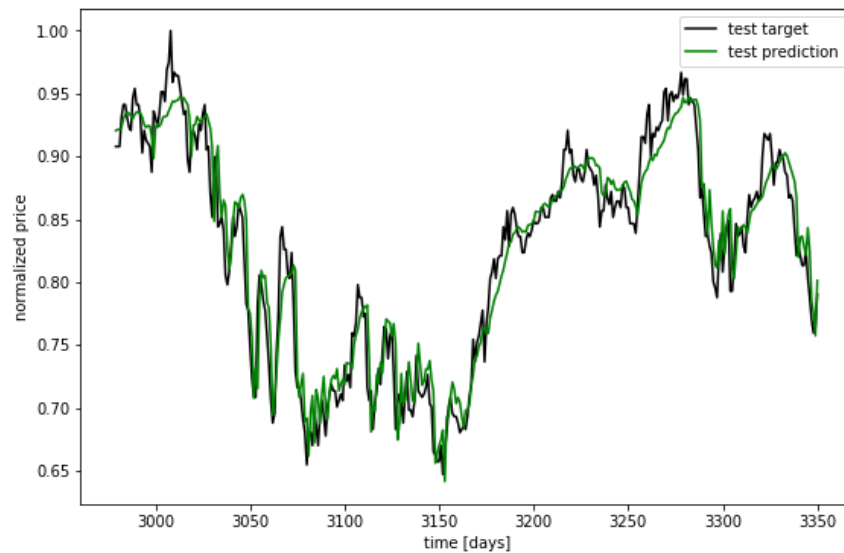


Gambar 17. Pergerakan harga saham pada LSTM berdasarkan nilai *close* dengan epoch 50.

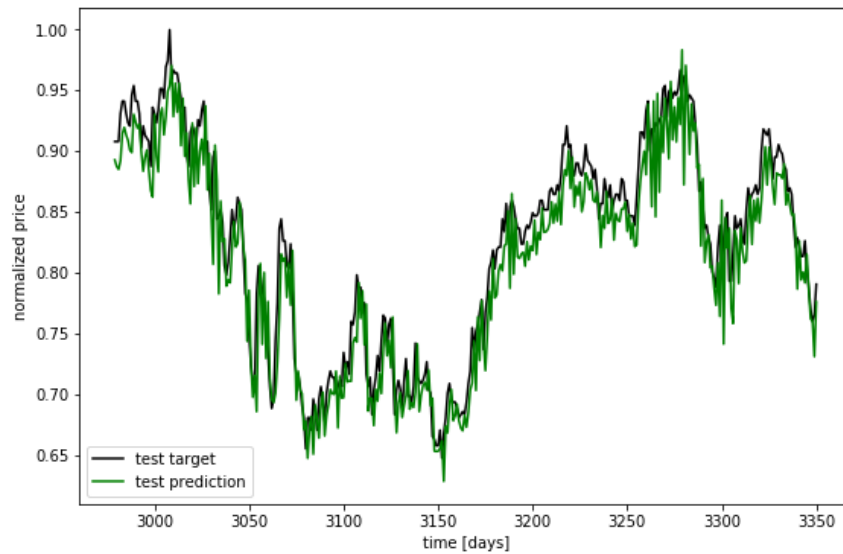


Gambar 18. Pergerakan harga saham pada RNN berdasarkan nilai *close* dengan epoch 50.

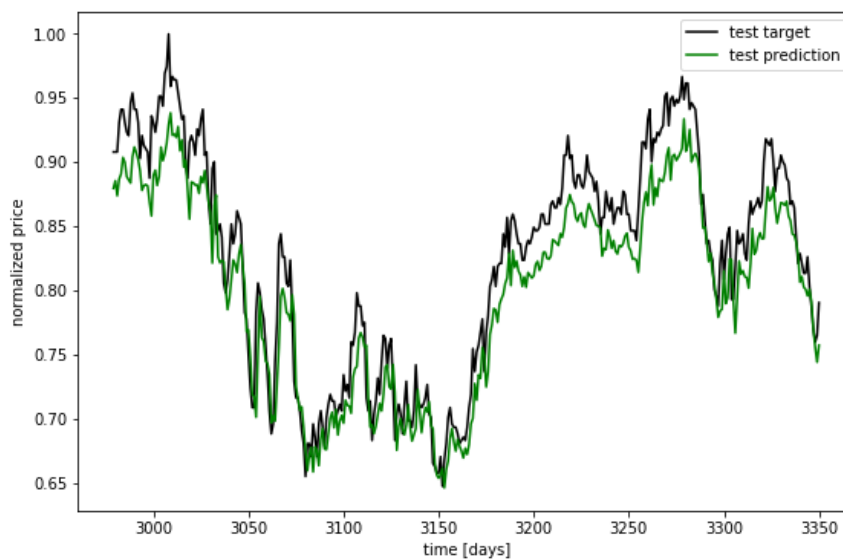
Kemudian ditampilkan analisa hasil model yang telah diuji berdasarkan nilai *close* untuk model GRU, LSTM, RNN yang ditunjukkan oleh Gambar 19., Gambar 20., dan Gambar 21.



Gambar 19. Prediksi pergerakan harga saham pada *data testing* GRU berdasarkan nilai *close* dengan epoch 50.



Gambar 20. Prediksi pergerakan harga saham pada *data testing* LSTM berdasarkan nilai *close* dengan epoch 50.

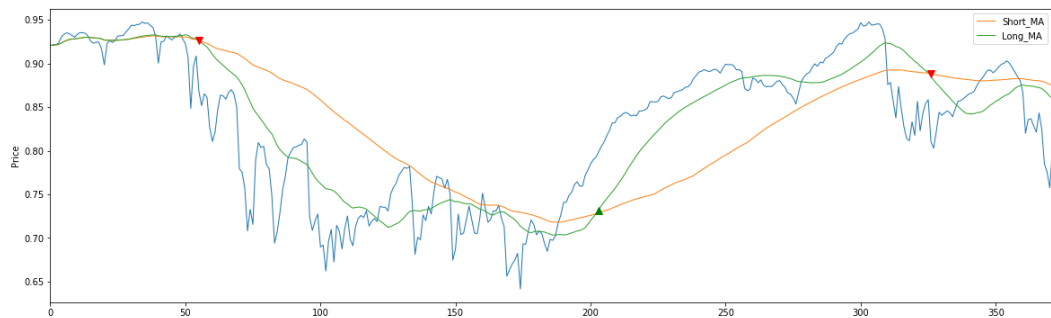


Gambar 21. Prediksi pergerakan harga saham pada *data testing* RNN berdasarkan nilai *close* dengan epoch 50.

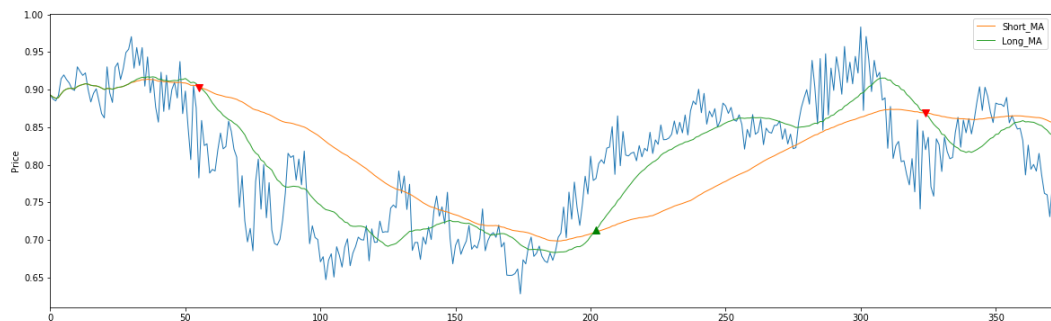
Dari hasil visualisasi grafik tersebut terlihat perbedaan yang cukup signifikan dari hasil masing-masing model. Untuk hasil model RNN yang ditunjukkan Gambar 18. dan Gambar 21. terlihat hasil prediksi yang berada di bawah data sebenarnya atau bisa disebut masih adanya error atau gap antara data prediksi dan data sebenarnya. Untuk hasil model LSTM yang ditunjukkan Gambar 17. dan Gambar 20. terlihat hasilnya cukup baik dibandingkan hasil model RNN meskipun hasil data prediksi dari data test masih berfluktuatif yang

ditunjukkan pada Gambar 20. Kemudian untuk hasil model GRU yang ditunjukkan Gambar 16. dan Gambar 19. terlihat hasil yang diperoleh lebih baik daripada hasil model RNN dan LSTM.

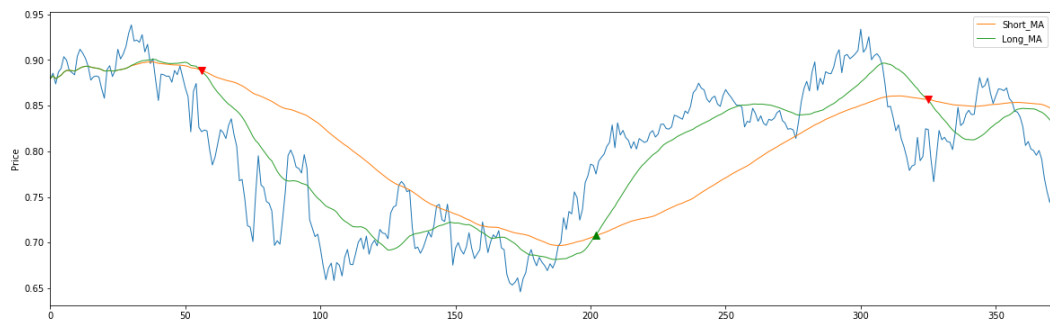
Kemudian analisis juga dilakukan dengan menghasilkan signal *buy* atau *sell* menggunakan indikator *moving average* (MA) berdasarkan nilai *close* dengan model GRU, LSTM, RNN pada saham yang masing-masingnya melalui visualisasi dalam bentuk grafik seperti ditunjukkan Gambar 22., Gambar 23., dan Gambar 24.



Gambar 22. Prediksi signal *buy* dan *sell* saham pada *data testing* GRU berdasarkan nilai *close* dengan epoch 50



Gambar 23. Prediksi signal *buy* dan *sell* saham pada *data testing* LSTM berdasarkan nilai *close* dengan epoch 50



Gambar 24. Prediksi signal *buy* dan *sell* saham pada *data testing* LSTM berdasarkan nilai *close* dengan epoch 50

Pada Gambar 22., Gambar 23., dan Gambar 24. nilai *short* MA didistribusikan sebanyak 30 dan *long* MA sebanyak 90 sehingga membentuk dua garis yang saling berpotongan. Tiap perpotongan akan menghasilkan signal *buy* atau *sell*. Jika *long* MA turun dan memotong *short* MA dari atas ke bawah maka signal yang dihasilkan adalah *sell* sedangkan jika *long* MA naik dan memotong *short* MA dari bawah ke atas maka signal yang dihasilkan adalah *buy*. Dari grafik MA ini juga dapat diprediksi bahwa pergerakan harga saham pada hari selanjutnya masih dapat menunjukkan trend penurunan karena garis *long* MA belum memotong garis *short* MA.

Hasil pengujian dan analisa model menunjukkan suatu nilai berupa nilai MSE dan RMSE dari *data training*, *data validation* dan *data testing* seperti ditunjukkan pada Tabel 1.

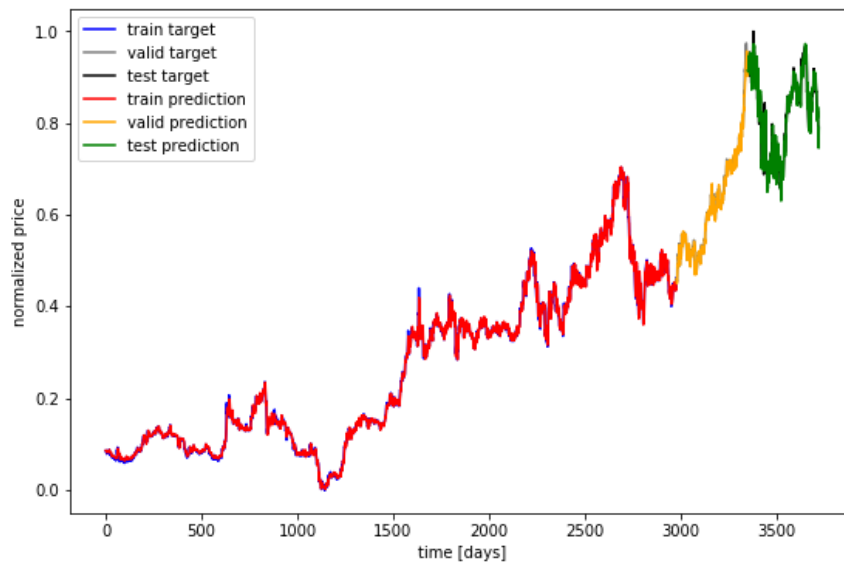
Tabel 1. Pengujian masing-masing model untuk *epochs* 50

No	Model	Training		Validation		Testing	
		MSE	RMSE	MSE	RMSE	MSE	RMSE
1	LSTM	0.000470	0.021684	0.000414	0.020335	0.000603	0.024562
2	GRU	0.000432	0.020786	0.000411	0.020266	0.000537	0.023166
3	RNN	0.000461	0.021476	0.000406	0.020153	0.000720	0.026832

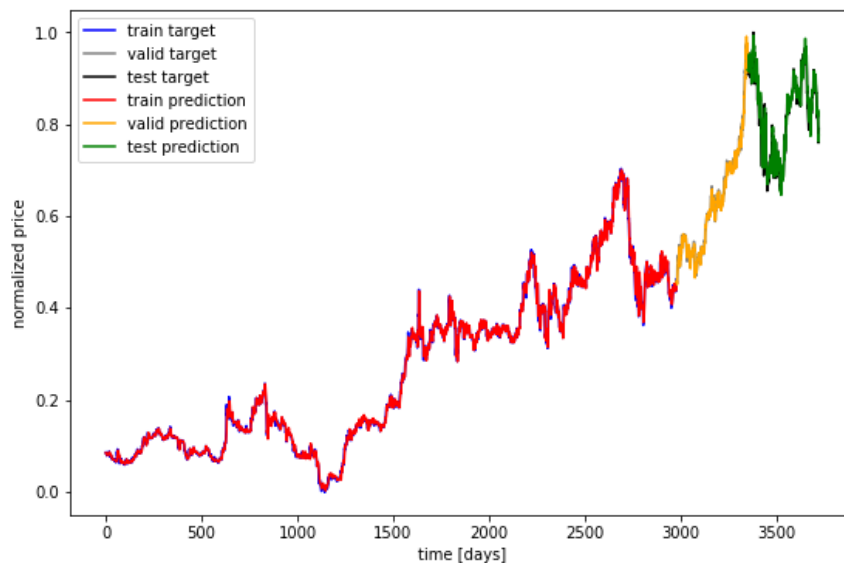
Dari hasil pengujian yang ditunjukkan pada Tabel 1. dapat diperoleh bahwa model yang memiliki nilai MSE dan RMSE paling optimal adalah model GRU kemudian model LSTM dan model RNN. Model GRU memperoleh nilai optimal untuk semua distribusi data yaitu *data training*, *data validation* dan *data testing*.

5.2 Pengujian 2

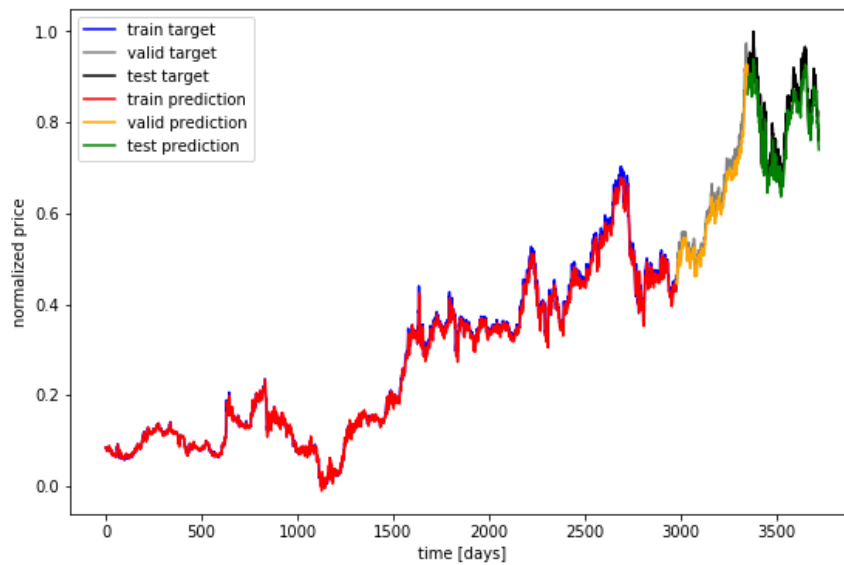
Pada pengujian ini dilakukan pengujian pada masing-masing model yaitu RNN, LSTM dan GRU dengan parameter *epochs* adalah 100 *epochs*. Parameter lainnya adalah 100 neuron, 100 *batch size*, 5 *hidden layers* dan *learning rate* 0.0001. Kemudian analisa hasil model yang telah divalidasi melalui visualisasi grafik berdasarkan nilai *close* model GRU, LSTM, dan RNN yang ditunjukkan oleh Gambar 25., Gambar 26., dan Gambar 27. dengan epoch 100.



Gambar 25. Pergerakan harga saham pada GRU berdasarkan nilai *close* dengan epoch 100.

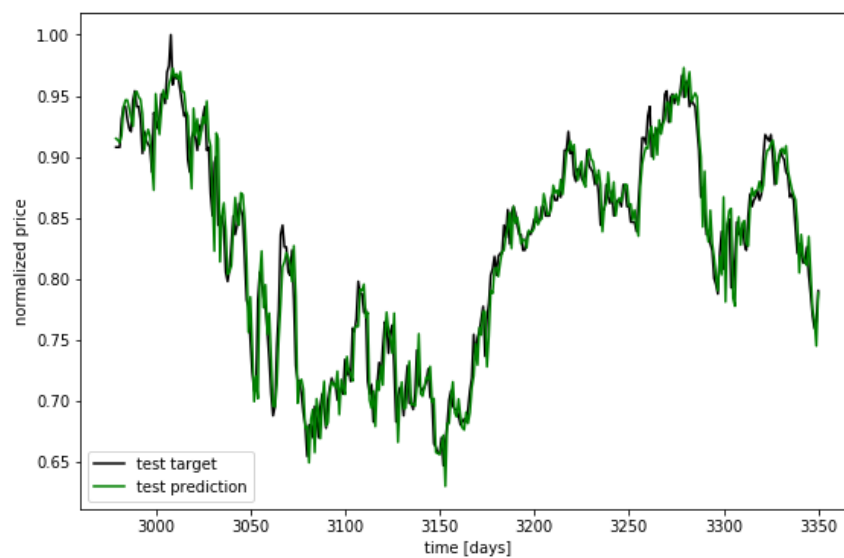


Gambar 26. Pergerakan harga saham pada LSTM berdasarkan nilai *close* dengan epoch 100.

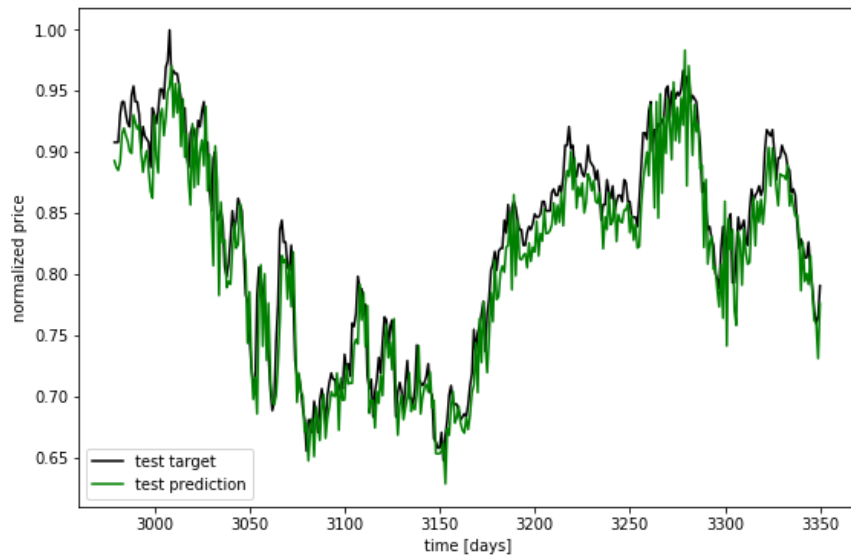


Gambar 27. Pergerakan harga saham pada RNN berdasarkan nilai *close* dengan epoch 100.

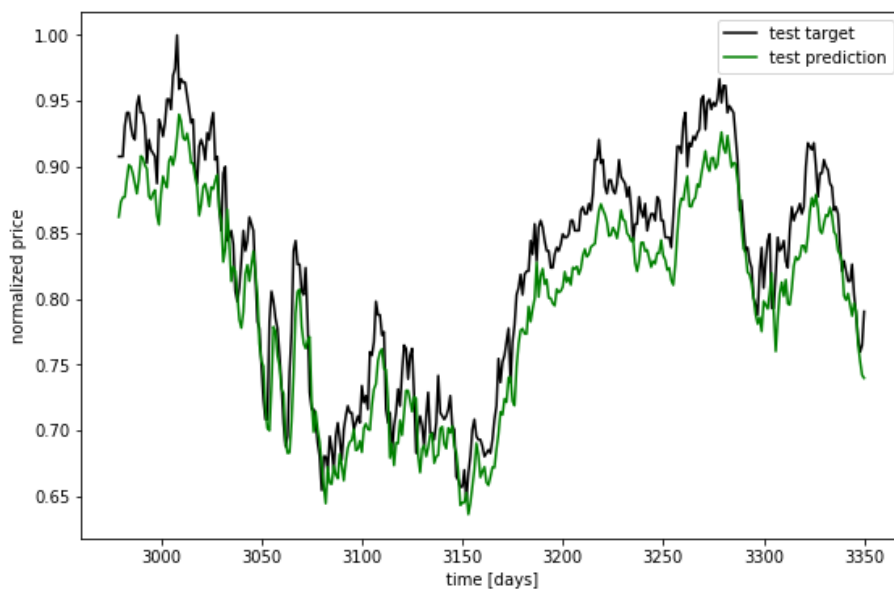
Kemudian ditampilkan analisa hasil model yang telah diuji berdasarkan nilai *close* untuk model GRU, LSTM, RNN yang ditunjukkan oleh Gambar 28., Gambar 29., dan Gambar 30.



Gambar 28. Prediksi pergerakan harga saham pada *data testing* GRU berdasarkan nilai *close* dengan epoch 100.



Gambar 29. Prediksi pergerakan harga saham pada *data testing* LSTM berdasarkan nilai *close* dengan epoch 100.

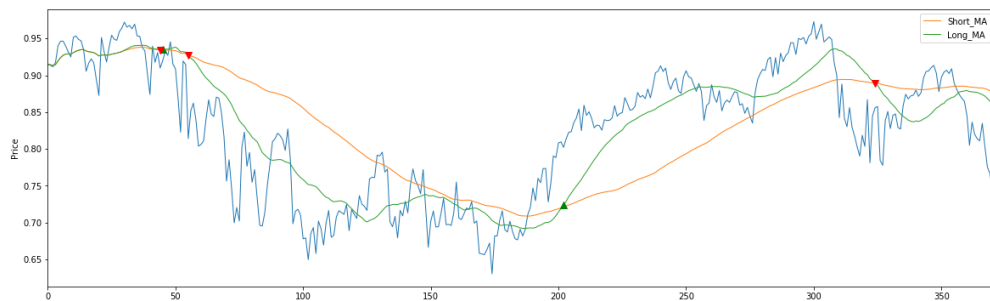


Gambar 30. Prediksi pergerakan harga saham pada *data testing* RNN berdasarkan nilai *close* dengan epoch 100.

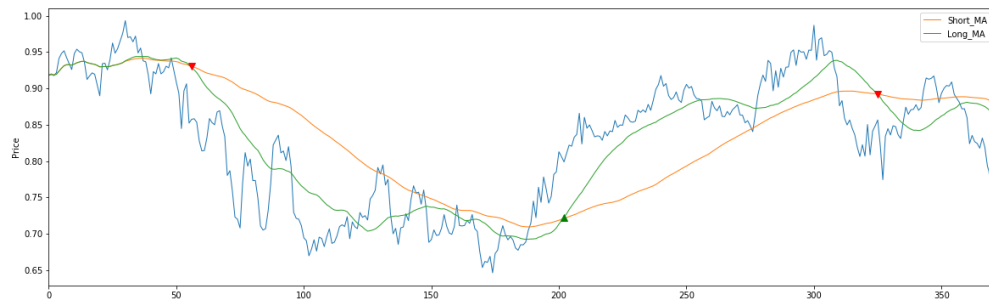
Dari hasil visualisasi grafik tersebut terlihat perbedaan yang cukup signifikan dari hasil masing-masing model. Untuk hasil model RNN yang ditunjukkan Gambar 27. dan Gambar 30. terlihat hasil prediksi yang berada di bawah data sebenarnya atau bisa disebut masih adanya error atau gap antara data prediksi dan data sebenarnya. Untuk hasil model LSTM yang ditunjukkan

Gambar 26. dan Gambar 29. terlihat hasilnya cukup baik dibandingkan hasil model RNN meskipun hasil data prediksi dari data test masih berfluktuatif yang ditunjukkan pada Gambar 29. Kemudian untuk hasil model GRU yang ditunjukkan Gambar 25. dan Gambar 28. terlihat hasil yang diperoleh lebih baik daripada hasil model RNN dan LSTM.

Kemudian analisis juga dilakukan dengan menghasilkan signal *buy* atau *sell* menggunakan indikator *moving average* (MA) berdasarkan nilai *close* dengan model GRU, LSTM, RNN pada saham yang masing-masingnya melalui visualisasi dalam bentuk grafik seperti ditunjukkan Gambar 31., Gambar 32., dan Gambar 33.



Gambar 31. Prediksi signal *buy* dan *sell* saham pada *data testing* GRU berdasarkan nilai *close* dengan epoch 100



Gambar 32. Prediksi signal *buy* dan *sell* saham pada *data testing* LSTM berdasarkan nilai *close* dengan epoch 100



Gambar 33. Prediksi signal *buy* dan *sell* saham pada *data testing* RNN berdasarkan nilai *close* dengan epoch 100

Pada Gambar 31., Gambar 32., dan Gambar 33. nilai *short* MA didistribusikan sebanyak 30 dan *long* MA sebanyak 90 sehingga membentuk dua garis yang saling berpotongan. Tiap perpotongan akan menghasilkan signal *buy* atau *sell*. Jika *long* MA turun dan memotong *short* MA dari atas ke bawah maka signal yang dihasilkan adalah *sell* sedangkan jika *long* MA naik dan memotong *short* MA dari bawah ke atas maka signal yang dihasilkan adalah *buy*. Dari grafik MA ini juga dapat diprediksi bahwa pergerakan harga saham pada hari selanjutnya masih dapat menunjukkan trend penurunan karena garis *long* MA belum memotong garis *short* MA.

Hasil pengujian berupa nilai MSE dan RMSE dari *data training*, *data validation* dan *data testing* seperti ditunjukkan pada Tabel 2.

Tabel 2. Pengujian masing-masing model untuk *epochs* 100

No	Model	Training		Validation		Testing	
		MSE	RMSE	MSE	RMSE	MSE	RMSE
1	LSTM	0.000438	0.020919	0.000270	0.016436	0.000385	0.019624
2	GRU	0.000418	0.020453	0.000291	0.017047	0.000380	0.019488
3	RNN	0.000433	0.020807	0.000548	0.023408	0.001188	0.034462

Dari hasil pengujian yang ditunjukkan pada Tabel 2. dapat diperoleh bahwa model yang memiliki nilai MSE dan RMSE paling optimal adalah model GRU kemudian model LSTM dan model RNN. Model GRU memperoleh nilai optimal untuk distribusi data yaitu *data training* dan *data testing* sedangkan *data validation* model LSTM lebih optimal.

6. Kesimpulan

Model pembelajaran mesin ANN dibangun untuk memprediksi pergerakan harga saham dengan pengujian model yaitu, RNN, LSTM dan GRU serta komposisi parameter *epochs* yaitu, 50 dan 100. Pada penelitian ini pengujian dibagi atas dua yaitu, pengujian 1 menggunakan *epochs* 50 dan pengujian 2 menggunakan *epochs* 100. Dari kedua pengujian yang dilakukan, hasil terbaik diperoleh dari model GRU dengan 100 *epochs* yang diukur melalui nilai MSE dan RMSE dengan masing-masing nilainya yaitu, 0.000380 dan 0.019488. Nilai MSE dan RMSE ini menunjukkan model GRU lebih baik dari model LSTM dan RNN untuk memprediksi data *time-series* seperti pada penelitian ini yaitu memprediksi pergerakan harga saham.

7. Referensi

1. Pengantar Pasar Modal. Sumber: <https://www.idx.co.id/investor/pengantar-pasar-modal/>
2. Saham. Sumber: <https://www.idx.co.id/produk/saham/>
3. *Understanding LSTM Networks*. Sumber: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
4. Roondiwala, M., Patel, H., Varma, S. 2017. *Predicting Stock Prices Using LSTM*. International Journal of Science and Research (IJSR)
5. Zhang, Lin. 2016. *Theory, Methodology, Tools, and Applications for Modeling and Simulation of Complex Systems*. Springer + Business Media, Singapore.
6. Hiransha Ma, Gopalakrishnan E.Ab, Menonab, V,K., Soman K,P. 2018. *NSE Stock Market Prediction Using Deep Learning Models*. International Conference on Computational Intelligence and Data Science (ICCIDS).
7. David M. Q. Nelson, Adriano C. M. Pereira, Renato A. de Oliveira. 2017. *Stock Market's Price Movement Prediction with LSTM Neural Networks*. International Joint Conference on Neural Networks (IJCNN).
8. Shah, D., Campbell, W., Zulkernine, F., H., 2018. *A Comparative Study of LSTM and DNN for Stock Market Forecasting*. IEEE International Conference on Big Data, At Seattle, Washington, USA.
9. *Recurrent Neural Network (RNN) dan Gated Recurrent Unit (GRU)*. Sumber: <https://socs.binus.ac.id/2017/02/13/rnn-dan-gru>